DEVELOPMENT OF AN ANDROID APPLICATION THAT ENHANCES HEALTHY COMMUNICATION BETWEEN CHILDREN AND PARENTS

Ilias stavros panagiotopoulos

This dissertation was submitted in part fulfillment of requirements for the degree of MSc Advanced Computer Science

DEPT. OF COMPUTER AND INFORMATION SCIENCES

UNIVERSITY OF STRATHCLYDE

AUGUST 2019

Reg.No: 201657956

DECLARATION

This dissertation is submitted in part fulfilment of the requirements for the degree of MSc of the University of Strathclyde.

I declare that this dissertation embodies the results of my own work and that it has been composed by myself.

Following normal academic conventions, I have made do acknowledgement to the work of others.

I declare that I have sought, and received, ethics approval via the Departmental Ethics Committee as appropriate to my research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to provide copies of the dissertation, at cost, to those who may in the future request a copy of the dissertation for private study or research.

I give permission to the University of Strathclyde, Department of Computer and Information Sciences, to place a copy of the dissertation in a publicly available archive.

(please tick) Yes [X] No [ ]

I declare that the word count for this dissertation (excluding title page, declaration, abstract, acknowledgements, table of contents, list of illustrations, references and appendices is

I confirm that I wish this to be assessed as a Type 1 2 3 4 5 Dissertation (please circle)

Signature:

Date: August 17th, 2019

AKWNOLEDGEMENTS

I am much obliged to Dr. Marilyn Lennon for her supervision and her support throughout the whole project.

This dissertation is Dedicated to the memory of George Karousos, a wonderful man, a great doctor.

ABSTRACT

Effective communication plays a significant role in human relationships. Communication between children and parents plays a significant role during the development stages of the child, and unhealthy communication affects children's mental health, by creating problems related with emotional regulation self-esteem, self-efficacy, and interpersonal skills. This dissertation examines the role of healthy communication in children-parent relationships and tries to examine how mobile applications technology can be used in order to enable an effective and healthy communication between them. By examining the nature of healthy communication, this dissertation tries to find ways for improvement, using mobile applications technology by creating an android app that enhances healthy communication between children and parents. In order to find what technologies are used for the development of an application that enhances healthy communication between children and parents, an agile software development approach was adopted, and the basic phases of design, implementation, testing and evaluation were implemented and analyzed. The methodology that was followed during the project was also analyzed, for each phase of agile software development. The analysis of parents and children as users that try to communicate effectively, gave us the requirements in order to create features for the application that enhance healthy communication, like capturing mood or mental and physical health information of the child, which will be accessible to the parent through his mobile phone in real time. Model view presenter architecture was used during the development phase making the development process easier, by separating the UI logic from the construction code. Google firebase mobile and web application platform was used for the backend services like Realtime database and cloud messaging. The application was successfully tested with firebase test lab and evaluated using heuristic evaluation, finding usability problems that could be fixed in a future release.

Table of Contents

Table of figures

Chapter 1: Introduction

The obstacles in communication between children and parents, can affect children's mental health by creating emotional regulation self-esteem, self-efficacy, and interpersonal skills problems. The control of these issues costs the UK millions of pounds each year. More specifically it is estimated that Mental ill-health is costing the UK more than £94bn every year, including social support, treatment cost, and the high unemployment rates in these groups including from people who cannot work according to the Organization for Economic Co-operation and Development's (Boseley, 2019). It's very interesting to examine how mobile applications technology can be used, in relationships between children and parents in order to enable an effective and healthy communication. Technology and more specifically mobile phones technology, can be a critical factor in the creation of a healthy communication between children and parent, improving parent availability when they are not in each other premise, and prompt connection using physical proximity. Calling or texting is an outcome of the modern structural-functional changes in mobile phones, whether nowadays are used more advanced technologies and applications who intend to monitor children's mobile phones, increasing at the same time many important issues regarding their privacy, and raising ethical issues. (Lenhart, et al. 2010). Nowadays smartphones can improve parent's availability, as the child can rich him easily provide them a direct and immediate connection. Affective awareness includes a feeling of closeness between friends and family, while peripheral awareness is related to the attendance of the presence and activity of others. (Benford & Bowers, 1994). Mobile phones usefulness goes beyond text or voice communication nowadays. Mobile phones provide an excellent way to stay connected, and could save lives, providing the user with feelings of security in case of emergency, embedding advanced technology like GPS and internet access, which permits advanced features (Streetdirectory.com, 2019). This research tries to examine how Mobile technology can enable healthy communication between parents and children, by creating an android application that will implement features and technologies that will enhance healthy communication between them.

1.1 Aim

This project has as its aim to develop an android application which enhance healthy communication between children and parents. More specifically the main objectives of this dissertation are to:

- Design
- Implement
- Test
- Evaluate an android application which enables healthy communication between children and parents

1.2 Objectives

To accomplish these points, a few targets must be accomplished first. These targets are:

- Review the literature and theory of “healthy communication” among children and parents and find gaps in current knowledge. Understand how healthy communication can affect parent’s children’s relationship (Chapter 2)
- Examine the methodology and techniques which will be used during the software lifecycle and analyze the approach that will be adopted, for the application development (chapter 3)
- Examine Children and parent as users, finding creative features for the application, understanding the parent-child as a dynamic relationship representing user’s needs. (chapter 4)
- Explore and apply the design patterns, principles of software engineering and design features that the mobile application will implement. (Chapter 4)
- Perform Functional testing on the application (chapter 4)
- Evaluate the usability of the android application using heuristic evaluation. (chapter 4).

1.3 Research questions

More particularly, the following research questions need to be approached:

1. What is the definition of “Healthy communication” between parents and children?
2. What methods/ functionality and technologies can be implemented in a mobile application to increase a parent’s availability while out-with physical proximity to support healthy communication and overcome Chronic difficulties?
3. What other similar apps are available on the market?
4. What usability problems were found during evaluation phase and how affect the usability of the application?

1.4 Overview of the Dissertation structure

Literature review in chapter 2, will try to present the literature in an organized way, using background research and identifying gaps in current Knowledge. Healthy communication’s various forms, teenager’s and parent’s relationship problems and obstacles that prevent an effective communication will be analyzed, focusing especially on child-parent relationship. In chapter two the main methodology and the research approach that was used for the implementation of the application, and a plan for each phase of the implementation are going to be examined. Design, implementation, testing and evaluation plan are going to be analyzed examining the methodology that is going to be used and the specific plan that is going to be followed in each phase. In analysis (chapter 4) the basic phases of software lifecycle

(design, Implementation, testing, Evaluation) are going to be conducted according to plans which will be analyzed on methodology chapter. In the last chapter a summary of findings and the resulting conclusions are going to be analyzed, followed by recommendations.

Chapter 2: Literature review

2.1 Aim

In Latin communicate, means "to share". A Universal definition of communication is "the act of transferring meanings from one entity or group to another applying common symbols, signs and semiotic rules (En.wikipedia.org, 2019). The current project tries to find and analyze the various mobile phone technologies that can be used in a mobile application to enhance healthy communication between children and parents, and literature review examines the nature of effective communication, and how can affect the relationship between children and parent. While communication in general as well as the different forms of communication are going to be reviewed, this literature review is focusing on communication between children and parents. Analyzing children and parent's behavior and how they communicate, this literature review will be used for the design of a mobile application, that enhances healthy communication between children and parents, analyzing children needs and trying to find creative ideas for the interaction of the user and the parent. The literature review is going to be used during the various phases of lifecycle and most specifically during the design phase of the android application, in order to understand the user and its context, as well as to enhance the process of requirements gathering, and user analysis. The basic aim is to understand the importance of healthy communication between children and parents, analyzing various forms of communication, and finally recognizing children and parents' behaviors which affect their relationship giving at the same time creative ideas for the design, and functionality of the application.

2.2 Literature review Search strategy

In order to conduct the literature, review the basic terms needed to be defined and key concepts to be addressed. A list of keywords and synonyms were used to find sources from databases as a research strategy. Books, articles or sources like forums and tutorials were utilized to get more insight concerning Healthy communication between children and parent. Specific keywords were used during the research in order to organize and analyze the literature. Searching creatively and Identifying all the relevant information sources is the first step. The main sources are libraries, indexes and electronic databases, and the Internet. The main keywords or pairs of keywords that are used for searching literature were:

- Healthy communication teenager
- Communication mobile phones children
- Apps children parents

Using google search was achieved to find a representative number of articles that made the sample illustrative in the field of study. A variety of analysis methods (content analysis and frequencies) were used. First, examining the definition and exploring each form of communication, the literature review tries to provide the context of communication in general. Then the various forms of communication inside a family, and problems or solutions for an effective or healthy communication are following. Next analyzing Bowlby's theory of attachment, the important role of communication in child parent relationship is going to be addressed. After having a basic understanding of the role of communication, there is a try to examine and find ways to build and improve healthy communication between children and parents. Some examples of effective communication are given, and advices for what a parent is best to do, or what should not do are analyzed. Then the obstacles in communication, the impact on their life's, and common problems that teenagers are facing are analyzed. The nature of digital communication and the specific fundamental features of it, are going to be analyzed next examining how Mobile phones are used in everyday life by children and parents in digital era. Finally, a conclusion section where the most important findings of the literature review and gaps in the current knowledge are going to be addressed.

2.3 The act of communication and its various forms

Communication takes places in logical verbal and nonverbal levels and happens when two people share information in a two-way street. Communication needs both persons that take part to engage (the person that is communicating, and the person that is being communicated) (Schmitz, 2019). Family Times journal which is published by New Mexico State University's Family and Consumer Sciences department defines Communication as sharing of feelings, ideas, and thoughts between people. Communication doesn't have to involve words and can be even a smile or a frown. Moreover, silence can be a way of communication (Runcan et al., 2019). Acquiring right set of skills, the miscommunication can be avoided, permitting a clearly and efficiently communication and deterring the cause of incomplete tasks, relationship frustration and hurt of feelings. In the following lines the main forms of communication are going to be analyzed. One form which may sometimes be the most important part of communication is listening. Listening it's not the same as hearing. It's important to separate the meaning of hearing and listening. Listening includes the effort to understand what the other person trying to say and it's not the same as the sense of hearing. There are many factors that influence listening and skills that needed to be a good listener like focus, interest, patience and many more. On the other hand, the form of verbal communication it's not as simple as just talking. The words that someone uses play a significant role and should be chosen wisely.It needs skills like using appropriate tone, consider the audience, or respond

appropriately (Schmitz, 2019). According to peter ducker who is widely regarded as the “Father of Modern Management “ the most important part of communication is hearing what isn’t said” (Ian et al., 2019). Nonverbal communication is a form of communication that shows the importance of pay attention to people actions as well as their words. Facial expressions, body language, eye contact, posture or body position, pace of speed or speech, are nonverbal cues helping understand what the speaker is saying. Feelings play an important role in communication. Emotional awareness is a form of communication that gives the ability to understand feelings of other people influencing the way they communicate. Emotional awareness involves consider other people’s feelings as well as your own feelings. Empathy plays a crucial role in this kind of communication. Written communication is o form of communication which is used very often in now days has its own set of rules. Except proper grammar and punctuation, emotions can be expressed inside writing but it’s very complicated since things like sarcasm often don’t being interpreted correctly (Schmitz, 2019).

2.4 The importance communication in child parent relationship

When we are talking about healthy communication or effective communication, we are not talking about simply exchanging ideas and information. It includes understanding of intentions and emotions behind the information (Robinson, 2019). More specifically effective communication is a combination of listening and speaking. Listening well to other people in a family, supports them to speak about what’s most important in their lives. It’s not difficult to get careless about listening, particularly in families. Of course, it is supposed that we know what the other person means or acts to listen while doing something else. Listening needs attention. Putting beside anything you’re doing you are giving the impression you’re ready to listen. Listening demands openness and respect. Feedback reassures the listener to know if they’ve really understood what was said. Listening is not easy when strong emotions are present. Being with a family member who is undergoing a tough time lets them know you care. Sometimes a hug or a soft touch shows support when someone doesn’t know what to say (Communication is the key to healthy family relationships, 2012). According to Bowlby, the process of healthy communication creates the building blocks for enhancing security in children and empower them to search and develop their sense of autonomy and independence. During the early years, the mother is a child’s ego and superego while the child obtains the capacity for self-regulation. These functions are operating also during infancy and early childhood, but during this phase of life, the child is dependent on his mother performing them for him. She orients him in space and time, permits the satisfaction provides his environment of some impulses and limits others. She is his ego and his super-ego. Increasingly he learns these arts himself, and as he does, the skilled parent transfers the roles to him. This is a continuous slow process, which begins when he first learns to walk and feed himself, and not ending completely until maturity is reached. Ego and super-ego development are therefore absolutely hounded up with the child’s

primitive human relationships (Bowlby, 1951, p. 53). Bowlby grounding the available empirical evidence concluded that to grow up mentally healthy, “the infant and young child should experience a warm, intimate, and continuous relationship with his mother (or permanent mother substitute) in which both find satisfaction and enjoyment” (Bowlby, 1951, p. 13). In a study that was conducted in Romania where 100 parents participated was found that the frequency of communication between the child and parent makes a deep impact thus much in the development of children, as in the communication with the parent. Parents need to recognize children needs in order to satisfy them. Sometimes it is difficult to sympathize with the children as there are many obstacles that prevent them to have good relationship with their children (Runcan, Constantineanu, Ielics, & Popa, 2012). Lack of communication can have a bad impact on the operational of family functions. Communication builds and sustains relationships between children and parents and through the feedback obtained cooperation and interaction become stronger and more efficient creating a successful relationship where needs of children can be identified and satisfied.

Effective and Fair relationships mutual acceptance and understanding can be created between children and parents during communication process. So, in order to have a healthy and effective relationship with mutual acceptance and understanding between parents and children communication must be present, hence it is a basic element in the relationship between the parent and the child. It's more important in the modern family for the parent or child to know how to communicate instead of just engaging in the communication process. Is very important here to have a successful message received by the child. The failure of the child to not "perceive the real meaning of the message" is usually related to parent’s ability and it's his fault because he didn't adopt an appropriate form to communicate the message to the child. As more and more confidence gained members of a family become increasingly open to others and with mutual communication, not one-sided offers significantly to the formulation of bonds in the child and parent interaction. The communication between parent and children, shows that parents are interested in their children's lives and that are valuable to them. Furthermore, Communication is essential so that family members respect each other’s wants and needs. Without communication they will not know what each other think or feel which can create many problems in family life. Apart from a single expression, the Communication between children and parents expresses more important communication interactions and is a mean through which parents try to be understandable to a child, in terms of the precise content of the communication as well as the general content (Runcan, Constantineanu, Ielics, & Popa, 2012).

2.5 Ways to improve communication

Communication takes places in logical verbal and nonverbal levels. To have effective communication, there should not be an inconsistency or disagreement between these levels. If there is discrepancy the message will not have an effect. At the logical level happens 7% of the act of communication while 38% happens at

paraverbal level. Paraverbal level includes the volume of speaker, the speed of speech, or tone). 55% happens at non-verbal (movement, position, clothing, facial expression) (Kumar, 2013). In order to answer the question of how communication should take place within the family we must find what is required for communication between children and parent. Several possible answers can be formulated to this question which are going to be analyzed. During the communication, an appropriate and calm body language and an appropriate vocabulary should be used. But what an appropriate vocabulary means? During the communication parent should use an appropriate vocabulary which is acceptable in social terms, and words that the child can fully understand. The creation and initiation of psychological comfort in communication is achieved using a calm tone during the conversation so the message will be better and faster acceptable and understood by the child instead of using an angry tone (Patricia Luciana Runcan a , Corneliu Constantineanu b , Brigitta Ielics c , Dorin Popa d, 2012). There is no doubt that parenting is hard work and needs a lot of effort, but there are some things that a parent can follow to retain the lines of communication open and secure a good connection. A parent should be available to its children and notice times when the children are ready for communication. There are specific times in the day that children are most likely to talk for instance, before dinner, in the car, or at bedtime. Showing to the children that he cares about what happens to their lives is very important as well as finding time each week for a one on one activity avoiding programming other tasks during that time. learning about Children interests like Activities and favorite music is very important as it shows interest in them. Prefer to open conversations by telling them what you have been thinking about instead of initiating a conversation with a question. It's preferred to listen and not interrupt the children when they are talking about their concerns letting them finish their point before you answer. It is very important to answer without anger because usually children tune out someone who looks angry or defensive. Sometimes it's hard to hear but it's very important to listen to their point of view, stopping what you are doing and express interest to the conversation, before a response is given. Ensure that you understand them accurately repeating what you have heard. Reply in a style your kids will hear Soften strong reactions; Communicating his opinion without invalidating theirs is very important recognizing that it's okay to have a different opinion or disagree, without arguing who is right, concentrating on the child's feelings rather than his own, preferring to say "I know you disagree with me, but this is what I was thinking about". There are many things that a kid needs from a conversation like simple listening, help to solve a problem, help in dealing with feelings, or simple advice and the parent needs to ask them. Imitation is a basic learning strategy that is used by children. It will be used when they will follow their parent guidance in how they deal with anger, work through hard feelings, and solve problems. Kids learn from their personal choices and many times testing the parent by saying him a tiny part of the story about what is bothering them. The parent must listen mindfully to what they say and support them to share the rest of the story. The key for a healthy connection with the children is to listen and talk, but sometimes it could be hard to keep a good

connection with the children since parents sometimes have other very serious issues to encounter. A mental health professional might be helpful if there are problems in communication for an extended period (https://www.apa.org, 2019). Body language plays an important role in the support or rejection of a request suggested by a child or parent, and it should be consistent with the message. Consistently is important to avoid confusion that may appear in the children mind, and to produce a message that could be believed and understood. An example is when a parent tells his child that forgives it, but he seems angry, or if the message is followed by a frown the child gets confused and the message fails. Contrary if the message of forgiveness is followed by a smile the message is comprehended understood and believed by the child. The child feels happy and connected with its parents when it has the chance to have an effective daily talk with them which leads to peaceful and complicated development. (Patricia Luciana Runcan a , Corneliu Constantineanu b , Brigitta Ielics c , Dorin Popa d, 2012).

2.6 Obstacles in communication

There are many parents that find it doubting to communicate with their children when they are apart. Situations such as divorce, illness or lengthened absences for work make having a healthy parent-child relationship an even bigger hurdle. The impact of separation can be destructive to children, and leave parents feeling weak. During the school year, absences are especially difficult when children want their parents to be there for school activities and sports, or when need help with their homework. Regardless of the reason why a parent is away, children still need to know that their parent loves them. Undoubtedly parents may feel their own difficulties because of separation from their children. However, it’s very important to put the kids first, especially during times of separation; parents must concentrate on providing their children with a nurturing environment. Teenagers are faced with life-changing decisions and the consequences almost every day. The middle school and high school years are the most difficult for teenagers because they are neither children nor adults. Their bodies are suddenly changing, and they also face tremendous emotional swings. Many teens are seeking answers to difficult questions such as what they’ll do with their lives and how they feel about themselves. There are many Important issues for teenagers like the importance for parents to talk with them, to pay attention to them before they get into trouble, to help them learn their body and emotional changes, to talk about peer pressure, drugs, and alcohol, to get support at home, school and in activities and the most important to be treated with respect and to be fair to them. (Communication is the key to healthy family relationships, 2012).

2.7 Nature of digital communication

After the millennium, digital communication has seen a massive grown. More specifically parent-child relationship is affected by four specific fundamental features of digital communication which are:

- Persistence: Most of the times interaction in digital communication can be pleasant. However, sometimes communication and especially digital communication may have an unpleasant interaction, having a continual digital footprint, which might influence the relationship between child and parent. In the digital era, regular communication has transformed into “forgive but persist” rather than “Forgive and forget”.

- Changeability: Most of the times Young parents are very comfortable and happy with digital communication. However older parents are usually not very familiar with digital technology. According to a recent U.S report there is a digital divide with some communication gaps and mismatch which although it's minor it could be an issue leaving the parent often in a state of delay since the turnover rate of technology tools is high, and children adapt much easier to new technologies.

- Scalability: the element of friends and social circle should be involved as well in the relationship between children and parent which makes the support of the parent in the social life complicated. Nowadays wide range of social media and various digital services make the social circle global and temporary which affects the parent-child relationship.

- Access to data: access to data and information is easy nowadays. Discipline a child was restricted in the past but now using mobile phones it is easier. An example is an ease with which a parent can find his daughter at the movie theater and the embarrassment that the child may feel.

Internet and digital technologies have made the parent child relationship more complicated making the child feels that he lives in a global village. Digital era is enormously different from the previous one, however digital tools have certainly helped to parent. Parents should change their parenting approaches activities and beliefs regarding the digital era in which children are born into, making right choices in order to improve the wealth that these tools produce into the parent-child relationship and avoid the traps that can be created (Huffpost.com, 2019).

2.8 Mobile applications technology nowadays

The growth in the use of cell phones in extension to calling and texting can be viewed to be as an effect of the modern structural- functional changes in the cell phones; such as advanced storage facility internet availability, etc. Mobile phones help in keeping in touch parents and children when they are not together, while the main way of communication through mobile phones is “text messages”. It has been found that teenagers send and receive more messages than any other age group. They claim that texting helps them in relating their parents with their while family and friends. “Phone calls” are the primary function of “mobile phones” and it is primarily used by teens to converse with their parents. Another important thing is that one of the main reasons behind letting the children own mobile phones is to “monitor” them

when they are not in each other premises (Lenhart, et al. 2010). Lately, mobile applications are being used as instruments to modify people's behavior and convince them to engage in more pleasant in daily physical activities as well. Fitness and wellbeing mobile applications aiming at creating dynamic relationships among users, taking a fun oriented approach, where everyone can train together and get prompted to have healthier habits. Real Personal trainers take part as well to make the workout motivating and safer (Callie Thomson and Jane Nash, 2016). While a comprehensive market research will be conducted during design, some basic features of application that addressed to parents and teenagers are going to be mentioned. There is an absence on the market of applications that that aims to enhance healthy communication between children and parents for android platform. Most applications try to find a balance between monitoring teenagers while keeping them out of trouble by keeping an eye on teenagers' devices and tracking their children mobile phones. Details like location, messages, calls, notes and browse history or even social media are shared with the parent. Some application offers features like capture the screen of their child and take screenshots, or record calls and the surrounding conversations. Many apps can also restrict the device and block apps remotely. Most of the application monitor all the crucial details of the device remotely, without accessing the device (Spyzie.com, 2019). While there are many apps on the market that help with specific problems of teenagers and children generally, there are not many that trying to enhance healthy communication and alleviate the relationship problems that were mentioned earlier.

2.9 Summarize the main findings

But how mobile phones technology can contribute to surpass the obstacles that are faced during communication between children and parent, and enhance the creation of a healthy and effective relationship? More specifically how a mobile application can contribute to the relationship between a parent and child, and how healthy communication can be succeeded using a mobile phone application? In order to answer that question, we need to examine the important aspects of literature review that afterwards will be converted into ideas during design phase of the application. As we saw communication has various forms and real Participation, not just speaking or listening, including empathy, is a critical factor for an effective communication. We analyzed the main forms of communication like verbal communication, nonverbal, listening written, and emotional awareness. Whether communication is healthy or effective depends on many factors and skills that both participants need to have. For example, someone can be careless about listening someone else, or not giving attention when someone is speaking making the communication non effective. But if healthy communication plays an important role into adults' relationships how can affect a child-parent relationship? Bowlby suggests that young child should encounter an intimate continuous and warm bond with his mother or mother substitute in which both find pleasure. In order to grow a mentally healthy child. The child, which is dependent on his mother learn from her, new skills,

while she orients him in space and time until maturity reached. Child's Ego and super ego are hounded up with the child's primitive human relationships. The frequency of communication as well as the ability of parents to empathize and meet children needs plays a critical role to a healthy communication. Inside a family a basic element of communication is cooperation which means participation in discussion and not only being present. The respect that shows each other plays also an important role. When the message is misunderstood, or the communication is non effective its parent's failure who didn't adopt the appropriate form. The parent must be prepared to communicate with the child and not use anger against the child when the communication is not successful. As it was analyzed most of Most of the percent of communication happens in the non-verbal which includes movements position clothing or facial expression, then is the paraverbal including level volume of speed and speech tone and only a minimum 7% happens in the logical level. But what skills needs a parent to be effective in communication? As we saw Property language vocabulary and calm body language are necessary during a healthy communication. Must be used words that are fully understood by the child and acceptable in social terms accompanied with a calm tone of voice. Availability is also a critical factor in success of communication. The parents need to be available when the child is most likely to talk and show that he cares about him. Some advices for healthy communication include an open conversation with the child, letting it know about what the parent is thinking and let them talk about their concerns. Also important is to Let them finish before the parent answer stopping what he is doing and give priority to child Repeating what he have heard without invalidating their point of view. Showing them that is okay to have different opinions. Children use imitation as a learning strategy and body language as we saw plays a significant role in communication process. Sometimes kids don't tell what they are thinking or tell part of the story testing their parent trying to find Guidance on what to do help dealing with feelings. Many times, even if the parent wants to be a good parent and have a healthy communication with his child, there are some obstacles that prevent him from do that. Situations like divorce illnesses making the children vulnerable during important development stages of their lives, like school years where many body changes happen to teenager and the guidance or love and their parents is valuable. The digital technology and more specifically mobile phones can play a vital role in the enhancement of a healthy relationship between child and parent. The main methods that the mobile phones are used nowadays are messages and voice video call. Many kinds of different message applications for mobile phones are used by teenagers today. The huge explosion of use of mobiles phones has raised many concerns like privacy issues, unpleasant interaction, changeability and persistence. All these issues should be considered during the development of an application which focuses on the creation of a healthy relationship between children and parent. For example, unpleasant interaction may be a reason for a child to not use an application. There are many applications on mobile stores that can modify people's behavior by creating a dynamic relationship between users by gathering data. Ux plays a significant role in the design of an application creating an experiential,

emotional, effective, hedonic and aesthetic features of an interaction of a human with a product. For example, feelings can be expressed using specific icons (emoticons) and specific widgets and graphic UI can create a friendly environment.

Although there is many information that help us understand the dynamics of relationships between children and parents, there are no specific researches of how android mobile applications can enhance healthy communication between parent and children. There are not many apps that are used for this scope in the android market. There are only some generic apps that are capturing data in order to modify behavior or monitor their child activity to keep them safe. This research will try to find solution to problems that affect healthy communication, by creating an application that enables healthy communication between children and parent. Ideas for how specific problems regarding the parent-child relationship could be alleviated using specific features to the app that will help the relationship between parent and children, will be analyzed and implemented during the various phases of this project

Chapter 3: Methodology

This section provides a description of the methodology, the techniques, and best practices that are going to be used in each phase. Firstly, after presenting various software development approaches, we analyze the approach that will be adopted by the current project, providing a plan for each phase of the software lifecycle.

3.1 Software development approaches:

Software undergoes many steps when developed during the software development process. Firstly, the requirement phase takes place where the objectives are concluded providing a requirement document to the team of what the program will have the ability to do. After that the user interface stage and the architecture of the application is designed. The implementation phase comes next and developers with designers must build the application based on the previous phases. Testing and verification come next where a working copy is used to confirm that the application is error free. Organization's quality controls will be met, making sure that issues that are found during this stage are fixed and settled. There are various techniques and methods which are used during the developing process. Some examples of these methods or techniques are illustrated below:

- Agile software development
- Incremental development
- Prototyping
- Waterfall
- Rapid application development
- Top-down programming
- Object oriented

- Unified process
- System testing
- Spiral development

(En.wikibooks.org, 2019)

3.2 Software development Approach adopted

For this project the agile model was picked as the suitable methodology as it underpins changes in requirements and gives visit to small releases followed by improved versions of application. little steady releases which are improved in the following ones. Many increments will be produced to create a successful application, each one of which is going to be improved in the coming version. In each increment we will have changes in requirements, or improvements in design which will prompt new programming cycles. In the following section the phases of the agile model that are taking place are going to be introduced:

- Requirements specification: Here the requirements of the application and a comprehensive description of the system is going to be identified and be developed. For this project Requirement specification document will be a part of design chapter.
- Design: The important components of the system and system's architecture are going to be defined.
- Implementation: Implementation of system design (coding).
- Testing and evaluation: During the testing phase the system code is tested regarding its specification, and after that the evaluation phase comes where the success of the previous phases is evaluated and sometimes modified if needed.

Agile Model.

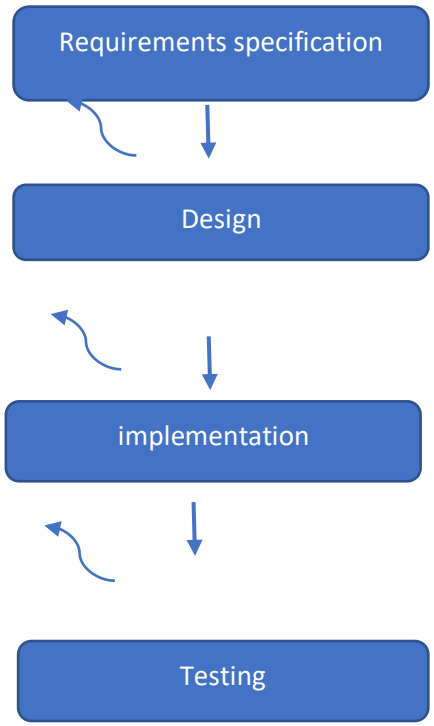


Figure: 2.1: Agile model

The order of phases is not absolute during the use of the agile model and if new requirements appear during implementation process a redefinition of these requirements is possible, going back to the requirements section.

3.3 Plan of Methodology chapter

In order to describe and analyze the methodology that will be implemented during the different phases of agile software lifecycle, a plan for each phase will be used, describing how each phase of software lifecycle is going to be implemented, and what methodology techniques and best practices are going to be used.

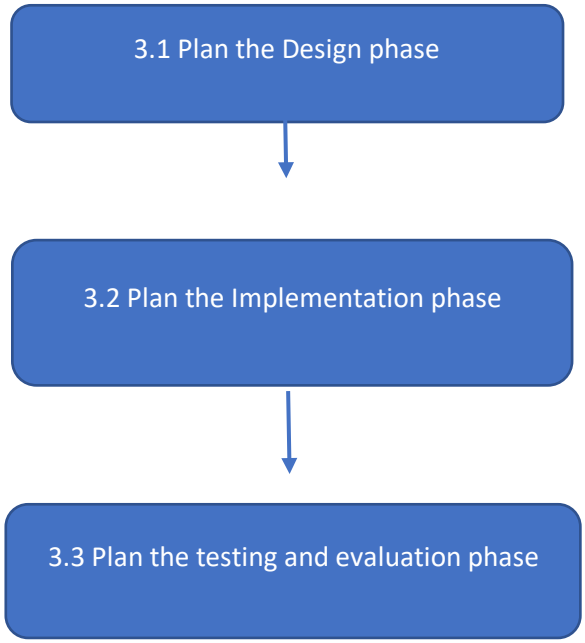


Figure: 2.2 Plan of Methodology chapter

3.4 Plan for design phase

In this chapter the Design phase is going to be analyzed in terms of methodology tools and best practices that will be used. In order to create the requirements, document the following phases using specific methodology need to be completed:

- Literature review and background analysis and research.
- Stakeholders analysis.
- Defining user context and tasks, understanding users and context.
- Market research.
- Gathering requirements plan (Task analysis, Uml diagrams).
- Implementing requirements document.
- Low fidelity prototype.
- High fidelity prototype.

