

LIST OF CORRECTIONS

Note: Flet sammen med common report. Meget af det der staar
her burde egentlig staa i common report [9](#)

Note: Find god overskrift [10](#)

Note: Hvorfor kraever det kun occasional feedback? [10](#)

Note: der mangler at skrives om TDD [10](#)

Note: Hate this word [11](#)

Note: DIEB reference [11](#)

Note: hate this word [11](#)

Note: Mangler TAIL [11](#)

Note: Sprint overskrifter [13](#)

Note: indsaet ref til der hvor vi fandt ud af at vi skulle have
authentication [17](#)

Note: Design - 'internal architecture' skal nok skrives færdig før
man kan skrive noget her. [21](#)

Frontpage!!!

Aalborg University

Department of Computer Science

Student report
Selma Lagerlöfs Vej 300

9220 Aalborg East

<http://www.cs.aau.dk>

Title:

Project title

Theme:

Theme

Project Term:

P6, spring 2012

Project Group:

sw605f12

Synopsis:

Synopsis synopsis

Students:

Kasper Møller Andersen
Magnus Stubman Reichenauer
Rasmus Steiniche
Thomas Kobber Panum

Supervisor:

Ulrik Nyman

Copies: 6

Pages: HUSK AT INDKOMMENTER
FØR VI AFLEVERER

Finished: June 4th, 2012.

This rapport and its content is freely available, but publication (with source) may only be made by agreement with the authors.

Kasper Møller Andersen

Magnus Stubman Reichenauer

Rasmus Steiniche

Thomas Kobber Panum

PREFACE

QUOTATIONS are the words of another person along with a source. The source of the citation will either be in the text immediately before or after, or could potentially be incorporated into the quote as shown in the example below:

REFERENCES are references to sections, figures, code snippets, chapters or parts written elsewhere in the report. This could look like the following:

This is explained in Section X.Y.

CODE EXAMPLES are written in a special environment so they are easy to read and recognize. Whenever there is a sequence of three dots ("...") in a code snippet, it means that we have omitted some content, which is not important in that specific context.

CONTENTS

I PROLOGUE	1
1 REPORT STRUCTURE	3
2 PROBLEM DEFINITION	5
II PROCESS	7
3 PREANALYSIS	9
3.1 Working Process	9
3.1.1 Multiproject work form	9
3.1.2 Launcher work form	10
3.2 Understanding	10
3.2.1 Interviews	10
3.2.2 Field Observations	11
3.3 Prototyping	11
4 ITERATIVE PROCESS	13
4.1 Sprint 1 - March 19th to March 23rd	13
4.2 Sprint 2 - March 26th to April 4th	13
4.3 Sprint 3 - April 10th to April 19th	13
4.4 Sprint 4 - April 23rd to May 2nd	14
4.5 Sprint 5	14
III PRODUCT	15
5 DESIGN	17
5.1 Internal Architecture	17
5.2 External Architecture	17
5.2.1 Launcher	17
5.2.2 GIRAF GUI Components	18
6 IMPLEMENTATION	21
6.1 Activity Structure	21
6.2 Logo Activity	21
6.3 Authentication Activity	23
6.4 Home Activity	23
6.5 Profile Select Activity	23
7 TEST AND VERIFICATION	25
7.1 Unit Test	25
7.2 Usability Test	25
IV EPILOGUE	27
8 REFLECTION	29
8.1 Remarks to the next group of developers	29
9 CONCLUSION	31

V APPENDIX	33
A IMPLEMENTATION	35
A.1 Logo Activity	35
A.2 Profile select activity	35

LIST OF FIGURES

Figure 1	Activity overview page from one of the first prototypes.	12
Figure 2	Flow diagram	17
Figure 3	External Architecture	18
Figure 4	GIRAF GUI Components Architecture Example	19
Figure 5	Activity diagram	21
Figure 6	Landscape profile select activity screenshot	24
Figure 7	Portrait Logo Activity only shown when user do not have a valid session.	35
Figure 8	Landscape Logo Activity only shown when user has an valid session in and start the application again.	36
Figure 9	Portrait profile select activity screenshot	37

LIST OF TABLES

LISTINGS

- Listing 1 The Logo Activity and how it uses Thread and synchronized [22](#)
- Listing 2 This is code [23](#)
- Listing 3 This is code [23](#)
- Listing 4 How the launcher makes the intent for launching an application [24](#)

ACRONYMS

XP Extreme Programming

GUI Graphical User Interface

Part I
PROLOGUE

REPORT STRUCTURE

In deciding upon a report structure, two main different approaches were considered.

