Developing an Agenda Application for Children with ASD

Kim Feenstra Kuiper
s3023303
kimfeenstrakuiper@student.ru.nl

Supervisors:
prof. dr. ir. Peter Desain
Artificial Intelligence, Radboud University Nijmegen

Dr. Franc Grootjen
Artificial Intelligence, Radboud University Nijmegen

Guido Ongena, MSc
R&D, Dr. Leo Kannerhuis

July 26, 2013
Children with autism spectrum disorders can benefit from having careful structure in their lives [Palmen et al., 2012]. Designing the Agenda Coach, a digital personal agenda which provides structure and extra information to the user, is done in 3 steps. Interview questions were built by combining knowledge of ASD [Campbell et al., 1996], [MacDuff et al., 1993] and HCI [Tognazzini, 1992], [Rogers et al., 2011]. The interviews were conducted on ten persons which have experience with children with ASD. Design and functional requirements were abstracted from the results of these interviews. A prototype is built and tested on two children with ASD. This resulted in positive feedback on the usage of the Agenda Coach.
I am deeply grateful to my brother, who gave me the primary inspiration to write this thesis.

Special thanks to my friends, family and supervisors who provided inspiration, motivation and help during the writing of this thesis.
Chapter 1

Introduction

Our world is full of changes. People with an autism spectrum disorder can experience a lot of problems in processing these changes. This thesis describes the research of designing the Agenda Coach, a digital personal agenda which provides structure and extra information to the user.

The first inspiration for choosing this topic, is my brother who has autism (PDD-NOS) and the lack of structure in his daily activities struck me. Even if structure is provided, changes in (for example) dinner time cause him troubles. For other disorders it may be necessary to provide more information about activities. For example: how to pack your bag before going to school. An Agenda Coach can help to accomplish this. Such a digital coach should be able to customize such as to the amount of provided information and notifications, to prevent irritations. The coach should provide the patient all the needed information about daily activities and give notifications when an activity starts or changes.

Early December 2012 the Informa Healthcare archive published an article titled: “A personal digital assistant for improving independent transitioning in adolescents with high-functioning autism spectrum disorder”. This article describes how the first developments in a so-called Agenda Coach (from the Dr. Leo Kannerhuis) are received by ASD patients: four young adults tested the Coach2Care Agenda Coach using a WIFI connection. The utilisation of the Agenda Coach resulted in a significant improvement of independent transitioning between their daily activities at their treatment facility, and medium effect is found for the baseline intervention [Palmen et al., 2012]. Despite the significant improvements, most participants showed irritability while using the Agenda Coach. The reasons for this ranged from problems
with connectivity to recognizing the script of the application. The Agenda Coach has not yet been put into practice because of the low user experience.

The Dr. Leo Kannerhuis is an organization experienced in ASD and would like to have an Agenda Coach which provides a schedule and helps diagnosed adolescents.

The functionality for the new Agenda Coach consists of a calendar on which all activities are displayed clearly and user friendly. The user is notified when an activity starts or changes, and extra information about the activities are shown. The child, parents and/or therapist will have access to see and change the schedule. In this way they know what the patient is doing and help can be provided if needed. How to build such an application has not been studied sufficiently, for example it is unclear if the application should include pictures or the way the user wants to be addressed. The purpose of this thesis is responding to the questions above and in consequence offering additional information to the theory of building more user friendly applications with extended usage.

In this chapter I am going to collect more knowledge about ASD in literature. Secondly I will investigate how new applications are constructed. Goals that have to be taken into account to get a higher user experience will be gathered. Different gathering techniques and test plans will be discussed. Then I am going to summarize some results that have already been discovered in research concerning applications for youth with ASD. Finally I will propose my research questions to build a useful application for children with autism.

Chapter 2 will describes which methods I used to gather the requirements for the application. In chapter 3 I will explain the requirements, and in section 3.3 the built prototype. Chapter 4 will reflect on the conducted tests and their results. Finally the conclusions will be presented in chapter 5.

1.1 Autism Spectrum Disorder

Since 1938, L. Kanner has studied children who acted and responded different from others. These unique responses were recorded by L. Kanner and in 1943 the first patients with ASD where characterized. The behavior of children with ASD were described by their parents as: “self-sufficient”, “like in a shell”, “happiest when left alone”, “acting as if people were not
there”, “perfectly oblivious to everything about him”, “giving the impression of silent wisdom” and “acting almost as if hypnotized”. [Kanner et al., 1943]

The definitions of autism have changed over the years, but the described behavior did not. The definitions changed because of new discoveries during research (Kanner, Asperger, Itard, etc), and cultural and political shifts [Sula Wolff FRCP, 2004].

Some results of researches conclude there is proof that ASD is caused by a development disorder [Abrahams and Geschwind, 2008] or that advanced age of parents (including father and mother) is correlated with an increased ASD risk [Grether et al., 2009].

According to Baio, in the UK 1 out of 110 children has a disorder in the autistic spectrum. Boys have 4 times more often such a disorder compared to girls. Symptoms of autism can get visible around the age of 24 months, but a diagnosis is made at earliest at the age of 54 months [Baio, 2012]. In two widely used diagnostic systems (DSM-IV and The International Classification of Disease-10) the umbrella classification for autism-like disorders is Pervasive Developmental Disorder. This includes Autism Disorder, Asperger Syndrome and Pervasive Developmental Disorder-Not Otherwise Specified (PDD-NOS). Those different types are differentiated by the intensity of the symptoms, age first symptoms, language delay and intellectual level [Lord and Bishop, 2010]. A difference is made between high functioning autism (IQ > 70) and low functioning autism (IQ < 70) [Palmen et al., 2012]. In the rest of this thesis I will refer to Pervasive Developmental Disorder as ASD (Autism Syndrome Disorder) since this is used in literature [Putnam and Chong, 2008].

Diagnosis of ASD is accomplished in a clinical way by recognizing, mostly complex, patterns of behavior. Some studies result in having difficulties defining borderlines of the different groups of autism [Wing, 1996]. Vague borderlines occur because most subgroups have one or more similar symptoms in common with different intensity. Most people have some autistic features; this does not mean they have an autistic disorder. One needs more specific symptoms to be diagnosed with ASD. There is no model-autism, because most diagnosed have a different spectrum of autistic features. This makes it hard to cure or help children with ASD as they all need a personalized treatment [De Bruin and Gelderland, 2004]. A way to help most of these children is to give a structured schedule which explains what they should do at which time. Research has shown that it works properly for children with high functioning autism [Bryan and Gast, 2000] and it sometimes results in a positive way for children with low functioning autism [Pierce and Schreibman, 2009].
1.2 Designing Human Computer Interaction

There are several theories about how to design user friendly software. The following information is gathered from two books with slightly different approaches and suggestions, including their suggested papers. The book ‘Tog on interface’ by Tognazzini is focused on brainstorming, user testing and theoretical aspects [Tognazzini, 1992]. The book ‘Interaction design beyond human computer interaction’ by Rogers et al. gave a very detailed overall view of designing interactive software [Rogers et al., 2011]. Both books refer often to [Nielsen, 1989] since he has done a lot of research in testing software.

The process of creating an interactive design requires by four steps [Rogers et al., 2011]:

1. Specify the needs and requirements
2. Develop a design that meets those requirements
3. Build the interactive design
4. Evaluate the design

There should be interaction between those four steps. When the needs change, or requirements are added, the design should be changed. If the evaluation suggests improvements, those options should be implemented or at least be considered for further research.

P. Heckel and Tognazzini state about step 1, 2 and 3 that one should keep in mind the users limits, problems and solutions [Heckel, 1991], [Tognazzini, 1992].

According to Olson and Moran there are 4 data gathering options to specify the needs and requirements for the audience [Olson and Moran, 1995]:

1. Questionnaires
2. Interviews
3. Focus on groups and workshops
4. Observations

For this application I chose for interviews since this is an exploring research. Benefits are the qualitative results and it will not be as time consuming as workshops or observations. Some drawbacks are the lack of quantitative results and I will only receive the opinion of a selected group. This can be mitigated by carefully selecting a various group of subjects.

