TU/e

Project FINGERPAINT

ATP-1.0

Acceptance Test Plan

Authors:
Tessa Belder (0739377)
Lasse Blaauwbroek (0749928)
Thom Castermans (0739808)
Roel van Happen (0751614)
Benjamin van der Hoeven (0758975)
Femke Jansen (0741948)
Hugo Snel (0657700)

Junior Management:
Simon Burg
Areti Paziourou
Luc de Smet

Senior Management:
Mark van den Brand, MF 7.096
Lou Somers, MF 7.145

Technical Advisor:
Ion Barosan, MF 7.082

Customer:
Patrick Anderson, GEM-Z 4.137

Eindhoven - June 23, 2013

Abstract

This document complies with the Software Engineering Standard as specified by the European Space Agency (ESA) [1].

This is the Acceptance Test Plan (ATP) of the FINGERPAINT project, developed in the context of the Software Engineering Project (2IP35). This document contains a detailed specification for the Acceptance Tests (AT). These tests have to be executed by the client, prof.dr.ir. P.D. Anderson, to determine whether the delivered product meets the user requirements described in the User Requirements Document (URD [2]). It also describes the environment needed to perform these tests. After a test has been performed by the client, a report on the results needs to be written. These reports will be included in this document. If all tests pass, the acceptance test phase can be finished successfully.

Contents

Document Status Sheet

Document Status Overview

General

Document title:	Acceptance Test Plan
Identification:	ATP-1.0
Author:	Tessa Belder, Lasse Blaauwbroek, Roel van Happen, Femke Jansen
Document status:	Approved by the customer

Document History

Version	*Date*	*Author*	*Reason of change*
0.0	14-June-2013	Tessa Belder, Lasse Blaauwbroek, Roel van Happen, Femke Jansen	Initial version.
0.1	18-June-2013	Tessa Belder, Lasse Blaauwbroek	Fixed inconsistensies between the AT's and the application.
1.0	18-June-2013	-	Approved by the customer.

Document Change Records Since Previous Issue

General

Date:	18-June-2013
Document title:	Acceptance Test Plan
Identification:	ATP-1.0

Changes

Page	*Paragraph*	*Reason to change*
-	-	Approved by the customer.

Chapter 1

Introduction

This chapter lists general information about this document.

1.1 Purpose

The Acceptance Test Plan (ATP) describes the plan for testing FINGERPAINT against the user requirements as described in the URD [2]. The acceptance tests make sure that FINGERPAINT complies with the requirements in the URD [2]. Therefore this document needs to be agreed upon by Group Fingerpaint and the client, prof.dr.ir. P.D. Anderson. These tests should be executed in the Acceptance Test (AT) phase of the FINGERPAINT project, as described in the ESA Software Engineering Standard [1].

1.2 Overview

Chapter 2 mentions the items to be tested together with the general criteria for the AT. A detailed specification for each test case is given in chapter 3. The procedures for the execution of the tests are explained in chapter 4. After execution of the tests, reports on the results will be presented in chapter 5. Appendix A contains a step-by-step user manual for the server setup. The client needs to perform all steps in this manual before executing the Acceptance Tests.

1.3 List of definitions and abbreviations

1.3.1 Definitions

Client Prof.dr.ir. P.D. Anderson.

1.3.2 Abbreviations

2IP35	The Software Engineering Project
AT	Acceptance Test(s)
ATP	Acceptance Test Plan
CPR	Capability Requirement
ESA	European Space Agency
SEP	Software Engineering Project
URD	User Requirements Document

1.4 List of references

[1] ESA, *ESA Software Engineering Standards.* ESA, March 1995.

[2] Group Fingerpaint, "User requirements document," *SEP*, 2013.

[3] Group Fingerpaint, "Acceptance test plan," *SEP*, 2013.

Chapter 2

Test plan

This chapter describes what items are tested with the acceptance tests, and how these items must be tested. Specific information about each test can be found in chapters 3-5.

2.1 Test items

In the acceptance tests, the FINGERPAINT application is tested to see if it fulfills the requirements as described in the URD [2]. In addition, the URD [2] contains use cases that describe the desired behavior.