1. Traditional analysis, design, & implementation-structured product oriented report
2. "Diary" iteration-structured process oriented report

The strength of the first considered approach is the clear way the product would be presented and it would be easy to understand the analysis, design, and implementation done of the product. The weakness is the ability to represent the reasons for the design choices and the implementational details, as these have emerged from a changing workflow which have alternated between analysis, designing, and programming activities, which were required by the used development method.

The second considered approach's strength is the clear and logical reasoning of the understanding which have emerged from the iterative process. The weakness is the lack of a clear structured way to easily grasp and understand the final state of analysis, design, and implementational work done.

We have chosen a mix, where we first focus on process oriented activities, their order of happening, and what the effect of them were, followed by a product oriented structure of analysis, design, and implementation.

2

PROBLEM DEFINITION

Part II
PROCESS

3

PREANALYSIS

This chapter describes the preparations that were made before the main design and implementation began. These preparations include research and decisions about what work form to use, e.g. agile or traditional, research into the use domain of the product, and some initial prototypes of the product.

3.1 WORKING PROCESS

The working process of this project is based on agile development methods, mainly Scrum and XP. There are two work forms used by us in this project, the one used by the entire multiproject group and the one used by our project group.

3.1.1 *Multiproject work form*

The work form of the multiproject group is designed around Scrum of Scrums. With it, the work for all the project groups is split into sprints, and each project group works in synchronization with the other project groups. Each sprint is started with a meeting, where each project group presents their progress so far and what they are going to accomplish in the coming sprint. These meetings are also used to vote on the length of the coming sprint, knowledge sharing and decision making for decisions that affect the entire multiproject group.

Each sprint is ended with an evaluation meeting, used to briefly sum up and evaluate on the sprint, such that each project group receives feedback on how they might improve their next sprint, before that sprint is planned. When two or more project groups need to work on something not related to the multiproject group, no formal meeting is required, and open discussion amongst the project groups is encouraged through an open door policy. This has been important, as many project groups rely on each other to provide services. This has also led to continuous, informal integration testing of the system, as each project group increased their reliance on the others.¹

¹ FiXme Note: Flet sammen med common report. Meget af det der staar her burde egentlig staa i common report

3.1.2 *Launcher work form*

² The work form used in our project group is primarily based on XP. XP is built around having a customer available at all times, but this has not been the case for this project. We found it unnecessary to compensate for this however. This was due to the nature of the project, which, in the case of the launcher, was focused a lot on usability and GUI, subjects that only require occasional user feedback to develop properly.³ The GUI focus also made it easier to fill out the backlog without the customer, and planning poker – a Scrum practice – was used to determine the size of each task.

Other XP conventions did make it into the work form though, e.g. pair programming. Pair programming has been a great tool in keeping the development pace up, as assisting each other in this manner makes it easier to discover problems and solutions early, while also reducing overhead in communicating code to the rest of the team.

Another XP convention that helped reduce this overhead was refactoring. This involves going through existing code to rewrite parts that are complex from a readability point of view, in order to simplify the code and make it easier to understand. This is essential, as the project will be handed over to a new team later on, and high readability helps ensure that the project is useful to them.

Lastly, the “whole team” and “sustainable pace” conventions were used as well. The reason for this was to ensure that our work remained high in quality, and led to fairly stringent work rules, where the team agreed to work through the day together, but not work at home. Exceptions were made in case of illness.

Of final note is that “collective code ownership” was also employed. However, this is a demand from the study regulation, and not something that was deployed based on personal judgement.

⁴

3.2 UNDERSTANDING

This section will address the tools used and the work done for understanding the customer domain.

3.2.1 *Interviews*

One of the method used for gathering understanding of the customer domain is interviews. The customer domain contains two kind of users: Guardians and children with austim. Guardians have exten-

² FiXme Note: Find god overskrift

³ FiXme Note: Hvorfor kraever det kun occasional feedback?