There are four types of interviewing the subject [Rogers et al., 2011]:

1. Open-ended (unstructured)
2. Structured
3. Semi-Structured
4. Group interview

A semi-structured questionnaire is chosen for this exploring research, consisting of only open-ended questions. Open-ended questions will exclude any bias of my own vision of the application. The semi-structured questionnaire can show the subject different possibilities. Some people do not know what they need and what is possible until they see it. Tognazzini suggested to make a careful selection of the critical population, and try to get all emotions out of the population. Using these emotions will help you discover what is preferred and needed.

The qualitative results of the interviews should be concluded and result in a list of requirements. “A requirement is a statement about an intended product that specifies what it should do or how it should perform” [Rogers et al., 2011]. Mostly these requirements are shown in a functional document which states what a system should do, and a non-functional document which contains all the constraints there are on the software development.

After describing the requirements the application can be built. We should keep in mind that the application is built for the user. The user application should get a high usability and user experience. Combining the usability and user experience goals of Rogers [Rogers et al., 2011], and the usability goals of Palmer [Palmer, 2002] resulted in the following goals:

- Navigation
- Supportive/Creativity
After building the application it should be tested on usability and experience. A good application could be figured out in ten minutes for his first use [Nelson, 1980]. If this is not the case the behavior of the application is not consistent with the behavior of the user. The best way to test an application is to create a situation and test a prototype for this specific situation. The user should use the application while thinking out loud, the user says everything he wants, does and thinks. [Nielsen, 1989]. We can test horizontally by displaying the full range, and vertically by carrying the user deep into the program. Since this application is new in its kind we will use the vertical test.

Things we should keep in mind during the whole process is to iterate between all stages. We also should keep in touch with the user during the process. Try to test the user as often as possible. Watch out for the three enemies of software engineering: Rationalization (this blocks the vision of the future), Assumption (it makes life dull and cries out for periodic review) and denial (this prevents for improving the software). [Tognazzini, 1992] We should always brainstorm to release our creativity. Even if we have already some ideas about the application this is an important step that should not be forgotten [De Bono, 2010].

1.3 ASD and user experience

Frost and Bondy (1994) developed a Pictogram Exchange Communication System (PECS) which helps children to communicate and understand behavior. Children with ASD can also be helped with a structured schedule. Visualizing the schedule through pictograms or photographs is a method that often shows a positive result [Campbell et al., 1996] [MacDuff et al., 1993].
The visualization can be done on paper or in an application. The result of the experiment of K. Pierce and L. Schreibman is that the usage of icons on paper helps children to successfully manage their behavior in the absence of a treatment provider. Some children learn from those pictures: after a while they manage to do activities without help. On the other hand some other children can not manage behavior without the icons any more [Pierce and Schreibman, 1994].

There is enough proof that using of a visual schedule helps children with ASD doing their daily tasks [Campbell et al., 1996] [MacDuff et al., 1993]. Some parents already use scheduling applications, but applications are not especially developed for people with an autistic disorder. “He would also benefit from software that helped him generate, organize and write his own thought and ideas” [Putnam and Chong, 2008].

I expect that software specifically developed for people with ASD will be more effective than normal software because it will combine visualizing with more methods of treatment. The idea I used is the what, where, who, when, how and why from the book ‘Geef me de 5’ [De Bruin and Gelderland, 2004]. This book describes easily how parents can handle their children with ASD. Giving all the needed information in a structured way will cause less confusion in the head of the child.

1.4 My Research

As read before, children with ASD need a structured schedule explaining how they need to do which activity. Those schedules are at the moment mostly offline and need to be changed by hand when an activity changes. Building a digital agenda can help youth diagnosed with ASD keeping track of their schedule, even if an activity changes over time. Therefore the main question will be:

*Is it possible to build a mobile application that helps children with ASD during the day?*

Where children is in the age group of 10 up to 20 years. Building a mobile application for children with ASD from scratch gives a lot of freedom and choices to be met with. One should keep in mind the user when designing the requirements. Children with ASD have a different perception compared to children without this disorder. Therefore need to be find out what functional
aspects they need, and what the design should look like. This results in two subsequent questions of the main question:

Which functional aspects specified for children with ASD should be taken into account?

Which non-functional aspects specified for children with ASD should be taken into account?

Last but not least need the different ways to implement this mobile application to be considered. Currently there are many different technologies that can be used to build a mobile application and they all have their own benefits and drawbacks. Therefore the last sub question is:

Which technologies should be used to implement this Agenda Coach?

To answer these questions I will combine the knowledge of section 1.1 and apply the guidelines section 1.2 in interviews. During this interviews will be checked if the knowledge gathered in section 1.3 is still correct and if it should be updated. You can read the set up and results of the interviews in chapter 2.
Chapter 2
Gathering the Requirements

2.1 Method

To give the user a high user experience, I researched the requirements to support children diagnosed with ASD in planning. Discovering these needs for specifying the requirements is done by conducting semi-structured interviews. The semi-structured approach was chosen because this is an exploratory research. During these interviews I tried to create a brainstorm session to gain new information and innovative ideas. The usability goals [Rogers et al., 2011] were used as inspiration to formulate questions. In figure 2.1 you can find which other literature is used to develop the questions. With the interview I have isolated the functions which I think are preferred by children with ASD. The questions can be found in appendix C. These functions will be processed in the requirements. The requirement will meet the usability goals and user experience for these youngsters. The results of this research are the requirements for an Agenda Coach application specially designed for children with ASD.

The interviews were conducted with ten subjects. This number was chosen because of the amount of expected innovative information the respondents would give. The only criteria for the selection of the subjects was having a close connection to youth diagnosed with ASD.

The interviewed consisted of 3 groups: teachers and treatment providers, parents of children with autism and adolescents with autism. The five specialists on ASD consisted of one teacher in primary special education, one teacher in secondary special education, one graduating educational science
student who focuses on youth with diagnosed ASD and two adults who have various work experiences with people with this diagnosis. The parents group included three subjects with children having ASD, one parent of an 11-year-old boy with Asperger, one parent of a 15-year-old girl with Asperger and one parent of a 12-year-old girl with PDD-NOS. The last group contained two university students with ASD, a 19-year-old boy with PDD-NOS and a 21-year-old girl with Asperger. Those subjects were found by connections through friends and family.

Eight interviews were conducted face to face, one with the use of online calling, one using the chat function, both via Skype. All spoken interviews were recorded so I could focus on conducting the interview. After all the interviews were done they were transcribed and summarized in one document, sorted by subject.
2.2 Results

The combined answers of the subjects gave suspected and unsuspected results. Most subjects agreed with some specific ideas, while for different ideas all disagreed. This occurred even though they had no idea of each others opinion and vision. Below the results of the interview are described per subject. Between the parentheses the amount of subjects that agreed with the requirement are listed.

Youth and children with ASD often have problems giving structure to their work (9). You can see this by the lack of structured behavior and doubt. The solution most respondents have is to visualize the planning in a scheme or calendar (9). One subject noticed problems keeping track of the planned schedule. If there is a clear planning those kids will know what is expected from them, and how they should perform these tasks. This structure results in less stress.