Each one of these phases will be analyzed in depth, focusing of how they were conducted, the methods that were used, and the plan that was followed.

3.4.1 Literature review and background analysis and research

Literature review is a valuable source for the design of the application. It gives a deep understanding of user and his needs. Science is a collective effort and Literature reviews play a crucial role in research. The basic aim of the literature review chapter is to present the literature in organized way, using background research and identifying gaps in current Knowledge with respect to the subject. (The Importance of the Literature Search, 2008). The literature review will try to synthesize the information into a summary, and critically analyze the information gathered. The structure of the literature review in chapter 2 is the following:

- Introduction
  The topic is defined providing an appropriate context and stating the scope of the review. In this dissertation the literature review will examine the healthy communication between children and parents using mobile phone technology, and not healthy communication generally.
- Literature Reviews Main Body
  The Main body organizes the literature with argument and evidence, using common themes, evaluating the current knowledge, and reviewing critical features of the current body of literature. It Is also important link the research with existing knowledge identifying important flaws and gaps in the current knowledge. The literature review of this project will try to examine what is communication in general, and how healthy communication affects the relationship between parents and children going from a general wider view (healthy communication) to the specific

focus of the research (healthy communication between children and parent using mobile phones)

- Conclusion
  During the conclusion, the crucial features of the current literature are summed, the gaps in current knowledge in existing knowledge are recognized outlining areas for future study.

3.4.2 Stakeholders

After literature review the next step for the design phase is to find and analyze the stakeholders. Stakeholders are persons or organizations which have a valid interest in the system, and they may be affected by it either directly or indirectly. During the stakeholder analysis the attitudes of the stakeholders, regarding the potential changes will be evaluated. First, we must find our stakeholders by reviewing the literature review. Next, we are going to examine their power and influence trying to understand them in-depth.

3.4.3 User context and tasks definition

Here detailed information about the indented users, their tasks and environmental constraints are analyzed. Capabilities or qualities of the user and/or their context that might influence the design of the system like Age or Gender, Environment/Context, Culture, Globalization Vs Fragmentation, are identified. We will Find out what user wants, what customer will pay for Understand the tasks and context we are designing for.

3.4.4 Personas

Personas are going to be used in order to illustrate various user types and understand their needs experiences behaviors and goals. These fictional characters will be based on research and will identify different needs and expectations making the design less complex, in order to create a pleasant user experience for target group users. Personas characteristics and attitudes will be a product of literature review.

3.4.5 Market analysis

Using a process of elimination instead of only looking for one great app idea, we will start our market research with more decent app ideas trying to find what is best trying to eliminate others. Using market research, we will try to identify a market area, capturing information about each application and identifying existing systems, using a qualitative approach to capture information about each. We will try to find commonalities that are a must, and gaps, reviewing the android app store, using specific words and keypairs, checking reviews, download rates, the size of competition and identifying opportunities. Also, there is going to be a research on demographics and segmentation, market need, target group, and barriers to entry.

The market research will be completed before the development of the application in order to have an essential understanding of the user. Play store will be used as our source for existing android applications and a table with download rates, user's evaluation and information that will help us find applications functionality, or flaws that will help us to

3.4.6 Eliciting, Collecting, and Developing Requirements

After market analysis phase has finished, we should have a good understanding of the behavior that the system should have. We should have identified the desired functionality, the constrains that are imposed, and the business objectives.

3.4.7 Use Cases and Scenarios

Using use cases and uml diagrams, a list of actions is going to be defined, describing the interaction between the user (actor) and the system in order to achieve a task. Use cases will be used to discover and represent functional and nonfunctional requirements. The tool in the webpage Diagrams.io will be used for the implementation of the uml diagrams.

3.4.8 Hierarchical Task Analysis

Task analysis will be used to illustrate how a task is accomplished resulting in a hierarchical representation of the steps that needs to be performed in order a user to achieve a specific goal. First of all, the tasks that the application must support will be identified finding the user goals, what users are trying to achieve, and most importantly what users should do to achieve these goals, examining the workflow they follow to perform their tasks.

3.4.9 Implementing requirements document

After we have a good structure of the data, we are going to implement functional and nonfunctional requirements that will be used in the implementation phase. Here we are moving from defining to analysis making platform and architectural decisions

Through methodical gathering and analysis of data collected, concepts will become apparent, tagging them with codes which have been extracted by data. Then codes will be grouped into concepts and then into categories.

Requirements specification Document

The requirement specification document (srs) will describe the nature of the application, providing a manual of the project. First, the purpose of the project will be analyzed. Next the scope and the functional and nonfunctional requirements should

be identified. It should contain and explain all the application features, and functions. The basic questions need to be answered in order to create a srs:

- What exactly the application do?
- What qualities and specific characteristics possess?

The SRS will describe in detail both the functionality of the system and the technical solutions (user interface specifications/ technical specifications).

3.4.10 Functional requirements

The functional requirements (business rules, external interfaces) are system functions that the developers must implement in order to allow users accomplish their tasks. Generally, describes what the system should do analysing the characteristics in order to meet the needs of stakeholders.

3.4.11 Non-Functional Requirements

Non-functional requirements are known as quality attributes or characteristics, and define how the system must behave (performance, security etc) representing significant quality-related issues.

3.4.12 Prioritizing requirements

In order to set priority in our requirements we will use MoSCoW Method. First will be listed the most important requirements followed by less significant. The Moscow method it's an acronym and is based on first letters of words (Must, Should, Could, Would) figure. The M defines a requirement that has to be satisfied (Must), The S for a requirement that should be included if possible but it's not time critical, The C for a requirements that is desirable but the solution will be still acceptable if it is not implemented, and finally the W stands for Would and represents a requirement that stakeholders want to have but not in the current version (ToolsHero, 2019).

REQUIREMENT CATEGORY	EXPLANATION
MUST (M)	Defines a requirement that has to be satisfied for the final solution to be acceptable e.g. The HR system "must" store employee leave history
SHOULD (S)	This is a high-priority requirement that should be included if possible within the delivery time box. Workarounds may be available for such requirements and they are not usually considered as time-critical or Must-haves. e.g. The HR system "should" allow printing of leave letters
COULD (C)	This is a desirable or nice-to-have requirement (time and resources permitting) but the solution will still be accepted if the functionality is not included e.g. The HR system "could" send out notifications on pending leave dates.
WON'T or WOULD (W)	This represents a requirement that stakeholders want to have but have agreed will not be implemented in the current version of the system, that is, they have decided it will be postponed e.g. The HR system "won't" support remote access

Figure 3.1: MoSCoW Method

3.4.13 Sketches and wireframes

The actual design process will start with wireframes making them the backbone of the entire project and creating the structure and later making it easier to build individual parts. We will draw the sketches in paper.

Wireframes

Wireframes will permit us to plan the layout according to how we want the user to process the information. Wireframing Interaction will also allow us to explore user interactions states of buttons Menu options Input methods Output formats loosely shaping the final product, giving us a reliable idea of where everything will eventually go. The sketches will be designed on a paper.

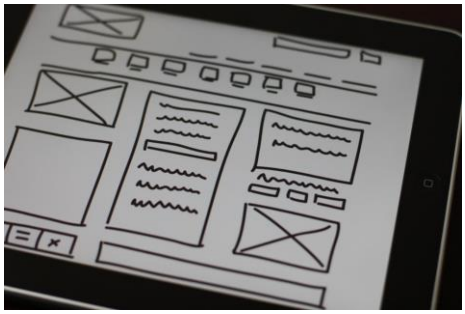


Figure 3.2: sketches

3.4.14 Prototypes

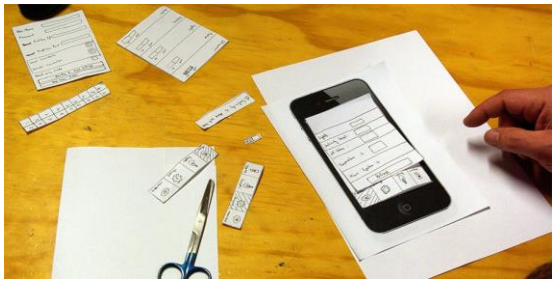


Figure 3.3: Prototypes

In order to move from defining to analysis (from requirements definition into analysis) we will evolve all or part of our essential user interface prototype into a traditional UI prototype. Here we will convert our sketches, into something a little more substantial. The prototype for this app will be implemented in android studio using visual editor. Following the sketches components of the user interface will be implemented with the tool and then a screenshot will be taken for each screen.

3.5 Plan for implementation phase

In this section the implementation plan is going to be analyzed. After description of some basic information about android architecture the implementation plan is going to be examined. The implementation plan is separated in to two parts. The first part is the user interface implementation plan where the implementation of the user interface is going to be analyzed and the second one the Logic implementation plan where basic architecture plan and best practices are going to be analyzed. Firstly, a general description of the Android operating system is going to be examined. The programming language that is going to be used in this project is Java and android studio will be the specific IDE (integrated development environment). The mobile application will be coded and built to run on the android devices (phone or tablets). The android studio IDE and its environment will be used which is a powerful and user friendly. Java is needed to write the programs; The android studio is the program (the environment) which will give us access to Android SDK (Software Development Kit). Android studio it's the program where we will write edit and save our code. SDK it's like an extension to the java code that runs fast on android devices using at the same time the hardware. Because of SDK the programs run and android as well android studio putting all together. Android studio also offers to run the application through an emulator or through a piece of hardware like an android device connected to the machine debugging and taking feedback as it runs (explaining crashes etc.). For this project an android phone xiaomi mi max 2 will be

used for debugging. Using an external device like a tablet or a phone to test the app and not a virtual machine will be more convenient since the external device is faster since it uses its own hardware. Also, you can feel and touch the application in real time (Hardware acceleration)

3.6 Implementation stages:

- Workspace Set up:
    - First, we must install android studio and create a new project.
    - Then Write the code: After android studio has been installed, we are going to use many tools that are going to help us build the application in order to write quality code, design the user interface or create resources.
- Build and run: During this phase, we will make an APK file which will be installed and run on the emulator or in an android device .
- Debug, profile, and test: While we continue writing our application, we will focus eliminating bugs and optimize app performance and creating tests. This is an iterative phase.

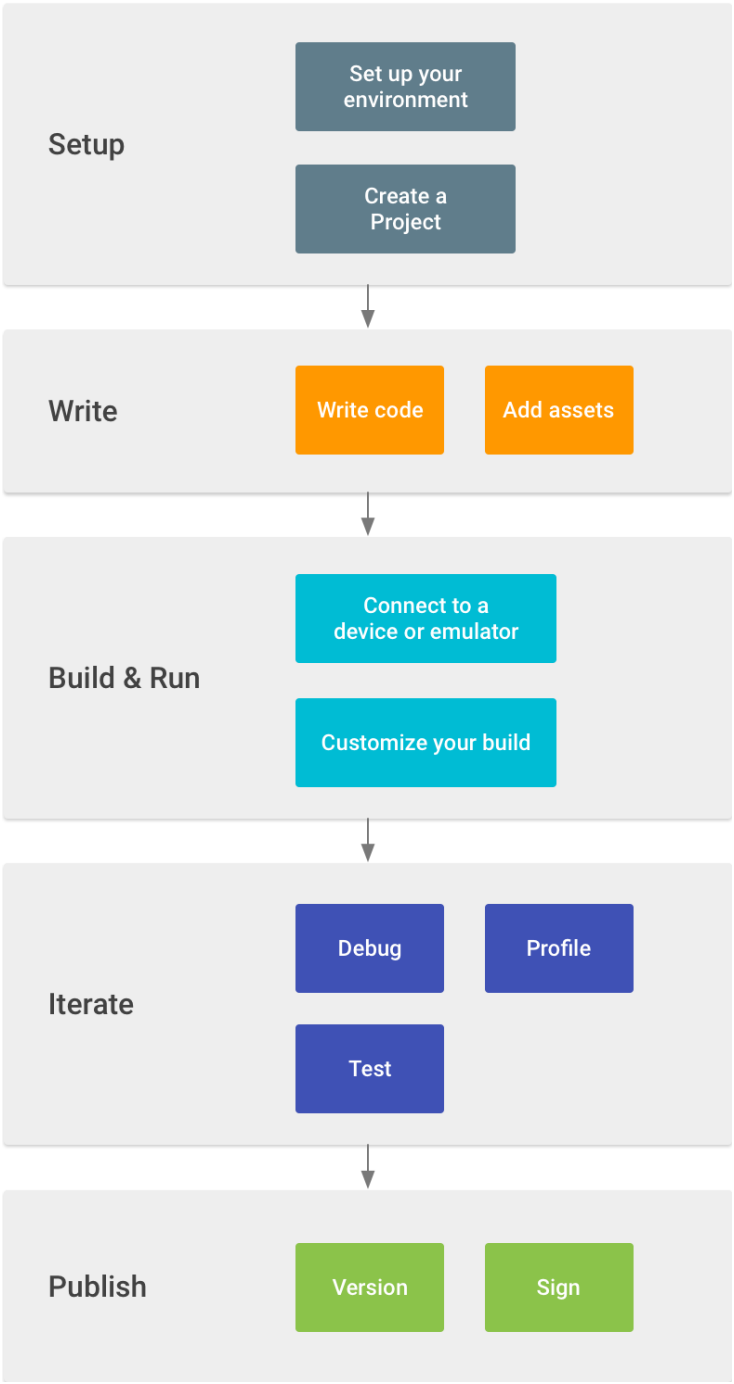


Figure 3-4: implementation stages

3.6.1 UI implementation

In order to implement the user interface, we need a basic understanding of the various layouts, a component that used in the xml files where the user interface is defined. The android studio visual editor is going to be used for the implementation of the user interface. The prototypes are going to be used as a guide for the structure of the various components inside layouts. The basic components can be found, and analyzed during the implementation phase

3.6.1.1 Material design

Material design is a design language which created by google and allows to create beautiful products using various material components. It focuses on a simple flat look which is responsive, moves and evolves while the user interact with the device and performs tasks. Seams, edges and shadows give meaning and show where the user can touch and interact with the interface. It can be used during android implementation. Everything that we need to build our application regarding material design can be found at official google developers website (Material Design, 2019).

3.6.1.2 Layouts

A layout is a container which defines the architecture for a user interface in our app, like in an activity or a component of UI. There are many layouts to choose from when we are designing the application. All the layout elements are based in a view and view group hierarchy.

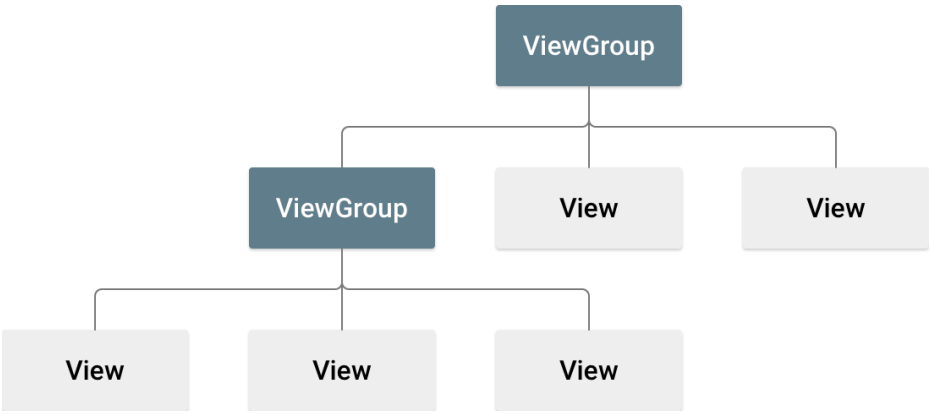


Figure 3.5: layouts

The viewgroup is a parent class of any view. It represents an object which is a container for other views. A View on the other hand draws something that user can see and interact with and is the basic building block for UI elements. Furthermore, the viewgroup is invisible and it's a container for view and other view group objects. The View objects are called "widgets". (Text View or Button). TheViewGroup objects are called "layouts" giving layout structure (Android Developers, 2019).

linear layout

The LinearLayout aligns all children in a single orientation, vertically or horizontally which means that the views will be either all in one row or on one column. The direction of the layout can be defined by attribute android:orientation (Android Developers, 2019).

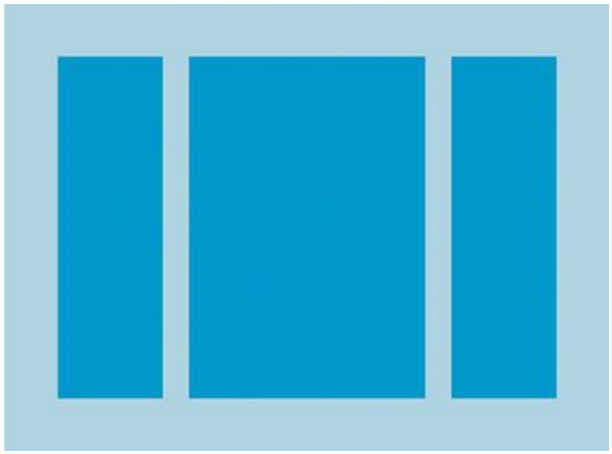


Figure 3.6: linear layout

Relative layout

The RelativeLayout displays child views in relative to sibling elements. For example, below another view, or to the left, or relative to the parent RelativeLayout. The position of each view can be specified as relative to sibling elements (such as to the left-of or below another view) or in positions relative to the parent RelativeLayout Using the centerinParent attribute a relative layout can center a child view. (Android Developers, 2019)

Figure 3.7: Relative layout

3.6.2 Logic implementation

Before we proceed to the best practices and methodology, we are going to see the Basic architecture of android operating system. As we see in the diagram in figure at the bottom of the software stack of android operating system is the Linux kernel which is optimized for mobile operating systems, in order to be effective in devices that have limited Cpu or memory. The use of Linux kernel makes android secure and take advantages of security features, allowing at the same time manufactures to

develop drivers. The hardware abstraction layer give access for hardware to java api framework (drivers for audio, camera, display etc.) On the top of the kernel we can find the android runtime with a set of core libraries which support behaviors of the operating system. Android run time run many virtual machines on devices with low memory. There is also the dalvik virtual machine which replaces the JVM and is designed for small devices. The byte code that dalvik read is different from the byte code which generated by java compilers. The next level is the application framework (java api framework) with native libraries and system components written in c and c++, which allows us to create android apps through apis written in java. These apis are the building blocks for android apps (view system, notification manager, activity manager) and control various components of the app like activities, which are the screens that the user can see, notifications, content providers telephone management and many more. At the top are the android applications or custom apps that we can download from android market or install manually to our phones (Android Developers, 2019).

3.6.2.1 Main components of android project

In order to proceed with the implementation first we should examine the basic structure of an android project. In android studio a project contains everything is required for an app like source code, test code, build configurations etc. The files are organized in android studio view by modules and file types:

- Manifests: contain the androidmanifest.xml
- Java: contains the java source code
- Res: contain the resources like UI strings, bitmap images, xml layouts.

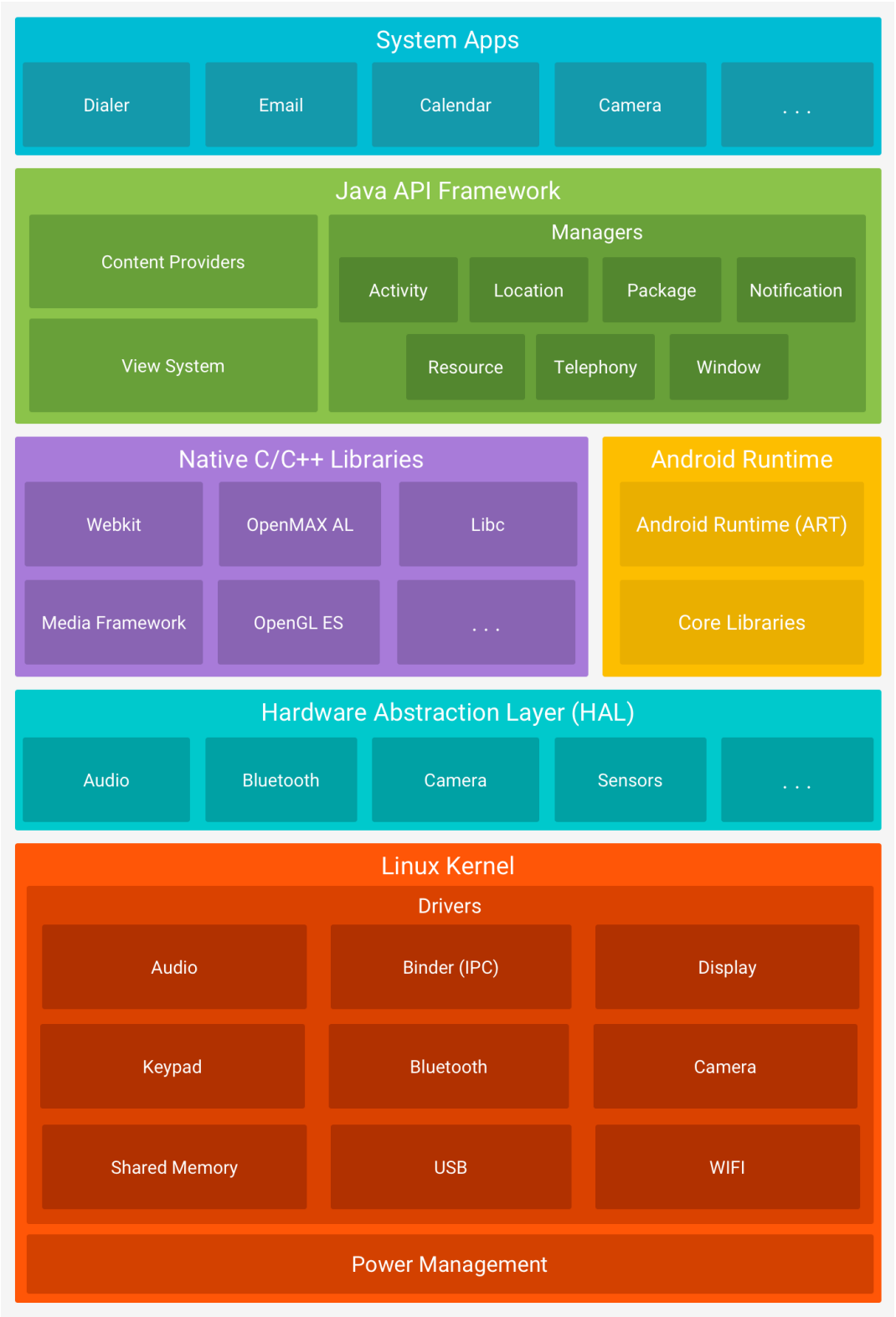


Figure 3.8: Android architecture

Activity

Activities and fragments are the most important parts of android application and illustrate a screen that the user will see, with the fragment being a kind of sub

screen, or an independent part of the screen. The lifecycle activity that's is going to described next ties together these fundamental parts.

Activity lifecycle

While the user navigates from one activity or fragment to another, the activity instances transitions through various stages of activity lifecycle.

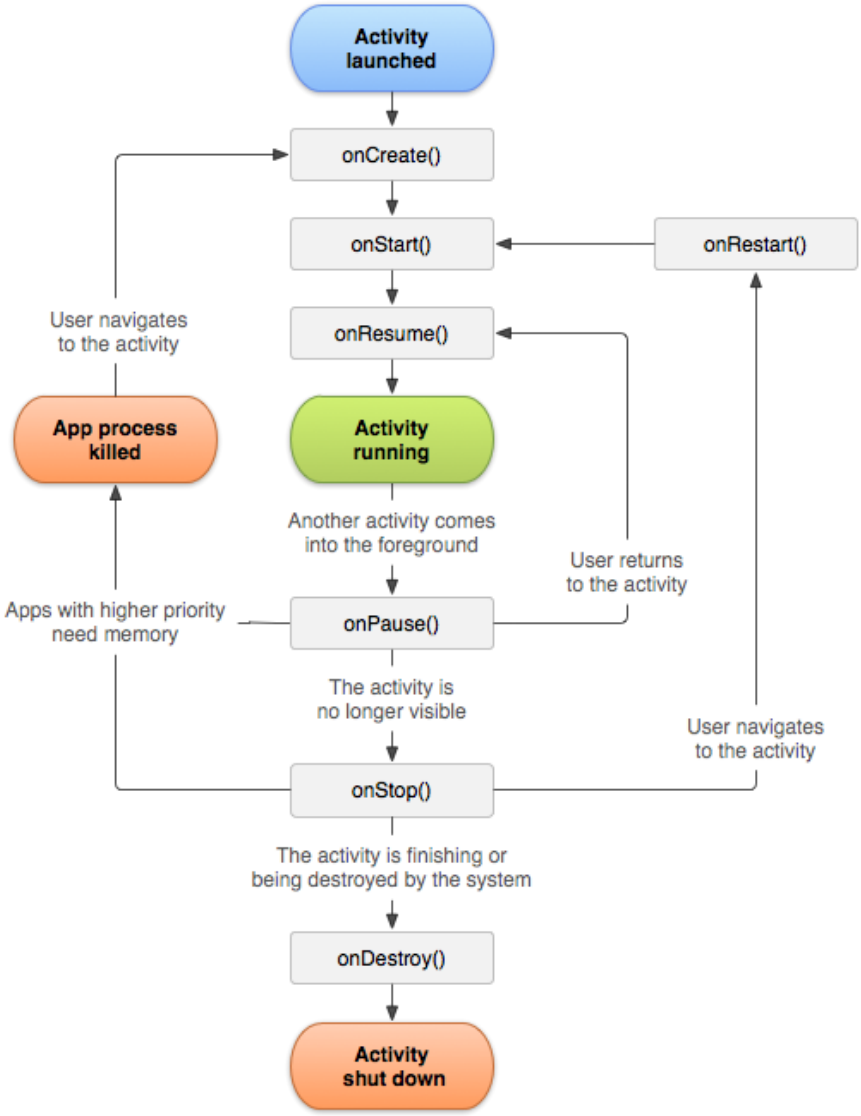


Figure 3.9: Activity lifecycle

The activity class has six callbacks to navigate between the various stages

onCreate(): This state is used to start an activity, or initialize it with oncreate(Bundle).In this method we perform the startup logic of our application.

onStart():This state makes the activity visible to the user, initializing the code that maintains the user interface.

onResume(): Here the activity enters the resumed state, and it's the state in which the user interacts with the app.

onPause(): This method is called when the user leaves the activity. In case of an event which interrupts the activity, the activity enters the paused state.

onStop(): The activity is no longer visible to the user. This for example happens when a new activity covers the screen.

onDestroy(). This state is called before the activity is destroyed when the activity is finishing. Each of these callbacks, are called by the system each time an activity enters a new state. (Android Developers, 2019)

3.6.2.2 Fragments

A fragment has its own lifecycle even if it must be hosted by an activity and its very powerful for a user interface design. Multiple fragments can be combined in a single activity. It relates to the activity that hosts it, but it can be shared by different activities. An example of a fragment is date picker and received its own events, enabling the user to input a date.it can be added or removed when the activity is running. (Android Developers, 2019)

3.6.2.3 Architectural Principles

Here we are going to discuss recommended architecture and best practices in order to build our app. Android app have a much complex structure than desktop apps. An android app contains various app components like fragments, activities, broadcast receivers, content providers, and services. All these components are declared in manifest file and android system uses this file to decide the overall user experience.

3.6.2.4 Separation of concerns

A big problem when developing an android application is that at the end the activity or the fragment doing too much containing complex code, making sometimes even impossible to understand the code, or fix something. The code is difficult to be understood to maintain and refactor. The activity or fragment are user interface-based classes and should only contain logic that handles user interface and operating system instructions like glue classes which represent the android operating system and the app. So, for a more effective application we should minimize the dependency on them.

3.6.2.5 Model View Presenter

MVP is an architectural pattern which helps in "separation of concerns". Decoupling some of the parts from each other The Model view presenter helps the developer giving him an easier way to think about the structure of the application. Furthermore, is a maintainable codebase and provides testability and modularity.

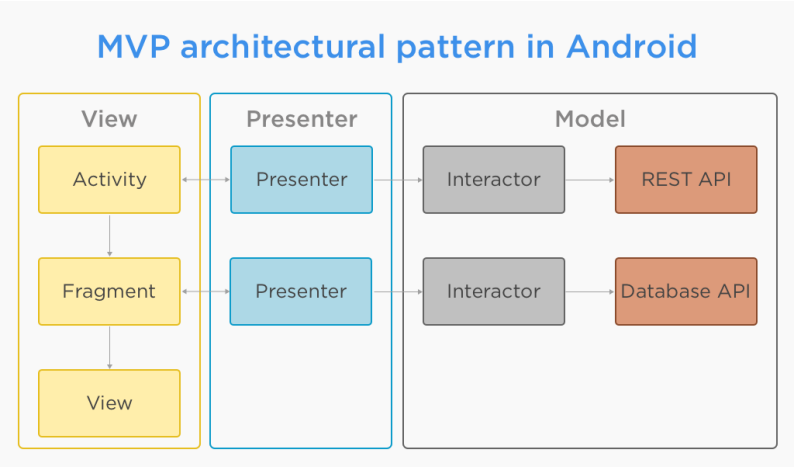


Figure 3.10: Model View Presenter

MVP Three main components

- Models: The model contains the data that will be displayed and is usually fetched from a local database or from the internet. Afterwards the data (using small simple classes) can be used by other components.
- Views: The views handle the interaction with the screen and what the user can see.It should not contain any business logic and should be responsible only for displaying content making it a more lightweight component compared to the controllers.
- Controllers: Controllers provide a way to connect views with models. The model is updated by the controller when something changes in the view. The same applies for the model. The controller will update the view when the model change. (raywenderlich.com, 2019)

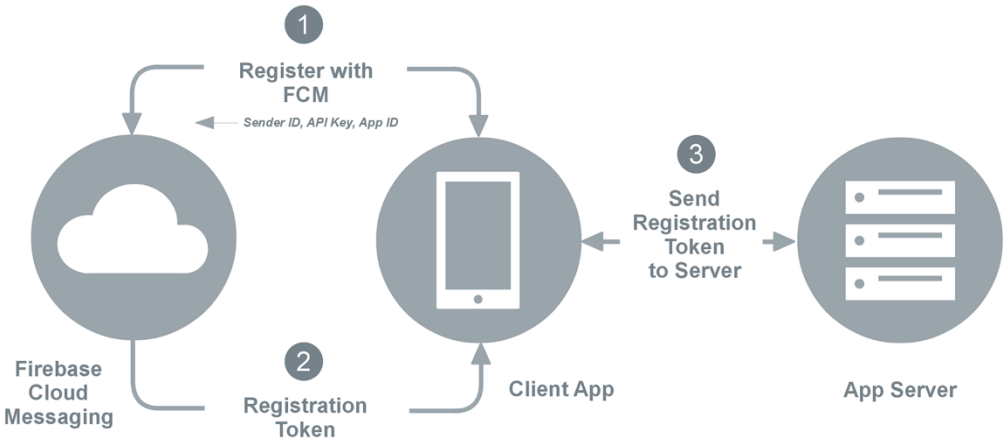


Figure 3.11: firebase

3.7.2.7 Firebase

Google firebase cloud solution which is based on google infrastructure Is a web application and mobile development platform. Gives the ability to the developers to focus on their user, supporting basic functionality like messaging, crash reporting, databases, analytics authentication, cloud storage, cloud functions Real-time database, test lab, in-app messaging, cloud messaging and many more. This functionality and technology are going to be used for this application. The app will be written using backend data storage, user event logging, and Realtime synchronization using Firebase. It provides JSON database where you can read and store data, and it hasn't rest method (Get, Post) working directly with the database. The firebase is like an actual database, as it stores files in documents, and allow queries. It is also a Realtime database which stores data as a big JSON object

3.7 Plan for Testing and evaluation

During the testing phase the firebase robo test will be used. Robo test is a tool which is available in firebase admin page under Firebase Test which is used to analyze the structure of the app's user interface simulating user activities. Robo test can capture logs, save screenshots of the application even save a video file from following the script that was used, showing the user operations that were performed. Analyze the logs screenshots and video we can fix bugs and find the root cause which is the reason that the app crashes. In order to create a script which will follow the robo to test the application we will use android studio. Under tools->firebase there is an option to record robo script and use it to guide robo test. The robo script will be saved in a Jason file which then we will upload to robo test page. The robo test script will be recorded in an external android device and not in a virtual machine. In order to create a virtual representation of robo crawl test lab uses the stats creating a crawl graph (Softwaretestinghelp.com, 2019).