2.2 Features to be tested

The features subject to testing are a part of the CPRs described in the URD [2]. Not all of these requirements are implemented, and only the implemented requirements can be tested. The implemented requirements are: **CPR1, CPR2, CPR3, CPR6, CPR7, CPR8, CPR9, CPR10, CPR11, CPR12, CPR13, CPR17, CPR18, CPR19, CPR20, CPR21, CPR22, CPR23, CPR24, CPR25, CPR29, CPR30, CPR31, CPR32, CPR33, CPR34, CPR35, CPR36, CPR37 and CPR40**.

2.3 Test deliverables

Prior to testing, the following documents/code should be completed:

- URD [2]
- ATP [3], should be finished up to the test reports (chapter 5).
- The Fingerpaint code

After the tests are concluded, test reports should be written. Problem reports should be written when necessary.

2.4 Testing tasks

Before the acceptance tests can be executed, the following needs to be done:

- The acceptance tests need to be written.
- The FINGERPAINT application should be functional.

2.5 Environmental needs

The hardware/software required to run the FINGERPAINT application is described in appendix A of the ATP [3].

2.6 Test case pass/fail criteria

The acceptance tests as a whole succeed if all the individual acceptance tests pass. Similarly if one test fails, the software is rejected. The test criteria are described in chapter 3.

Chapter 3

Test case specifications

This chapter gives a detailed description of all acceptance tests. These tests are strongly based on the use cases as described in Appendix B of the URD [2]. They cover all the user requirements from the URD [2], except for unimplemented and untestable requirements.

Requirements stating portability to a certain browser are also not stated; these requirements can be tested by testing all other requirements on the specified browser. For example: compatibility with Safari can be testing by running all acceptance tests in Safari. In a similar way, support for a certain Operating System can be tested by running the tests on that Operating System.

Requirements that are not implemented in the FINGERPAINT application are **CPR4, CPR5, CPR14, CPR15, CPR16, CPR26, CPR27, CPR28, CPR38, CPR39 and CPR41**.

The requirement that cannot be tested is **CPR40**.

Before executing these tests, one should read and follow the guidelines as described in chapter 4 of this document.

3.1 Choose a rectangular geometry and mixer type

AT1	**CPR1, CPR2**
Test items: Verify that the user can select a rectangular geometry and mixer when starting the application.	

Input specifications

1. Press the *Rectangle 400x240* button after loading the application.

2. Press the *Default* button.

Output specifications

- After step 1, a second selection row is displayed for the rectangle geometry.

- After step 2, both the selection rows close and both a menu and a drawing canvas in the shape of a rectangle show up.

3.2 Define an initial concentration distribution

AT2	**CPR6, CPR7, CPR8, CPR9, CPR10**

Test items: Verify that the user can define a concentration distribution by drawing in the drawing area.

Input specifications

1. Press somewhere in the drawing area.
2. Press *Select tool* in the menu bar on the right side of the application.
3. Press the button with a circle in the menu bar on the right side of the application.
4. Press somewhere in the drawing area.
5. Press the + button of the numberspinner in the menu bar on the right side of the application, until the number shown equals 10.
6. Press somewhere in the drawing area.
7. Press the button with a square in the menu bar on the right side of the application.
8. Press the - button of the numberspinner in the menu bar on the right side of the application, until the number shown equals 5.
9. Press the button which shows a black square on top of a white square in the menu bar on the right side of the application.
10. Press within the large black circle on the drawing area.
11. Press the button which shows a white square on top of a black square in the menu bar on the right side of the application.
12. Drag across the drawing area.
13. Press *Back* in the menu bar on the right side of the application.
14. Press *Clear canvas* in the menu bar on the right side of the application.

Output specifications

- After step 1, a small black square is displayed within the drawing area.
- After step 2, the menu on the right side of the application slides to the left, and a number spinner, a button with a circle, a button with a square, a button with a black square on top of a white square and a *Back* button are shown.
- After step 3, the button with a circle appears pressed down.
- After step 4, a small black circle is displayed within the drawing area.
- After step 6, a large black circle is displayed within the drawing area.