⁴ FiXme Note: der mangler at skrives om TDD

sive⁵ knowledge about the children with austim, and as they are the customers, we have chosen only to interview them. The structure of the interviews were semi structured, in order to create a solid base of questions, while having flexibility, incase the need of diving into subtopics⁶. Two types of interviews was performed: One conducted by non-fixed subgroups of the multi-project with the customers divided between each of the subgroups, and the other type was conducted by each of the project groups with each of their assigned customer. The first type of interviews was done at the university building, in separate rooms. The second type of interviews was performed at the customers workplaces.

The results from the first interview showed: Usability is critical, based on experience from previous software and the need for certificates in order to use the existing software in the same category. Flexibility is also critical, as the needs of each child with austim is individual.

3.2.2 *Field Observations*

As a part of the interviews conducted at the customers workplace, the environment was observed. The observation gave insight in: currently used physical and software tools, the robustness of the environment and the organization needed for the children with austim.

The impressions of the field observations were: The current software tools have limited flexibility and have an extensive⁷ cost. The physical tools used are robust and fulfill simple tasks. Some of the simple tasks the physical tools fulfill, could easily be solved by software solutions. The environment have the same robustness as a public school, as the same interior is used. The organization is done by the guardians with the children with austim having influence, and tools help the communication and enhance the understanding of the children with austim.

⁸

3.3 PROTOTYPING

This section will explain how important prototyping and sketching was in this project and how it helped in the preanalysis.

In this project a very big part have been prototyping and sketching. This was done to gather understanding about how the final designe would look like and how it could be improved further. The sketches

⁵ FiXme Note: Hate this word

⁶ FiXme Note: DIEB reference

⁷ FiXme Note: hate this word

⁸ FiXme Note: Mangler TAIL

was used to gather some information about how the guardians saw the design and if the design were right for them. The prototypes was improved, remade over and over because of the iterative workform. The most important part of the overall design happened in the sketching and prototyping phase, all activities and their screens was made this way.

A late version of a prototype was represented to the guardians from Egebakken. This was a papir-prototype and represented two things about the product: How the screens was ment to be implemented, designed and how the flow in the program was going to be. It was designed so that buttons could be seen and when pressed on a button it linked to a new page of papir. This was done so the guardians could try out the program even before it was made and talk about what they had in mind about the overall design of the launcher. This was done so the guardians could come with feedback on which screens or activities should be changed as it was important to hear what the costumer had to say. It was also done so it was possible to check out how the flow was and if it was good enough or had to be improved further. An overview of this prototype can be seen in [Figure 1](#)

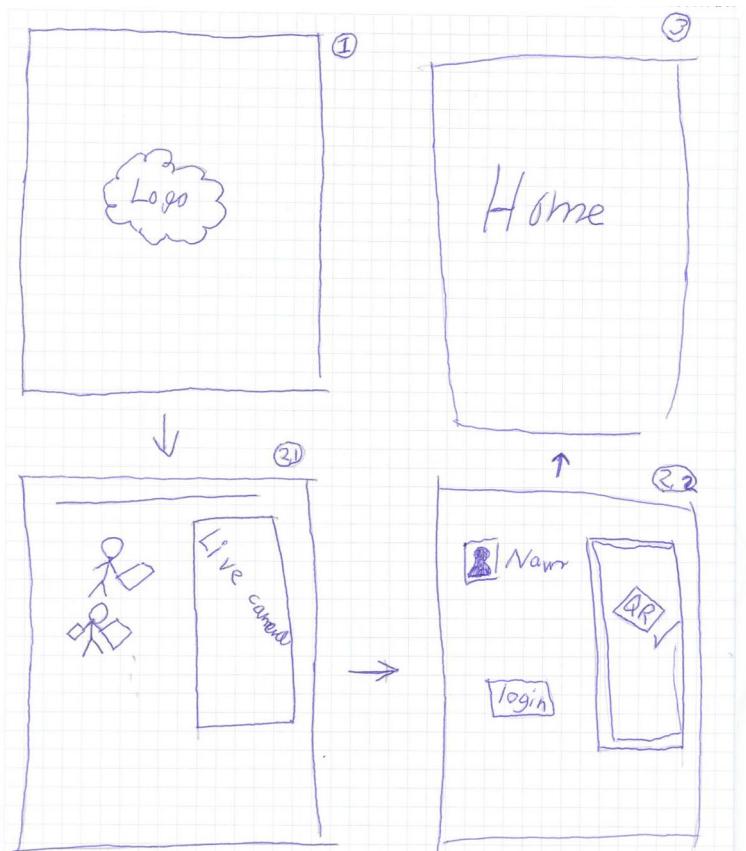


Figure 1: Activity overview page from one of the first prototypes.