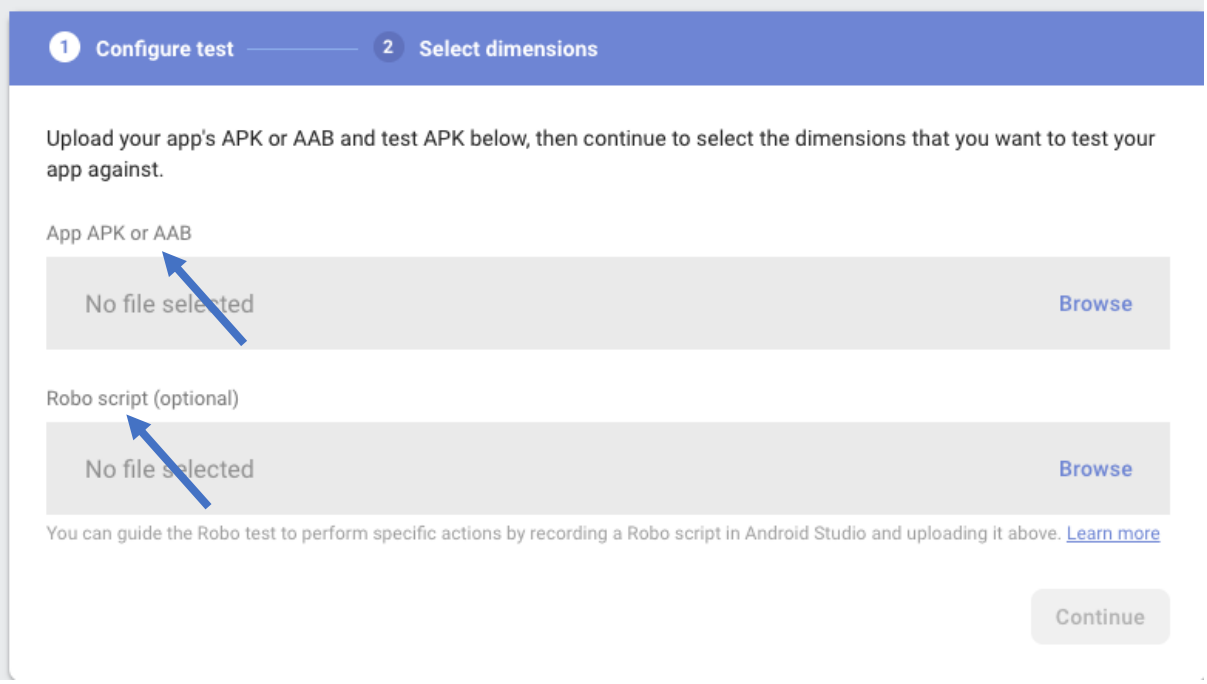


Figure: 3.12 configure Robo test

3.7.1 Robo test

In order to interpret the results robo records stats in each test crawl. The following stats actions, activities, and screens are going to be analyzed:

- Actions: It calculates the total number of actions that were performed during the crawl (Robo script actions)
- Activities: the total number of activities that were tested during the crawl.
- Screens: the total number of screens which were visited during the crawl.

All the files and output of the test lab will be included in the projects folder in a zip file video (video screenshots and the Jason file which was used as a script).

3.7.2 Evaluation

Heuristic evaluation will be used during evaluation of usability of the application for its simplicity and cost efficiency compared with other evaluation methods, in order to identify any issues that affect usability. The evaluation will be done by two evaluators using the application apk in an android device. The first evaluator will be the current's project researcher and the second evaluator will be a Phd computer science student from university of Strathclyde. The evaluators will inspect the user interface based on the 10 Nielsen heuristics for interaction design.

3.7.2.1 Evaluators process

First, we should give the evaluators a couple of tasks to use and stepping through carefully several times. They will examine details, flow and architecture focusing on 10 Nielsen principles. The violations of heuristics will be used to redesign a future version of the current application.

3.7.2.2 Pre-evaluation training

We will give the evaluators the domain knowledge that is needed. The evaluators are experts, so no assistance needed. No scenarios will be used (walk up and use)

3.7.2.3 Evaluation

The individuals evaluate and then aggregate results. They will use the application freely to gain a feel for the methods of interaction and the scope, identifying specific elements that they want to evaluate. Each evaluator will produce a list of problems explaining why, with reference to heuristic. Problems may be found on a single location of UI, problem with overall structure of UI, or something that is missing

3.7.2.4 Severity ratings

1. No usability problem
2. cosmetic problem
3. insignificant usability problem
4. significant usability problem; important to fix
5. usability disaster; must be fixed

3.7.2.5 Debriefing

Conduct with evaluators, observers, and development team members discussing general characteristics of UI collate their findings and establish a complete list of problems. Then suggest potential improvements to address major usability problems using brainstorm solutions based on the heuristics.

3.7.2.6 Analysis of 10 Nielsen's heuristics

- Visibility of system status.
    - Users should always be aware of the system status with efficient feedback Interactions. For example, when a phone is charging the charging icon is on the screen, or when there is no internet connection on internet explorer.

- Match between system and the real world.
    - The designers should make the system communicating with a familiar way for the user, using language and concepts that would find in real world, making the system easier to use. For example, a cart system reflecting the same shopping experience for the user.
- User control and freedom.
    - Users should have the freedom where backward steps are possible like Undo and redo actions with quick exit option on the menu bar.

- Consistency and standards.
    - It is essential to have a consistent user interface using graphic and terminology which is maintained across similar platforms. An example is a responsive site that dynamically changes content size across multiple platforms.

- Error prevention.
    - Preventing error from happening in the first place it's not the same as error handling. The systems should be designed so that potential error kept to a minimum. For example, flags can be used to prevent user enter an invalid input like google uses suggestions to inform the user for spelling mistakes.

- Recognition rather than recall.
    - The user should not have to remember information, and instructions of system should be visible. Generally, it's much better to use in our application memory recognition instead of memory recall. For example, in many sites we have breadcrumbs, which is a visible string of links which shows the current position within a website.so the user recognizes how he get there instead of remembering.

- Flexibility and efficiency of use.
    - The system should be flexible and efficient for advanced users, as well as for beginners. Generally, less interactions allow faster navigation. The interface should be easily customizable according to user needs. Using function keys, macros, of abbreviations this can be feasible, so that frequent actions are easily achieved. Another example is while a software is being installed the user is asked if wants to proceed with the installation or go to a custom installation for advanced features.

- Aesthetic and minimalist design.
    - Only necessary information and not irrelevant should be displayed for the current tasks, while giving a visible clear and unambiguous navigation to the user. For example, apple provides only the basic information of feature hiding additional information under "More".

- Help users recognize, diagnose and recover from errors.
    - An understandable language should be used for errors and not technical terminology, ensuring that nothing gets lost in translation. An example is when in a login page the username or password is wrong,

but the system instead gives information which of the two or if both are wrong it says “username or password is incorrect”

- Help and documentation.
- Even if it’s better the system to be used without documentation and be user friendly, sometimes is necessary to provide documentation. Help should be easily located, and the information should be easy to search and not be too large (Mattsoave.com, 2019)

Chapter 4: Analysis

In analysis chapter we are starting our software lifecycle. First, the design phase using methods and techniques that we examined on methodology chapter will give us a requirements specification document and a high-fidelity prototype which will illustrate the user interface. In implementation first we need to implement and analyze the user interface and then we are going to examine the architecture of the application, using code snippets and flowcharts. First, we examine how firebase is going to be installed in our project and then we will examine the project structure and more specifically (Activities, adapters, Fragments, model classes). At the end we are going to view the technologies and the libraries that were used during the implementation. Next, we are going to test our application using robo from firebase testlab using scenarios that will be captures in android studio using an android phone. We are going to test each scenario and if pass or fail each stage. Final we are going to evaluate the usability of application using Nielsen heuristics.

4.1 Design

During the design phase the basic methodology that was analyzed in the methodology chapter is going to be applied. First, stakeholders need to be addressed and analyzed, as well as Define user context and tasks from literature review.

4.1.1 Stakeholders

The key stakeholders are children and parents. The child needs an application to have continuous connection with a parent and communicate with him whenever they want using text messages and recording information regarding its health and emotional wellness and feelings, which will be accessed by its parent. The parents need to have access on information from their mobile phones and communicate or answer children’s incoming messages. Both children and parents need a continuous connection with predefined features. Software development team. Analysts, programmers, designers, quality assurance representatives, testers, maintainers, project managers, trainers. The developers team need information about the

requirements of the application, specific technologies and restrictions regarding the functionality of the application.

4.1.2 User context and tasks

The main users of the application are children and their parents. The children users should be from 6 years up to 17 years old. They should have a basic understanding of how a mobile phone works in order to interact with the application successfully. Could be of any gender and from any environmental or family context, without specific criteria or limitations in parenting behavior, family function or parenting style. The users could be of any ethnicity. The users should have access to internet connection on their mobile phones in order to use the application. Parents needs an application that will inform them with graphical statistics regarding their child wellness. Information like mood, water intake, sleep and health will be available to parents in real time. They will also can send and receive live messages and predefined messages. Children need an application that will provide them continuous connection with their parent. They must record their information provided by the application, and send predefined messages, or Realtime messages providing them a sense of security and attachment.

4.1.3 Personas

Alex (Child)

Figure 4-1: Child persona

Alex is 10 years old. He tries to deal with his parent's divorce the last two years. He lives with his mother, but he feels alone since his father left home and he is trying to cope with anxiety, feelings of anger and sadness. Sometimes he thinks that his parent's divorce it's his fault and wish he could have prevented arguments between his parents. His sleep has been affected the last year since the divorce and his academic performance has been declined. Alex needs to communicate with his father more often and needs to feel that he understands him. However, he finds it difficult to share his feelings with him when they are together. Alex's parents are very worried with the problems that their child encounters, and especially his father who wants to find a way to be informed about his son mental and physical health. He uses a mobile phone and communicates with his father via messenger applications, but he needs something that would allow him to share his feelings and share information with his father without need to send him messages because it makes him feel uncomfortable.

Mary(child)

Figure 4-2: child persona

Mary is13 years old. She is coming from a wealthy family. She feels that her parents don't have time for her as they are very busy, working all the time. She needs to speak to them about the problems she faces especially in school regarding the bullying that anticipate because she is overweight. Mary is attending sessions with a psychologist and her parents try to help her, as her sleep and her mood are not very good. She tries to find a way to send how she is feeling, but using messenger is very annoying for her. She wants to be sure that her parents will have her attention, having access to her moods and her feelings. She needs an application that will be capable to record her feelings and moods, which will be accessible by her parents without her participation in order to feel safe.

Georgia (child)

Georgia is 15 years old. She feels that her father is overprotective as he sends messages through messenger to her all the time. She finds messenger app annoying as her phones accept notifications all the time from her parent with the characteristic sound of Facebook messenger. She wants to chat on messenger only with her friends. She Wants a way to communicate which her father only when is important. She doesn't want to tell how she feels all the time, but she doesn't have problem to express her feelings.

Steve (Father)

His little daughter has diabetes type 1. He needs a way to have access to his daughter information, like feelings or moods without being overprotective as she already anticipates some psychological issues related to her health problem. It's essential for him to have daily information for the psychological and health profile of his child since he cooperates with doctors and the information to them are valuable.

Cynthia and george(mother)

Cynthia and george are parents of 2 boys. Their youngest boy (10 years old) expresses aggressive behavior toward his friends, and he is not very social. His

relationship with his parents is not bad but his mother trying to understand why her child is so aggressive. The paediatrician suggested to begin a medication therapy for ADHD, but their parents think that the boy doesn't need the medication, he just needs a closure connection with them in order to guide him how to respond in stressful situations. They need to find ways that will have access to information about their child emotional well-being before they decide if they are going to put him on medication.

4.1.4 Market analysis

As we examined in literature review mutual respect, listening with participation, availability, emotion, and understanding are essentials for a healthy communication. In this section a market research is going to be conducted. Searching the android market someone can find many apps that supports the situations that was described above, implementing specific functionality and features. The problem that this project is trying to solve is find ways to enhance healthy communication between parents and children, using various technologies, like android operating system components that are supported by the android platform. The people who will benefit from this application will be the parents and children. The application will be used to capture mood, water intake, and data of the child that will be then be accessible to be illustrated by graphs and visualizations to the parents, to make assumptions and judgements for their child mental and physical health. Other functionalities this application will support is a chat messenger, and a way to express feeling or thoughts using predefined messages. The parent would be able to access the information from his mobile device and be available wherever he is. The application will be used in everyday life and is intended to become a part of their daily routine trying to enhance healthy communication between parent child and create healthy bonds.

In order to research the market and find applications that support the functionality that was mentioned in the previous paragraph, the following keywords and phrases were used, "mood track, sleep track, monitor app". The market research has also as its scope to make a research on the technologies, weaknesses and strengths of each app taking, ideas for the design process of the Parent-child application. During the market research the phrase "child track" gave the most relevant applications. In order to research all the features of the application that are going to be implemented different keywords were used.

Research keywords: child track, mood track, sleep track, child messenger

Applications Review

App name	Features	Technology	Review

Diary - Mood Tracker	Quick data entry, easy customization and the ability to correlate activities and moods. Edit delete mood, mood rating scale	Backup and restore data via Dropbox. pin-base passcode, Notification reminder	(4.2) Graph little problematic, more reminders instead of one, need google storage, Calendar time bug, Too general app
Mood Tracker	Simple and elegant, Facebook sharing, History	Graphs, Emoticons	(3.8) needs more emotion, need edit button, need for synchronize between devices
Free Mood Tracker	Track daily mood swings, Export graph to pdf, backup data, Free without advertisements	Graphs, calendar	(3.9) Need multiple entries in a day, Personalization of moods
UP! - Mood Tracker for Burnout and Bipolar	Bipolar mood tracker, automatic physical activity recordings, no registration needed	Graphs, emoticons, History, Pdf reports	(4.5) Need tracking meds, more moods, anger irritation needed, location needed, it is slowing my phone
Kid Finder	Live location on map, see whom the kid called last, see whether kid is on phone and what app is being used, See last app used by the kid	QR code,	4.4) trial, need sos button, tracking problem, not working properly,
Chat with parents	Audio messages, Family connect, sample messages,	Register with code	

Family Locator	Real-time location of your family & friends, sos with location, messages	Notifications, safe zones, groups, stolen phone, location history, GPS tracking functionality and internet mapping	4.0) Bug database “continually stating that both users are offline”. Position keeps shifting although I'm sitting still. Location updating is very slow, not update the locations frequently
Track Child Location	-It will save child location on a regular interval. -Parent can add multiple children. -Parent can see location of child/children on map.	You can add one child in one device. GPS must be on in child device. Internet must be available in device.	4.4)
InNav Child			

Figure 4.3: Market research applications

The essential features that are supported from most of the applications that are available on the market are: Mood, food, sleep, water intake tracking. Some of the messenger apps supporting location messages (tracking location). Graphs and emoticons are essential for these kind of apps UI. Some of the application support push notification. The technologies that are used by most of the applications are:

Local Database: most of the application are using a local database, with data being stored in the local storage inside the phone’s storage. some of them support a backup feature, or a restore data via Dropbox.

Hardware: Gps functionality or location by WIFI/gsm is a must to send location to another user.

Other features: location history, pdf reports export, register with code by SMS, or QRcode instead of registering, pin base passcode.

Some of the flaws that many apps should improve:

- Location accuracy: There are any problems regarding the application location services. Most of them focus on delay of the location or false location.
- Simplicity: Graph are little problematic, and the users are not satisfied by the user experience.
- More reminders instead of one: Some application using reminders for tracking data to the app.
- Need google storage: The need for google storage, or cloud storage to transfer data through devices or backup and restore them.
- SOS button: a SOS button with predefined functionality
- More moods/ Personalization of moods:
- Need multiple entries in a day
- Many devices, many children: the need to support more than one child

There are some gaps observed in the market analysis that need to be analyzed in order to find new features for a new healthy communication application. Generally, there is an absence of child mood tracking applications specific for child target group. Water intake tracking for general user but not for a specific target group as it is mentioned. These applications are for general purpose and they aren't designed for a specific target group like children-parent. Most application are addressed to adults or generally to the public and not for a specific target group like children. In most of the application there is an absence of a sos button, or panic button. The parent should be available on his phone each time the child needs him. Using a panic or sos button the communication could be direct and immediate.

4.1.5 Hierarchical Task analysis

Child-Parent:

1. **In order to login manually**

- Select child or parent
- Type your email password
- Press sign in

2. **In order to login with google**

- Select child or parent
- Type your email
- Type your password
- Press login with Google

3. **In order to create account**

- Select child or parent
- Type your email
- Type your password
- Press Submit

4. **In order to logout**

- Click on logout
- Click ok on confirmation

Child:

5. **In order to capture mood**

- Click on capture mood
- choose mood
- Click ok

6. **In order to send predefined message**

- Click on send message
- select message from list

7. **In order to capture sleep**

- click on record sleep
- select hours and quality
- click submit

8. **In order to capture water intake**

- click on water intake
- select number of glasses
- click send

9. **In order to capture how you are feeling**

- click on how you are feeling
- select feeling

10. **In order to send location**

- click on send location

11. **In order to chat**

- Click on chat button
- Select parent
- Type your text
- click send

Parent:

12. **In order to track mood**

- click on track mood
- view moods
- view mood chart (Daily, weekly, monthly)

13. **In order to see predefined message**

- click on view messages sent
- view messages

14. **In order to track sleep**

- click on track sleep
- view sleep entries
- click on view chart
- view sleep charts (Daily, weekly, monthly)

15. **In order to track water intake**

- click on track water intake
- View water intake
- Click on View chart
- view water intake charts (Daily, weekly, monthly)

16. **In order to track how you are feeling**

- click on child's health track
- view health
- view chart

17. **In order to see locations received**

- click on locations sent
- click on specific location
- view location on map

18. **In order to chat**

- click on chat
- select child

- type message
- click send

19. In order to delete history activity

- Click delete on specific activity
- Click confirm

4.1.6 Use cases

Parent user use cases

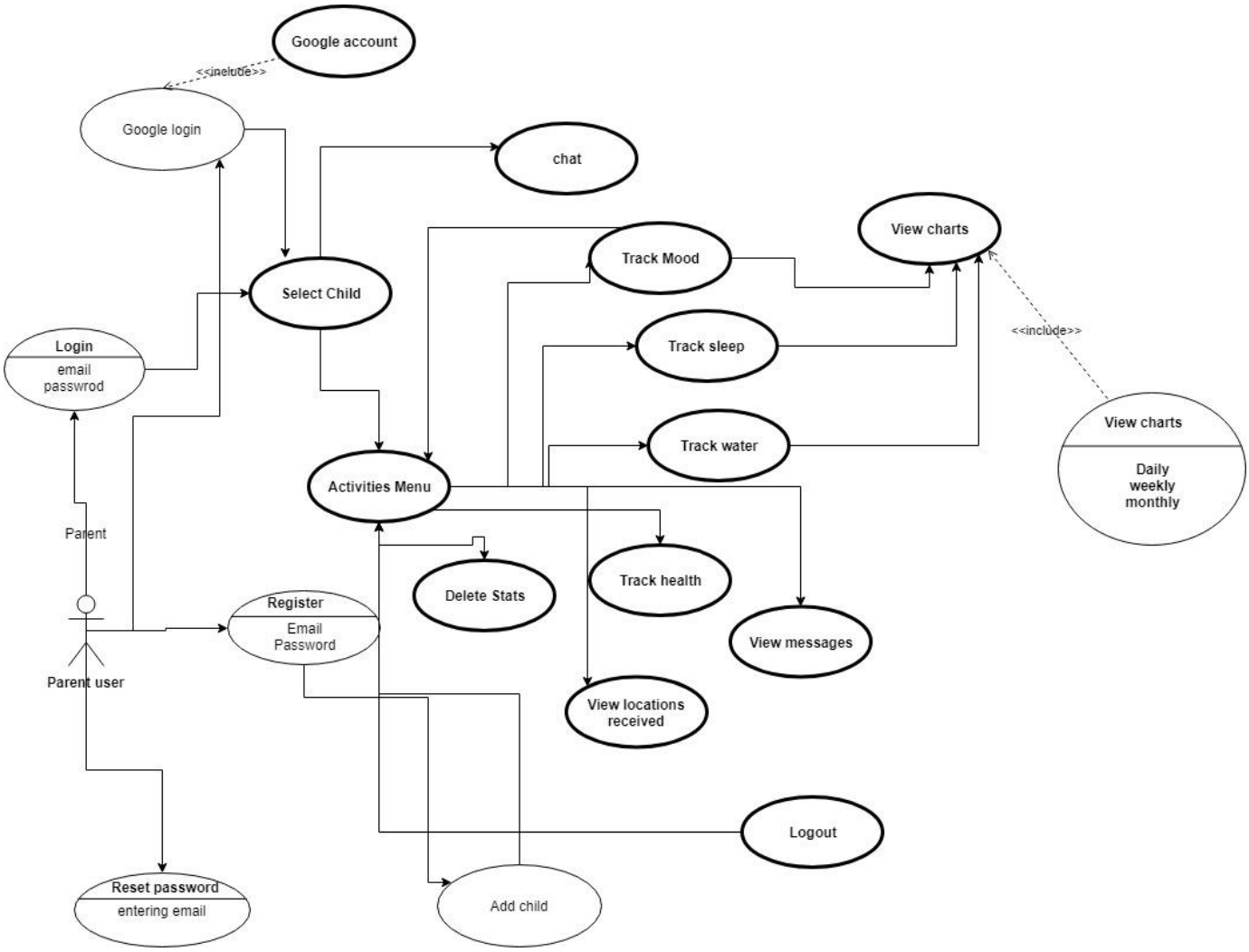


Figure 4.4 Parent Uml diagram

Child user use cases

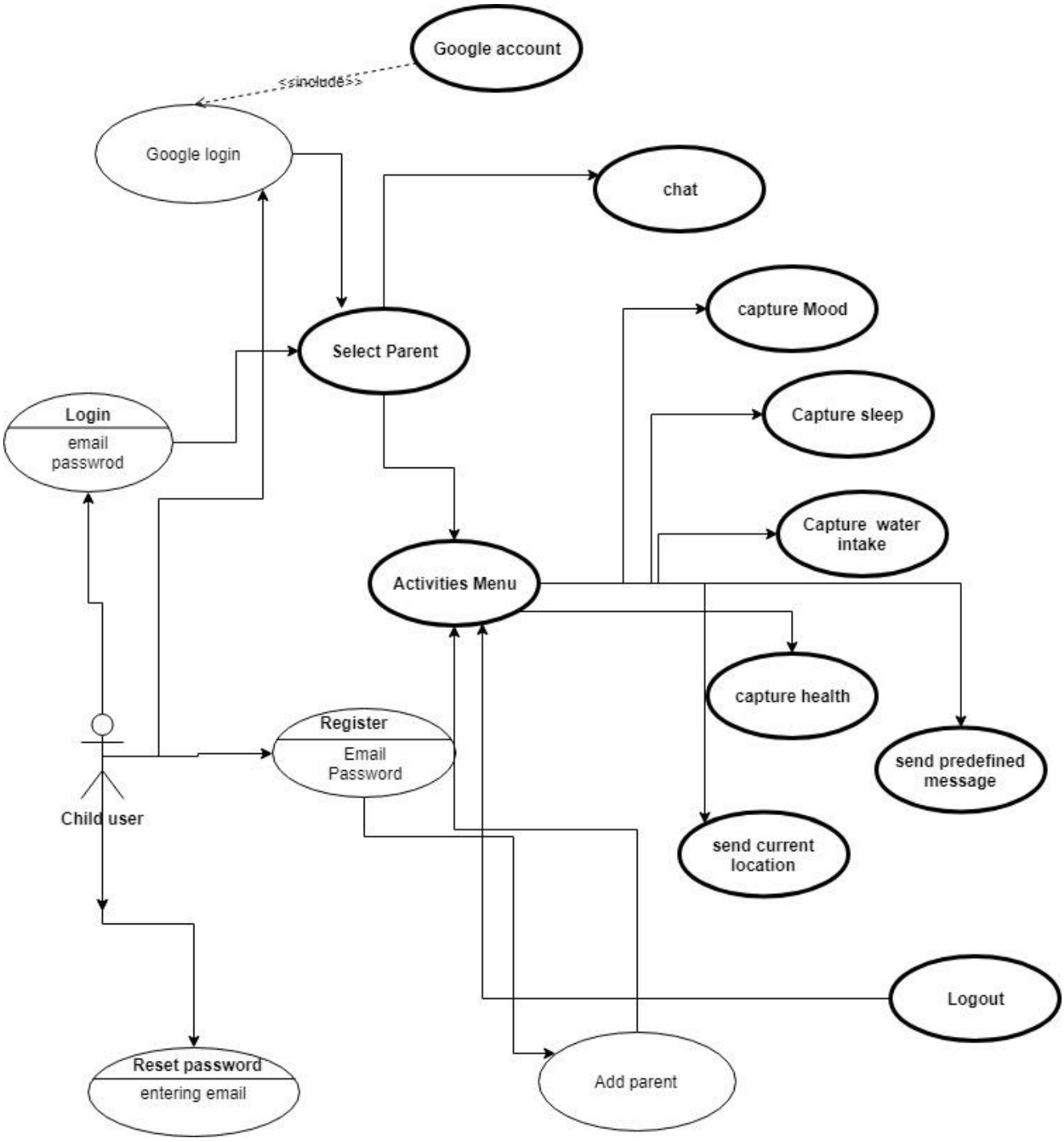


Figure 4.5: Child uml diagram

4.1.7 Main functionality of the application

The application will provide different user interface for the parent and different for the child. When the application starts will prompt to select the parent or child mode. When the parent mode is selected, and the username-password is correct then the

parent goes to the main screen of the app. From there he can choose to see statistics about the child's information like mood, sleep etc. He also can receive messages from the child and answer. When the child selects the child mode the login screen appears. If the username and password are correct it goes to the main screen. From there it can track mood, food and water intake etc. There is also the ability to send predefined messages to the parent or open the messenger and send a message and its location. Each time the child track something like mood a push notification is sent to the parent.

Features overview of the app:

- Parents and children can login/signup using Google or traditional form filling
- Parents can add as many children as they want using children email address
- Children can add as many parents as they want using parent' email address
- App's dashboard should dynamically change according to user type i.e. the dashboard is different for both parents and children.
- Parents and children can chat to each other in real time.
- Children can send predefined messages, record their water intake, mood, sleep, send their and location.
- Parents can delete everything children have sent to them, including interactive daily, weekly and monthly charts.
- Parents receive notifications of their children' activity.

4.1.8 Functional Requirements

Child-Parent

1. (M) login manually
   The user should be able to log in using an email address and a password
2. (M) login with google
   The user should be able to log in using a google email address and a password
3. (M) create account
   The user should be able to register (create a new account) using an email and a password
4. (M) Password forgot
   The user should be able to retrieve his password and receiving it to his email address which is registered with.
5. (M)Log out
   The user should be able to logout when he presses the logout key.
6. (S)The user should be able to add more than one connection with more than one parent or more than one child.

Child

- (M)Add Parent
  The child should be able to add a parent after he creates a new account. The system should ask for the parent connection only when there is not a parent connected account to the child email account when the account is created.
- Capture mood

  The child should be able to capture its mood.

- (M)Send predefined message
  The child should be able to send predefined messages.

- (M)Capture sleep
  The child should be able to capture its sleep habits, entering hours of sleep and quality.

- (M)Capture water intake
  The child should be able to capture its water intake entering number of glasses.

- (M)Capture how he is feeling
  The child should be able to capture his feelings, and select by a predefined list, but choose only one each time and not send all the list.

- (M)Send location
  The child should be able to send its location.

- (M)Chat
  The child should be able to send and receive messages in messenger.

- (S)Receive push notifications for messages
  The child should be able to receive push notifications each time a new message comes

- •(S)Change profile picture
  The child should be able to upload its picture to his account profile

- (W) settings
  The child should be able to change settings regarding the predefined messages. There also should be an option to disable push notifications sound.

- (W) SOS Button

The child should be able to send an alert to the parent when presses the SOS button

Parent

- (M)Add child

  The parent should be able to add a child after he creates a new account. The system should ask for the child connection, only when there is not a child connected account to his email account, ore after registration.

- (M)Track mood
  The parent should be able to track his child mood and view statistics (Daily, Weekly, Monthly).

- (M)Delete mood statistics
  The parent should be able to track his child mood (Daily, Weekly, Monthly).

- (M)View messages sent
  The parent should be able to view messages sent.

- (M)Track sleep
  The parent should be able to track his child sleep and view statistics (Daily, Weekly, Monthly).

- (M)Track water intake
  The parent should be able to track his child water intake and view statistics (Daily, Weekly, Monthly).

- (M)Track how child is feeling
  The parent should be able to track his child mood and view statistics (Daily, Weekly, Monthly).

- (M)view locations received
  The parent should be able to track his child mood (Daily, Weekly, Monthly).

- (M)Chat
  The parent should be able to send and receive messages in messenger.

- (S)Receive push notifications for messages
  The parent should be able to receive push notifications each time a new message comes

- (S)Change profile picture
  The parent should be able to upload his profile picture

4.1.9 Non-Functional Requirements

- (M)The GUI should be the same on mobile phones, as well as on tablets, and easy to use from both low experience users and high experience users. All the screens should have a similar style.
- (M)The system should have fast response time
- (M)The system must support the implementation of new features.
- (M)The system must have not software bugs and should inform the user about errors. The system should be responsive to the user and his actions
- (M)The system should show clear and detailed notification messages to the user. The notifications should be clear. (error messages etc.)
- (M)The system should be able to run on most android versions and all devices

Hardware Interfaces

The app is designed for the Android phone and market.