- After step 7, the button with a square appears pressed down.
- After step 9, the pressed button now shows a white square on top of a black square.
- After step 10, a smaller white square is displayed within the large black circle on the drawing area.
- After step 11, the pressed button now shows a black square on top of a white square.
- After step 12, a black line following the drag is displayed within the drawing area.
- After step 13, the menu on the right side of the application slides to the right, and the original menu items appear again.
- After step 14, the drawing area is completely white.

3.3 Save a concentration distribution

AT3	**CPR11**

Test items: Verify that the user can save a concentration distribution.

Input specifications

1. Drag across the drawing area to draw a happy smiley face.
2. Press *Distributions* in the menu bar on the right side of the application.
3. Press *Save* in the menu bar on the right side of the application.
4. Enter the text *Smiley* within the text area.
5. Press *Save.*
6. Press *Back* in the menu bar on the right side of the application.
7. Press *Clear canvas* in the menu bar on the right side of the application.
8. Drag across the drawing area to draw a sad smiley face.
9. Press *Distributions* in the menu bar on the right side of the application.
10. Press *Save* in the menu bar on the right side of the application.
11. Enter the text *Smiley* within the text area.
12. Press *Save.*
13. Press *Cancel.*
14. Enter the text *SadSmiley* within the text area.
15. Press *Save.*

16. Press *Back* in the menu bar on the right side of the application.
17. Press *Clear canvas* in the menu bar on the right side of the application.
18. Drag across the drawing area to draw a new happy smiley face.
19. Press *Distributions* in the menu bar on the right side of the application.
20. Press *Save* in the menu bar on the right side of the application.
21. Enter the text *Smiley* within the text area.
22. Press *Save.*
23. Press *Overwrite.*
24. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, a happy smiley face is displayed within the drawing area.
- After step 2, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Export image* and *Back* appear.
- After step 3, a popup panel with a textfield, a *Save* button and a *Cancel* button is displayed.
- After step 5, a *Save successful* message is displayed, which disappears after a few seconds.
- After step 6, the menu on the right side of the application slides to the right, and the original menu items appear again.
- After step 7, the drawing area is completely white.
- After step 8, a sad smiley face is displayed withing the drawing area.
- After step 9, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Export image* and *Back* appear.
- After step 10, a popup panel with a textfield, a *Save* button and a *Cancel* button is displayed.
- After step 12, a new popup panel with a message, a *Overwrite* button and a *Cancel* button is displayed.
- After step 13, the first popup panel is displayed again.
- After step 15, a *Save successful* message is displayed, which disappears after a few seconds.
- After step 16, the menu on the right side of the application slides to the right, and the original menu items appear again.

- After step 17, the drawing area is completely white.
- After step 18, a happy smiley face is displayed within the drawing area.
- After step 19, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Export image* and *Back* appear.
- After step 20, a popup panel with a textfield, a *Save* button and a *Cancel* button is displayed.
- After step 22, a new popup panel with a message, a *Overwrite* button and a *Cancel* button is displayed.
- After step 23, a *Save successful* message is displayed, which disappears after a few seconds.
- After step 24, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.4 Define and execute a single mixing step

AT4	**CPR18**

Test items: Verify that the user can define and execute a single mixing step on an initial concentration distribution.

Input specifications

1. Press the + button of the numberspinner with the *Step size* label in the menu bar on the right side of the application, until the number shown equals 5.
2. Press *Clear canvas* in the menu bar on the right side of the application.
3. Drag across the drawing area to draw a large black dot in the middle of the drawing area.
4. Drag the top wall to the right.
5. Press *Clear canvas* in the menu bar on the right side of the application.

Output specifications

- After step 2, the drawing area is completely white.
- After step 3, a large black dot is displayed withing the drawing area.
- After step 4, the black dot has moved, and now has a different shape.
- After step 5, the drawing area is completely white.