Most subjects think that they, and the children, will benefit from the app if it is used (9). One subject doubted if the application would be used. Some subjects point out that the application will be only used after familiarization (3). The subjects suspect to benefit from the application for several personal reasons:

- Trust that your child knows what to do when parents are not at home.
- No new planning must be printed when a planning changes.
- Let the child make a conscious choice whether to ignore a planned activity

With only a small amount of my vision on the application, the respondents were asked which functionality they would like to have it. These are the first reactions of the subjects, so the number of similar answers is low. Three subjects asked specifically for a detailed day schedule with the possibility of a weekly overview. Two teachers asked for showing the connection between the exams. Three subjects asked for a detailed task analysis (how the task should be completed). Half of the subjects mentioned that the calendar should be easy to be adjusted, copied or shared. Two subjects would like to have feedback during the activities to check if the child is still doing his task. Three people suggested to use visual graphic images, smileys or adding your own icons. Three subjects preferred a notification at the beginning and
ending of an activity. Two people would like to have a timer, telling them how long the activity would last. One subject preferred a function with which a child could choose an activity during his or her free time.

The subjects gave their suggestions for the lay-out of the application. Eight subjects suggested using a variety of colors ranging from calm tones to bright colors. Some of them gave the suggestion to make it personal by using themes (3). One teacher pointed out that her school uses different colors for different days. The subjects were asked to give their opinion upon using icons. Most of them agreed with using icons, though some told me they had no influence on the behavior of the child. Icons could be distracting, but also clarifying. If icons have to be used more, subjects prefer graphic images over photographs (6). Some teachers suggested that a picture of the specific teacher is shown during class (2). Two other subjects agreed with showing a picture of friends and family while having an activity with them. Adding your own icon is only presented by one subject, two other subjects did not have any remarks on this element. All the subjects suggested the application should be easy, simple, clearly and predictable. One subject suggested it had to look and feel like the already existing 'School Coach'.

The content of the application should be simple, clear and to the point, to get the least distraction. Most subjects thought it is a good idea to use the 'Geef me de 5'-approach (supply information about what, how, when, where and with who the activity is) but pointed out to keep the interface clean (8). Brainstorming with some subjects resulted in the solution to show this approach only when asked (by click)(4). The task description of the activity should be easy to adapt, since every child has different needs. Some subjects worried they had to do a lot of work for this task description, and all gave three different solutions: make the child choose between three different amounts of information, make it easy to copy task descriptions, make all task-descriptions ever made accessible on a website. One subject pointed out to avoid scrolling: everything has to be visible immediately.

Seven subjects stated clearly that there should be a notification when an activity ends and a new one starts. Suggestions for the time frame of the notification were either using a percentile to the time frame of the activity or making it personally adjustable. The notification alarm should be easily adjustable to a pop-up, different sounds or vibration. A teacher implied that she did not prefer sound or buzz notifications during class hours. When an activity is changed there will be chaos in the brain of the child diagnosed with ASD. To minimize this chaos there should be a clear reason for the
change told to the user (4) and there should be a (option to a) replacement (2).

No subject reacted positively to the question if the application should contain a rewarding system. Three subjects stated a reward should be given by the environment. Three subjects thought that a rewarding system including coins and new avatars where approved but not needed. One subject pointed out children may get sad or angry when they do not receive possible positive feedback (like coins). Four subjects informed it would be useful to give feedback if an activity is successfully finished. This can be shown by a green check mark. If an activity was not successful it is the best way to give it a neutral sign, this is suggested by one specialist.

On inserting checklists seven subjects responded negatively and three subjects neutral. Although a checklist shows clearly what is finished, most subjects do not see an added value using this option in the app. Two subjects suggested to make it optional. One subject imagines a pop-up of the next item on the checklist when the current item is finished.

Every subject suggested the application should include specific optional functions. This probably occurred because the ASD spectrum is very wide so no effect is the same. To make more functions optional a wider public can be reached. The functions proposed to be adjustable are: amount of information in the task-description, (no) graphic image, color of lay-out, (no) pop-up, type, (no) checklist, number of choices, own icons, hearing the upcoming activity, time frame of day part and sound of notification.

To summarize the above:

The following options should be able to personalized.

- An amount of information in the task-description
- (No) graphic image
- Color of the lay-out
- (No) pop-up
- Type
- Number of choices
- Usage of own icons
I visualize the planning in a scheme or calendar
I think I would benefit from using the application
The application will only be used after familiarization
I would like to see a detailed day schedule with the possibility of a weekly overview
I would like to give the child a detailed task description
I prefer pictograms over photographs
The application should be easy to be adjusted, copied or shared
The application should be easy, simple, clearly and predictable
Supply information when, what, where, how and with who the activity should be performed is a good idea
The child should get a notification when an activity ends and a new one starts
When an activity changes the child should get a notification with a reason
Inserting a checklist does not give an added value

<table>
<thead>
<tr>
<th>Statement</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>I visualize the planning in a scheme or calendar</td>
<td>9</td>
</tr>
<tr>
<td>I think I would benefit from using the application</td>
<td>9</td>
</tr>
<tr>
<td>The application will only be used after familiarization</td>
<td>3</td>
</tr>
<tr>
<td>I would like to see a detailed day schedule with the possibility of a weekly overview</td>
<td>3</td>
</tr>
<tr>
<td>I would like to give the child a detailed task description</td>
<td>3</td>
</tr>
<tr>
<td>I prefer pictograms over photographs</td>
<td>6</td>
</tr>
<tr>
<td>The application should be easy to be adjusted, copied or shared</td>
<td>5</td>
</tr>
<tr>
<td>The application should be easy, simple, clearly and predictable</td>
<td>10</td>
</tr>
<tr>
<td>Supply information when, what, where, how and with who the activity should be performed is a good idea</td>
<td>8</td>
</tr>
<tr>
<td>The child should get a notification when an activity ends and a new one starts</td>
<td>7</td>
</tr>
<tr>
<td>When an activity changes the child should get a notification with a reason</td>
<td>4</td>
</tr>
<tr>
<td>Inserting a checklist does not give an added value</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 2.2: Results of the interviews. N is the number of respondents who agreed to the statement
• Hear the upcoming activity
• Time frame
• Sound of notification
Chapter 3

The Application

Conducting ten interviews with experts in the field of ASD gave me insight into what kind of application is needed. The documentation of this application is given in two parts. The functional documentation will give insight into which functionality and design requirements the application should have. The technical documentation describes the so called ‘back-end’ of the application, the architecture and structure is explained here. Section 3.3 describes how the prototype is built.

3.1 Functional document

While conducting the interviews the general outline of the application became more clear. As can be read in section 2.2 some requirements are more preferable than others. This is reflected in the functional document using the MoSCoW approach. A distinction is made between functional, usability/design and reliability requirements. Used cases are added to clarify the needs and prevent misinterpretations. The full documentation can be found in the appendix chapter A and can be read as an independent document.

1The index of both technical and functional documents are based on the templates from the course Software Engineering. Since we built an application with a roughly comparable back-end some information will be the same as in the Software Engineering project. For more information: http://www.giphousescience.ru.nl/
3.2 Technical document

The technical document can be found in appendix chapter B. This document contains the back-end of the application. The back-end is built up by using the knowledge of software engineering combined with the desired functionality of the application.

3.3 Prototype

With more time and programming skills I could have built all requirements, but the most important were accomplished. After considering which requirements would be useful for implementation to test the prototype, a restriction was decided upon to implement only the following ones for the prototype:

1. All upcoming activities are shown.
2. Extra information of an activity is available by clicking on the activity.
3. The starting time of all activities is shown.
4. The pictograph of all activities is shown.
5. The time spent and still left for an activity is shown through a filling progress bar.
6. When an activity ends a notification is given to the user.

The application is programmed in Java, most programmers know this language and unexperienced engineers can easily understand the written code. This results in a prototype with code which is easy to understand for computer experts. Java in combination with XML in Eclipse is well supported and gives the ability to program mobile applications. Running and testing the software was immediately done on a Samsung Galaxy Tab 2 10.1 with Android 4.0.3.