4

ITERATIVE PROCESS

4.1 SPRINT 1 - MARCH 19th TO MARCH 23rd

¹ The focus of this sprint was on usability. This includes creating use cases, evaluating login methods, and designing visual elements of the system. By the end of the sprint, a usable design had been created. Managementwise, the meetings held in the multiproject group until this point had been marked by lengthy discussions and knowledge sharing, which made the meetings bloated and inefficient. This improved by now, as most major decisions about the project had been made and knowledge sharing had been done for most relevant subjects. The meetings also improved as the organization of the multiproject group improved, and better restrictions were put in place to help alleviate unnecessary discussion.

There were issues with miscommunication however. For example, we presented ideas at a meeting in this sprint, but did not clearly communicate which parts we would aim to implement, and which we considered part of our vision for the system, but would not pursue.

4.2 SPRINT 2 - MARCH 26th TO APRIL 4th

The focus of this sprint was to get implementation started. As a base design was already in place, ressources were mainly put into getting to know the system and the requirements for making a launcher. This led to features like app arrangement, app loading and a login system being the priority, with services for the other project groups not yet becoming important. At the end of the sprint, these features had been roughly implemented.

An issue came up at a meeting, as one of the project groups had trouble with their members showing up to the meetings on time. A quick effort was made to solve the issue, and though some disagreement occurred, several possible resolutions were presented by the multiproject group.

4.3 SPRINT 3 - APRIL 10th TO APRIL 19th

The work done in this sprint revolved around improving the features that were implemented in the previous sprint, while adding new features and services. Tasks worked on in this sprint include the ability to launch applications, providing profile information for applications,

¹ FiXme Note: Sprint overskrifter

and refining login screen and design. This is also where integration with the Oasis library has started, increasing our reliance on the Oasis group.

4.4 SPRINT 4 - APRIL 23rd TO MAY 2nd

4.5 SPRINT 5

fish

Part III

PRODUCT

The *product* part explains the state of the product as of our delivery, and not the “finished” state of the product.

5

DESIGN

5.1 INTERNAL ARCHITECTURE

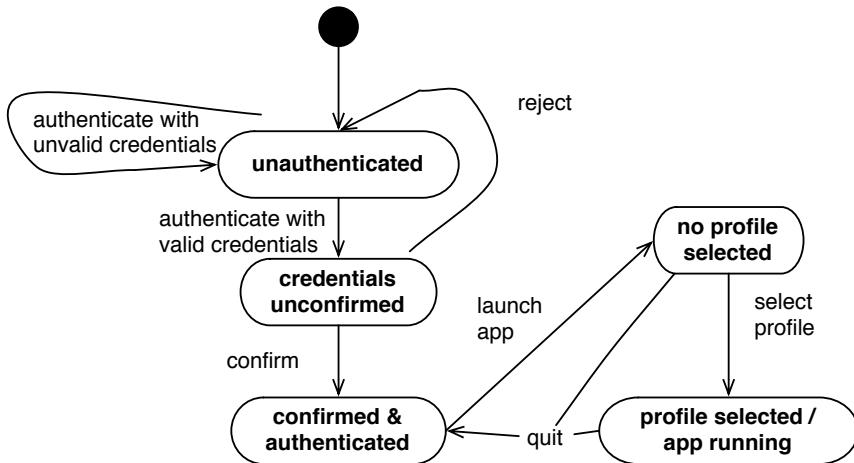


Figure 2: Flow diagram

Figure 2 shows the state of the launcher. The first three states handles the distinction between users which are allowed to access the launchable applications, and those who are not.¹

Based on Figure 2, ..

5.2 EXTERNAL ARCHITECTURE

Two external architectures will be covered: The launcher architecture, and the GIRAF GUI Components library architecture.

5.2.1 Launcher

The external architecture for the launcher, can be seen in Figure 3. The launcher provides with two services: Launching apps, and providing information about the profile that is authenticated. The *Launch app* service allows external apps to run and be stopped, ie: *launch*. The *Profile* service provides the possibility of external apps to retrieve the current authenticated profile. This is necessary for external apps in order to use the profile specific data in Oasis.