3.5 Define and execute a mixing protocol

AT5	**CPR17, CPR19, CPR20, CPR21, CPR29, CPR32**

Test items: Verify that the user can define and execute a mixing protocol on an initial concentration distribution.

Input specifications

1. Press the + button of the numberspinner with the *Step size* label in the menu bar on the right side of the application, until the number shown equals 6.25.
2. Drag across the drawing area to draw a large black dot in the middle of the left half of the drawing area.
3. Press *Define protocol* in the menu bar on the right side of the application.
4. Drag the top wall to the right.
5. Drag the bottom wall to the right.
6. Press the + button of the spinner for the number of steps in the menu bar on the right side of the application, until the number shown equals 10.
7. Press *Clear protocol* in the menu bar on the right side of the application.
8. Drag the top wall to the right and the bottom wall to the left.
9. Press the + button of the spinner for the number of steps in the menu bar on the right side of the application, until the number shown equals 10.
10. Press *Mix Now*.
11. Press *Stop defining protocol* in the menu bar on the right side of the application.

Output specifications

- After step 2, a large black dot is displayed withing the drawing area.
- After step 3, the menu on the right side of the application slides to the left, and two number spinners and *Stop defining protocol*, *Mix Now*, *View performance graph*, *Clear Protocol*, and *Protocols* buttons appear.
- After step 4, a text area appears in the menu bar on the right side of the application and shows the text *T[6.25]*.
- After step 5, the text area in the menu bar on the right side of the application shows the text *T[6.25] -B[6.25]*.
- After step 7, the text area in the menu bar on the right side of the application has disappeared, and the spinner for the number of steps shows 1.

- After step 8, the text area in the menu bar on the right side of the application appeared again and shows the text *T[6.25] B[6.25]*.
- After step 10, the drawing area shows a black spiral distribution. Also the *View performance graph* button in the menu bar on the right side of the application is now enabled.
- After step 11, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.6 Export the resulting distribution image of a mixing run

AT6	**CPR33**
Test items: Verify that the user can export the resulting distribution of a mixing run to their device.	

Input specifications

1. Press *Distributions* in the menu bar on the right side of the application.
2. Press the *Export image* button.
3. Follow the standard procedure to download a file.
4. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Export image* and *Back* appear.
- After step 2, the browser's download menu appears.
- After step 3, the image is saved at the chosen location.
- After step 4, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.7 Save a mixing protocol

AT7	**CPR22**
Test items: Verify that the user can store the current mixing protocol in local storage	

Input specifications

1. Press *Define protocol* in the menu bar on the right side of the application.
2. Press *Protocols* in the menu bar on the right side of the application.
3. Press the *Save* button.
4. Type the name *Testprotocol* in the textfield of the popup.
5. Press the *Save* button.
6. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and two number spinners and *Stop defining protocol*, *Mix Now*, *View performance graph*, *Clear Protocol*, and *Protocols* buttons appear.
- After step 2, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove* and *Back* appear.
- After step 3, a popup shows up with a textfield, a *Save* button and a *Cancel* button.
- After step 5, the popup closes and a *Save successful* message is displayed, which disappears after a few seconds.
- After step 6, the menu on the right side of the application slides to the right, and the protocol menu items appear again.

3.8 View the performance of a mixing run

AT8	**CPR34**
Test items: Verify that the user can view the mixing performance of an executed mixing run.	

Input specifications

1. Press the *View performance graph* button.

Output specifications

- After step 1, a popup shows up with a graph. In this graph, the mixing performance is shown on the y-axis, and the corresponding protocol step is shown on the x-axis. In addition, there is a *Close* button and an *Export Graph* button.

3.9 Export the performance graph of a mixing run

AT9	**CPR35**
Test items: Verify that the user can export the performance graph of a mixing run to their device.	

Input specifications

1. Press the *Export Graph* button.
2. Follow the standard procedure to download a file.
3. Press the *Close* button.
4. Press *Stop defining protocol* in the menu bar on the right side of the application.