I decided to program requirement 1 by using a vertical scrolling list. Using a list view makes sure the upcoming events are shown and the amount of shown tasks depends only the screen size. Every element consists of a progress bar, a starting time, a name and a picture of the activity, see figure D.1a.
To fulfill requirement 2, I chose to change the list view into an extendable list view. An extendable list view consists of group views like a list view, but added child views which can be individually expanded. See figure 3.1b. The view is now a vertical scrolling list of two levels. The child views are filled with extra information of the activity shown in the parent view. To open the extra information you can easily click on the parent view. For closing, one can click on the parent node again. An extra feature which was not a requirement is a limit of opening a maximum of one child view at the time: If the extra information of activity one is opened and you click on activity two, the extra information of activity one is automatically closed and the child of activity two is opened.

The progress bars in requirement 5 are all updated when the percentage is changed in the associated thread. The progress is calculated by the time already passed divided by the total time of the activity. The minimum value of the progress bar is 0, the maximum is 100, although the progress can be negative or bigger than 100. Progress bars sometimes give unexpected results (filled while there is no progress). This is probably caused by the fact that every progress bar has its own thread, since the issue only occurred when I
inserted more than 4 progress bars. A solution could be to delete the progress bar if the progress has reached 100. To succeed in deleting an object in the extendable list view there should be a GUI-thread in the main view.

Requirement 6 is difficult to implement since it needs a (temporary) change in the interface of the application. To implement a change you need a GUI thread in the main view. A problem could occur when a thread of the view awakes, since the threads of the progress bars need to keep running. This can be solved by pause the progress bar-threads. This solution is not used in the prototype, unfortunately meaning that requirement 6 is not implemented.
Chapter 4

Testing

4.1 Method

The application is tested one day on two male subjects with ages 12 and 15 year with high functioning ASD. Both subjects are male. Subject 1 is 12 years-old and is diagnosed with classic Autism. Subject 2 is 15 years-old and is diagnosed with PDD-NOS. Subject 1 is found through friends, subject 2 is related family. These subjects are selected because their parents agreed with a one day testing of the application. Their parents e-mailed me a personal schedule for the subjects for one day. I implemented this schedule in the prototype and sent them the application via e-mail. The subjects were instructed in advance that they had to use the application on Friday. The only given instruction for the application was the explanation of the possibility to click on activity. Parents did not give any further instructions. If there were questions or hesitations they had the possibility to contact me. None of the subjects contacted me during the experiment. Both subjects used the application on a 10.1 inch tablet with 4.0.3. Android.

All tasks were implemented in the application and the progress bar filled while time was passing. Subject 1 had eight activities consisting of: waking up, breakfast, gaming, learning French, lunch, reading, gaming and filling in a questionnaire. The activities where scheduled from 9:30 to 16:15. Subject 2 had twelve activities scheduled from 10.00 until 19.00 namely: waking up, breakfast, gaming, cleaning, lunch, gaming, packing his bags, grocery shopping, gaming, spare time, dinner and filling in a questionnaire.

After using the application they had to fill in a questionnaire on the com-
puter. The parents had to fill in a different questionnaire. The questionnaires for the children consisted of nine scale (10-1) questions, eleven open questions and four multiple choice questions. The questionnaires for the parents had four open questions.

4.2 Results

The results are derived from the filled-in questionnaires of the children and their parents. The results are inconclusive but give a strong indication that the usage of the Agenda Coach provides structure and help.

Both subjects enjoyed (scored: 8) using the application and thought it was helpful (scored: 7) during the day. Both parents noticed a motivated use of the Agenda Coach in the beginning of the day. Every task was conducted in the given time frame. Later that day the subjects became more idle with the time frame and carrying out activities.

Subject 1 wanted to have a digital clock with the current time. Both subjects wanted to see in digital time how much time was left for the momentary activity. Subject 2 preferred to see only one progress bar, namely the currently running one. He liked the clarity of the pictograms and starting times. He also expressed positive feelings about the extendable list view, it is clear for him how to use the application. Both parents found it a clear and obvious application.

Not all activities of subject 1 did start on the set time, his reason was he enjoyed the last activity and he did not enjoy the upcoming one. Subject 2 did not accomplish all activities for the same reason. Subject 2 and both parents give a possible solution in the form of a notification for an activity transition. In this case a subject has an extra tangible reminder. Subject 1 did not want to have notifications.

Extra information on how to perform an activity was positive rated by subject 1 (scored: 8). The information who (scored: 6) to perform the activity with was not needed in his opinion. Where (scored: 7) to perform the activity was a good addition to the activity. Subject 2 found the information on how (scored: 7) to perform an activity too detailed. Who (scored: 6) the activity is to be performed with did not interest him. The information where (scored: 8) to perform the activity was useful.

Subject 1 might use the application every day. Even though he thinks it is a useful application, he does not want to plan every day this detailed. His
parents thought the application is probably going to be useful for homework. Subject 2 wants to use the application for school tasks or really busy days. The parent of subject two raised the idea of making the schedule together with subject 2, so the subject knows what will happen during the day. After longer use the parent expects more familiarity with the application and thought this would give better results.

The results per subject can be found in table 4.2. These results are iterative processed in the functional and technical document. This is done because there should be interaction between the four design steps of the application [Rogers et al., 2011], as described in section 1.2.

<table>
<thead>
<tr>
<th>Statement</th>
<th>Subject 1</th>
<th>Subject 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overall impression</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>Lay out</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Usefulness of the pictures</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Usefulness of the time line</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Would like to receive a notification</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Usefulness of the extra information how to perform</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>Usefulness of the extra information with who to perform</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Usefulness of the extra information where to perform</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>Started all activities in time</td>
<td>no</td>
<td>yes</td>
</tr>
<tr>
<td>Finished all activities</td>
<td>yes</td>
<td>no</td>
</tr>
<tr>
<td>Enjoyment of using the Agenda Coach</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Usefulness of the Agenda Coach</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 4.2: Test results per subject
Chapter 5

Conclusions

Since it is difficult for parents to provide an up-to-date schedule for children with ASD aged between 10 and 20 years I designed an Agenda Coach. This Agenda Coach should provide a day schedule for the child with all the necessary information easily shown. This agenda should be easily adjustable by the parents. In the process of designing the application I uncovered which functional and design requirements were preferred. Those requirements were abstracted from ten semi-structured interviews with experts on the field of ASD. Knowing these requirements a way was found to implement these requirements. The so-called ‘back-end’ of the application needed to be designed. Combining these requirements together resulted in a small prototype, which was tested on two children with ASD.

Which functional aspects specified for children with ASD should be taken into account?

The functionality the experts suggested varied from vague ideas to small detailed functionality. The functions which were named the most were where:

- Notifications when an activity is ending, the child needs to prepare himself for the transition to another activity.

- A clear visualization of the time left for this activity.

- Usage of pictograms for the activities (but only when the user prefers to use them).

- A clear and obvious way to provide extra information for activities.
All functional requirements can be found in section [A.4.1]

Which non-functional aspects specified for children with ASD should be taken into account?

The experts opinion on the design requirements were consistent. The requirements suggested by almost all subjects were: a clear and simple interface, the user should feel attracted to the application, the application should have predictable behavior. It is recommended that colors, sounds and amount of information can be adapted for personal use. The last requirement will make the application appealing to the users. By adapting the amount of information the user will get the feeling the application can help him. The option to play sound with notifications should be personalized because some children with ASD are highly sensitive, sounds can distract or annoy them. An overview of all design requirements can be found in section [A.4.2].

Which technologies should be used to implement this Agenda Coach?

The back-end of the application should be adapted to the functional and design requirements. The application should function on IOS and Android systems, both for different screen sizes on mobile phones and tablets. A framework which can be used for multiple systems is PhoneGap. The parents should be able to adjust the schedule of the child, both should have access to the same schedule. The child receives a notification when a parent changes the schedule. This is done through a web server connection. When there is an internet connection, the schedule is saved in the database. If there is no internet connection available the schedule is called from the database.

Is it possible to build a mobile application that helps children with ASD during the day?