¹ FiXme Note: indsæt ref til der hvor vi fandt ud af at vi skulle have authentication

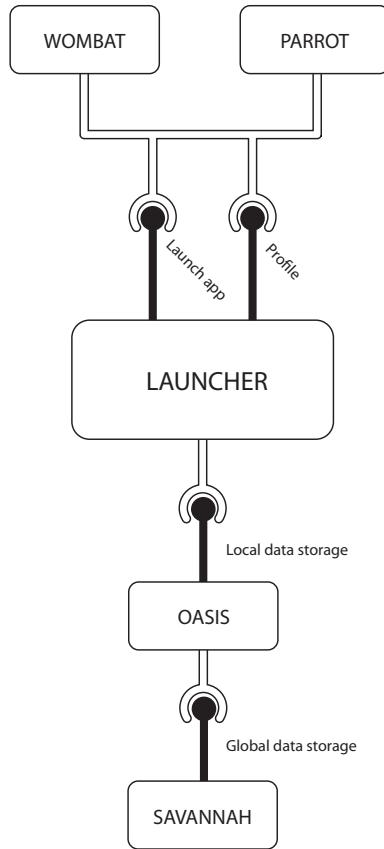


Figure 3: External Architecture

The Oasis database is the single required service provider for the launcher. It allows for authenticating profiles and store settings for the launcher relative to the profile.

5.2.2 GIRAFT GUI Components

The GIRAFT GUI Components architecture is not final, and is based on existing Android architecture. An example of the architecture can be seen in [Figure 4](#). The philosophy behind the architecture is to use existing Android UI components, with a new layout and possible added functionality. The example shown in [Figure 4](#) highlights a possible way to incorporate this philosophy, assuming the need for a customized button *GButton*.



Figure 4: GIRAf GUI Components Architecture Example

6

IMPLEMENTATION

6.1 ACTIVITY STRUCTURE

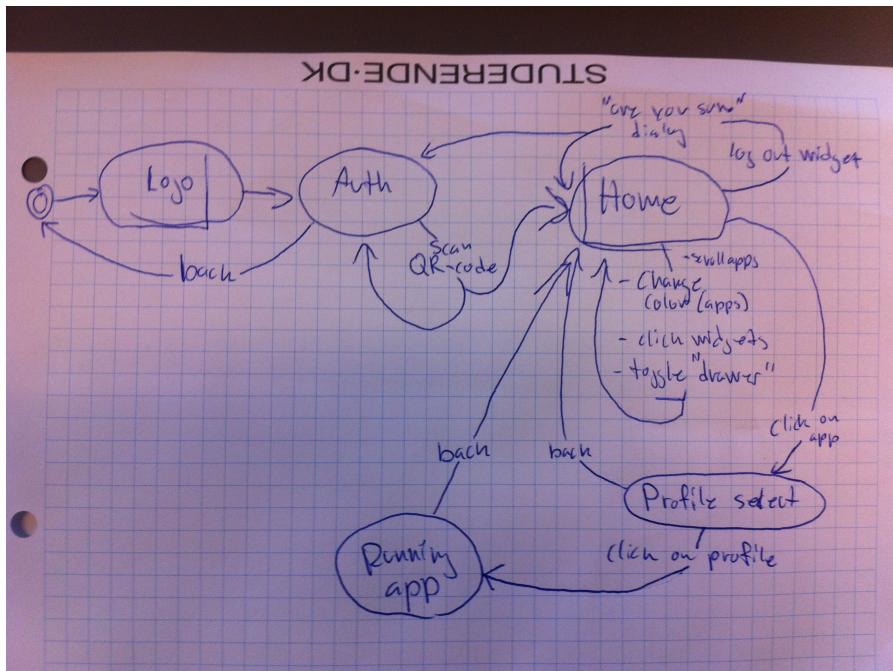


Figure 5: Activity diagram

Figure 5 shows the activity structure ... ¹

6.2 LOGO ACTIVITY

This section will explain the Logo Activity and how it is used.

This activity is used and only seen by the user when the launcher starts up.