Communications Interfaces

The application will utilize the networking hardware of the user's device through network communications services provided by the Android operating system. Network communications capability will be used to connect to remote databases and/or web services for authentication and importing the user's class schedule information.

Performance Requirements

System

The application will run on all Android devices running 4.1 (JellyBean) or Later. It will be around 10mb in size. The application will respond to the size of the screen and/or window the application is running in.

Response Time

The application should take less than 4 seconds when running on an Android phone and less than 8 second when on an emulator or tablet. The application will run fine until the user begins to multi-task between 3 or more processes.

Other non-functional attributes

6.1 Security

Users can authenticate by logging in using their username and password

6.4 Maintainability

The development team will follow best practices for clean code and software modularity in order to make the application as maintainable as possible.

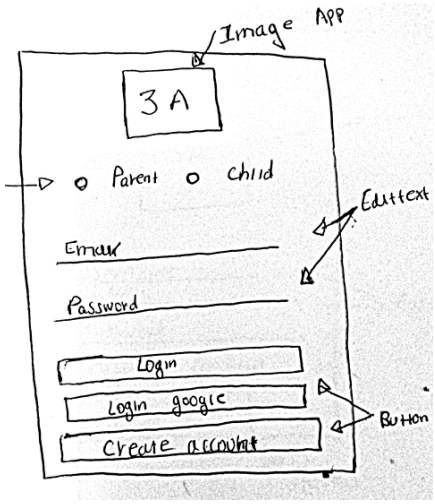
6.8 Application Affinity/Compatibility

The application shall be compatible with Android 4.1 or any later version.

4.1.10 Sketches

The user interface will be drawn in paper. During the process best practices of material design and instructions from google android developers were followed. The sketches show the various components as well, but not in depth. Description about the functionality of the prototype will be given later in the high fidelity prototype section.

Sketches samples



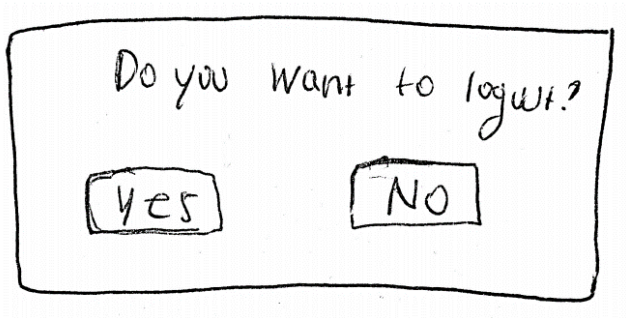


Figure 4.6: sketches 1

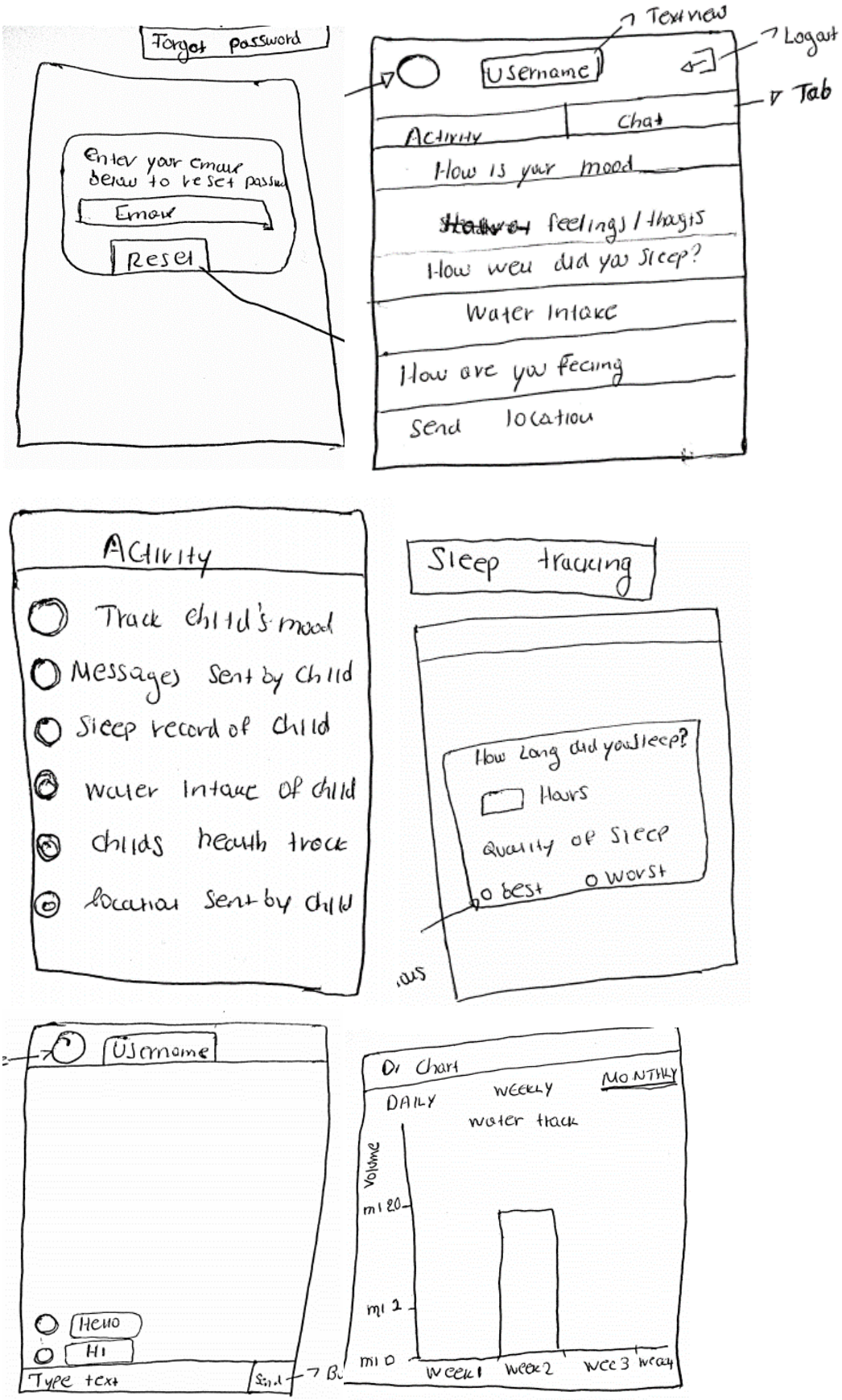


Figure 4.7: sketches 2

4.1.11 High fidelity Prototypes

Here we are going to see the high-fidelity prototypes which were designed based on sketches. The android studio visual editor was used to arrange the various components in the screen. android studio visual editor.

Login screen

Figure 4.8 login screen

This should be the main screen of the application. When the app starts this screen will appear on the screen. The user (Child or parent) should be able to select between child and parent mode. Then after entering his email and password he should press on the specific login button (Google login or login manually). In case the user has forgotten his password, by pressing the forgot password and typing his email the system should send him his password. The user should be able to create account clicking on create account button and by providing his email address and a password.

Signup screen

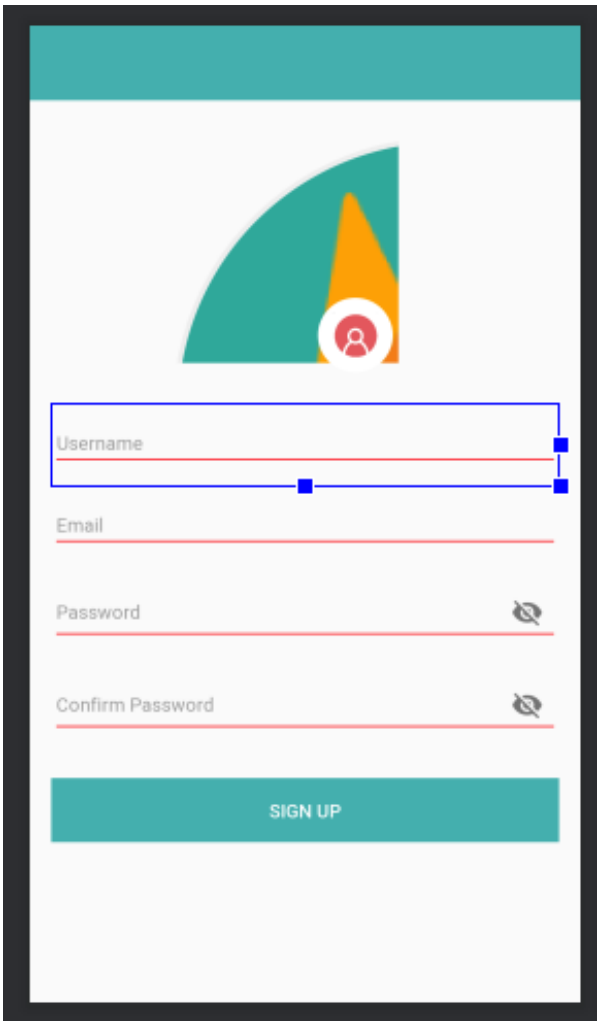


Figure 4.9: Signup screen

When the user (Child or parent) presses the signup button a screen should be appeared in order to put his username, his email, and his password (confirm his password), and then by clicking on signup the user should launch the main screen of the application which will prompt to add a parent or a child in order to continue.

Add parent/child after signup

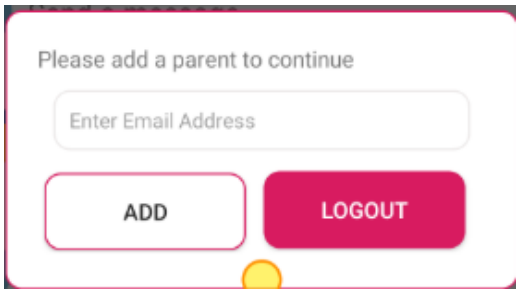


Figure 4.10: Add user

After the user has registered, or if there is no email connected to his account when the user logins, there will be a prompt to add email of user (Parent/Child).

Forgot password screen

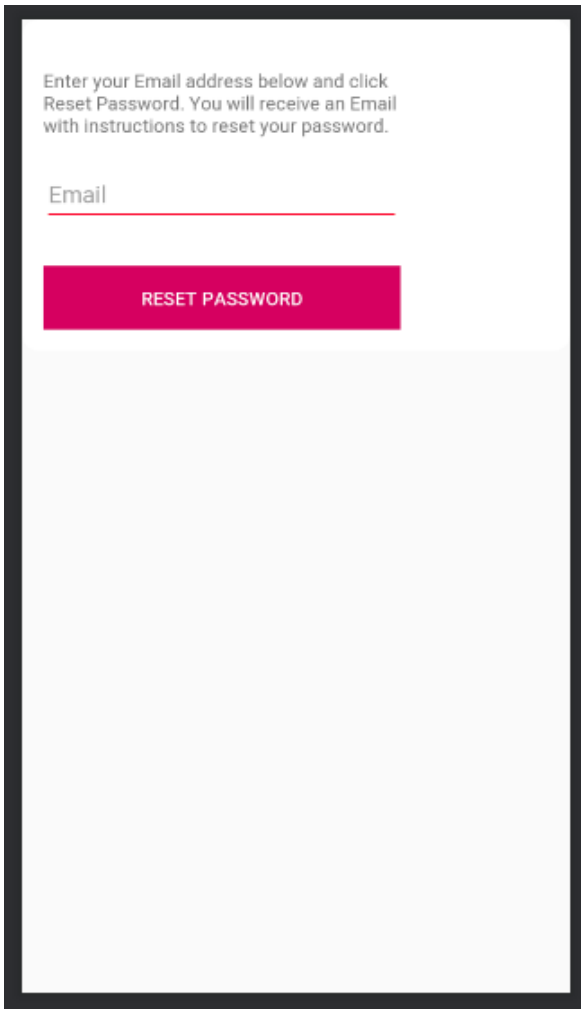


Figure 4.11: Forgot password screen

In case the user (Child or parent) has forgotten the password by entering his email address and clicking reset password should receive his password in his email.

Add second User screen

This screen allows the user to add another user (parent to add child and child add parent).

Logout

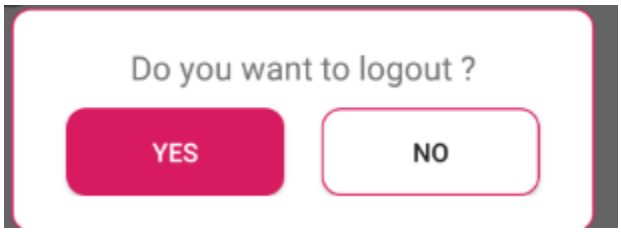


Figure 4.12: Logout

The user should be able to logout after pressing the logout button. Then in order to avoid selection by error he can choose yes or no.

Main screen child

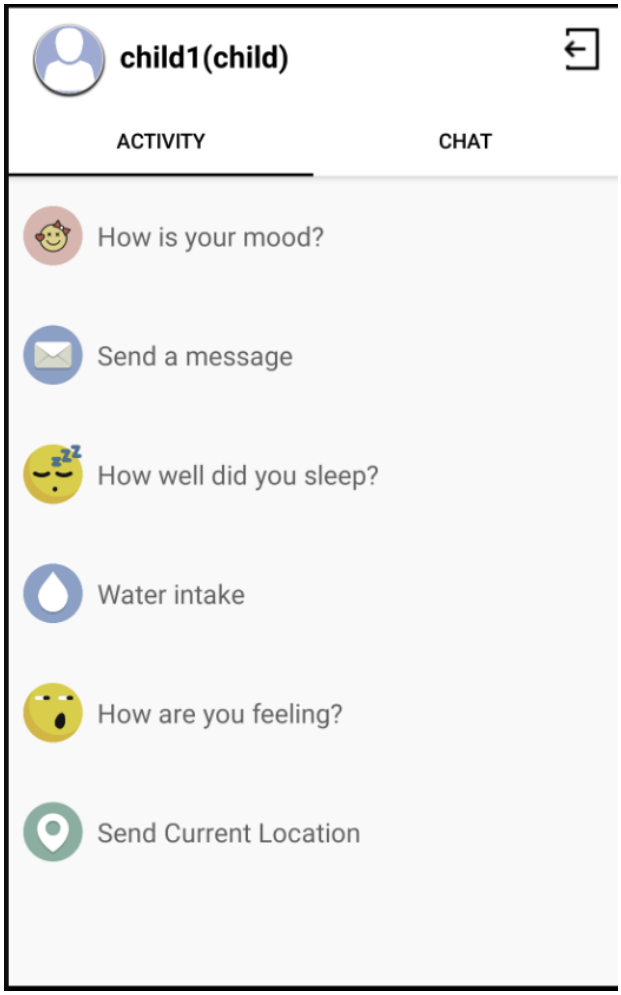


Figure 4.13: Main screen child

This is the main screen of the child. Here the child can record statistics regarding his mood, send predefined messages, capture the quality and hours of sleep, water intake volume, express how he is feeling, and send its current location to the parent.

Home screen child

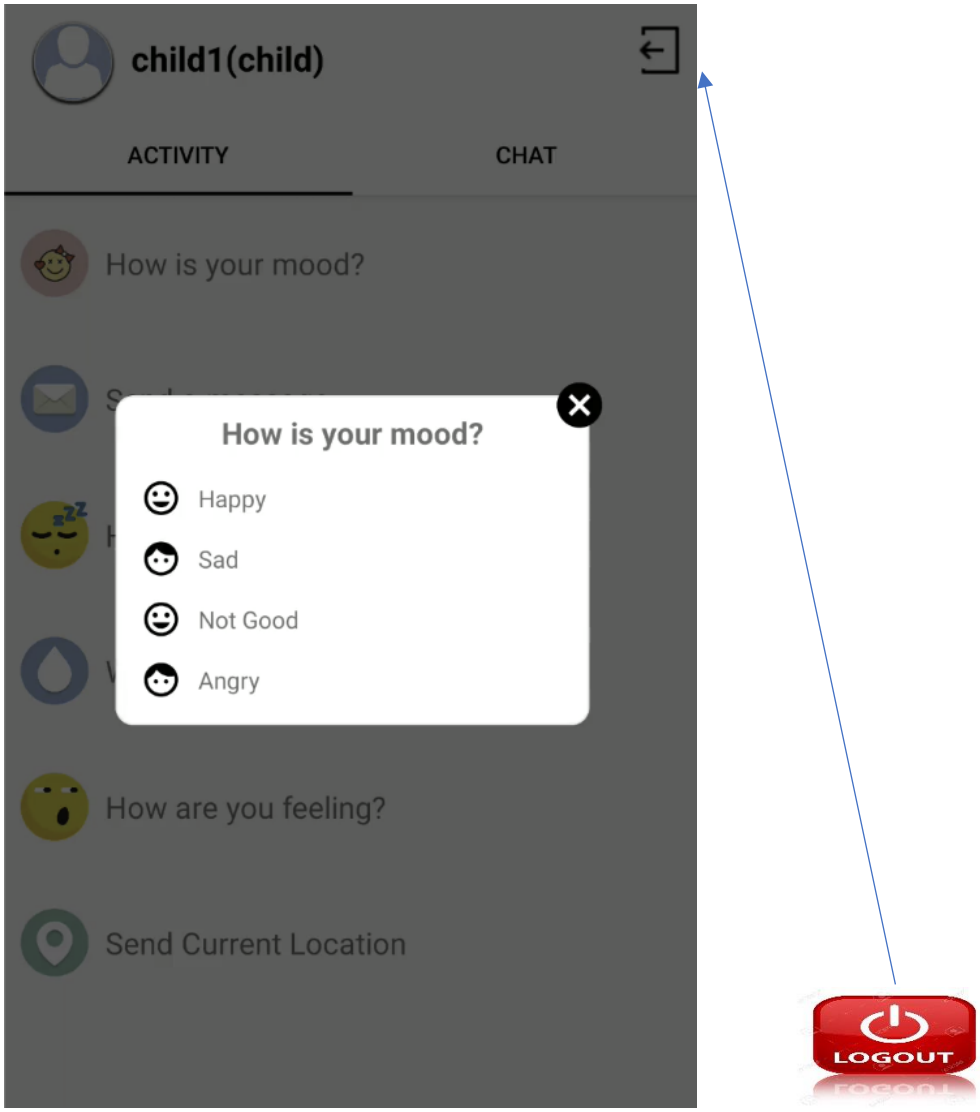


Figure 4.14 home child screen

The child after pressing on “how is your mood can select from a list of moods, by clicking on the specific mood.

Send predefined message

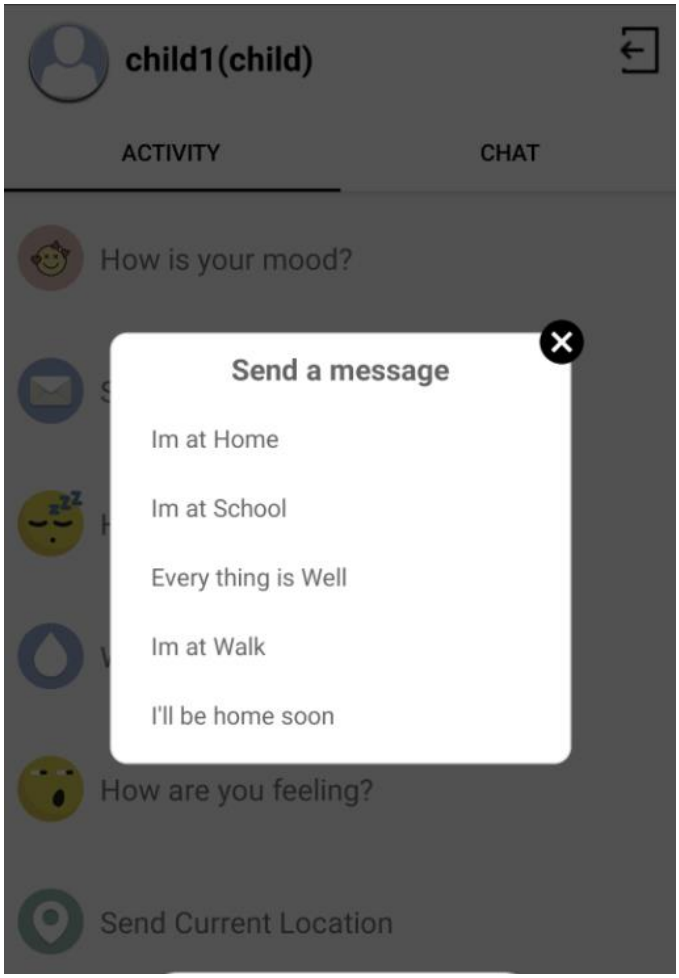


Figure 4.14: Send predefined message

The user can send a predefined message to his parent. The predefined messages are fixed and don’t change in the current version.

Capture sleep

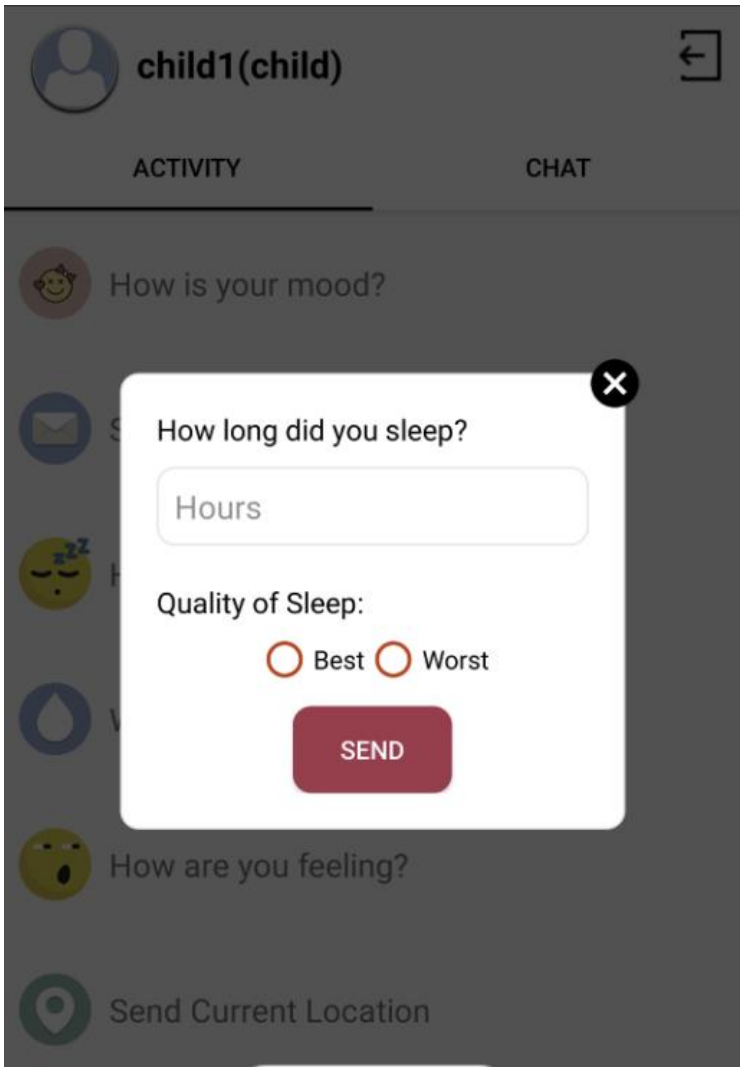


Figure 4.15: Capture sleep

The child can capture the number of hours that he slept, and the quality of the sleep. Then by clicking on send the information are saved on the database and will be accessible by parent.

Capture water intake

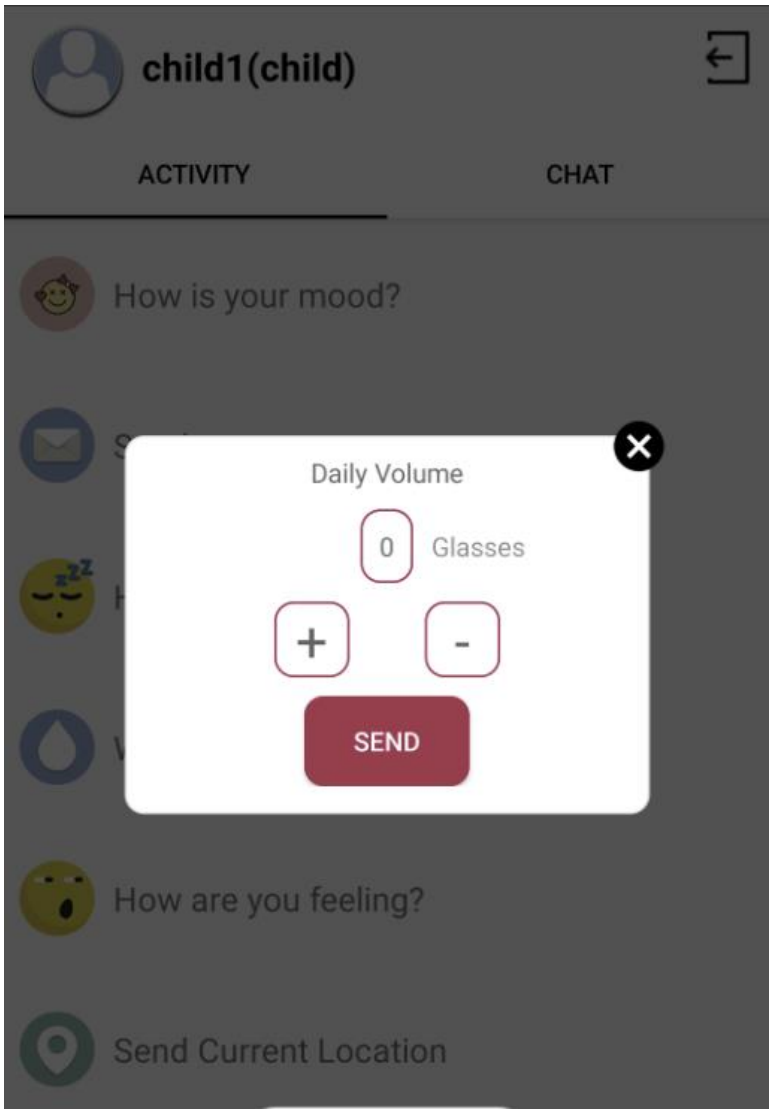


Figure 4.16: Capture water intake

The child can capture the volume of glasses of water daily, incrementing or decrementing the number by choosing (+, -).

Capture how you are feeling

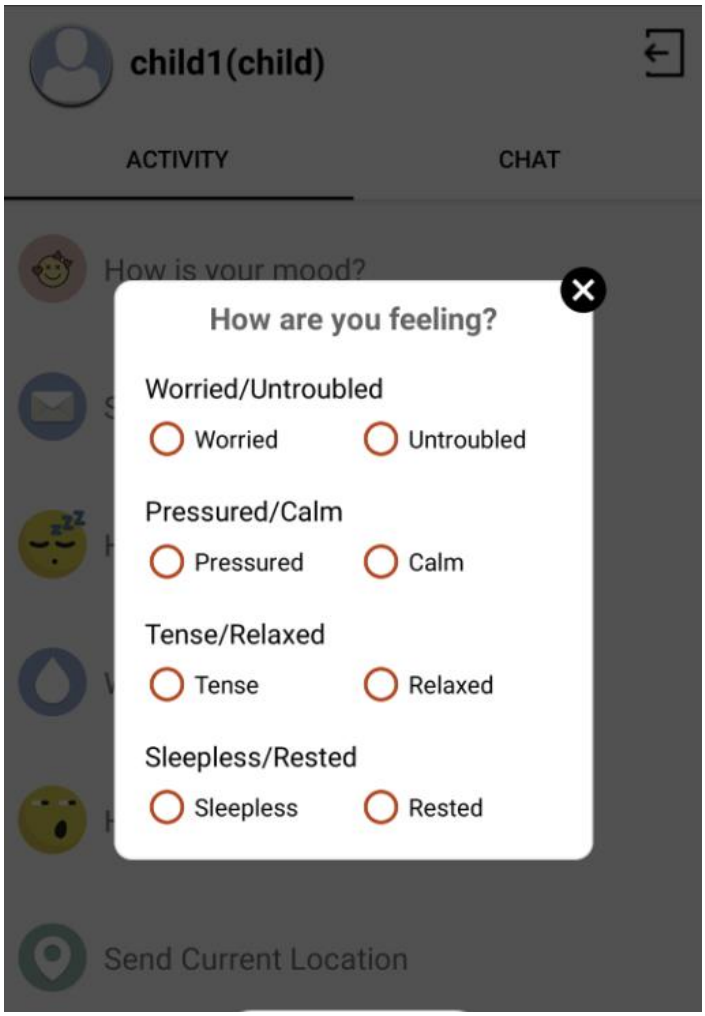


Figure 4.17: Capture how you are feeling

The child can choose how he is feeling. Each time he clicks on a radio button the information is automatically send. So, he can choose which entries want to send, and don’t have to fill all of them in order to send them.

Send location

click on send location on the home screen.

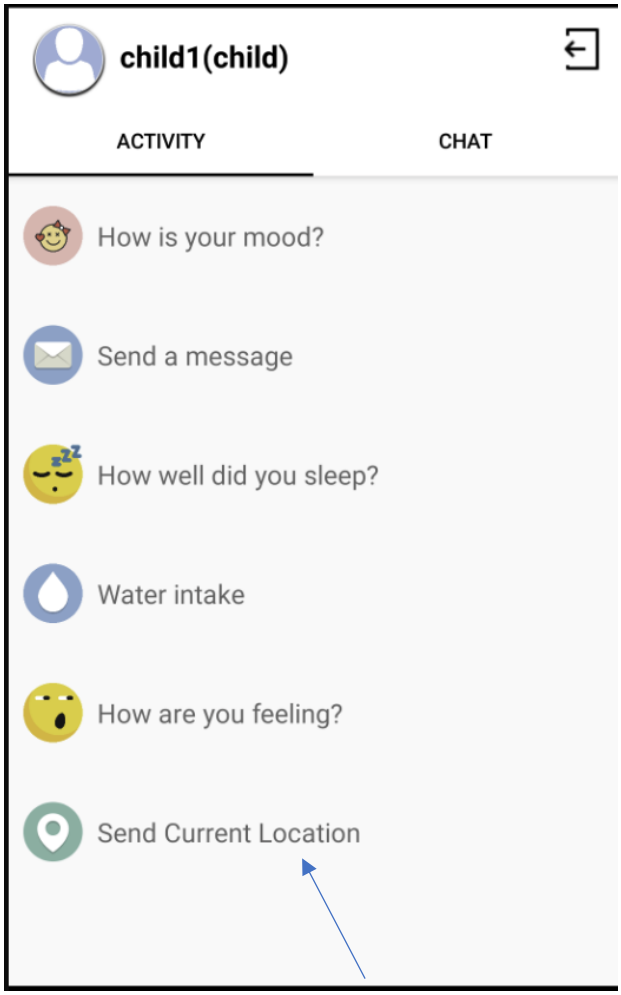


Figure 4.18 Send location

The child can send its current location clicking on the send location button. He should have enabled the location services on his phone.