The prototype of the Agenda Coach was tested on 2 boys with ASD (PDD-NOS and Classical Autism). It is hard to test such an application on children with low functioning ASD, because they are used to their own structure. Providing this structure via an application is a transition they need to be prepared for. Although the results are not conclusive they show positive results. Both children enjoyed using the application and noticed useful aspects. Both boys, however, thought the application was too detailed but useful for particularly busy days. The parents also thought the application was useful especially for school work or active days. The answer to the main research question is affirmative.
5.1 Discussion

Prior studies noted the importance of scheduling [Campbell et al., 1996] [MacDuff et al., 1993] for children with ASD. Some parents reported there should be more software for persons with ASD specifically [Putnam and Chong, 2008]. Combining those two studies together in building an application with a working approach to autism [De Bruin and Gelderland, 2004] seems an easy way to positive results. Though it is important to create useful requirements for this users for in the future.

I think this result is gathered in the right way because it generated some bright and innovating ideas. Interviewing ten persons was a sufficient amount of subjects because the level of innovative ideas descended after six or seven interviews. The last interviewed person gave nothing but positive confirmation to the already given ideas. I found a conflict between the read literature and real life with the usage of pictograms [Campbell et al., 1996] [MacDuff et al., 1993]. Parents and treatment providers use pictograms because literature implies a positive result, though there are more parents who does not notice this positive result.

To test the effectiveness of an application it should be used for a longer time. Testing an equivalent application [Palmen et al., 2012] is done with four participants for at least two months. The test subjects, children with ASD, needed to make transition when testing the Agenda Coach. Not every child with ASD can make this transition from no schedule or an offline schedule to a schedule on a tablet or mobile phone. Therefore it is decided to test this application only on children with high functioning ASD and parents agreed in participating. It was preferred to test this application on at least five subjects, but due to problems with finding subjects I settled for fewer participants. To measure the amount of assistance of the application, it should be tested for a longer period of time and with more subjects to get meaningful results.

5.2 Future research

The results of this research are that a lot of parents and treatment providers would like to have an application that can provide structure. For future research I suggest an improvement of the prototype or a full development of the application. With this application at least six children with various
forms on the Autism spectrum should test the application for a longer time period. After this test one can observe whether this application is helpful for children with ASD. If needed, some requirements of the application can be adapted.
Appendix A

Functional Design

A.1 Project description

A.1.1 About the project

The application has the goal to help children with diagnosed autism spectrum disorder during their daily activities. This application will give them the information during the activities and structure they need. To be more specific, it is possible to enter the ritual of waking up. The application will tell the child exactly what to do, for example how to brush your teeth, if they would like. The application will be a phone application, the parents and children can see and adapt the planning every second of the day. The child will receive a notification if a parent changes the schedule.

A.1.2 Starting point

Since there is no application available with any calendar interfaces at the Dr. Leo Kannerhuis, it will be started from scratch.

A.2 Project requirements

Parts

GUI The graphical user interface shows the upcoming activities, and when asked a description of these activities. The interface should be simple, clean
and predictable. To attract the subjects they can choose a color scheme for
the interface.

**Offline storage and synchronization** Because we can not guarantee that
the mobile always has an internet connection, the application must still be
usable when there is no internet connection. Offline storage of the upcoming
event and their description must be available. When there is internet connec-
tion and new activities are inserted, they have to be automatically updated
on the mobile device of the child.

**Stakeholders**

The primary stakeholder is the Dr. Leo Kannerhuis center. They would like
to develop new applications that provide help that fit the needs of children
with autism. They already have several applications that provide help in
public transport and school. These two coaches are part of the Coach2Care,
a mobile digital coach.

A different stakeholder is the group of children with an autistic spectrum
disorder. They need structure and sometimes help with different activities,
this Agenda Coach application will provide this information. Since this is
an mobile application they can aces their schedule when ever and where ever
they want. Every child with ASD has different needs, the application has
the possibility to adapt to those different needs. The application should be
appealing by the users.

Parents and treatment provides are also stakeholders. Their work and
effort should be enlightened by the usage of the Agenda Coach application.
The application will provide information instead of the parents. Although,
the parents and treatment providers should be able to insert activities and
extra information.

**A.2.1 Environment**

The application must be able to run on android and IOS, mobile phones and
tables. The application will be distributed using an app store.
A.2.2 Software

Building this application will be most easy by the usage of software that can run on IOS and Android system. One possibility is to program in the open source framework PhoneGap. This combines HTML, CSS and JavaScript and compiles without SDK’s, compilers and hardware. PhoneGap is free, for downloading see [http://phonegap.com/](http://phonegap.com/) Though the programmer is free to find his own solution to make the application available for multiple systems.

A.3 Use cases

Use case 1, Retrieving information from web server

<table>
<thead>
<tr>
<th>Use Case</th>
<th>01 Retrieve information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Information retrieval to the mobile application from the server</td>
</tr>
<tr>
<td>Actors</td>
<td>Children, parents an treatment providers</td>
</tr>
<tr>
<td>Precondition</td>
<td>- There must be an internet connection.</td>
</tr>
<tr>
<td></td>
<td>- The user has to be logged in.</td>
</tr>
<tr>
<td>Post condition</td>
<td>New information is shown in the application</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user want to see the schedule</td>
</tr>
<tr>
<td>Basic course of events</td>
<td>- The user has the application opened</td>
</tr>
<tr>
<td></td>
<td>- The application indicates connection to the internet</td>
</tr>
<tr>
<td></td>
<td>- The application synchronizes information</td>
</tr>
</tbody>
</table>

Use case 2, view day schedule

<table>
<thead>
<tr>
<th>Use Case</th>
<th>02 View schedule</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Upcoming activities are shown in the display</td>
</tr>
<tr>
<td>Actors</td>
<td>Children, parents an treatment providers</td>
</tr>
<tr>
<td>Precondition</td>
<td>The child must be logged in</td>
</tr>
<tr>
<td>Post condition</td>
<td>Upcoming activities are shown in the application</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user wants to see the schedule</td>
</tr>
<tr>
<td>Basic course of events</td>
<td>- The user has the application opened</td>
</tr>
<tr>
<td></td>
<td>- The application shows the upcoming activities</td>
</tr>
</tbody>
</table>
Use case 3, View description of activity

<table>
<thead>
<tr>
<th>Use Case</th>
<th>03 Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Extra information about an activity is shown</td>
</tr>
<tr>
<td>Actors</td>
<td>Children</td>
</tr>
</tbody>
</table>
| Precondition | - The child must be logged in  
               - The upcoming activities must be visible |
| Post condition | Description of the upcoming activity is shown |
| Trigger | The user needs extra information of the upcoming activity |
| Basic course of events | - The user has the application opened  
                            - The application shows the upcoming activities  
                            - The user touches an activity from which they prefer more information  
                            - In the application unfolds the touched activity, and the description is shown |

Use case 4, Personalize by changing settings

<table>
<thead>
<tr>
<th>Use Case</th>
<th>04 Personalize</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Change settings to fit the needs of the child</td>
</tr>
<tr>
<td>Actors</td>
<td>Children, parents an treatment providers</td>
</tr>
<tr>
<td>Precondition</td>
<td>The child must be logged in.</td>
</tr>
<tr>
<td>Post condition</td>
<td>The settings are visible and ready to be changed</td>
</tr>
<tr>
<td>Trigger</td>
<td>The user wants to change the settings.</td>
</tr>
</tbody>
</table>
| Basic course of events | - The user has the application opened  
                           - The menu button is pressed  
                           - The menu is opened  
                           - The ‘settings’ button is pressed  
                           - The settings are shown with example how it is, and how it could be  
                           - Settings can be changed |

Use case 5, Inserting a new activity

<table>
<thead>
<tr>
<th>Use Case</th>
<th>05 New activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Add a new activity in a time frame</td>
</tr>
</tbody>
</table>
### Use case 6, Copying a day