Output specifications

- After step 1, the browser's download menu appears.
- After step 2, the image is saved at the chosen location.
- After step 3, the popup showing the performance graph closes.
- After step 4, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.10 Save a mixing run

AT10	**CPR30**
Test items: Verify that the user can save the results of a mixing run.	

Input specifications

1. Press *Results* in the menu bar on the right side of the application.
2. Press *Save* in the menu bar on the right side of the application.
3. Enter the text *Spiral* within the text area.
4. Press *Save*.
5. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Compare performance* and *Back* appear.
- After step 2, a popup panel with a textfield, a *Save* button and a *Cancel* button is displayed.
- After step 4, a *Save successful* message is displayed, which disappears after a few seconds.
- After step 5, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.11 Load a previously saved concentration distribution

AT11	**CPR13**

Test items: Verify that the user can load a previously saved concentration distribution in the drawing area.

Input specifications

1. Press *Distributions* in the menu bar on the right side of the application.
2. Press *Load* in the menu bar on the right side of the application.
3. Press *SadSmiley.*
4. Press *Load* in the menu bar on the right side of the application.
5. Press *Smiley.*
6. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Export image* and *Back* appear.
- After step 2, a popup panel with *SadSmiley*, *Smiley* and a *Close* button is displayed.
- After step 3, the popup panel is closed, and the sad smiley that was drawn at AT3 is displayed in the drawing area.
- After step 4, a popup panel with *SadSmiley*, *Smiley* and a *Close* button is displayed.
- After step 5, the popup panel is closed, and the secondly drawn happy smiley from AT3 is displayed in the drawing area.
- After step 6, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.12 Load a previously saved mixing protocol

AT12	**CPR24**

Test items: Verify that the user can load a previously saved mixing protocol.

Input specifications

1. Press *Define protocol* in the menu bar on the right side of the application.
2. Press *Clear protocol* in the menu bar on the right side of the application.
3. Press the + button of the spinner for the number of steps in the menu bar on the right side of the application, until the number shown equals 10.
4. Press *Protocols* in the menu bar on the right side of the application.
5. Press *Load* in the menu bar on the right side of the application.
6. Press *testprotocol*.
7. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and two number spinners, a text area and *Stop defining protocol*, *Mix Now*, *View performance graph*, *Clear Protocol*, and *Protocols* buttons appear.
- After step 2, the text area in the menu bar on the right side of the application has disappeared, and the spinner for the number of steps shows 1.
- After step 4, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, and *Back* appear.
- After step 5, a popup panel with *Testprotocol* and a *Close* button is displayed.
- After step 6, the popup panel is closed.
- After step 7, the menu on the right side of the application slides to the right, the protocol menu items appear again, and *T[6.25] B[6.25]* is displayed in the text area on the right side of the application.

3.13 View multiple performance results in one graph

AT13	**CPR36**

Test items: Verify that the user can view multiple performance results from previous mixing runs in one graph.

Input specifications

1. Press *Mix Now* in the menu bar on the right side of the application.
2. Press *Stop defining protocol* in the menu bar on the right side of the application.
3. Press *Results* in the menu bar on the right side of the application.
4. Press *Save* in the menu bar on the right side of the application.
5. Enter the text *MixingRun2* within the text area.
6. Press *Save.*
7. Press *Compare performance* in the menu bar on the right side of the application.
8. Select *Spiral.*
9. Press *Compare.*
10. Press *New Comparison.*
11. Select both *Spiral* and *MixingRun2.*
12. Press *Compare.*

Output specifications

- After step 1, the resulting concentration distribution is displayed in the drawing panel.
- After step 2, the menu on the right side of the application slides to the right, and the original menu items appear again.
- After step 3, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Compare performance* and *Back* appear.
- After step 4, a popup panel with a textfield, a *Save* button and a *Cancel* button is displayed.
- After step 6, a *Save successful* message is displayed, which disappears after a few seconds.
- After step 7, a popup panel with *Spiral*, *MixingRun2*, a *Compare* button and a *Cancel* button is displayed.
- After step 9, a popup panel with a graph of the performance of the *Spiral* mixing run, a *New Comparison* button, an *Export graphs* button and a *Cancel* button is displayed.
- After step 10, The first popup panel is displayed again.
- After step 12, a popup panel with a graph of the performance of both the selected mixing runs, a *New Comparison* button, an *Export graphs* button and a *Cancel* button is displayed.