The important thing about this activity is that it runs in a `Thread()` and uses `synchronized()` to wait an amount of time until the Authentication Activity or the Home Activity is most certainly done. Which activity that should be started is decided by the `Tools.sessionExpired(mContext)` method. This method returns true or false over if the users login is still valid or not. If the login is still valid and the id which is still logged in, in the `sharedPreferences` do not return null. It will only

¹ FiXme Note: Design - 'internal architecture' skal nok skrives færdig før man kan skrive noget her.

return null if the user do not exsist in the system. The user should be taken to the Home Activity else they should be presented with the Authentication Activity so they can login. When the right Intent is build, the activity is started and the thread is stopped ad finish. All this can be seen in [Listing 1](#) and the final design of the implementation can be seen in [Figure 7](#) and [Figure 8](#) on page [page 35](#). Unfortunately due to errors in the Oasislib the out commented code was not tested.

```

1      mLogoThread = new Thread() {
2          @Override
3          public void run() {
4              try {
5                  synchronized(this) {
6                      wait(Data.TIME_TO_DISPLAY_LOGO);
7                  }
8              } catch(InterruptedException e) {}
9              finally {
10                  Intent intent;
11
12                  if (Tools.sessionExpired(mContext)) {
13                      intent = new Intent(mContext,
14                                         AuthenticationActivity.class);
15                  } else {
16                      intent = new Intent(mContext, HomeActivity.class)
17                                         ;
18
19                      SharedPreferences sharedpreferences =
20                          getSharedPreferences(Data.TIMERKEY, 0);
21                      long guardianID = sharedpreferences.getLong(Data.
22                                         GUARDIANID, -1);
23
24                      /* Following did we not have time to test due to
25                         errors in the Oasislib */
26
27                      //if ((new Helper(mContext)).profilesHelper.
28                      //getProfileById(guardianID) != null) {
29                      //    intent.putExtra(Data.GUARDIANID, guardianID);
30                      //}
31                      //else {
32                      //    intent = new Intent(mContext,
33                      //                        AuthenticationActivity.class);
34                      //}
35
36                  }
37
38                  startActivity(intent);
39                  stop();
40                  finish();
41              }
42          }
43      };
44

```

[Listing 1](#): The Logo Activity and how it uses Thread and synchronized

6.3 AUTHENTICATION ACTIVITY

 Listing 2: This is code

6.4 HOME ACTIVITY

 Listing 3: This is code

6.5 PROFILE SELECT ACTIVITY

This section will explain the Profile Select Activity and how it makes some of the external architecture for others to use.

The final implementation of the design is made so it is simple and easy to use. What can be seen on this screen is the giraf logo and colors to secure consistency and then the children connected to the currently logged in guardian profile.

This can be seen in [Figure 6](#) and in [Figure 9](#) on page [page 37](#).

The external architecture is build using intents which Android have defined as “An intent is an abstract description of an operation to be performed.” [?]. These intents can hold almost any data Java can handle and is easy to use. The `putExtra()` method puts a data type e.g. long in the intent with some kind of tag, which is of the type `String`. This activity puts three extra attributes in the intent before calling `startActivity` for the intent with the selected application in it. These are a `childID` which holds the id of the selected child, `mGuardianID` which holds the current logged in guardians id and the color which the user have choosen for the application icon. All other applications in giraf can look at the intent they were started with and use this information. How the intent is build can be seen in [Listing 4](#).