<table>
<thead>
<tr>
<th>Use Case</th>
<th>06 Copy a day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Copy all activities of a day to a new day</td>
</tr>
<tr>
<td>Actors</td>
<td>Children, parents and treatment providers</td>
</tr>
</tbody>
</table>
| Precondition | - The child must be logged in  
- The ‘Change schedule’ button is pressed in the menu option  
- The day view is opened |
| Post condition | The activities of the selected day are copied to a new day |
| Trigger | The user wants to copy all activities of a day |
| Basic course of events | - The user selects a day by pressing the name of it  
- A new menu is opened  
- The user selects ‘copy all activities of this day’  
- The user selects an other day by pressing the name of it  
- A new menu is opened  
- A time frame is selected  
- The user selects ‘Paste all activities’  
- All the activities are copied |
Use case 6.1, Copying an activity

<table>
<thead>
<tr>
<th>Use Case</th>
<th>06.1 Copy an activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Copy an activity</td>
</tr>
<tr>
<td>Actors</td>
<td>Children, parents and treatment providers</td>
</tr>
</tbody>
</table>
| Precondition| - The user must be logged in  
|            | - The ‘Change schedule’ button is pressed in the menu option  
|            | - The day view is opened |
| Post condition| An selected activity is copied |
| Trigger    | The user wants to copy an activity |
| Basic course of events| - The user selects an activity by pressing it  
|            | - A new menu is opened  
|            | - The user selects ‘copy activity’  
|            | - The user selects a new time frame by selecting it  
|            | - A new menu is opened  
|            | - The user selects ‘Paste activity’  
|            | - The activity is pasted |

Use case 7, Delete a day

<table>
<thead>
<tr>
<th>Use Case</th>
<th>07.1 Delete a day</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description</td>
<td>Delete a selected day</td>
</tr>
<tr>
<td>Actors</td>
<td>Children, parents and treatment providers</td>
</tr>
</tbody>
</table>
| Precondition| - The user must be logged in  
|            | - The ‘Change schedule’ button is pressed in the menu option  
|            | - The day view is opened. |
| Post condition| The activities on the selected day are deleted |
| Trigger    | The user wants to delete all activities during a day |
| Basic course of events| - The user selects a day by pressing the name of it  
|            | - A new menu is opened  
|            | - The user selects ‘delete all activities today’  
|            | - All activities during this day are deleted  
|            | - The application asks if he should notify the child |

Use case 7.1, Delete an activity
<table>
<thead>
<tr>
<th>Use Case</th>
<th>07.1 Delete activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Delete a selected activity</td>
</tr>
<tr>
<td><strong>Actors</strong></td>
<td>Children, parents and treatment providers</td>
</tr>
</tbody>
</table>
| **Precondition** | - The user must be logged in  
- The ‘Change schedule’ button is pressed in the menu option  
- The day view is opened. |
| **Post condition** | The selected activity is deleted |
| **Trigger** | The user wants to delete an activity |
| **Basic course of events** | - The user selects an activity by pressing it  
- A new menu is opened  
- The user selects ‘delete activity’  
- The activity is deleted  
- The application asks if he should notify the child |

**Use case 8, Change an activity**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>08 Change an activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Description</strong></td>
<td>Make a change in an activity</td>
</tr>
<tr>
<td><strong>Actors</strong></td>
<td>Children, parents, treatment providers</td>
</tr>
</tbody>
</table>
| **Precondition** | - The user must be logged in  
- The ‘Change schedule’ button is pressed in the menu option  
- The day view is opened |
| **Post condition** | The user has changed an activity |
| **Trigger** | The user has to change the information of the activity |
| **Basic course of events** | - The user selects an activity by pressing it  
- A new menu is opened  
- The user selects ‘Change activity’  
- The input activity menu is opened  
- The user can change the given input of the activity  
- The user had the option to fill in why the activity is changed  
- The user can choose no notify the child of this change  
- When clicked ‘save’ the application is saved |

**Use case 9, Find a free time activity**

<table>
<thead>
<tr>
<th>Use Case</th>
<th>09 Free time activity</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Case</strong></td>
<td>09 Free time activity</td>
</tr>
</tbody>
</table>

41
Description: Find an activity to do during free time

Actors: Children

Precondition: The user must be logged in

Post condition: The user has chosen an activity

Trigger: The user wants fill spare time with an activity

Basic course of events:
- The user presses the menu button
- The menu is opened
- The user selects ‘Free time’
- A new window is opened and asks how many spare time the user has
- The user answers
- The applications asks if the user is alone.
- The user answers
- The application asks if the user wants to go outside or stay inside
- The user answers
- The application show all possible activities that can be done
- The user chooses an activity

---

A.4 Requirements

A.4.1 Functional requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
<th>MoS-CoW</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>The user is able to log in to the application</td>
<td>M</td>
<td>Secure access for all users to schedule.</td>
</tr>
<tr>
<td>F2</td>
<td>The user is able to insert a new activity and their extra information</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>F3</td>
<td>The user is able to save an activity and their extra information</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>F4</td>
<td>The user is able to insert a saved activity and their extra information which can be chosen from the ‘saved list’</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Description</td>
<td>Setting</td>
<td>Notes</td>
</tr>
<tr>
<td>---</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------</td>
<td>------------------------------------</td>
</tr>
<tr>
<td>F5</td>
<td>The user is able to copy an existing activity or day</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>F6</td>
<td>It is possible to retrieve more information of the activity by clicking on the activity.</td>
<td>M</td>
<td>Provide information to the user</td>
</tr>
<tr>
<td>F7</td>
<td>A notification is given a certain time (around 5% of the total time) before start of a new activity</td>
<td>M</td>
<td>This reminds the user for the ending of the current activity and the upcoming activity.</td>
</tr>
<tr>
<td>F8</td>
<td>Though a pop up is asked if everything goes as planned during an activity which requisite focus</td>
<td>C</td>
<td>This question makes sure the child stays focused on his work.</td>
</tr>
<tr>
<td>F9</td>
<td>A feedback question is asked after the activity, asking if he finished the activity</td>
<td>S</td>
<td>This question makes sure the user stopped doing the activity and provides feedback to the children it sends and the parents.</td>
</tr>
<tr>
<td>F10</td>
<td>The user can select an activity during spare time through answering questions</td>
<td>S</td>
<td>This option helps the child to fill free time.</td>
</tr>
<tr>
<td>F11</td>
<td>The user should be able to change the following settings: (No) pictures Color scheme Notifications given by: buzz / sound / no notifications Sound</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>M</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>F12</td>
<td>If the user changes an activity is asked if the child should be notified, also is asked why the activity is changed</td>
<td>S</td>
<td></td>
</tr>
<tr>
<td>F13</td>
<td>If requested the user gets a notification if an activity is changed, and why it is changed</td>
<td>S</td>
<td>Update the user of a changing program</td>
</tr>
<tr>
<td>F14</td>
<td>It is possible to import activities from Google calendar</td>
<td>C</td>
<td>Extra information is not yet available but could be inserted by changing the activity</td>
</tr>
</tbody>
</table>
While having internet connection, the application checks automatically if the schedule is still up to date.

If there is no internet connection available, the last seen schedule is shown.

This provides the user always a structure.