3.14 Export the performance graph of multiple mixing runs

AT14	**CPR37**

Test items: Verify that the user can export the performance graph of multiple mixing runs to their device.

Input specifications

1. Press the *Export graphs* button.
2. Follow the standard procedure to download a file.
3. Press the *Close* button.

Output specifications

- After step 1, the browser's download menu appears.
- After step 2, the image is saved at the chosen location.
- After step 3, the popup showing the performance graph closes.

3.15 Load the results of a previously saved mixing run

AT15	**CPR32, CPR34**

Test items: Verify that the user can load the resulting distribution, protocol and performance graph of a previously saved mixing run.

Input specifications

1. Press *Load* in the menu bar on the right side of the application.
2. Press *Spiral*.
3. Press the *View performance graph* button.
4. Press the *Close* button.
5. Press *Stop defining protocol* in the menu bar on the right side of the application.

Output specifications

- After step 1, a popup panel with *Spiral*, *MixingRun2* and a *Close* button is displayed.
- After step 2, the popup panel is closed, and the black spiral distribution that was obtained at AT5 is displayed in the drawing area. Also, the menu bar on the right side of the application navigates to the protocol menu items. The text *T[6.25] B[6.25]* is displayed in the text area on the right side of the application and the spinner for the number of steps shows 10.

- After step 3, a popup shows up with the graph from AT8. In addition, there is a *Close* button and an *Export Graph* button.
- After step 4, the popup showing the performance graph closes.
- After step 5, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.16 Remove a mixing run from local storage

AT16	**CPR31**

Test items: Verify that the user can remove a previously stored mixing run from local storage

Input specifications

1. Press *Results* in the menu bar on the right side of the application.
2. Press the *Remove* button.
3. Press the `X` button next to the *MixingRun2* name.
4. Press the *Close* button.
5. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Compare performance* and *Back* appear.
- After step 2, a popup shows up with a table containing the names *Spiral* and *MixingRun2*, an `X` button next to each of them, and a *Close* button under this table.
- After step 3, a *Delete successful* message is displayed, which disappears after a few seconds. Also, the *MixingRun2* file is removed from the table.
- After step 4, the popup closes.
- After step 5, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.17 Remove a concentration distribution from local storage

AT17	**CPR12**

Test items: Verify that the user can remove a previously stored concentration distribution from local storage

Input specifications

1. Press *Distributions* in the menu bar on the right side of the application.
2. Press the *Remove* button.
3. Press the `X` button next to the *Smiley* name.
4. Press the *Close* button.
5. Press *Back* in the menu bar on the right side of the application.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove*, *Export image* and *Back* appear.
- After step 2, a popup shows up with a table containing the names *Smiley* and *SadSmiley*, an `X` button next to each of them, and a *Close* button under this table.
- After step 3, a *Delete successful* message is displayed, which disappears after a few seconds. Also, the *Smiley* file is removed from the table.
- After step 4, the popup closes.
- After step 5, the menu on the right side of the application slides to the right, and the original menu items appear again.

3.18 Remove a mixing protocol from local storage

AT18	**CPR23**

Test items: Verify that the user can remove a previously stored mixing protocol from local storage

Input specifications

1. Press *Define protocol* in the menu bar on the right side of the application.
2. Press *Protocols* in the menu bar on the right side of the application.
3. Press the *Remove* button.
4. Press the `X` button next to the *Testprotocol* name.
5. Press the *Close* button.

Output specifications

- After step 1, the menu on the right side of the application slides to the left, and two number spinners and *Stop defining protocol*, *Mix Now*, *View performance graph*, *Clear Protocol*, and *Protocols* buttons appear.
- After step 2, the menu on the right side of the application slides to the left, and the new buttons *Save*, *Load*, *Remove* and *Back* appear.
- After step 3, a popup shows up with a table containing the name *TestProtocol*, an `X` button next to this name, and a *Close* button under this table.
- After step 4, a *Delete successful* message is displayed, which disappears after a few seconds. Also, the *Testprotocol* file is removed from the table, and the text *No saved files* appears.
- After step 5, the popup closes.