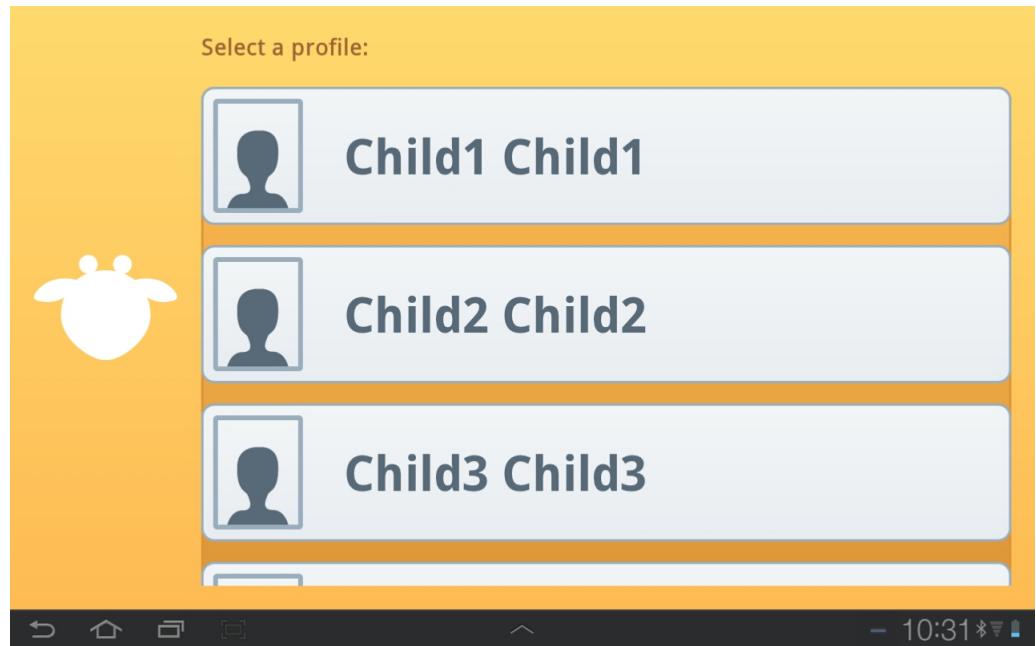


Figure 6: Landscape profile select activity screenshot

```

1  listOfChildren.setOnItemClickListener(new OnItemClickListener() {
2      public void onItemClick(AdapterView<?> parent, View view, int
3          position, long id) {
4          final long childID = ((Profile) parent.getAdapter()).
5              getItem(position)).getId();
6
7          Intent intent = new Intent(Intent.ACTION_MAIN);
8          intent.addCategory(Intent.CATEGORY_LAUNCHER);
9          intent.setComponent(new ComponentName(mPackageName,
10                  mActivityName));
11         intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK
12             | Intent.FLAG_ACTIVITY_RESET_TASK_IF_NEEDED);
13
14         intent.putExtra(Data.CHILDID, childID);
15         intent.putExtra(Data.GUARDIANID, mGuardianID);
16         intent.putExtra(Data.APP_COLOR, mAppColor);
17
18         startActivity(intent);
19     }
20 });

```

Listing 4: How the launcher makes the intent for launching an application

7

TEST AND VERIFICATION

7.1 UNIT TEST

7.2 USABILITY TEST

Part IV
EPILOGUE

8

REFLECTION

8.1 REMARKS TO THE NEXT GROUP OF DEVELOPERS

9

CONCLUSION

Part V
APPENDIX

A

IMPLEMENTATION

A.1 LOGO ACTIVITY

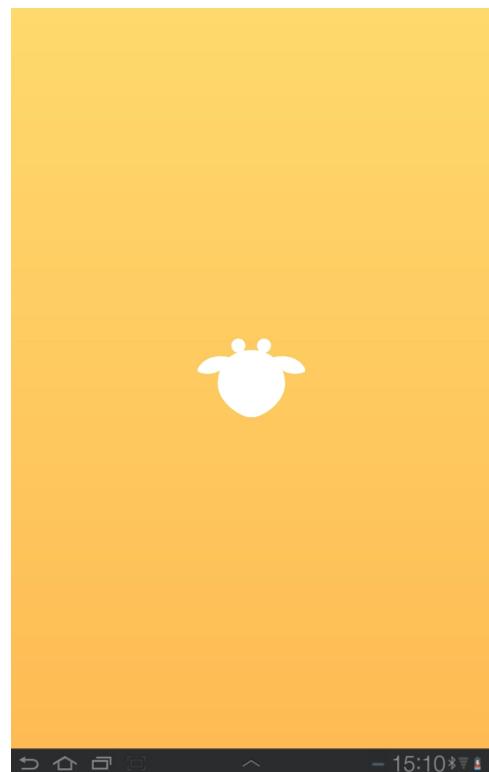


Figure 7: Portrait Logo Activity only shown when user do not have a valid session.