### A.4.2 Usability/design requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
<th>MoS-CoW</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>The interface should be clean and simple</td>
<td>M</td>
<td>Provide the schedule to the user</td>
</tr>
<tr>
<td>U2</td>
<td>The next upcoming events are shown in a list, with the present activity on top</td>
<td>M</td>
<td>Provide extra information to the user when needed</td>
</tr>
<tr>
<td>U3</td>
<td>The finished events are only visible while scrolling up</td>
<td>S</td>
<td>Gives the user the ability to look back what they have done</td>
</tr>
<tr>
<td>U4</td>
<td>The application is able to show extra information of an upcoming event</td>
<td>M</td>
<td>Provide extra information to the user when needed</td>
</tr>
<tr>
<td>U5</td>
<td>The application shows how long the present activity will last in a bar including the left minutes</td>
<td>M</td>
<td>Visualize information to the user</td>
</tr>
<tr>
<td>U6</td>
<td>The application shows all starting and ending times in the upcoming events</td>
<td>M</td>
<td>Provide information to the user</td>
</tr>
<tr>
<td>U7</td>
<td>If the user is in ‘change schedule’ view, he can see a day schedule and easily switch to a week schedule</td>
<td>C</td>
<td>Provide an overview to the user</td>
</tr>
<tr>
<td>U8</td>
<td>If requested a pictogram of the activity should be shown</td>
<td>M</td>
<td>Visualize information to the user</td>
</tr>
<tr>
<td>U9</td>
<td>The application is shown in a chosen color scheme</td>
<td>S</td>
<td>Provide feedback to both users</td>
</tr>
</tbody>
</table>

44
If an activity is finished properly the background of the activity is changed in a positive color (like green). If this is not the case the background changes in a neutral color (like gray).

If preferred by the user, a picture is shown with who the activity has to be performed with.

### A.4.3 Reliability requirements

<table>
<thead>
<tr>
<th>ID</th>
<th>Requirement</th>
<th>MoS-CoW</th>
<th>Goal</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>The application must also be usable in an offline mode, when there is no connection to the web server.</td>
<td>M</td>
<td>Provides the user a schedule</td>
</tr>
</tbody>
</table>
Appendix B

Technical design

The goal of the Agenda Coach is to help children, between 10 and 20 years, with autism by giving structure and provide extra information to activities. This all will be shown in a clean and simple interface in a mobile application.

This document describes the technical design of the Agenda Coach. It contains a description of the goal of the project and a detailed explanation of the technical choices that are made. This document is an expansion of the functional design.

B.1 Project requirements

B.1.1 Design Principles

This application is for children with a diagnosis of autism spectrum disorder. While keeping this in mind the application should not contain small details and distracting activities. To make sure the application is understood by most of the users it should work intuitively.

The application must be simple, clean, clear and predictable.

B.1.2 Design Patterns

MVVM

The pattern I suggest is MVVM. It is an adjusted version of MVC. The application will not have a separated controller class. The controller will be
replaced by a View Model. The user can control the application by making changes in the view.

Repository

Since there always has to be an schedule available we will use a Repository-pattern to create a data abstraction layer between the database and the view. If the schedule changes in the web server, the repository should be updated as well.

B.1.3 Topology diagram

![Topology Diagram](image)

Figure B.1: Topology

To update the data in the database with the data in the web server, the application synchronizes with the web server. The application stores the data of the application in the local database. This is visualized in figure B.1.

B.2 Components

B.2.1 Web server connection

The web server connection component must be able to receive all media objects and corresponding meta data from the existing web server. This component should also be able to automatically recover when the connection to the server is lost.

The offline synchronization component checks frequently whether there is an internet connection. If that is the case, an updated schedule is transferred to the offline database.
B.2.2 GUI

The application should have a clean and easy to use interface. Basically there are just 5 views: the log in view, the main view to use the schedule, the change view to adapt the schedule, the insert view to insert information for an activity and the free time activity view which helps choosing an activity. This division makes sure the child does not accidentally changes the set schedule. Using the ‘menu’ button one can switch through the main, change and free time activity views. The insert view can be reached though clicking a time frame in the change view. Those views are visualized in figure B.2.

The user is able to choose a color scheme (in settings). This color scheme is used in the full application.

Log in view

The log in view is a simple view that allows the user to log in to his account. Parents and treatment providers can also log in on the same account. This view only shows when there a user has not logged in already. After the log
in view the user is directed to the main view.

Main view

The main view has a simple (extendable) list view interface. Every parent item of the list consist an activity, start time, (if preferred) pictogram and if the activity is active at the moment a progress bar. The items are shown in chronological time order starting from the current time. Already passed activities are reversed chronological ordered above the current activity and only shown if the user scrolls up in the main view. There is a maximum for scrolling up for only 12 hours. The colors of positive finished activities is green, otherwise gray. The colors for the upcoming events are given are chosen by the color scheme.

An activity can contain extra information. This extra information can be shown by clicking on the name of the activity. The extendable list view shows the child of the clicked parent. This child can be closed by clicking on the parent again, or click on a different parent whereby this child closes and the new child opens.

Change view

If one wants to change the schedule he can open the change view by clicking on the menu button in the main view. The same schedule is opened but one can see that he can make changes in this schedule. One can make changes by:

1. Clicking an activity
2. Click a day
3. Select a time frame by holding
4. Drag an activity to a new time frame

When action 1, 2 or 3 is performed a pop up will ask you which action you want to perform:

- Copy
- Change
• Delete

When action 2 or 3 is performed also the option paste appears. When acting with action 3 on an empty time frame the option Insert appears. When choosing the action the activity will be copied, pasted, deleted or in case of Change an insert, direct you to the corresponding insert view.

For acting activity 4 you will reschedule the activity and will automatically directed to the insert view, reschedule.

Insert view

In this view one has the option to insert or adapt an activity and his corresponding information. Depending on how directed from the change view to the insert view some information will already be shown.

For all activities the user is able to fill in the following information:

• Name of activity
• Picture of the activity
• Starting time
• Ending time
• How to perform the activity
• Where to perform the activity
• With who to perform the activity
• Needs reminder during the activity?

Insert new activity The selected time frame is already filled in, but can be adapted if needed. There can be chosen from a list of already saved activities. All fields will be filled with the knowledge of the chosen activity, but can still be adapted. If the activity and his information is not chosen from a list, it can be saved by pressing on the ‘save in activities’ button. One should now insert a name for this activity with the corresponding extra information. When the activity should be set in the schedule one can easily press the OK button.

Change existing activity The insert screen is opened with all already know information filled in. One can adapt the information he prefers to
change. When finished one can press the OK button. The application will ask a yes/no question if the other users should be notified with this change, when answering No nothing will happen, when answering Yes a pop up will appear. In this pop up one can send why and what is changed in this activity. All users will be notified with this change.

**Reschedule existing activity** This screen is the same as the change existing activity screen, though the time frame is already changed to the time where this activity is dragged to.

**Free time activity view**

One can reach this view by clicking Find free time activity in the menu. This view is a multiple choice environment: the applications asks questions and the child can answer them. The goal of this view is to find a suitable activity for the momentary spare time. The application will ask how many free time he has, what kind of weather it is and with how many people the activity should be performed. When these questions are answered the application will give 2 possible activities. The child can choose one of them.

**Menu**

The menu can be opened at any time and had the options: Settings, Change schedule, Find free time activity. By choosing one of these options the application will direct them to their corresponding views.

**B.3 Technical diagrams**
Figure B.3: Class diagram
Figure B.4: Sequence Diagram
Figure B.5: Data flow model
Appendix C

Interview

The interviews were conducted in Dutch, in this thesis they were translated to English. It was a semi-structured interview, during the interviews some extra questions were asked and some were left out.

Introduction

For my bachelor thesis at the Dr. Leo Kannerhuis and Radboud University, I am going to build a prototype application for children with ASD. It will be an application where all upcoming activities will be displayed and the user will be notified when an activity changes. Because the application is build from scratch there is a wide scale from functionality we can implement. By interviewing you I am researching which functional and design requirements are useful for children with ASD. Because you are experienced in children with ASD your opinion is important to make a good design for this target group.

Open ended questions

- What is your relation to children with Autism?
- Is planning a problem, how do you notice this?
- Do you have a solution for the planning problem?
- How do you feel about digital planners?
• Which functionality would you like to have if you could design your own digital planner for children with ASD?

• What do you suspect of the usage of such an application?

Specific questions

Navigation

• How would you imagine the layout?

• How do you feel about the usage of pictures?

• Would pictograms have a more positive result then pictures?

• Brain storming about navigation in the application.