3.19 Choose a square geometry and mixer type

AT19	**CPR3**

Test items: Verify that the user can select a square geometry and mixer when starting the application.

Input specifications

1. Press the *refresh* icon of the browser.
2. Press the *Square* button.
3. Press the *Default* button.

Output specifications

- After step 1, the initial menu of the application is displayed.
- After step 2, a second selection row is displayed for the square geometry.
- After step 3, both the selection rows close and both a menu and a drawing canvas in the shape of a square show up.

3.20 Define a mixing protocol for a square geometry

AT20	**CPR25**

Test items: Verify that the user can define a mixing protocol for a square geometry.

Input specifications

1. Press the + button of the numberspinner with the *Step size* label in the menu bar on the right side of the application, until the number shown equals 6.25.
2. Press *Define Protocol* in the menu bar on the right side of the application.
3. Drag the top wall to the right.
4. Press the - button of the numberspinner with the *Step size* label in the menu bar on the right side of the application, until the number shown equals 3.5.
5. Drag the bottom wall to the right.

Output specifications

- After step 2, the menu on the right side of the application slides to the left, and two number spinners and *Stop defining protocol*, *Mix Now*, *View performance graph*, *Clear Protocol*, and *Protocols* buttons appear.
- After step 3, the text area in the menu bar on the right side of the application shows the text *T[6.25]*.
- After step 5, the text area in the menu bar on the right side of the application shows the text *T[6.25] -B[3.5]*.

Chapter 4

Test procedures

For the tests to succeed, it is important that the tests are done in order. It is also important that the application is not closed between tests, unless a test states something else. Otherwise some steps done in earlier tests may have to be repeated for the latter tests to succeed. Furthermore, the tests should be executed on the device that hasn't run the application before. If this is not the case, i.e. the application has been used on the device before, all distributions, protocols and mixing results that might have been saved on the device should be removed before starting the acceptance tests. The set up required to execute the acceptance tests from chapter 3 is described in appendix A.

Chapter 5

Test reports

5.1 First acceptance test

5.1.1 Description

In the first acceptance test, the client tested all the acceptance tests. The tested acceptance tests are AT1, AT2, AT3, AT4, AT5, AT6, AT7, AT8, AT9, AT10, AT11, AT12, AT13, AT14, AT15, AT16, AT17, AT18, AT19 and AT20. The tests are supervised by Lasse Blaauwbroek, Roel van Happen and Thom Castermans.

5.1.2 Test results

Below the results are given for the tests:

AT1	pass
AT2	pass
AT3	pass
AT4	pass
AT5	pass
AT6	pass
AT7	pass
AT8	pass
AT9	pass
AT10	pass
AT11	pass
AT12	pass
AT13	pass
AT14	pass
AT15	pass
AT16	pass
AT17	pass
AT18	pass
AT19	pass
AT20	pass

Chapter 6

Requirements traceability matrices

In this chapter, we link implemented CPRs, described in the URD [2], to the tests described in this document. CPRs we did not implement are omitted from this matrix.

6.1 URD to ATP

The following matrix lists which components are linked to the CPRs in the URD.

CPR	Acceptance tests	CPR	Acceptance tests
CPR1	AT1	CPR2	AT1
CPR3	AT19	CPR6	AT2
CPR7	AT2	CPR8	AT2
CPR9	AT2	CPR10	AT2
CPR11	AT3	CPR12	AT17
CPR13	AT11	CPR17	AT5
CPR18	AT4	CPR19	AT5
CPR20	AT5	CPR21	AT5
CPR22	AT7	CPR23	AT18
CPR24	AT12	CPR25	AT20
CPR29	AT5	CPR30	AT10
CPR31	AT16	CPR32	AT5, AT15
CPR33	AT6	CPR34	AT8, AT15
CPR35	AT9	CPR36	AT13
CPR37	AT14		

6.2 ATP to URD

The following matrix lists which CPRs are implemented by each acceptance test.