A.2 PROFILE SELECT ACTIVITY

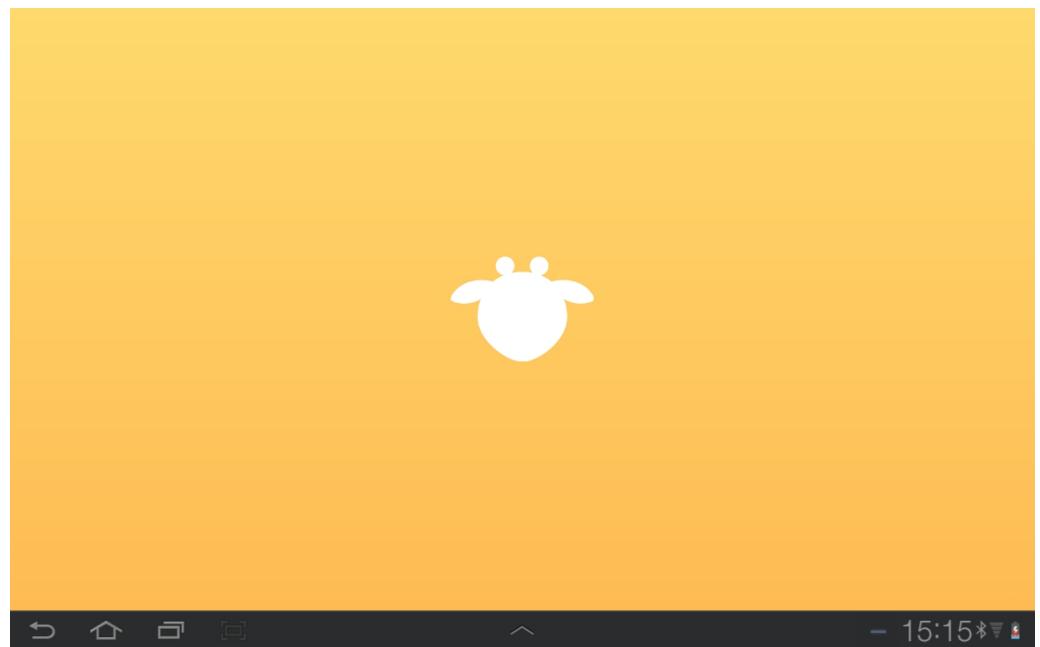


Figure 8: Landscape Logo Activity only shown when user has a valid session in and start the application again.

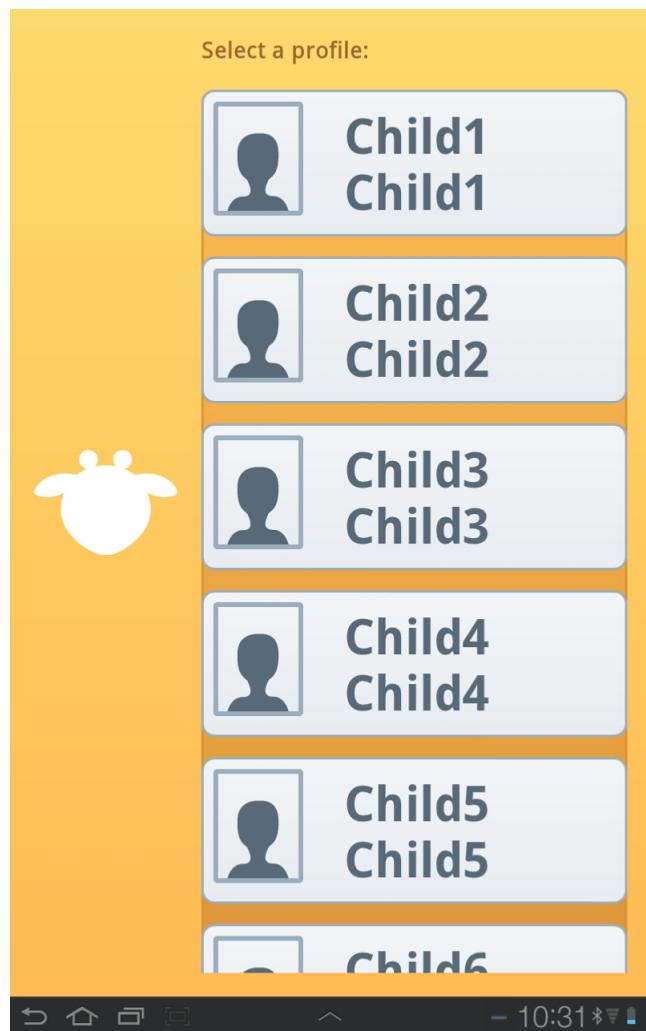


Figure 9: Portrait profile select activity screenshot