Parent home screen

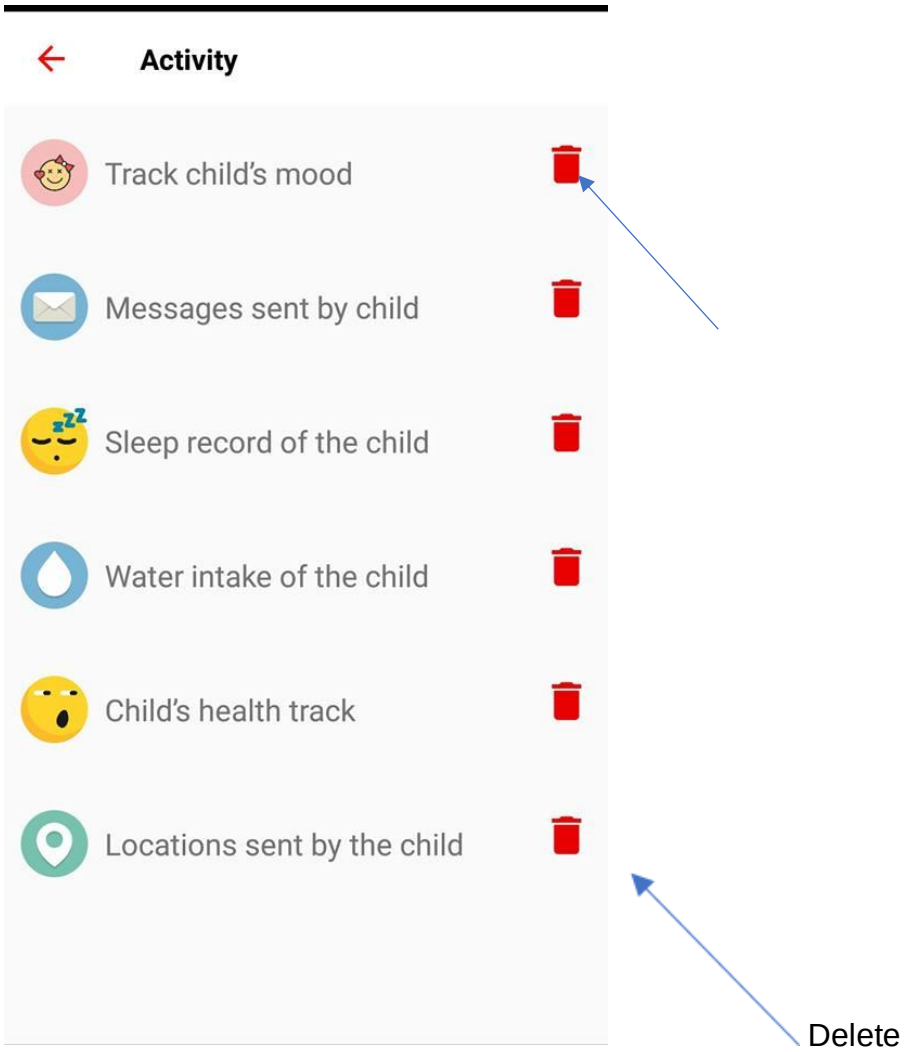


Figure 4.19: Parent home screen

This is the main screen of the parent user. The user should be able to see the information when presses on each button. He should be able to track child's mood, view predefined messages that have been sent, track sleep habits of the child, view water intake, track child's health, and view the locations that the child has been sent. For each function he should be able to delete the data by clicking on delete button next to the activity.

Track Child Mood

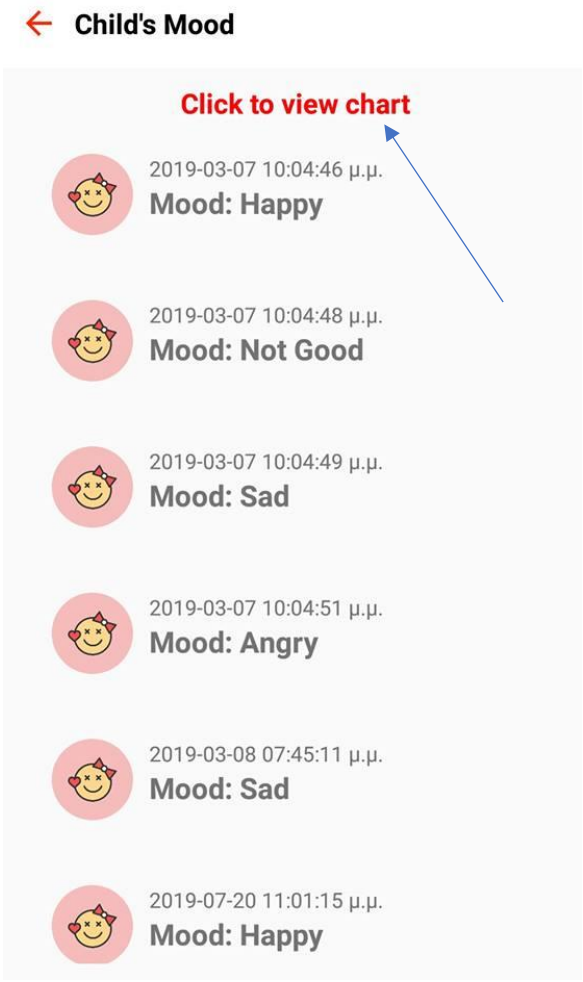


Figure 4.20: Track Child Mood

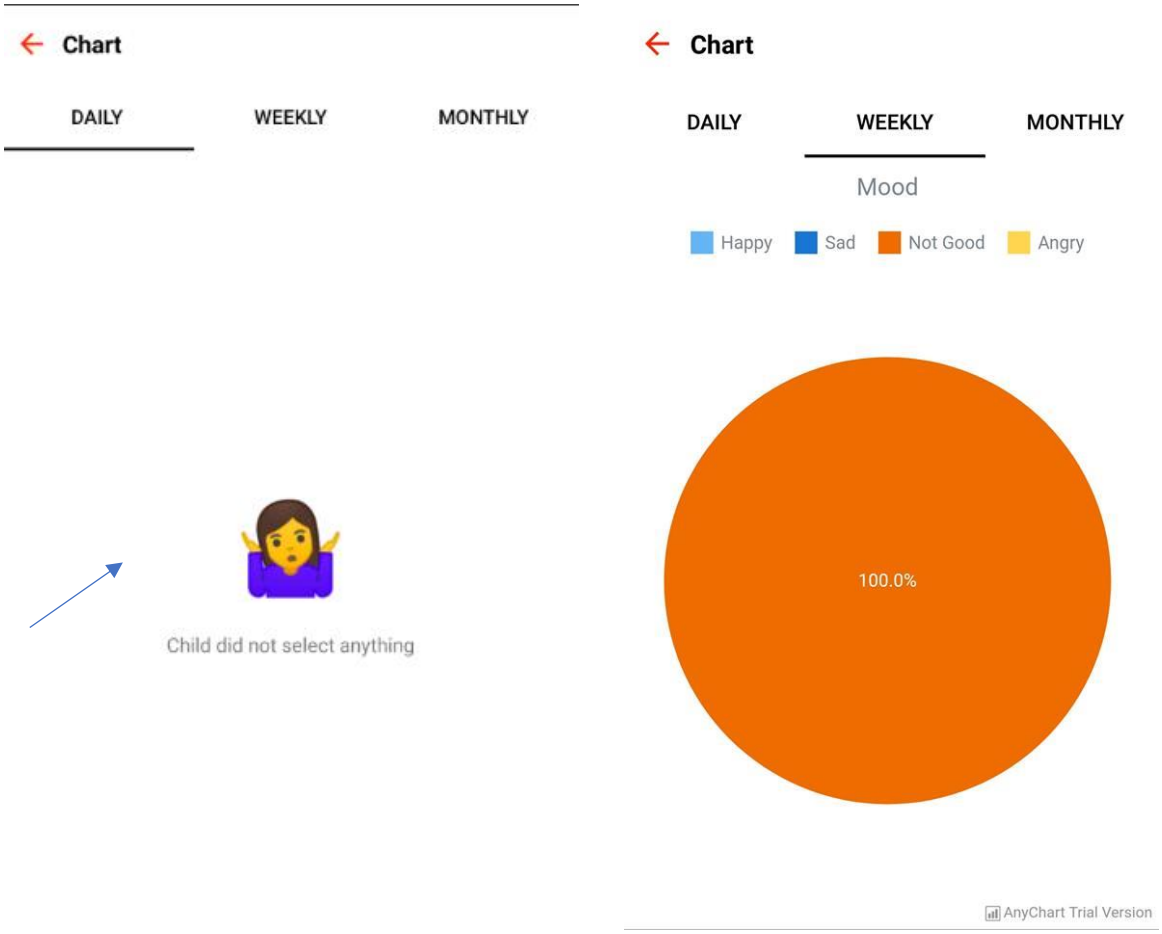


Figure 4.21: mood child daily, weekly, monthly

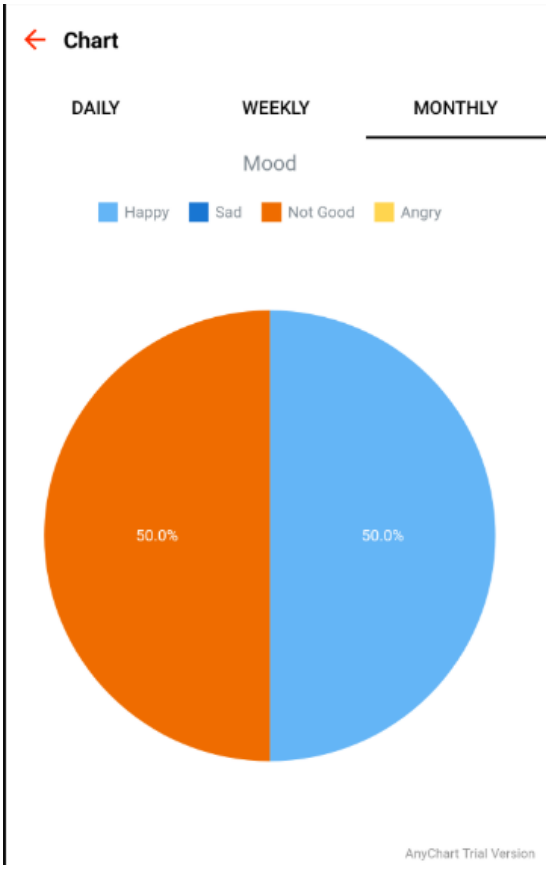


Figure 4.22: Mood tracking Daily/weekly Monthly

Here the parent can view graphic and more specifically a pie chart with percentage about each mood. The moods are: Happy, sad, Not good and angry. If there are no data in the server for the specific category a screen with a message provides information to the father.

Predefined message view

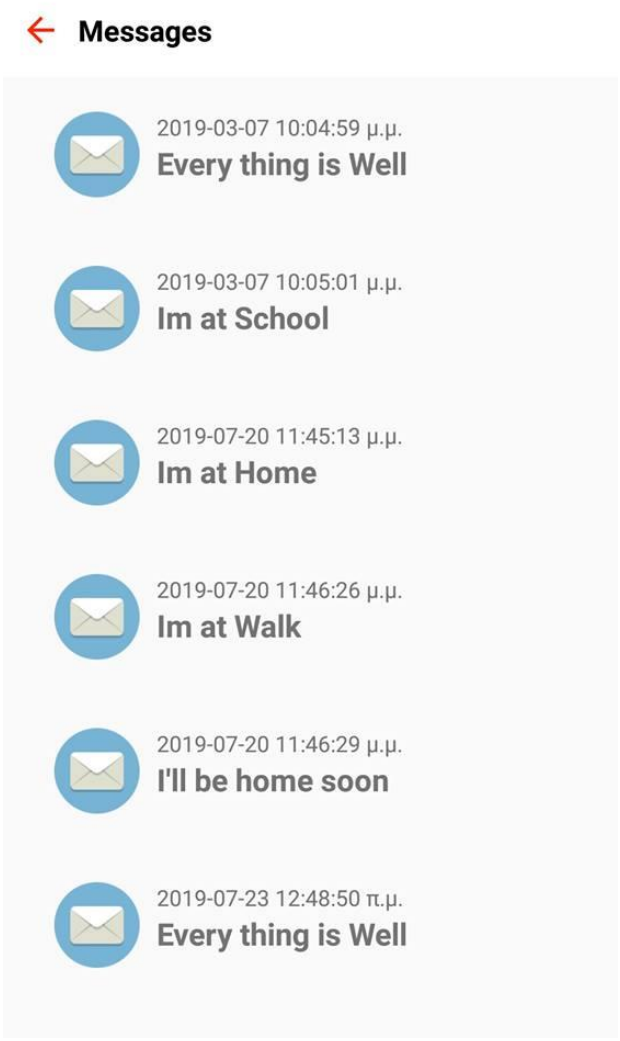


Figure 4.23: Predefined message view

Here the parent can view the predefined messages that have been sent by the child with specific date and time.

Track sleep

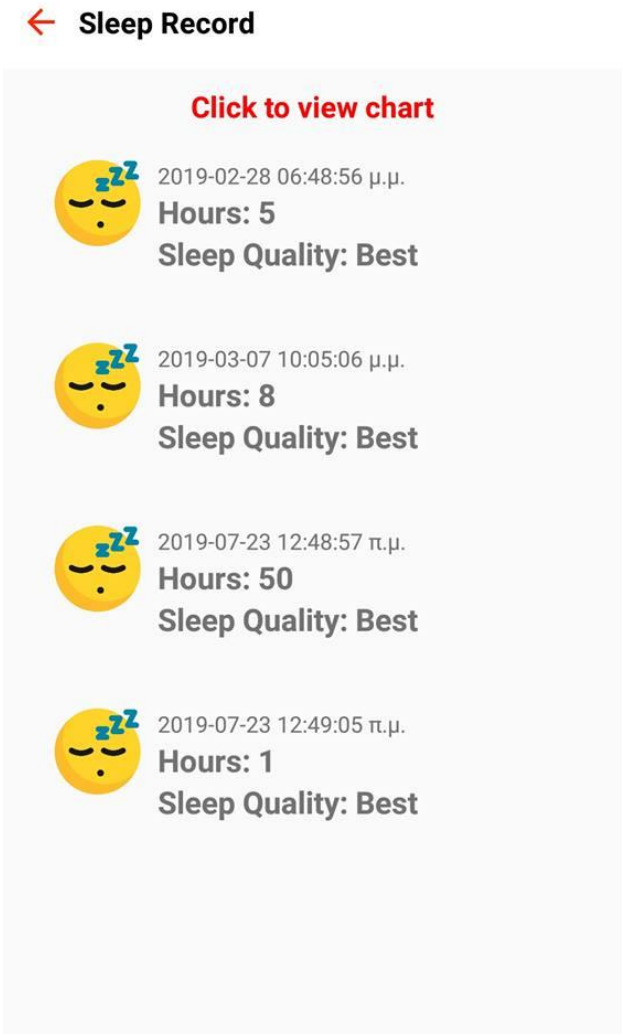


Figure 4.24: Track sleep

The parent should be able to view the quality of sleep, and how many hours he slept each date.

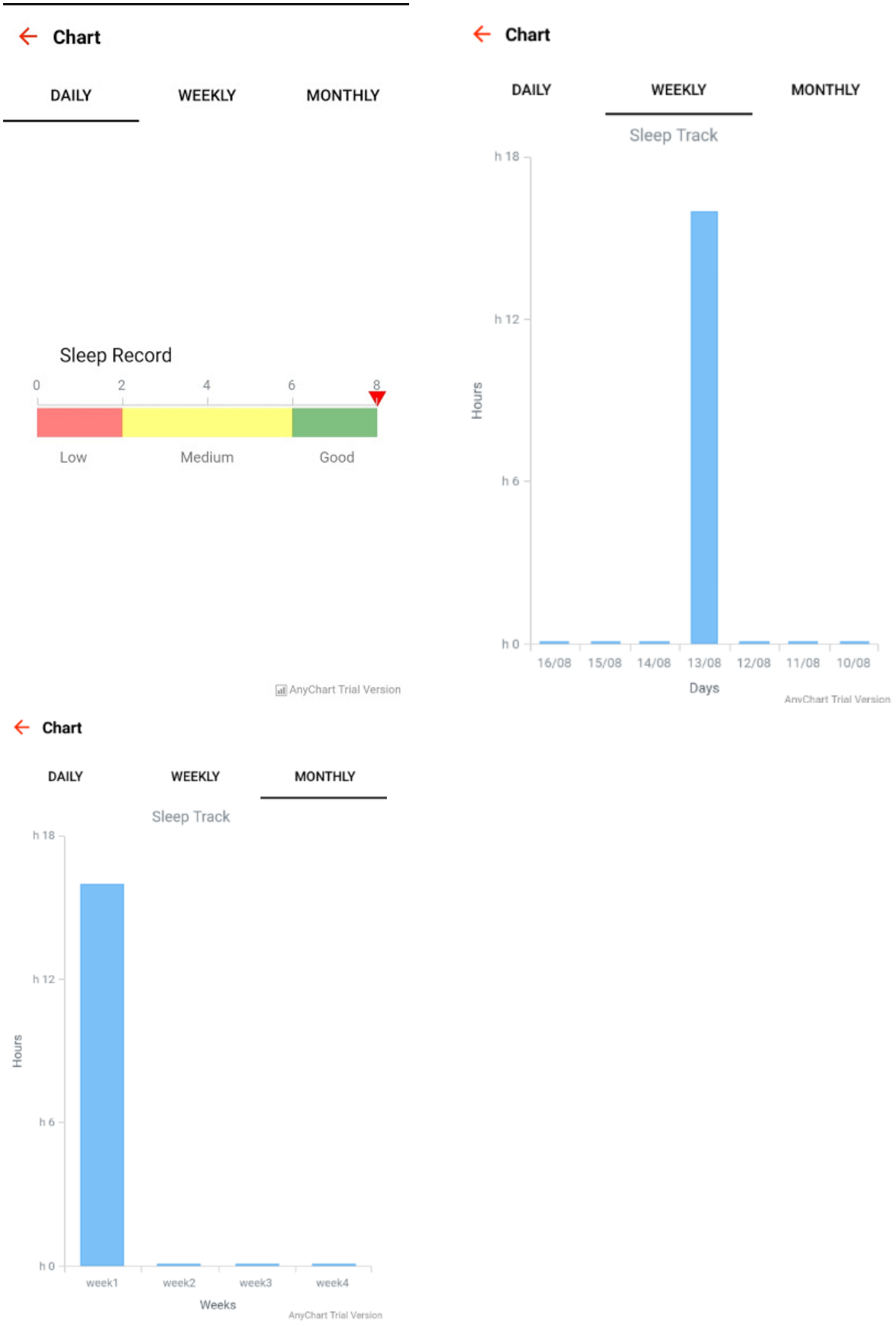


Figure 4.25: daily/weekly/monthly sleep graphs.

In this screen the parent can see information about the hours of sleep of his child. The date, the number of hours is available to the parent.

Track water intake

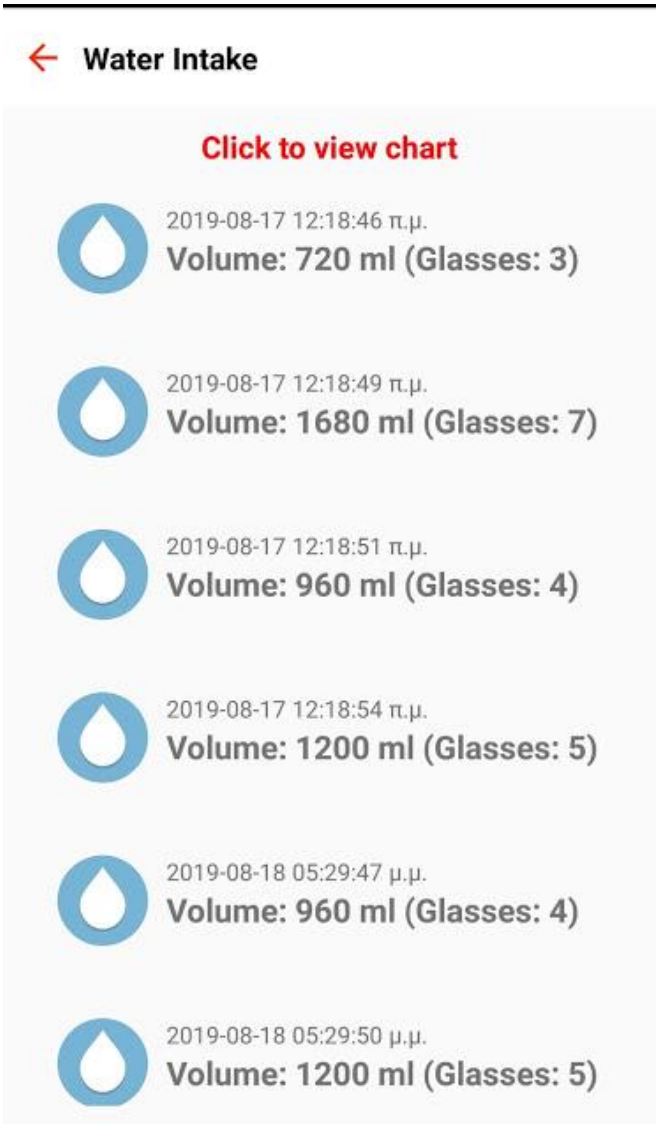


Figure 4.26: Track water intake

The parent should be able to view statistics about water intake (glasses. ml)

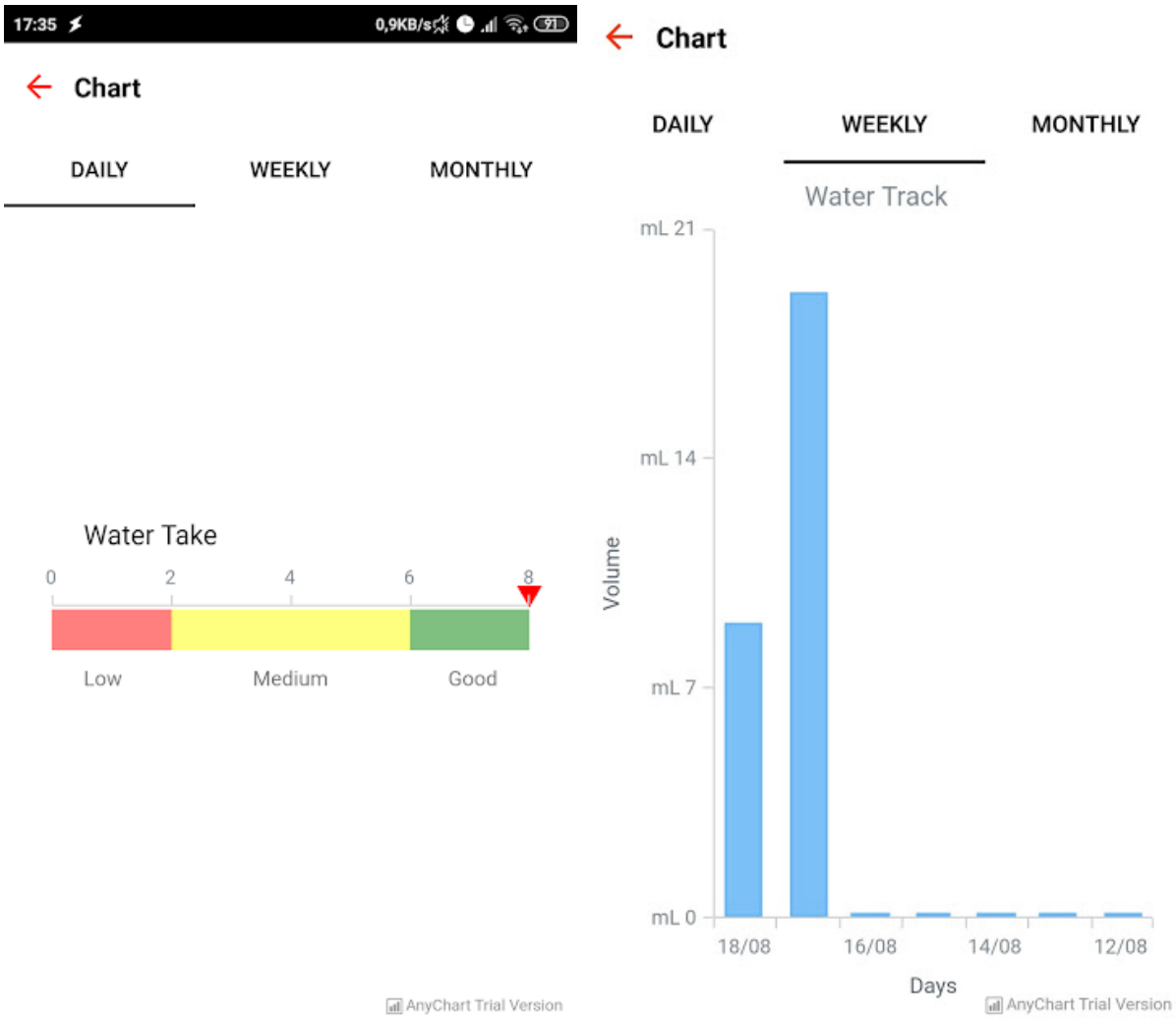


Figure 4.27: water intake charts

Track child’s health

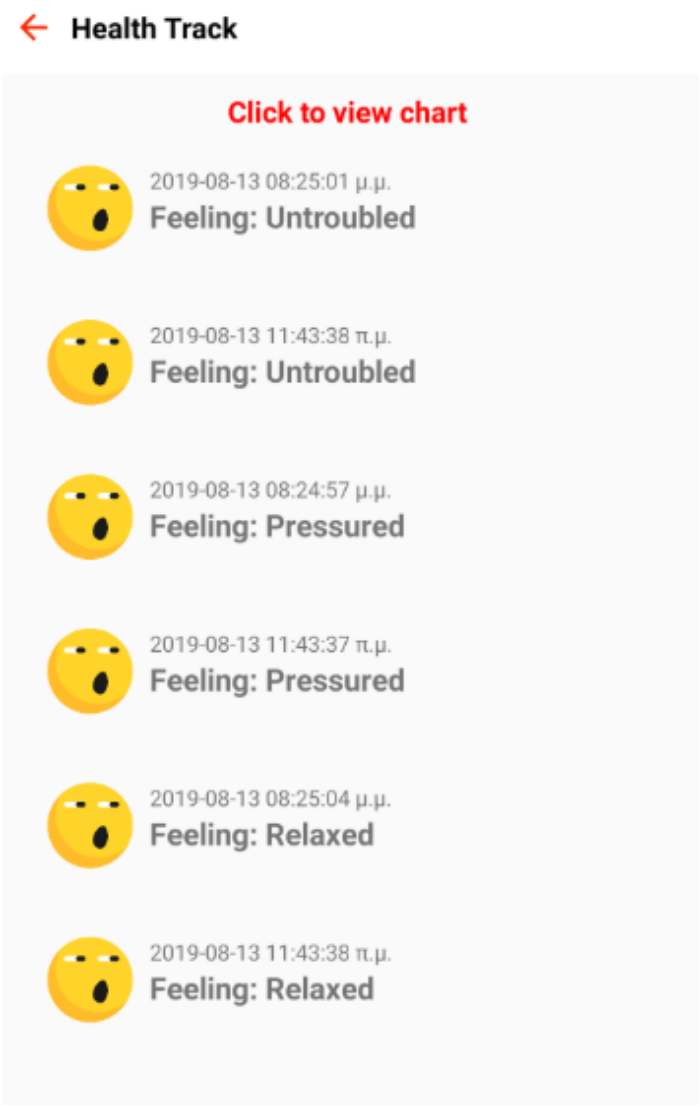


Figure 4.28: track child’s health

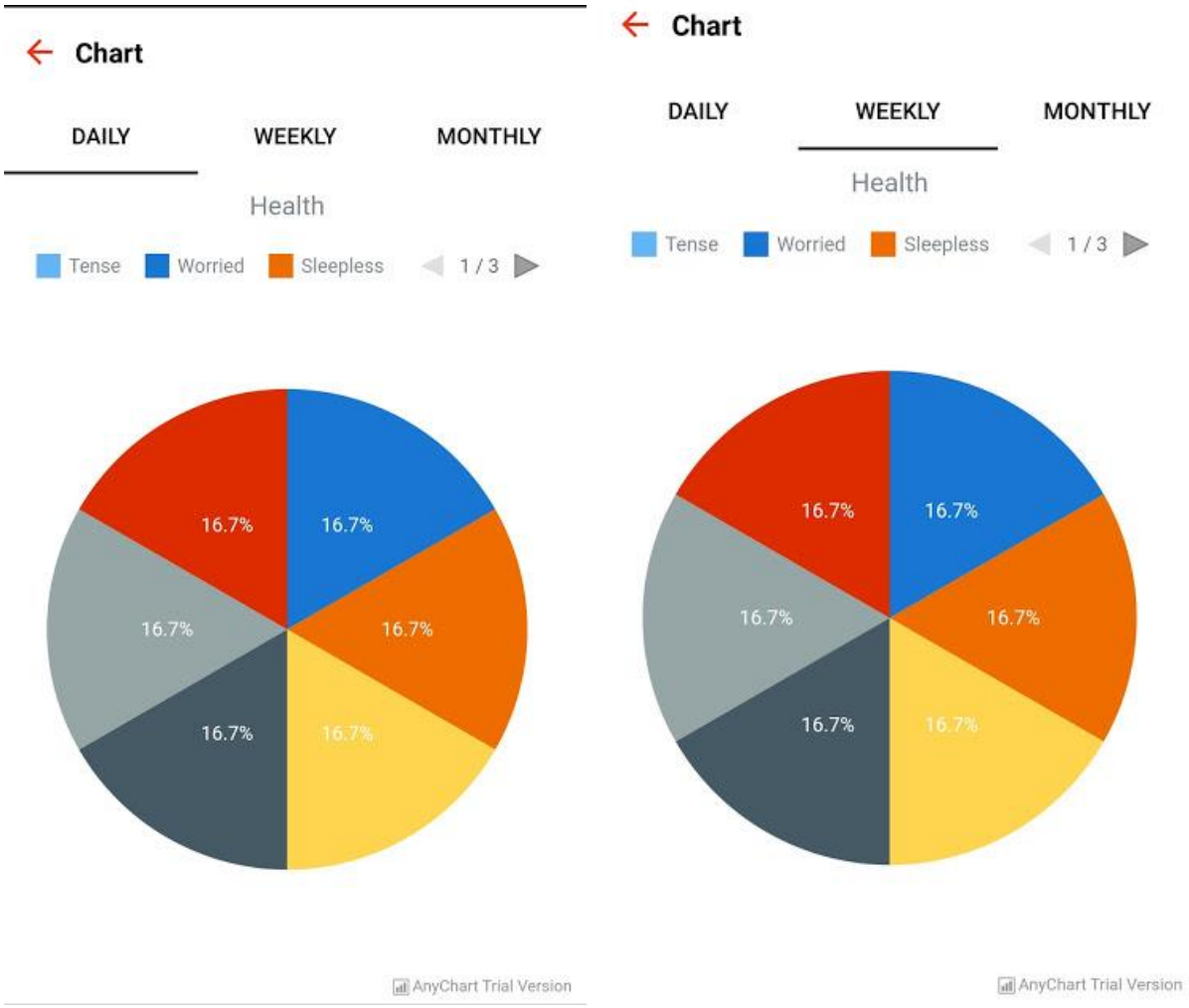


Figure 4.29: Health track chart-daily-weekly

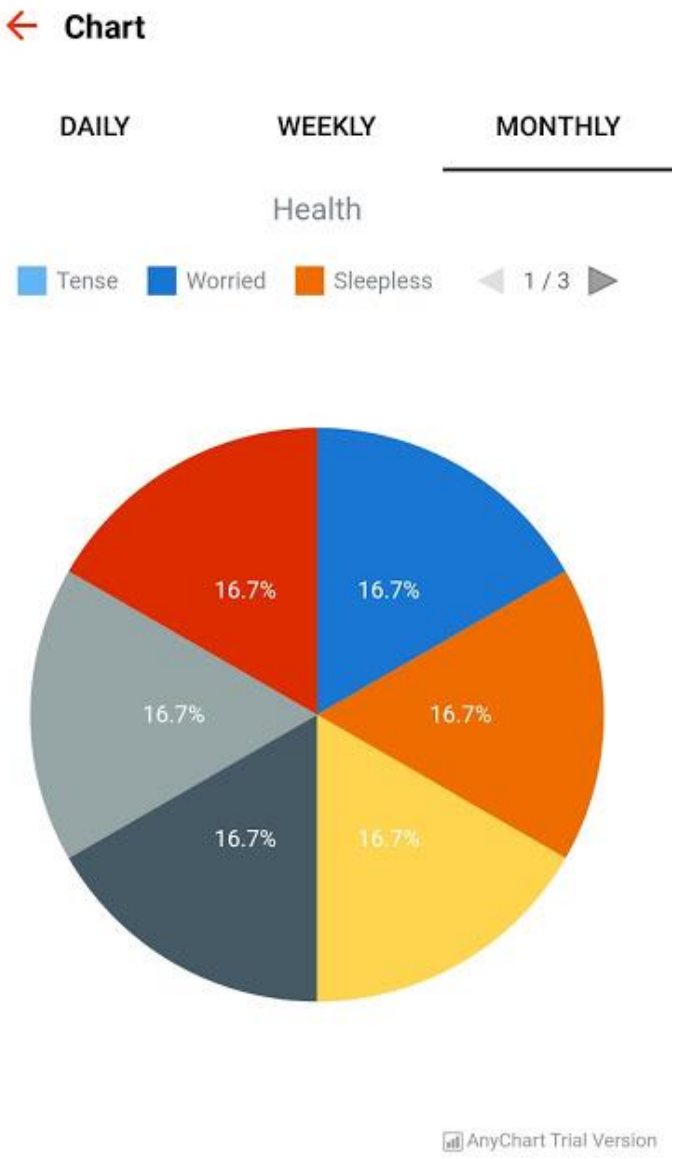


Figure 4.30: Health track chart monthly

The parent should be able to view all the entries recorded, of how the child is feeling, with the specific date and the feelings selected.