Acceptance test	CPR(s)
AT1	CPR1,CPR2
AT2	CPR6, CPR7 ,CPR8, CPR9, CPR10
AT3	CPR11
AT4	CPR18
AT5	CPR17, CPR19, CPR20, CPR21, CPR29, CPR32
AT6	CPR33
AT7	CPR22
AT8	CPR34
AT9	CPR35
AT10	CPR30
AT11	CPR13
AT12	CPR24
AT13	CPR36
AT14	CPR37
AT15	CPR32, CPR34
AT16	CPR31
AT17	CPR12
AT18	CPR23
AT19	CPR3
AT20	CPR25

Appendix A

Environment setup

The environment of the Fingerpaint application consists of a client with a browser and a server on which the application needs to be installed.

What follows is a manual on how to build the Fingerpaint application.

A.1 Server

We assume the server runs a Linux operating system. A Windows operating system can also be used, but that is not supported by this manual. The text below is tested on a Debian based system.

A.1.1 Tools installation

First the JDK, the Ant and Make build systems and the gfortran and gcc compilers need to be installed. To get the application from the git repository, git is also required. Installation can be done through the package manager of the operating system. On a Debian based system this is:

```
apt-get install openjdk-7-jdk gcc gfortran make ant git
```

A.1.2 Retrieving the application

To retrieve the application from the git repository, navigate to the folder where you want to place the source. We denote this folder as <APP-ROOT>. Now execute the following command:

```
git clone git@github.com:Fingerpaint/project-code.git
```

The repository is now initialized in <APP-ROOT>.

A.1.3 Compilation of the application

There are three possible options to compile the application. These options are the following and the name of the option you chose is denoted by <COMPILER-OPTION>.

run This option compiles the application and immediately starts a server running the application

jar This option compiles the application to a runnable jar file named FINGERPAINT.JAR. This file can then be run at any time to start the server running the application

war This option compiles the application to a war file named FINGERPAINT.WAR. This file can then be deployed in a jetty server. This deployment procedure is, however, not covered by this manual as it depends on the configuration of the server used.

For the first compiling option a port number may be chosen on which the server listens, denoted by <PORT-NUMBER>.

Extra compiler options may be given to the GWT-compiler in order to modify its behavior. These options are denoted by <GWT-OPTIONS>. This is, however, an advanced option and is not covered futher by this manual.

We assume source code of the application can be found in the <APP-ROOT> folder. Navigate in the console to the <APP-ROOT>/FINGERPAINT folder. We need to make sure the JAVA_HOME environment variable is correctly set. On some machines, this will be the case by default. The variable should be set to the home directory of the JDK that was just installed (we denote this directory as <JDK-HOME>). If the variable was already set correctly (or you choose to set it manually, not covered by this manual), you can ignore the option setting the <JDK-HOME> in the following command. Otherwise, you have to include it. You can now execute the following command to compile the project:

```
ant <compiler-option> [-Djava.jdk.home=<jdk-home>] [-Dgwt.args=<gwt-options>]
                      [-Drun.port.number=<port-number>]
```

After compilation has completed, either a war or jar file has been created, or the application has been deployed. If a jar file was created, the following section describes how to run the application.

A.1.4 Running the application

After compilation, move the FINGERPAINT.JAR to the folder where you want to run the application. We assume this folder is <DEPLOY-ROOT>. You can again chose a port number to run the application on, denoted by <PORT-NUMBER>. Navigate in the console to this folder and execute the following:

```
java -jar Fingerpaint.jar <port-number>
```

This starts the standalone fingerpaint server listening to the specified port on the system. After typing QUIT and pressing <ENTER> the server is stopped again.

A.1.5 Deployment of the application

The application can also be deployed in a jetty server. See section A.1.3 to compile the application into a war file. This file can now be deployed to a jetty server in a standard manner. However, this procedure is not explained in this manual, because it is highly dependent on the configuration of the server.