Supportive/Creativity

• Would adding your own pictures have an added value?

Site content

• What is the best way to approach children with ASD?

• How much information is needed when explaining an activity?

Interactivity

• Which functionality should be able to personalize?

Responsiveness

• Activity 1 is ending, which means the start of activity 2. What is the best approach of telling?

• An activity is changed, what is the best approach of telling?

• Would a vibrating function have a positive result?
Helpful

- Which amount of help and information should the application give?
- How do you feel to give the following information to the user: what, where, when, who and how?

Emotionally fulfilling / Rewarding

- Should children be able to check their finished activities?
- Should children be rewarded when they finished an activity correctly?

Fun

- What are the hobbies of your child?
Appendix D

Nederlandse Samenvatting

D.1 Inleiding

Kinderen met een autistimespectrum (ASS) hebben een duidelijke structuur en vaak extra uitleg nodig bij dagelijkse activiteiten. Deze problemen kunnen opgelost worden door een duidelijk tijdsschema aan te geven, waarbij gebruik gemaakt wordt van pictogrammen [Campbell et al., 1996], [MacDuff et al., 1993]. Als er veranderingen optreden moet het tijdsschema aangepast worden, daarom is het handig als het schema in een digitale applicatie zou zitten. Hierdoor heeft het kind altijd een schema en extra uitleg bij de hand. In deze thesis beantwoord ik de volgende vragen:

- Is het mogelijk een mobiele applicatie te maken die kinderen met ASS helpt gedurende de dag?
- Welke functionele aspecten zijn er nodig in een applicatie voor kinderen met ASS?
- Welke non-functionele aspecten zijn er nodig in een applicatie voor kinderen met ASS?
- Welke technologiën zijn er nodig om deze applicatie te programmeren?

De antwoorden op deze vragen worden beantwoord door interviews, het bouwen van een prototype en het testen van dit prototype.
Ik visualiseer de planning van het kind in een schema of kalender
Ik denk dat men voordeel zal hebben van het gebruik van de applicatie
De applicatie zal alleen gebruikt worden als het kind het vertrouwd is
Ik zou graag een gedetailleerd dag schema willen zien, met de mogelijkheid tot een weekoverzicht
Ik zou het kind graag een gedetailleerde omschrijving van de activiteit willen geven
Ik verkies pictogrammen over het gebruik van foto’s
De applicatie moet makkelijk aan te passen, te kopieren en te delen zijn
De applicatie moet makkelijk, simpel en een voorspelbaar interface hebben
Het geven van de informatie wat, wanneer, hoe en met wie de activiteit is lijkt mij een goed idee
Het kind zou een notificatie moeten ontvangen als een activiteit eindigt en er een nieuwe aankomt
Als een activiteit veranderd moet het kind een notificatie krijgen met een toelichtende rede
Het toevoegen van een checklist heeft geen toegevoegde waarde

Table D.2: Resultaten interviews. N is het aantal respondanten dat het eens was met de stelling

<table>
<thead>
<tr>
<th>Stelling</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ik visualiseer de planning van het kind in een schema of kalender</td>
<td>9</td>
</tr>
<tr>
<td>Ik denk dat men voordeel zal hebben van het gebruik van de applicatie</td>
<td>9</td>
</tr>
<tr>
<td>De applicatie zal alleen gebruikt worden als het kind het vertrouwd is</td>
<td>3</td>
</tr>
<tr>
<td>Ik zou graag een gedetailleerd dag schema willen zien, met de mogelijkheid tot een weekoverzicht</td>
<td>3</td>
</tr>
<tr>
<td>Ik zou het kind graag een gedetailleerde omschrijving van de activiteit willen geven</td>
<td>3</td>
</tr>
<tr>
<td>Ik verkies pictogrammen over het gebruik van foto’s</td>
<td>6</td>
</tr>
<tr>
<td>De applicatie moet makkelijk aan te passen, te kopieren en te delen zijn</td>
<td>5</td>
</tr>
<tr>
<td>De applicatie moet makkelijk, simpel en een voorspelbaar interface hebben</td>
<td>10</td>
</tr>
<tr>
<td>Het geven van de informatie wat, wanneer, hoe en met wie de activiteit is lijkt mij een goed idee</td>
<td>8</td>
</tr>
<tr>
<td>Het kind zou een notificatie moeten ontvangen als een activiteit eindigt en er een nieuwe aankomt</td>
<td>7</td>
</tr>
<tr>
<td>Als een activiteit veranderd moet het kind een notificatie krijgen met een toelichtende rede</td>
<td>4</td>
</tr>
<tr>
<td>Het toevoegen van een checklist heeft geen toegevoegde waarde</td>
<td>7</td>
</tr>
</tbody>
</table>

D.2 Verzamelen van requirements

Voor het verzamelen van de requirements voor de applicatie heb ik 10 semi gestructureerde interviews gehouden. De geïnterviewde personen hadden ervaring met kinderen met ASS en bestonden uit 5 leraren / begeleiders, 3 ouders en 2 adolescente met een vorm van autisme. De samenvattende resultaten kunnen gevonden worden in tabel D.2.

De volgende opties zijn in de interviews geopperd om de applicatie te personaliseren:

- De grootte van de taakomschrijving
D.3 De applicatie

Na het houden van de interviews is de applicatie ontworpen. Deze is in detail geformuleerd in een technisch en functioneel document. Vervolgens is er een prototype gebouwd met de belangrijkste requirements. Deze requirements waren:

1. De aankomende activiteiten zijn zichtbaar
2. Er is extra informatie beschikbaar over een activiteit
3. De starttijd van een activiteit is zichtbaar
4. Bij activiteiten worden bijbehorende pictogrammen weergegeven
5. Er wordt door een progressbar visueel duidelijk gemaakt hoe lang de activiteit nog duurt

In afbeelding [D.1] zijn twee screenshots te vinden. Zoals te zien is wordt requirement 1 vervuld door het laten zien van een lijst. Requirement 2 wordt volbracht door een uitklapbare lijst, deze is te activeren door op de activiteit te klikken. Requirement 3 en 4 door het laten zien van een klok en een pictogram. Door de progressbars wordt ook requirement 5 vervuld. De progressbars vullen naar de hoeveelheid tijd er al gepasseerd is. De applicatie zou
het beste geprogrammeerd kunnen worden via het PhoneGap framework. Hierdoor is de applicatie gelijk beschikbaar op IOS en Android. Ook moet er een schema beschikbaar zijn wanneer er geen connectie met internet is. Dit wordt gerealiseerd door een database die het recentste schema opslaat en weergeeft als de applicatie buiten het bereik van internet komt.

D.4 Testen

Het prototype is een dag getest op twee jongens met een vorm van autisme: een jongen van 12 met klassiek autisme en een jongen van 15 met PDD-NOS. Het programma werd van te voren ingevuld, geprogrammeerd en opgestuurd naar de jongens. Een programma bestond bijvoorbeeld uit: opstaan, ontbijten, computeren, Franse woordjes leren, lunchen, lezen, computeren. De ouders merkten bij de kinderen een positieve gedragsverandering op, waarbij de kinderen de meeste activiteiten uit zichzelf uitvoerden. Beide kinderen
vonden het erg leuk om de applicatie te gebruiken, maar vonden de dag indeling te gedetailleerd. Zij en de ouders verwachten de applicatie vooral te gebruiken op drukkere dagen of tijdens toetsweken.

D.5 Conclusie

Het is mogelijk een mobiele applicatie te maken die kinderen met ASS helpt gedurende de dag. Ondanks de niet significante resultaten, blijkt dat de ouders een positieve gedragsverandering opmerkten. Het design van de applicatie moet zo simpel en overzichtelijk mogelijk zijn. De belangrijkste functionele eisen zijn het visualiseren van de verstreken tijd, mogelijkheid tot extra informatie geven en notificaties bij het aflopen en beginnen van nieuwe activiteiten.
Bibliography