View locations received

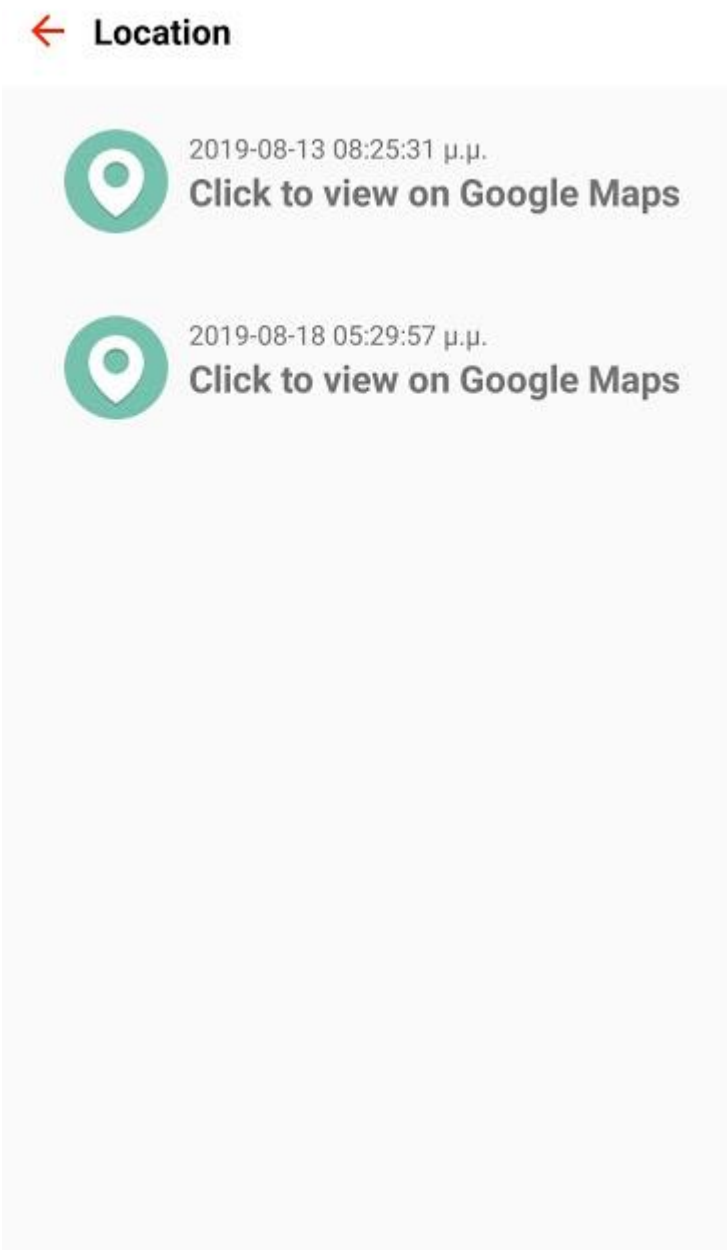


Figure 4.31 View locations received

The user can view the locations sent by the child. The location is appeared in the screen according to date and when the user presses on the “click to view on google maps” a screen with exact location is available.

Messenger/chat

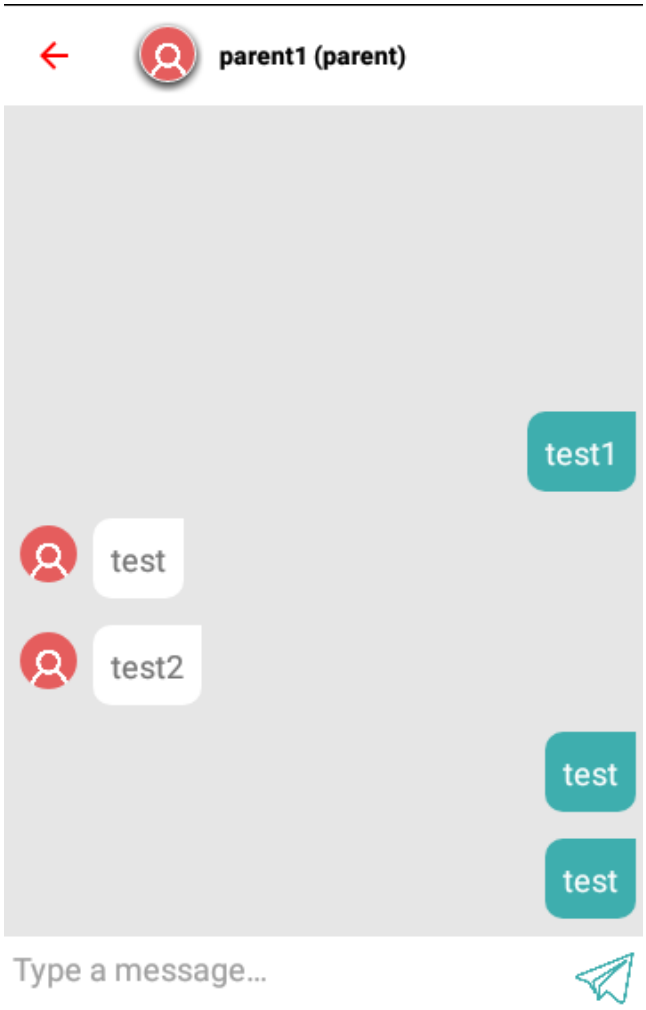


Figure 4.32: Messenger/chat

The parent can send messages to the child by typing the message and click on send.

4.2 Implementation

During the implementation phase the user interface, the logic of the application implementation, the testing and evaluation phases are going to be analyzed. First, we are going to start with user interface implementation which will be based on the high-fidelity prototype and sketches. We are going to examine the various layouts that are used. Xml files are designed using layouts and widgets like buttons and Edit text or views. For this application the Android studio layout editor will be used to implement all the .xml layout files. The basic components of each layout will be analyzed before implementing the code that is going to support the logic behind the user interface. More Information on each component can be found on the official android developers' site https://developer.android.com/. The interface is based on

activities which hold many different fragments. Later after the description of the basic layout we are going to see how main screens for the user and child are made using specific methods that we have seen in the methodology on logic implementation plan. We are going to analyze how the MVP architecture is implemented using adapters and data models. Next the structure of the android project and the most significant activities, adaptors, fragments, models, are going to be analyzed. The importance of firebase backend is going to be addresses with examples of code as well.

4.2.1 UI implementation

As it was examined in the methodology chapter the basic building blocks of an android application are activities. The activities (and fragments) are bonded with user interface layouts xml files using specific methods which we will be discussed in logic implementation. The interface is based on activities which hold many different fragments. Later after the description of the basic layout we are going to see how main screens for the user and child are made using specific methods and not, and how grid view, uses adapters and data models to retrieve data. The most significant activities components are going to be described. Then the fragments that are used as a part of a screen usually will be analyzed. The common layouts for child parent like login, register, forgot password are going to analyzed first.

Login screen ▼

Figure 4.33: Login screen

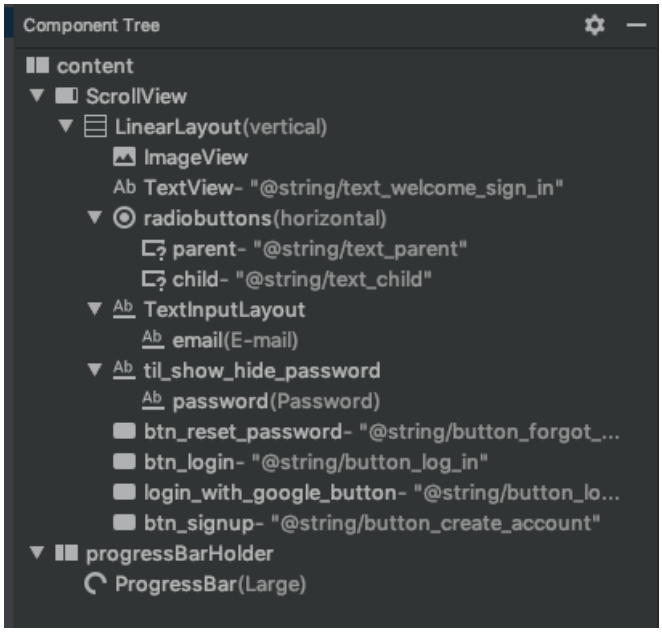


Figure: Login screen Component Tree view

This is the (activity_main xml) file of the main activity. It consists of a Relative Layout which holds the Scroll View. Scroll view is a viewgroup that allows to scroll the child elements inside it. All the components that will be described are contained inside scroll view. First, we create a linear layout which is contained inside the scroll view. The first widget which shows the image is an image view. Then there is a text view with text defined “please login as a parent or child”. Next is a radio group which holds the radio buttons for the selection of a parent or a child. Next is the text input an editetxt component to allow the user to add the email, and another one for the password. Then we have three buttons login, login with google, signup. Inside the same xml there is a progress bar which will show the progress of login operation.

Sign up screen

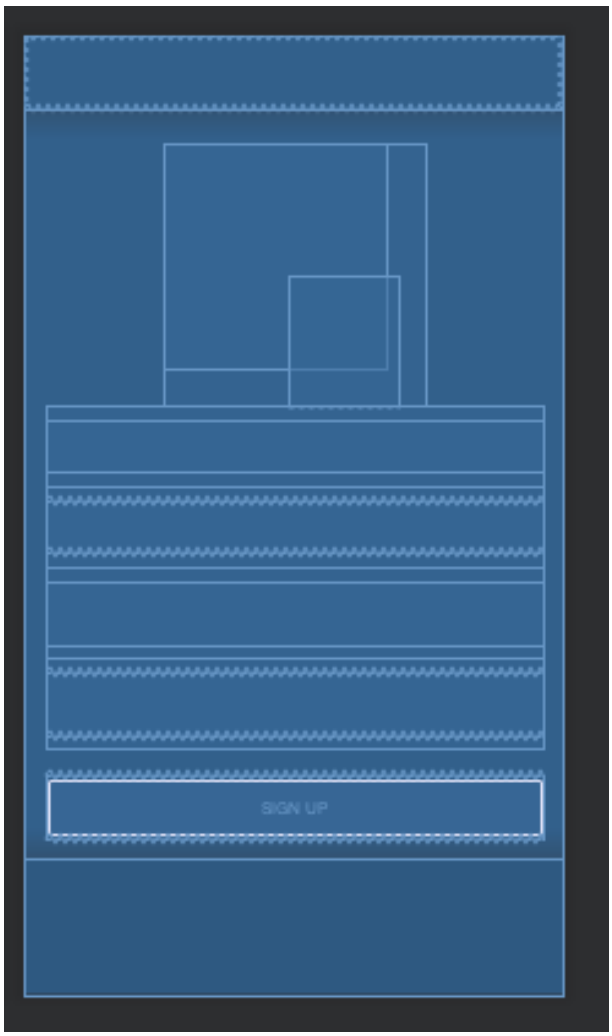


Figure 4.34 Sign up screen

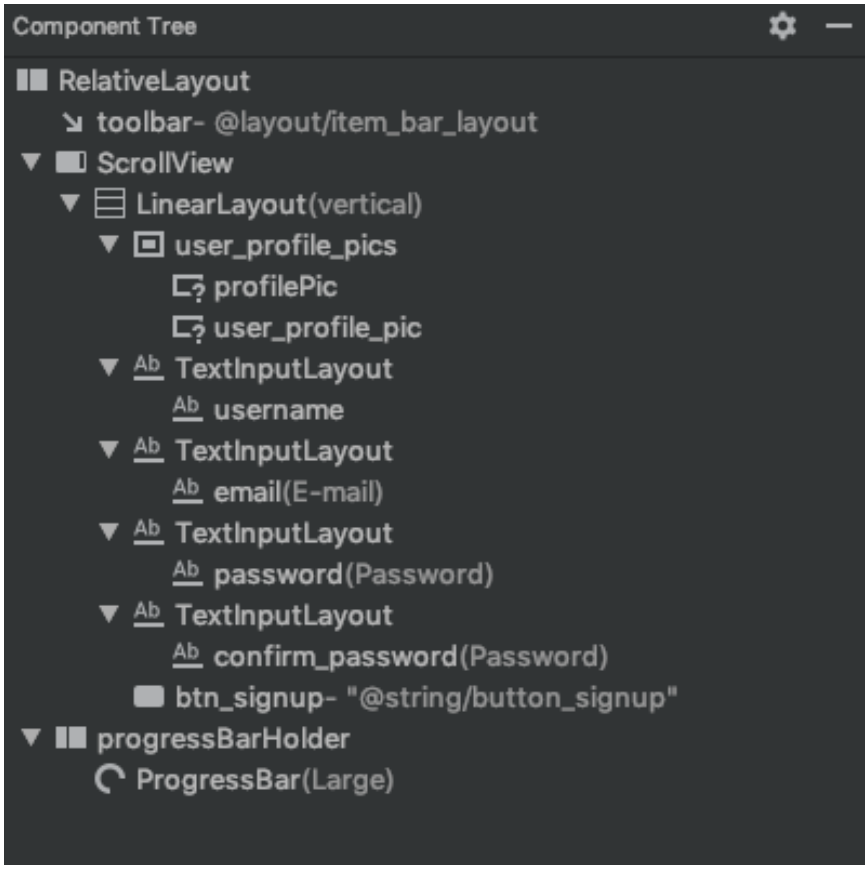


Figure 4.35: signup screen Component Tree view

Sometimes we need to use larger components that require a special layout. The sign up screen (Activity.Signup xml) has a Relative Layout and includes (item_bar_layout) a toolbar item which is another xml layout with a toolbar. <include/>. Inside scroll view like in the previous activity there are the components:

1. Framelayout to add pics of user
2. Circular image view (display image)
3. edittext widget for email, an editext for password, and an editext for confirm password
4. The Button signup follows.

Reset password

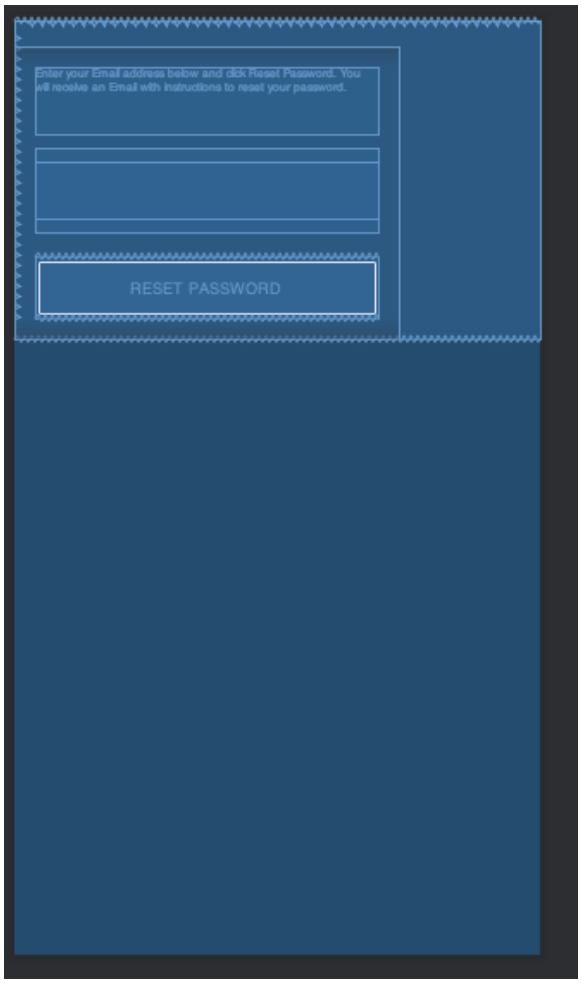


Figure 4.36: Reset password

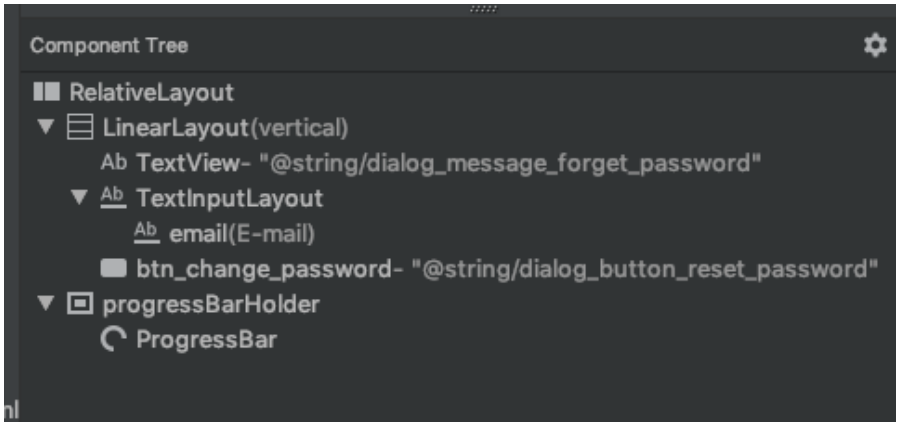


Figure: Reset password Component Tree view

Forgot password is implemented it's a dialogfragment. Consists of a relative layout, textview which displays the "enter your email" message an edittext for edit email and

the submit button. The progress bar is included in the layout in order to show the process of reset password.

Home activity for child

Activity dashboard layout

The activity dashboard layout is the layout bonded with the home activity for child. It is used both by parent home screen and child home screen. In the activity class fragment Fragments. ChildFragment has been imported in order to be used from the activity **(import** com.example.anonymous.child_parent_application.Fragments.ChildFragment;

```
if (AppVariableClass.getInstance().getUserStatus().equals("child")) {
    viewPagerAdapter.addfragment(new ChildFragment(), "Activity");
```

Fragment (Activity)

```
viewPagerAdapter.addfragment(new UsersFragment(), "Chat")
```

(chat)

Figure 4.37 Home activity for child

The activity dashboard layout contains a toolbar, a tab layout which provides a horizontal layout to display tabs, AND a vie pager that allows flip left and right through pages of data

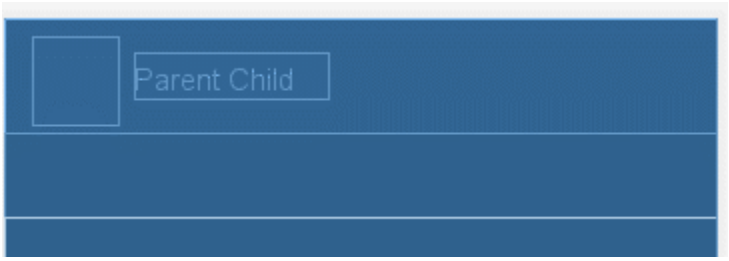


Figure 4.38: Tree view Activity dashboard

The child fragment includes layout gridview xml.

<include layout="@layout/layout_grid_view"/>

Child fragment

trackAdapter = new ChildTrack(AppVariableClass.*getInstance*().listOfTracks, getContext(), AppVariableClass.*getInstance*().listOfImageResource);

Here using child track adapter and the listoftracks arraylist which is included in appvariable class we bind data with fragment. (controller with fragment layout)

Figure: Fragment child

The grid data allows to display data as columns and rows creating easy to browse lists.

Appvariableclass:

public ArrayList<String> listOfTracks = new ArrayList<>();

```
public void dataLists() {
        if (listOfTracks.size() <= 0) {

            listOfTracks.add("How is your mood?");
            listOfTracks.add("Send a message");
            listOfTracks.add("How well did you sleep?");
            listOfTracks.add("Water intake");
            listOfTracks.add("How are you feeling?");
            listOfTracks.add("Send Current Location");
```

This array list will be used to show the options for selection on child main screen.

Home activity for Parent

The main screen of the parent has the same uses the same architecture as child's home screen.

Showhildtracks activity

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_child_tracks);


private void setGridView() {
    trackAdapter = new
ChildTrackFromAcitivityAdapter(AppVariableClass.getInstance().childMoodToShow
Parent, this, AppVariableClass.getInstance().listOfImageResource);
    gv_child_moods.setAdapter(trackAdapter);
}
```

Figure 4.39: *activity_child_tracks*

From appvariableclass:

```
public ArrayList<String> childMoodToShowParent = new ArrayList<>();

childMoodToShowParent.add("Track child's mood");
childMoodToShowParent.add("Messages sent by child");
childMoodToShowParent.add("Sleep record of the child");
childMoodToShowParent.add("Water intake of the child");
childMoodToShowParent.add("Child's health track");
childMoodToShowParent.add("Locations sent by the child");
```

Show child tracks

```
setContentView(R.layout.activity_show_child_tracks);
```

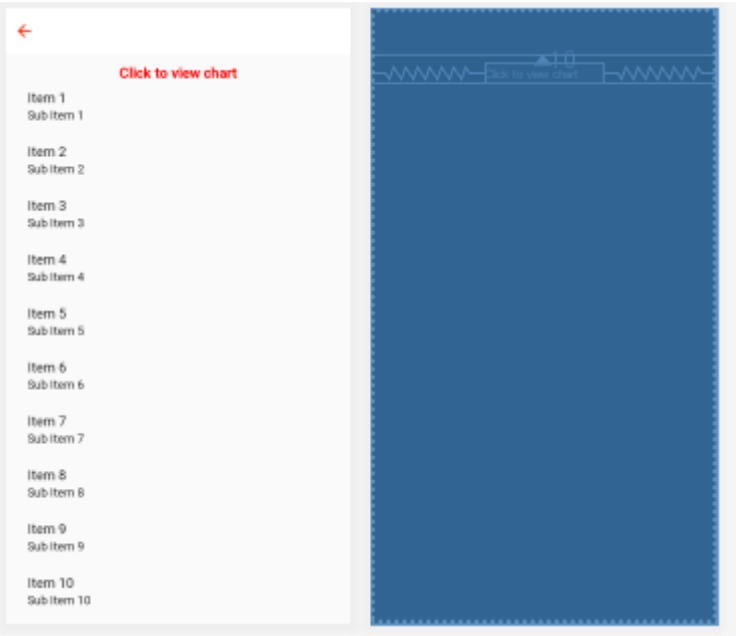


Figure 4.40: Activity activity_show_child_tracks

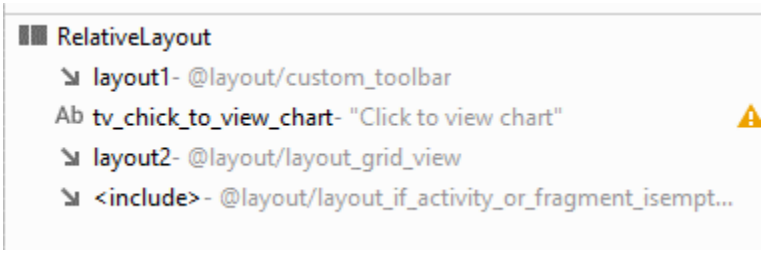


Figure 4.41: Activity activity_show_child_tracks treeview

Chat activity

Figure 4.42: Chat activity

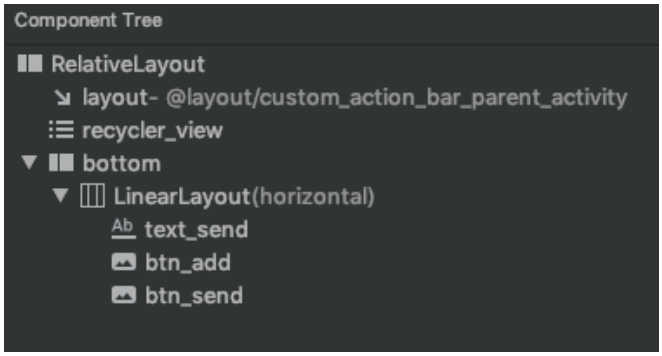


Figure4.43: Chat activity treeview

Here is the Activity chat.xml which is responsible for the messages. It consists of a recycle view to show messages, a button send text, a button adds, and the button send.

setContentView(R.layout.*activity_chat*);

public class MessageActivity extends AppCompatActivity implements BottomSheetDialog.BottomSheetListener {

BottomSheetDialog is a simple dialog which is styled as a bottom sheet.

Add user activity

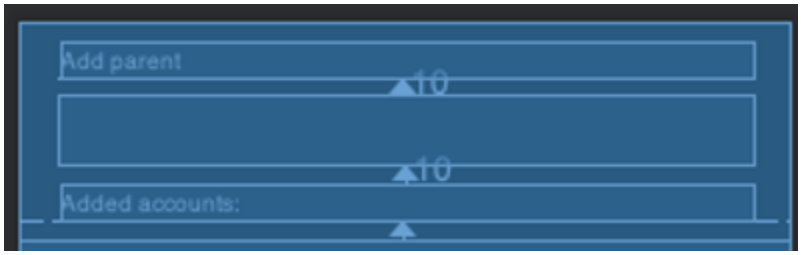


Figure 4.44: Add user

This fragment contains a textview “add user”, an autocomplete textview which retrieve the users from the database, a textview “added accounts, and” finally a recycler view which shows the added users.

User adapter

```
public ViewHolder onCreateViewHolder(@NonNull ViewGroup viewGroup, int i) {
    View view = LayoutInflater.from(context).inflate(R.layout.item_user, viewGroup,
false);
    return new ViewHolder(view);
```

Figure 4.44: Item user layout

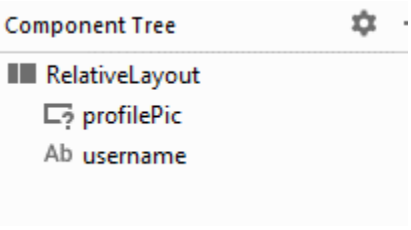


Figure 4.45 Treeview item

Fragments

Here we are going to analyze the most basic fragments.

Fragment used to capture sleep.

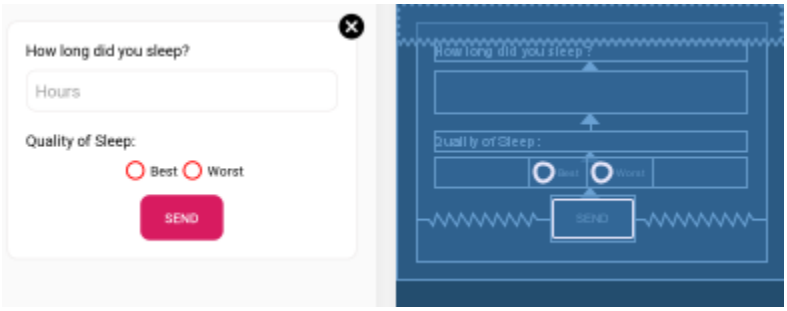


Figure 4.46: capture sleep fragment

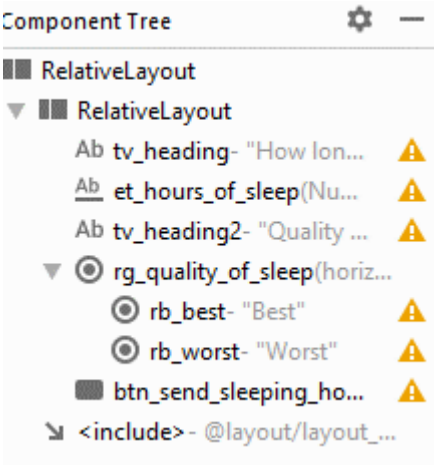


Figure: capture sleep tree view

```
private void layoutForSleepTrack(final Dialog dialog) {
    final EditText et_volume_of_water =
dialog.findViewById(R.id.et_hours_of_sleep);
    final RadioGroup rg_quality_of_sleep =
dialog.findViewById(R.id.rg_quality_of_sleep);
    final RadioButton rb_best = dialog.findViewById(R.id.rb_best);
    final RadioButton rb_worst = dialog.findViewById(R.id.rb_worst);
    final Button btn_send =
dialog.findViewById(R.id.btn_send_sleeping_hours_and_quality);

    rg_quality_of_sleep.setOnCheckedChangeListener(new
```

```
RadioGroup.OnCheckedChangeListener() {
        @Override
        public void onCheckedChanged(RadioGroup group, int checkedId) {
            View radioButton = rg_quality_of_sleep.findViewById(checkedId);
            int index = rg_quality_of_sleep.indexOfChild(radioButton);
            switch (index) {
                case 0:

AppVariableClass.getInstance().setChildQualityOFSleep(rb_best.getText().toString()
);
                    break;
                case 1:

AppVariableClass.getInstance().setChildQualityOFSleep(rb_worst.getText().toString(
));
                    break;
```

Fragment used to reset statistics

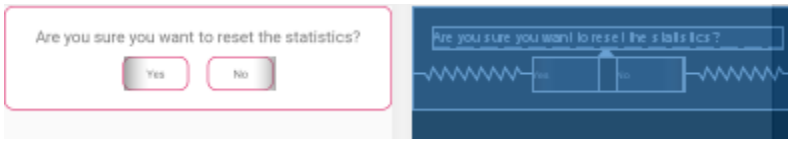


Figure 4.47: Reset statistics

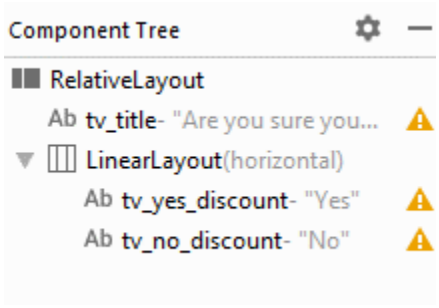


Figure: reset statistics Tree view

Reset statistics:

```
 private void CustomDialogBox(Context context, final int position) {

        final Dialog dialog = new Dialog(context);

        dialog.setContentView(R.layout.layout_confirm_delete_connection);
```

```
dialog.setCancelable(false);
TextView yes = dialog.findViewById(R.id.tv_yes_discount);
TextView no = dialog.findViewById(R.id.tv_no_discount);
yes.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        if (position == 0) {
            firebaseDeleteConnection("Mood");
        } else if (position == 1) {
            firebaseDeleteConnection("PredefinedMessages");
        } else if (position == 2) {
            firebaseDeleteConnection("SleepTrack");
        } else if (position == 3) {
            firebaseDeleteConnection("WaterTrack");
        } else if (position == 4) {
            firebaseDeleteConnection("AnxietyTrack");
        } else if (position == 5) {
            firebaseDeleteConnection("Location");
        }
        dialog.cancel();
```

Other fragments

Figure 4.48: Layout child track

Figure 4.49: Child water track

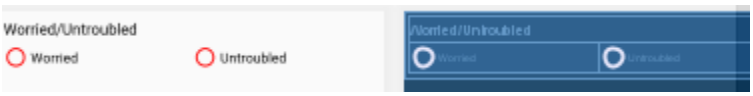


Figure 4.50 : anxiety track

Splash screen

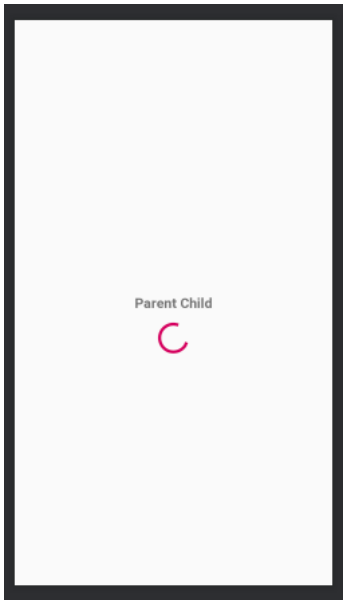


Figure4.50: splash screen

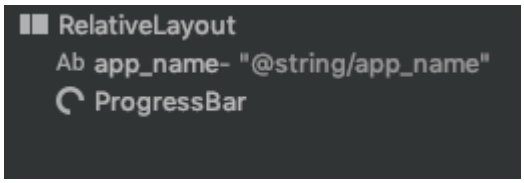


Figure 4.51: splash screen treeview

The splash screen is the first to be launched by the application. It creates an intent in order to call main activity.

Intent intent = new Intent(getApplicationContext(), MainActivity.class);

4.2.2 Logic implementation

Now we are going to analyze the architecture of the application, using code snippets and flowcharts. We are going to examine how MVP, and Firebase implemented, and other basic building blocks of android like activity lifecycle, adaptors.

4.2.2.1 Flowcharts

CHAT

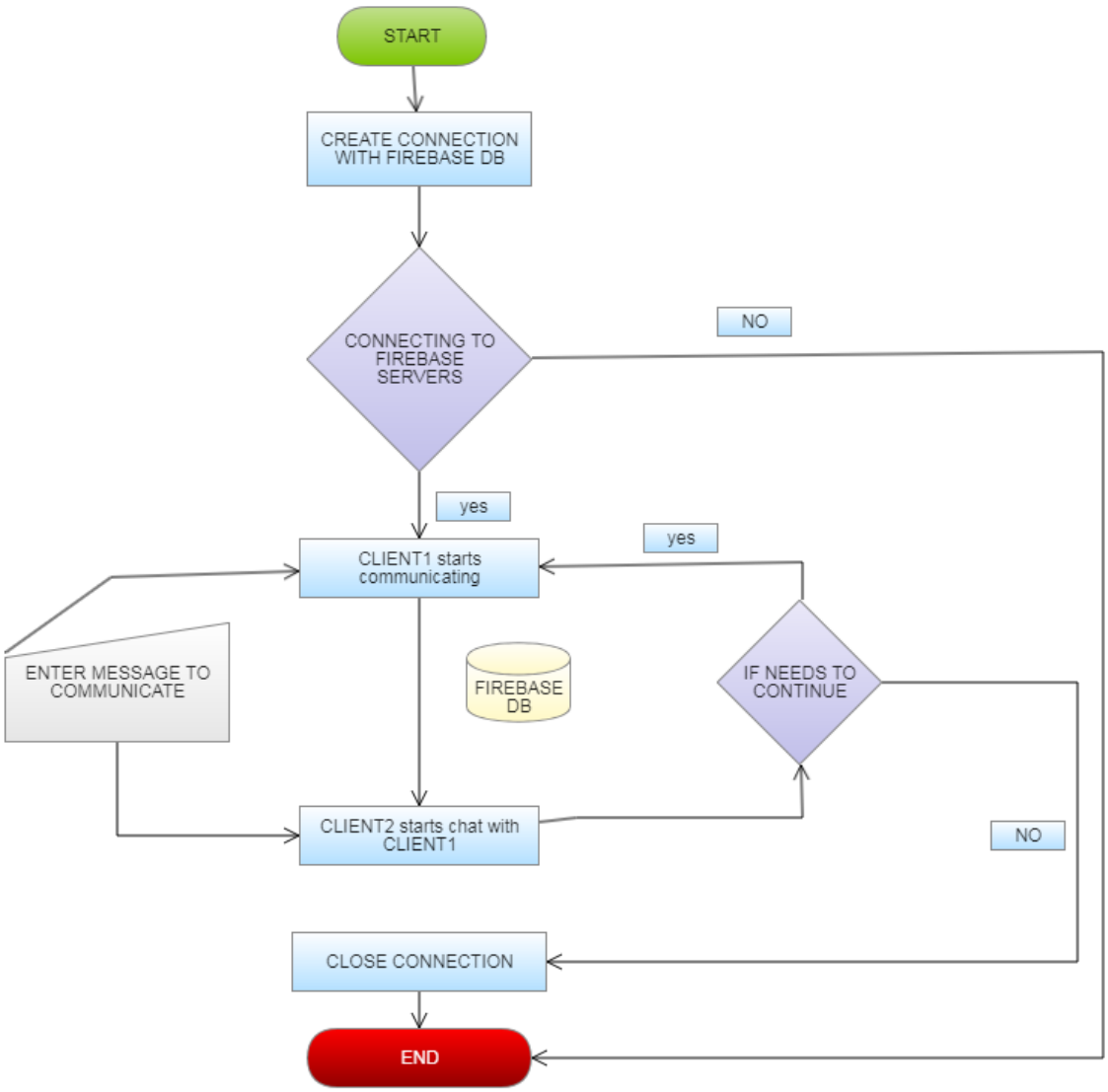


Figure 4.52: Chat flowchart

When the user tries to connect with firebase with his account email and password, a connection with firebase will be created. If the connection is successful, the client1 starts to chat with client 2. When the user leaves the chat, the connection will be closed.

Login/register

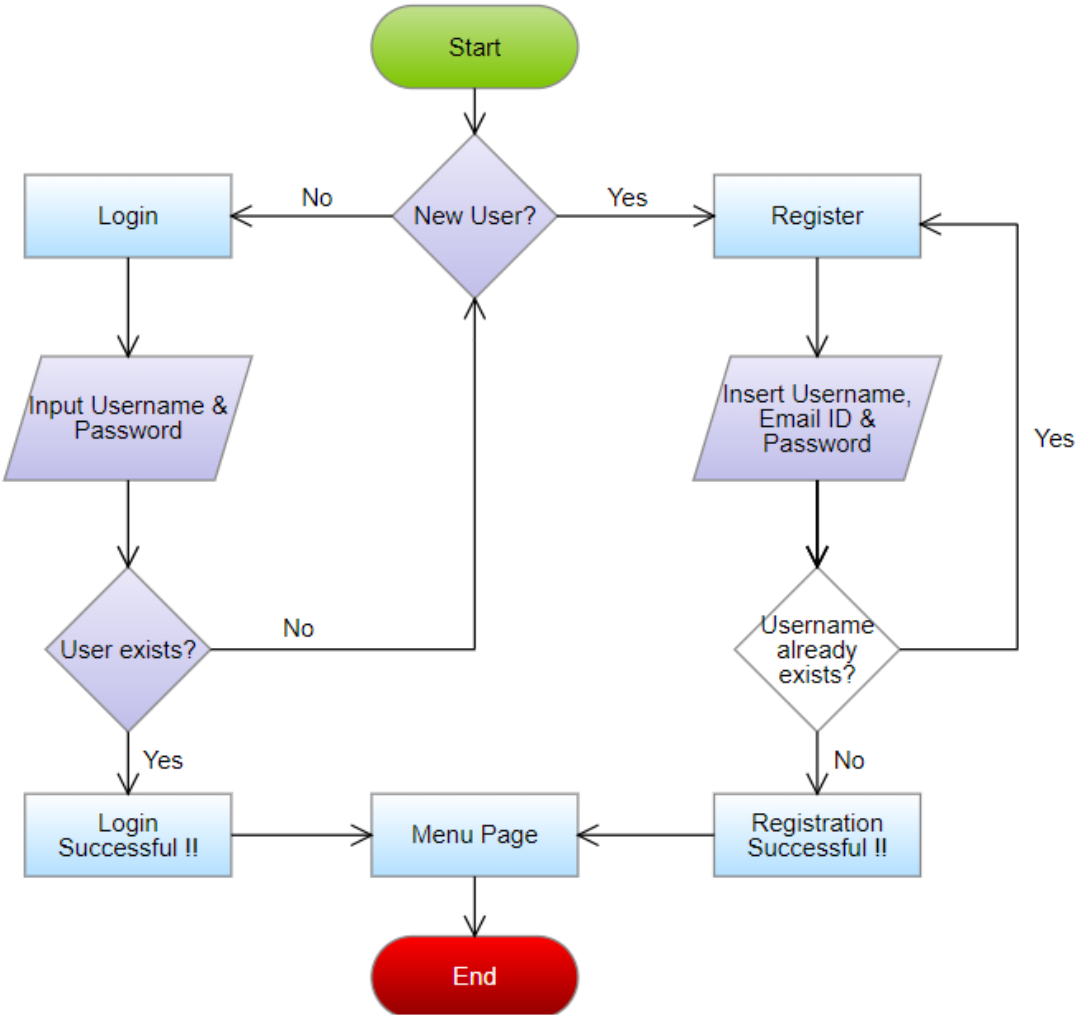


Figure 4.53: login/Register flowchart

Login

The user from the main screen enters his username and password in order to login if the user doesn't exist then he goes back to main activity. If the user exists successfully logins and goes to menu page, even if is child or parent.

Register

First the user should enter his username and his password. If the user already exists, he is being redirected to registration page. If there is no user with specific username the user directed to menu.

Save data/ notification

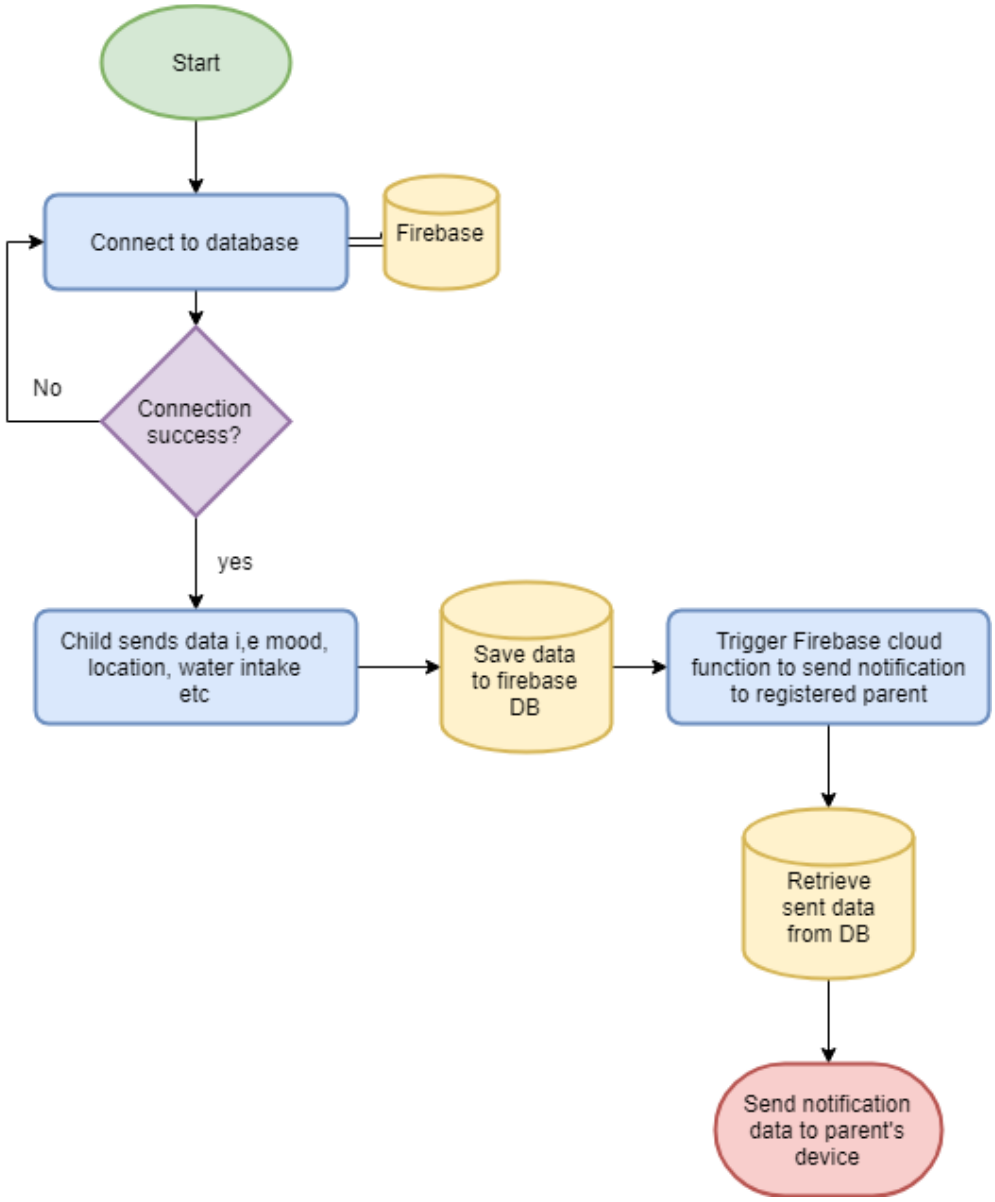


Figure 4.54: Send data/ notification flowchart

If the connection to the database is successful, the user can send data and save them to firebase. Then the firebase cloud function is triggered in order to send notification to the user.

Parent child dashboard

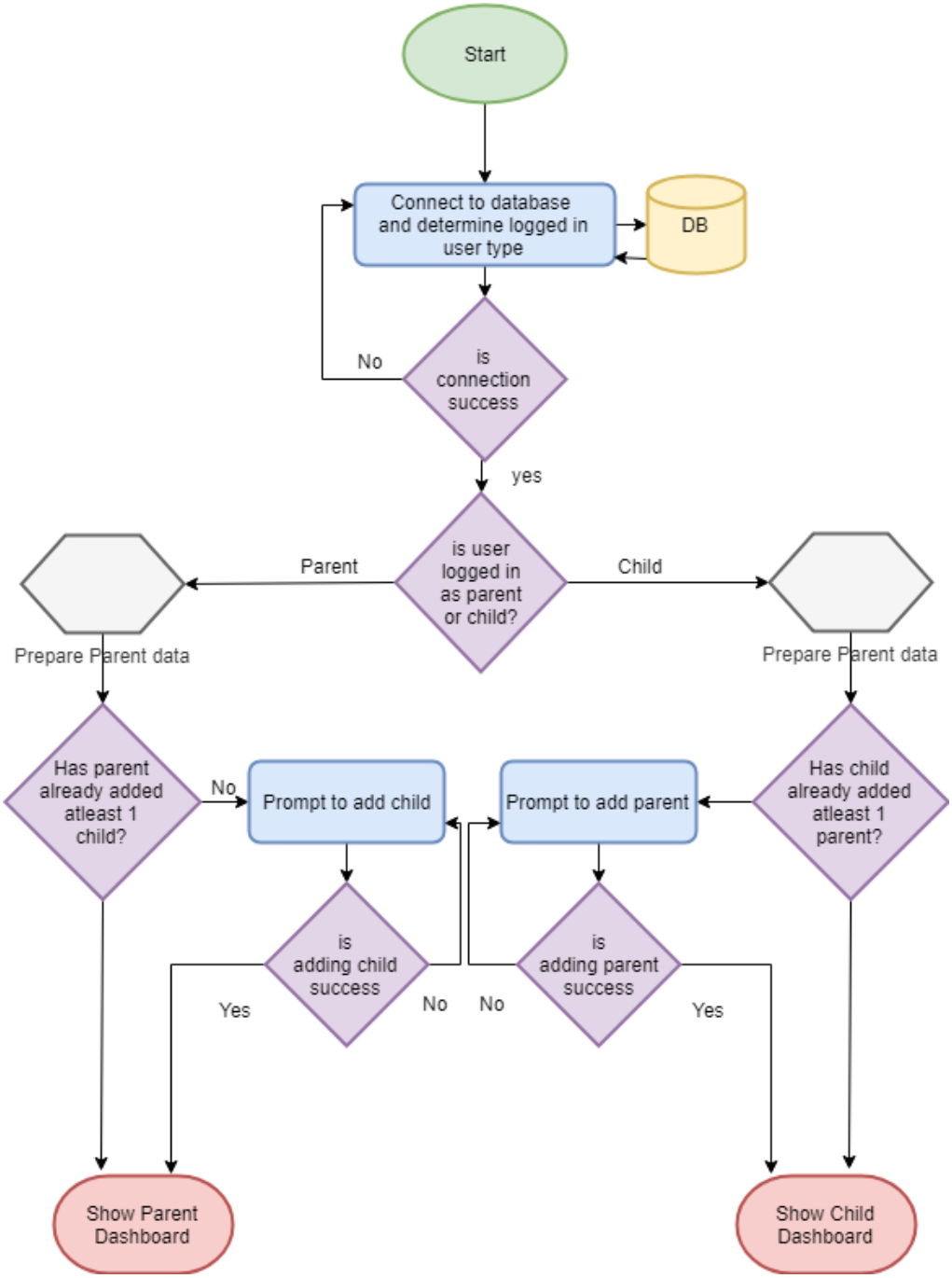


Figure4.55: Parent child dashboard flowchart

Child

The user connects and determines the logged in user type. If the connection is successful and the user is a child, it is checked if the child has already added 1 parent. If he hasn't the system prompts to add a parent in order to continue.

Parent

If the connection is successful and the user is a Parent, it is checked if the parent has already added 1 child. If he hasn't the system prompts to add child in order to continue.

4.2.2.2 Firebase installation

Firebase

Connecting app to Firebase
Before we can start connecting to any Firebase service, we need to install the Firebase sdk into the android app and add our app to Firebase project in Firebase console. Following two steps are necessary to make a connection between client (app) and server (Firebase).

- First, we must Install the Firebase SDK
- Then Add app to Firebase project in Firebase console

Install the Firebase SDK
In project-level (root-level) build.gradle, add rules in order to include the Google Services plugin. Also we should check to have Google's Maven repository included.

```
buildscript {
    // ...
    dependencies {
      // ...
      // Add the following line:
      classpath 'com.google.gms:google-services:4.2.0'  // Google Services
plugin
    }
  }

  allprojects {
    // ...
    repositories {
      // Check that you have the following line (if not, add it):
      google()  // Google's Maven repository
      // ...
    }
  }
```

Figure 4.56: Gradle

```
apply plugin: 'com.android.application'

  android {
    // ...
  }

  // Add the following line to the bottom of the file:
  apply plugin: 'com.google.gms:google-services'  // Google Play services
Gradle plugin
```

Figure 4.57: Google services

Final we must add firebase sdk dependency to the app-level Gradle file

```
dependencies {
  // ...
  implementation 'com.google.firebase:firebase-core:16.0.8'

  // Getting a "Could not find" error? Make sure that you've added
  // Google's Maven repository to your root-level build.gradle file
}
```

That's it. Sync the project with Gradle files and our app is ready to be linked to Firebase on the server side. We must complete the second step which is described below else we may encounter an error.

Add app to Firebase project in Firebase console

First thing we must do is to create a project on firebase console. We must Go to console.firebase.google.com and sign in with our Google account. Click Add project and enter our desired project name.

After our project is created, we click the Add App button and then select Android to launch the Android setup workflow.

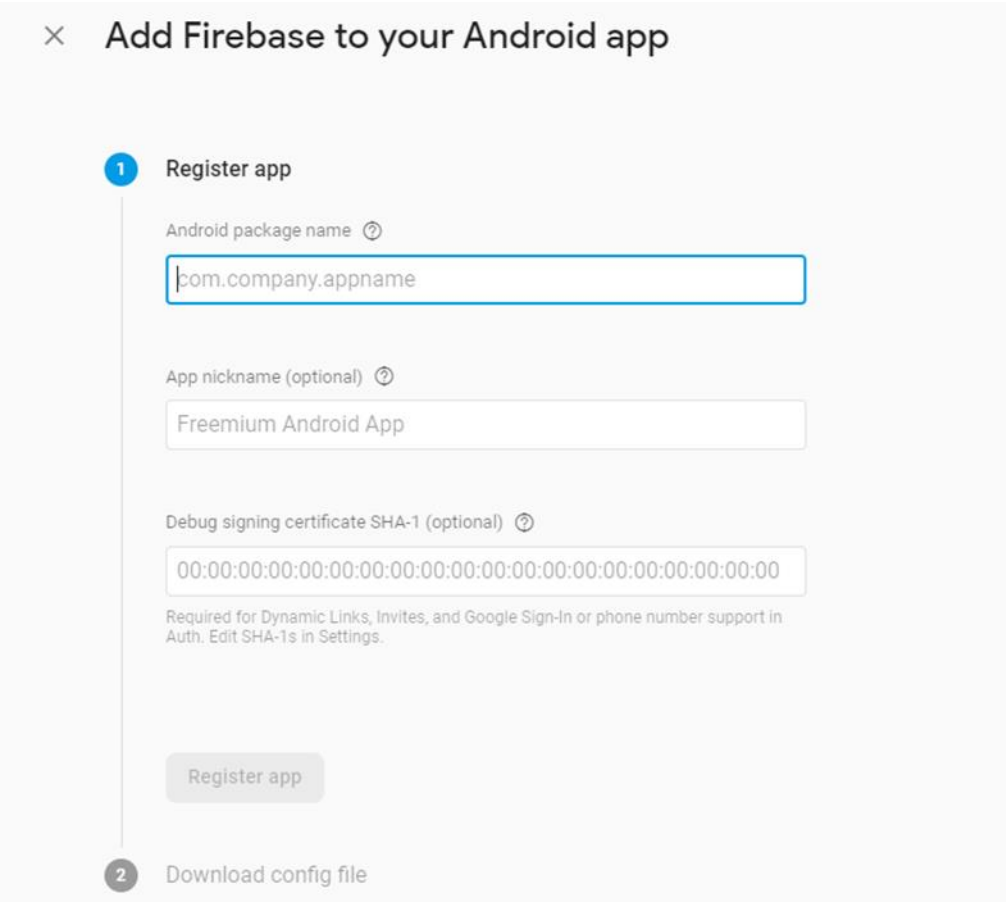


Figure 4.58: Add firebase to app

Next, we enter package name. It must be the same as app's package name in Android studio. Enter app's nickname. Debug signing certificate SHA-1 fingerprint is optional here. But we require it to sign in with google feature in our app. To get this certificate, we should go to Android studio and expand Gradle tab from top right corner window.

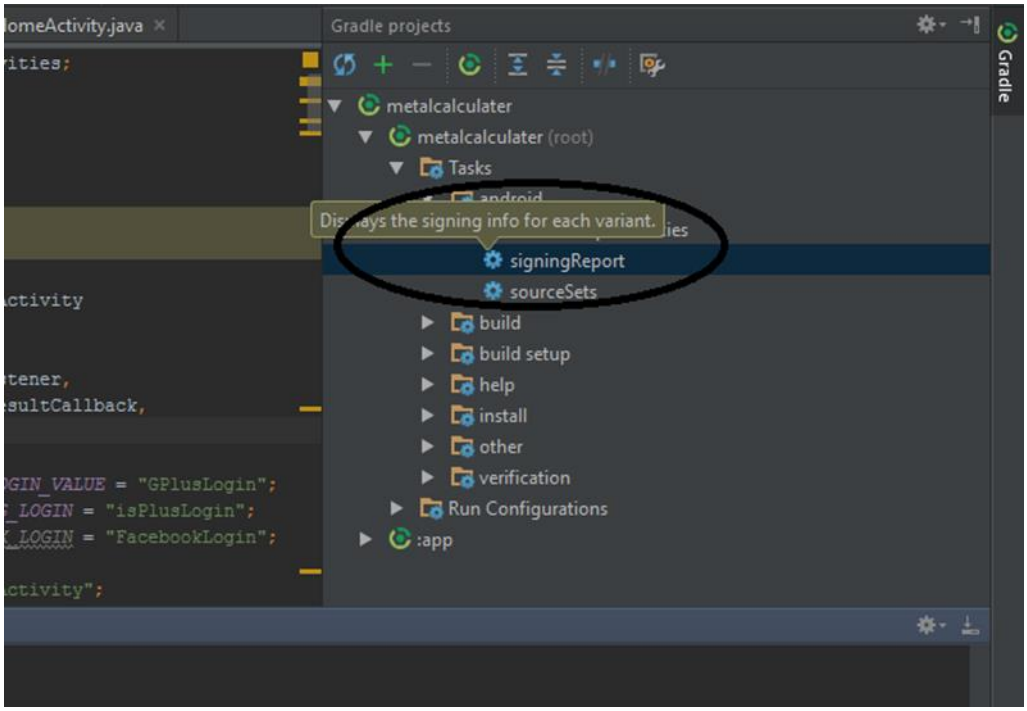


Figure 4.59 :gradle tab

now we can see the SHA-1 certificate under Run tab. Copy it and paste it in firebase console setup workflow.

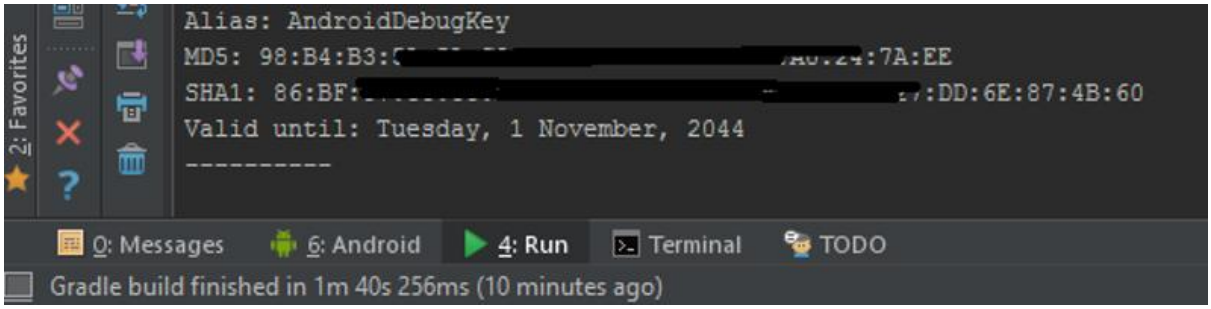


Figure: sha1 certification

Finally click register app. Firebase will now give a downloadable google-services.json file that we must include in app folder of our android studio project.

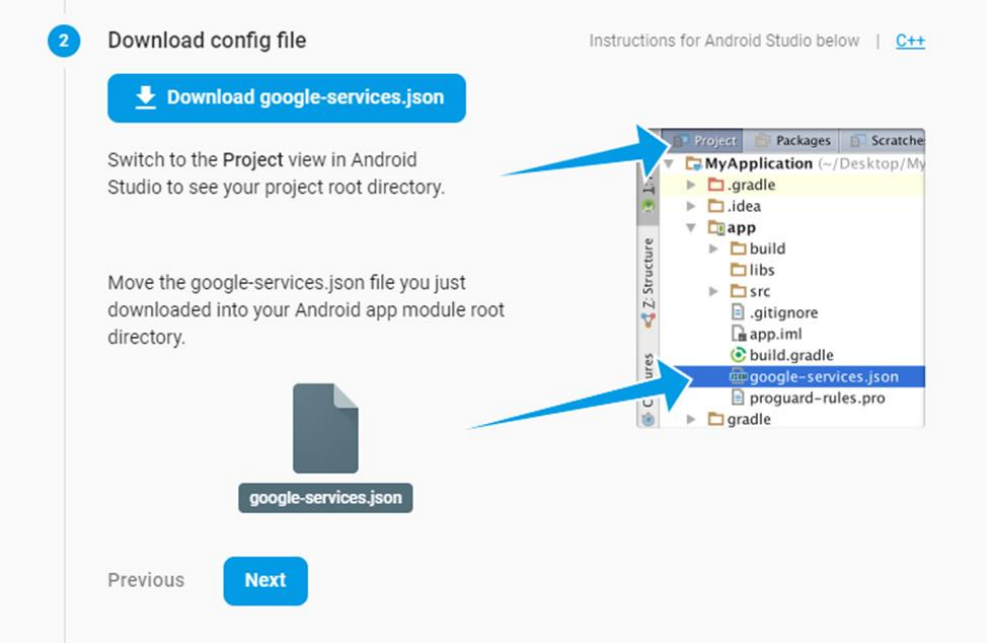


Figure 4.60: Jason google services

In order to add Realtime database we have to include its dependency to our build.gradle

4.2.2.3 Android Project structure

Here we are going to examine the structure of the android project and analyze important parts of the source code. There are many different packages including various type of classes inside the project (Activities, adapters, fragments, messageservice, model classes,).

4.2.2.3.1 Activities

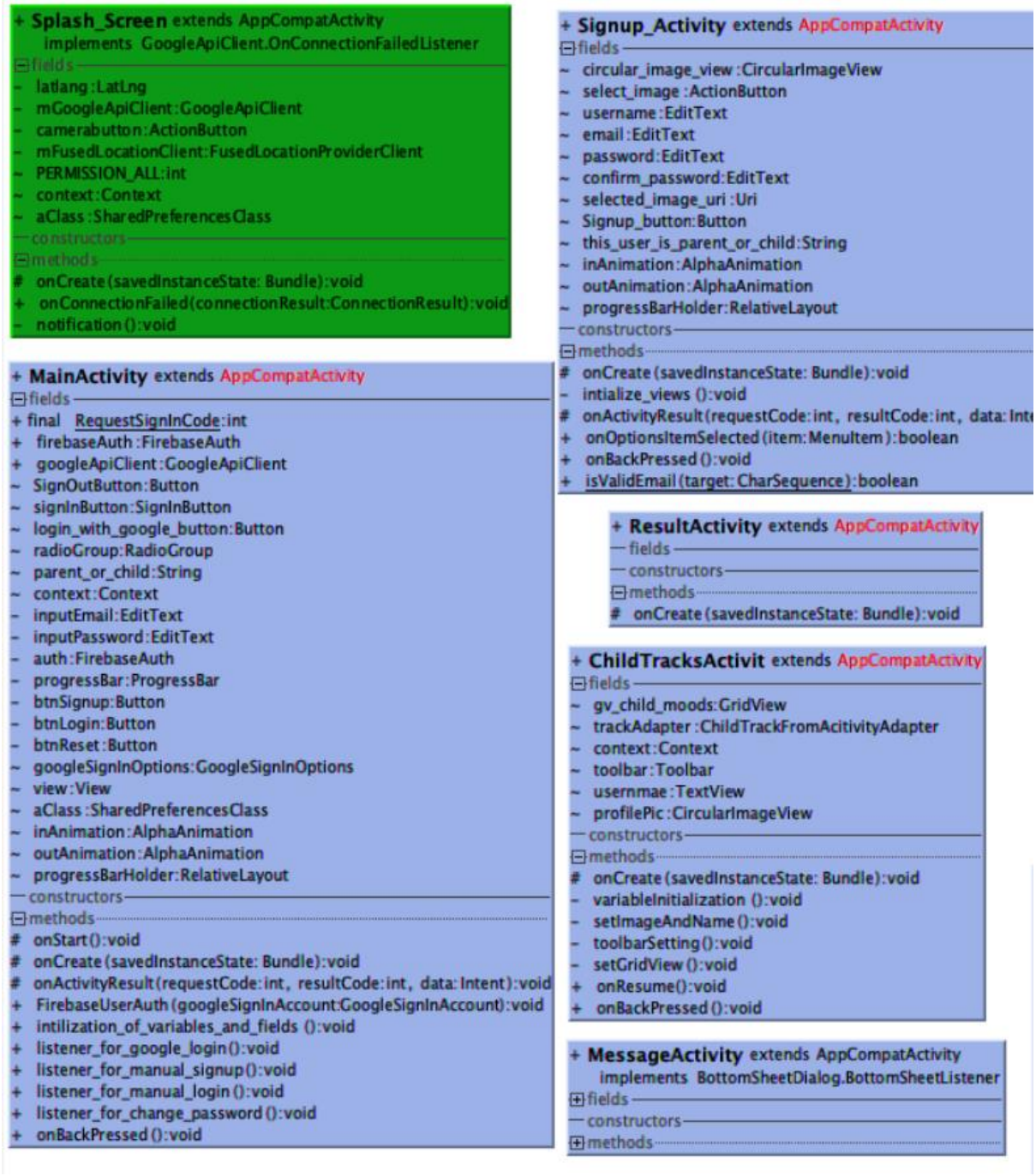


Figure 4.61: Activities

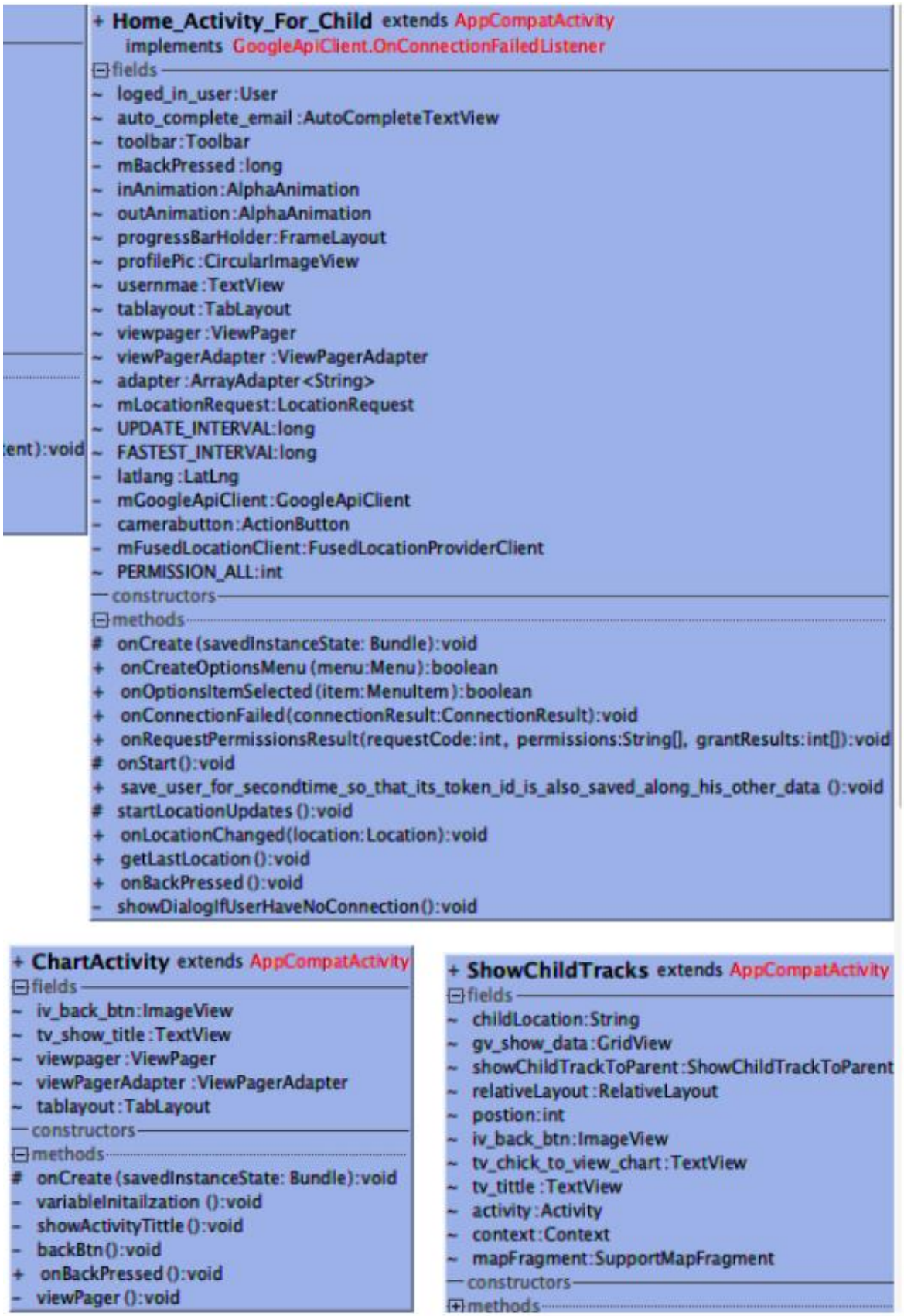


Figure 4.62 Activities 2

Activity package contains all activity classes.

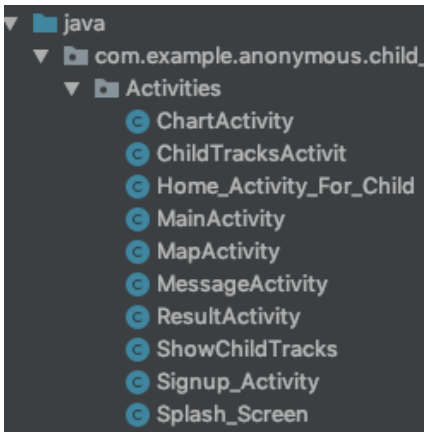


Figure 4.63: Activities

Examples of activities explained:

Splash screen Activity

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_splash_screen);

    mFusedLocationClient =
LocationServices.getFusedLocationProviderClient(Splash_Screen.this);
    //notification();
    //getentire();  // here
    AppVariableClass.getInstance().dataLists();

    mGoogleApiClient = new GoogleApiClient
            .Builder(Splash_Screen.this)
            .addApi(Places.GEO_DATA_API)
            .addApi(Places.PLACE_DETECTION_API)
            .enableAutoManage(new Splash_Screen(), this)
            .build();
    // get_device_location()



    context = Splash_Screen.this;
    Thread mythread = new Thread() {
        @Override
        public void run() {
            super.run();
```

```
            FirebaseUser firebaseUser = FirebaseAuth.getInstance().getCurrentUser();
            if (firebaseUser != null) {
                new
Firebase_Actions().getThisUsersAllConnections(firebaseUser.getEmail());

AppVariableClass.getInstance().setUserStatus(aClass.SharedPreferencesGetObject
("status", context));
                new Firebase_Actions().retrieve_specific_user(Splash_Screen.this,
firebaseUser.getEmail(), null, true, null, null, false);
            } else {
                Intent intent = new Intent(getApplicationContext(), MainActivity.class);
                startActivity(intent);
                finish();
            }
            //finish();


        }
    };
    mythread.start();
```

oncreate() method is used to start an activity. We analyzed the lifecycle in the methodology chapter. The activity has its own layout which here is setContentView(R.layout.*activity_splash_screen*); Using this method we can find any layout and start it with the activity. The specific layout is the splash screen layout. In firebase actions we are going to examine the firebase methods that are used.

```
Intent intent = new Intent(getApplicationContext(), MainActivity.class);
            startActivity(intent);
```

Redirects to main activity.

Chart Activity

This activity is responsible to statistics received from the child.

```
ublic class ChartActivity extends AppCompatActivity {
```

Appcompat activity enables the action bar and material design implementations (for example the toolbar)

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_chart);
    variableInitailzation();
```

```
    showActivityTittle();
    viewPager();
    backBtn();
}
```

On create method calls the layout(activity_chart) which contains a tab layout and a viewpager in order to show statistics in daily, weekly monthly

```
rivate void viewPager(){
    viewPagerAdapter.addfragment(new DailyFragment(), "Daily");
    viewPagerAdapter.addfragment(new WeeklyFragment(), "Weekly");
    viewPagerAdapter.addfragment(new MonthlyFragment(), "Monthly");
    viewpager.setAdapter(viewPagerAdapter);
    viewpager.setOffscreenPageLimit(4);
    tablayout.setupWithViewPager(viewpager);
```

view pager allows to slide between different screens or fragments. The daily weekly monthly chart is based on fragments.

Childtrackks Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_child_tracks);
```

```
private void setGridView() {
    trackAdapter = new
ChildTrackFromAcitivityAdapter(AppVariableClass.getInstance().childMoodToShowP
arent, this, AppVariableClass.getInstance().listOfImageResource);
    gv_child_moods.setAdapter(trackAdapter);
}
```

gridview() shows items in two dimensional scrolling grid. Using the adapter (ChildTrackFromAcitivityAdapter) the data from appvariableclass are bonded and the grid view is responsible to make them visible.

Home activity for child

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_dashboard);
```

```
if (AppVariableClass.getInstance().getUserStatus().equals("child")) {
    viewPagerAdapter.addfragment(new ChildFragment(), "Activity");
} else {
    viewPagerAdapter.addfragment(new ParentFragmentToGetChildTracks(),
"Activity");
}
viewPagerAdapter.addfragment(new UsersFragment(), "Chat");
viewpager.setAdapter(viewPagerAdapter);
tablayout.setupWithViewPager(viewpager);
```

using viewpager adapter here the fragments activity and chat are added

MAIN ACTIVITY

```
public void FirebaseUserAuth(GoogleSignInAccount googleSignInAccount) {

    AuthCredential authCredential =
GoogleAuthProvider.getCredential(googleSignInAccount.getIdToken(), null);
    firebaseAuth.signInWithCredential(authCredential)
            .addOnCompleteListener(MainActivity.this, new
OnCompleteListener<AuthResult>() {
                @Override
                public void onComplete(@NonNull Task<AuthResult> AuthResultTask) {
                    if (AuthResultTask.isSuccessful()) {
                        if (parent_or_child != null && !parent_or_child.equals("")) {
                            FirebaseUser firebaseUser = firebaseAuth.getCurrentUser();
                            User user = new User();
                            user.setUsername(firebaseUser.getDisplayName());
                            user.setEmail(firebaseUser.getEmail());
                            user.setPic_url(firebaseUser.getPhotoUrl().toString());
                            user.setPhone_number(firebaseUser.getPhoneNumber());
                            user.setAppid(FirebaseInstanceId.getInstance().getToken());
                            user.setThis_user_is_parent_or_child(parent_or_child);
                            inAnimation = new AlphaAnimation(0f, 1f);
                            inAnimation.setDuration(200);
                            progressBarHolder.setAnimation(inAnimation);
                            progressBarHolder.setVisibility(View.VISIBLE);
                            Static_information.show_progressbar(MainActivity.this);

                            new
Firebase_Actions().check_if_this_user_already_exists(MainActivity.this,
user.getEmail(), "", user, null, progressBarHolder, "google");
```

using firebaseAuth.getCurrentUser(); we get the current user. If there is no user signed in then it returns null.

Map Activity

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_marker_demo);

    latitude = getIntent().getDoubleExtra("latitude", 0);
    longitude = getIntent().getDoubleExtra("longitude", 0);
    SupportMapFragment mapFragment = (SupportMapFragment)
getSupportFragmentManager()
            .findFragmentById(R.id.map);
    mapFragment.getMapAsync(this);
```

Getting latitude and longitude for location

Message Activity

```
DatabaseReference database =
FirebaseDatabase.getInstance().getReference("parent_or_child");
    database.child("Messages").push().setValue(message);
    text_send.setText("");

}
```

Here a list of data is saved in the database using push database.child("Messages").push().

Showchildtracks Activity

```
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_show_child_tracks);
```

```
@SuppressLint("SetTextI18n")
private void gridView() {
    if (postion == 0) {
        tv_tittle.setText("Child's Mood");
        if (AppVariableClass.getInstance().childMoodTrack.size() != 0) {
            showChildTrackToParent = new ShowChildTrackToParent(this,
AppVariableClass.getInstance().childMoodTrack, postion );
```

```
            gv_show_data.setAdapter(showChildTrackToParent);
        } else {
            gv_show_data.setVisibility(View.GONE);
            relativeLayout.setVisibility(View.VISIBLE);
            tv_chick_to_view_chart.setVisibility(View.GONE);
        }
    } else if (postion == 1) {
        tv_tittle.setText("Messages");
        tv_chick_to_view_chart.setVisibility(View.GONE);
        if (AppVariableClass.getInstance().childPreDefinedMsg.size() != 0) {
            showChildTrackToParent = new ShowChildTrackToParent(this,
AppVariableClass.getInstance().childPreDefinedMsg, postion);
            gv_show_data.setAdapter(showChildTrackToParent);
        } else {
```

Show child tracks. Using showChildTrackToParent Adapter.

Signup Activity

```
User user = new User(usernmae_value, email_value, selected_image_uris, null,
app_id, this_user_is_parent_or_child, password_value);

Static_information.context_to_destory_of_home_Activity = Signup_Activity.this;
new Firebase_Actions().check_if_this_user_already_exists(Signup_Activity.this,
email_value, password_value, user, selected_image_uri, progressBarHolder,
"manuall");
// new
Firebase_Actions().register_user(Signup_Activity.this,email_value,password_value,u
ser,selected_image_uri,progressBarHolder);
```

Here checks if the user already exists.

4.2.2.3.2 Adapters

Adapters package contains all adapters that are responsible for bridging view and data with classes. An adapter works as a bond and connects an adapter view and the underlying data that interact with that view.it is also responsible to create view for each item in the data set

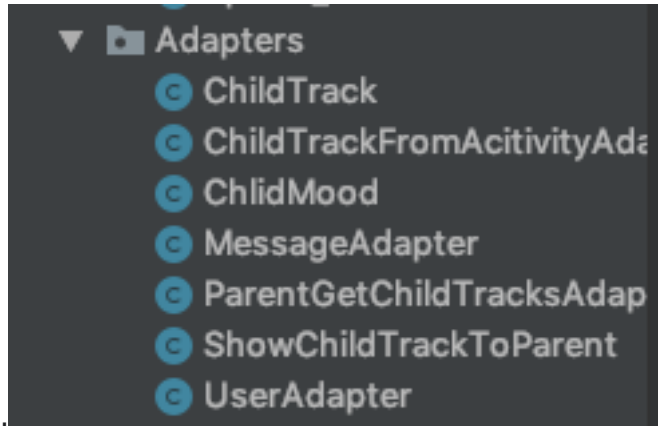


Figure 4.64: Adapters

Childmood Adapter

```
private void sendFireBaseChildWater(String tracks, String anxiety) {
    ChlidMoodClass moodClass = new ChlidMoodClass();
    String date = getCurrentDate();

moodClass.setChild_appid(AppVariableClass.getInstance().getLoggedInUserAppId()
);

moodClass.setChild_email(AppVariableClass.getInstance().getLoggedInUserEmail()
);

moodClass.setChild_username(AppVariableClass.getInstance().getLoggedInUserNa
me());
    moodClass.setChildAnxiety(anxiety);
    moodClass.setDateAndTime(date);
    DatabaseReference database =
FirebaseDatabase.getInstance().getReference("parent_or_child").child("ChildTracks"
)
            .child(AppVariableClass.getInstance().getLoggedInUserEmail().replace(".",
"").replace("@", "")).child("AnxietyTrack").child(tracks);
    database.push().setValue(moodClass);
    Toast.makeText(context, "Successfully sent ", Toast.LENGTH_SHORT).show();
}
```

```
private void sentPredefinedMsgForNotification(String msg) {

    Message message = new Message();
```

```
    for (int i = 0; i < AppVariableClass.getInstance().userEmailCheck.size(); i++) {

message.setReciever_appid(AppVariableClass.getInstance().userAppIdForNotificati
on.get(i));

message.setSender_appid(AppVariableClass.getInstance().getLoggedInUserAppId()
);


message.setReciever_email(AppVariableClass.getInstance().userEmailCheck.get(i))
;

message.setSender_email(AppVariableClass.getInstance().getLoggedInUserEmail()
);


message.setSender_username(AppVariableClass.getInstance().getLoggedInUserNa
me());


message.setSender_this_user_is_parent_or_child(Static_information.loged_in_user.
getThis_user_is_parent_or_child());


message.setReciever_phone_number(Static_information.loged_in_user.getPhone_n
umber());

message.setSender_phone_number(Static_information.loged_in_user.getPhone_nu
mber());

        message.setMessage(msg);

        DatabaseReference database =
FirebaseDatabase.getInstance().getReference("parent_or_child");
        database.child("Notification").push().setValue(message);
    }
```

4.2.2.3.3 Application Variables

App_Variables package contains functions that are repeatedly used throughout the code. Our Application classes is also in this package. Our Firebase server-side handling is also present here

Figure 4.65: App variables

```
ublic void dataLists() {
    if (listOfTracks.size() <= 0) {

        listOfTracks.add("How is your mood?");
        listOfTracks.add("Send a message");
        listOfTracks.add("How well did you sleep?");
        listOfTracks.add("Water intake");
        listOfTracks.add("How are you feeling?");
        listOfTracks.add("Send Current Location");


        childMoodToShowParent.add("Track child's mood");
        childMoodToShowParent.add("Messages sent by child");
        childMoodToShowParent.add("Sleep record of the child");
        childMoodToShowParent.add("Water intake of the child");
        childMoodToShowParent.add("Child's health track")
childMoodToShowParent.add("Locations sent by the child");
```

```
public ArrayList<String> listOfTracks = new ArrayList<>();
public ArrayList<Integer> listOfMoods = new ArrayList<>();
```

4.2.2.3.4 Fragments

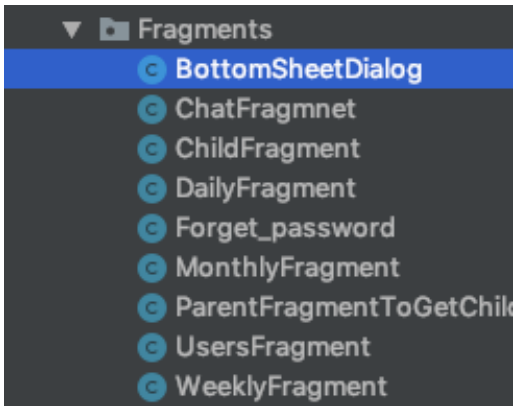


Figure 4.66: Fragments

Fragments package contains all fragment classes.

```
private void layoutPredefinedMessages(final Dialog dialog) {
    childMoodAdapter = new
ChlidMood(AppVariableClass.getInstance().listOfPredefinedMsg, getContext(),
AppVariableClass.getInstance().listOfLayoutUseInDialogBoxsGridview.get(1));
    gv_dialog_box.setNumColumns(1);
    gv_dialog_box.setAdapter(childMoodAdapter);
    gv_dialog_box.setOnItemClickListener(new AdapterView.OnItemClickListener() {
        @Override
        public void onItemClick(AdapterView<?> parent, View view, int position, long id)
{
            sentPredefinedMsg(position);

sendFireBaseNotification(AppVariableClass.getInstance().listOfPredefinedMsg.get(p
osition), "PredefinedMessages");
            dialog.cancel();
            Toast.makeText(getContext(), "Successfully Send",
Toast.LENGTH_SHORT).show();
        }
    });
```

set adapter

Messageservice

MessageService contains files for handling firebase remote messages.

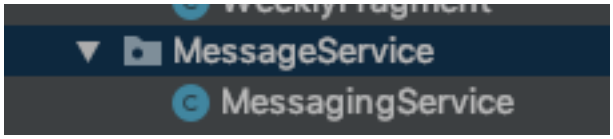


Figure 4.67: MessageService

4.2.2.3.5 MessageService

Contains files for handling firebase remote messages

```
private void sendNotification(String notificationTitle, String notificationBody) {
    Intent intent = new Intent(this, MainActivity.class);
    intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
    PendingIntent pendingIntent = PendingIntent.getActivity(this, 0, intent,
            PendingIntent.FLAG_ONE_SHOT);

    Uri defaultSoundUri=
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION);
    NotificationCompat.Builder notificationBuilder = (NotificationCompat.Builder) new
NotificationCompat.Builder(this)
            .setAutoCancel(true)   //Automatically delete the notification
            .setSmallIcon(R.mipmap.ic_launcher) //Notification icon
            .setContentIntent(pendingIntent)
            .setContentTitle(notificationTitle)
            .setContentText(notificationBody)
            .setSound(defaultSoundUri);


NotificationManager notificationManager = (NotificationManager)
getSystemService(Context.NOTIFICATION_SERVICE);
```

4.2.2.3.6 Firebase actions

```
ublic class Firebase_Actions {

    public void userLogin(final Context application_context, final Context context, final
FirebaseAuth auth, final String email, String password, final String parent_or_child,
final RelativeLayout progressBarHolder) {
        auth.signInWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
```

```
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            // progressBar.setVisibility(View.GONE);
            if (task.isSuccessful()) {
                getThisUsersAllConnections(email);
                retrieve_specific_user(context, email, progressBarHolder, true, null, null,
false);
            } else {
                progressBarHolder.setVisibility(View.GONE);
                Static_information.hide_progressbar((Activity) context);
                Toast.makeText(application_context, task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
            }
        }
    });
```

```
public void register_user(final Context context, String email, String password, final
User user, final Uri selected_image_uri, final RelativeLayout progressBarHolder) {
    FirebaseAuth.getInstance().createUserWithEmailAndPassword(email,
password).addOnCompleteListener(new OnCompleteListener<AuthResult>() {
        @Override
        public void onComplete(@NonNull Task<AuthResult> task) {
            // progressBar.setVisibility(View.GONE);
            Log.d("ratio", task.isSuccessful() + "  level_1");
            if (task.isSuccessful()) {
                //  Toast.makeText(context,"user registered
successfully",Toast.LENGTH_LONG).show();
                save_image(context, user, selected_image_uri, progressBarHolder);
            } else {

                if (task.getException() instanceof FirebaseAuthUserCollisionException) {
                    Toast.makeText(context, "A user with this email address is already
registered!!", Toast.LENGTH_SHORT).show();
                    progressBarHolder.setVisibility(View.GONE);
                    Static_information.hide_progressbar((Activity) context);

                } else {
                    Toast.makeText(context, task.getException().getMessage(),
Toast.LENGTH_SHORT).show();
                    if (Static_information.context_to_destory_of_home_Activity != null) {
                        ((Activity)
```

```
Static_information.context_to_destory_of_home_Activity).finish();
                    ((Activity)
Static_information.context_to_destory_of_main_Activity).finish();

                    Intent intent = new Intent(context, MainActivity.class);
                    context.startActivity(intent);
                }
            }

        }
    }
});
}
```

```
ublic void check_if_this_user_already_exists(final Context context, final String email,
final String password, final User users, final Uri selected_image_uri, final
RelativeLayout progressBarHolder, final String google_signin_or_manuall) {
    final FirebaseDatabase database = FirebaseDatabase.getInstance();
    database.getReference("parent_or_child").child("Users").child((email.replace(".",
"")).replace("@", "")).addListenerForSingleValueEvent(new ValueEventListener() {
        @Override
        public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
            User user = dataSnapshot.getValue(User.class);
            getThisUsersAllConnections(email);
            if (user == null) {
                if (google_signin_or_manuall.equals("google")) {
                    save_user(context, users, progressBarHolder);
                } else {
                    new Firebase_Actions().register_user(context, email, password, users,
selected_image_uri, progressBarHolder);
                }

            } else {
                if (google_signin_or_manuall.equals("google")) {
                    if
(!user.getThis_user_is_parent_or_child().equals(Static_information.loged_in_user_or
_signup_user_is_child_or_doctor)) {
                        progressBarHolder.setVisibility(View.GONE);
                        Static_information.hide_progressbar((Activity) context);
                        FirebaseAuth.getInstance().signOut();
                        Toast.makeText(context, "This user is already registered as " +
```

```
user.getThis_user_is_parent_or_child(), Toast.LENGTH_SHORT).show();
                } else {
                    retrieve_specific_user(context, user.getEmail(), progressBarHolder,
true, null, null, false);
                }

            } else {
                Toast.makeText(context, "This user is already registered",
Toast.LENGTH_LONG).show();
                progressBarHolder.setVisibility(View.GONE);
                Static_information.hide_progressbar((Activity) context);
            }
```

4.2.2.3.7 Model classes

Figure 4.68 Model classes

Figure 4.69: Model classes

Model example

```
public String getDateAndTime() {

    return dateAndTime;
}

public void setChildLocation(String childLocation) {
    this.childLocation = childLocation;
}

public String getChildLocation() {

    return childLocation;
```

Model using getters and setters (encapsulation)

4.2.2.3.8 Tech stack

Following are the programming languages, frameworks and tools used to develop this project:

- Android Studio for building client-side code
- Firebase for building server-side code
- Java for writing client-side code
- XML for app layouts
- Node.js for triggering notifications
- NoSQL Database for data management
- RestFul apis for calling firebase methods and functions

4.2.2.3.9 Libraries and apis

Following libraries and APIs are used in this project:

- Firebase authentication for login/signup system
- Firebase realtime database for storing and retrieving data
- Firebase storage for storing profile images
- Firebase messaging service for sending and receiving notifications
- Firebase crashlytics to detect app fatal and non-fatal errors
- Google Maps for detecting and sending precise location
- Firebase Cloud Function written in Node.js to trigger events in firebase database
- AnyChart open source library to read and write charts

4.3 Testing

For testing firebase test lab will be used and more specifically the Robo test tool. First, we need an android phone in order to run our application and android studio and select record robo script. Then we choose the android device as an emulator. First, we must record our scenarios.

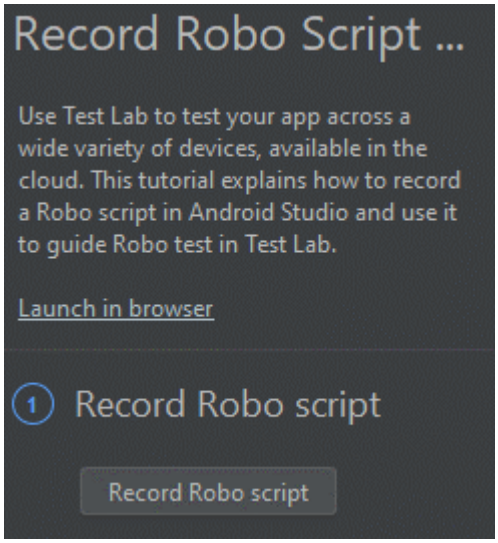


Figure 4.70: Android studio record robo script

The following scenarios are going to be recorded and tested:

4.3.1 Child parent test cases

Common functionality

1. **Create child account (PASS)**

- Select child or parent
- Type your email
- Type your password
- Press Submit

2. **Create Parent account (PASS)**

- Select child or parent
- Type your email
- Type your password
- Press Submit

3. **Forgot password (PASS)**

The test is running in android device using android studio and robo test recorder. Since we signup with child or parent users in order to create the accounts the first time, in order to record the script, the second time that the robo tries to do the same (create accounts) it can't, since user already exists. On the video which is available under the folder procedure can be seen. Even if the test can't create the users, they have successfully been created.

Figure 4.71: Common functionality scenarios

The same applies for add parent. When the user logins and there is no parent connected, the system asks for child or parent email address. Since in order to record the script the parent child have been successfully added the first-time test robo cannot test the add parent. We have to delete manually the connected user in order to test it, or we can edit the json script file with a username that has not been registered to see the results. The user successfully logins in the next test.

4.3.2 Child test cases

1. **login manually as a child (PASS)**
2. **Add parent**. (**PASS**)
3. **Choose how is your mood (PASS)**
4. **Send predefined message (PASS)**
5. **Capture how well did you sleep (PASS)**
6. **Capture water intake (PASS)**
7. **Capture how are you feeling (PASS)**
8. **Send current location (PASS)**
9. **Add chat user (Passed) (no available email in the database)**
10. **Send message (PASS)**

During the child test we encountered some problems with testlab firebase. Robo couldn't choose the right button to sign in manually, instead was using the login with google using test credentials. The reason is that robo tries to find sources from their name. The buttons googlelogin and login uses both the word "login". In order to work we have changed in the source of the app the id of the google sign in button, to "google". According to firebase documentation most often, Robo scripts fail because Test Lab isn't able to find a necessary element on the screen.

Json script.

{

"eventType": "VIEW_TEXT_CHANGED",

"timestamp": 1565716929659,

"replacementText": "child1@gmail.com",

"actionCode": -1,

"delayTime": 0,

"canScrollTo": true,

"elementDescriptors": [

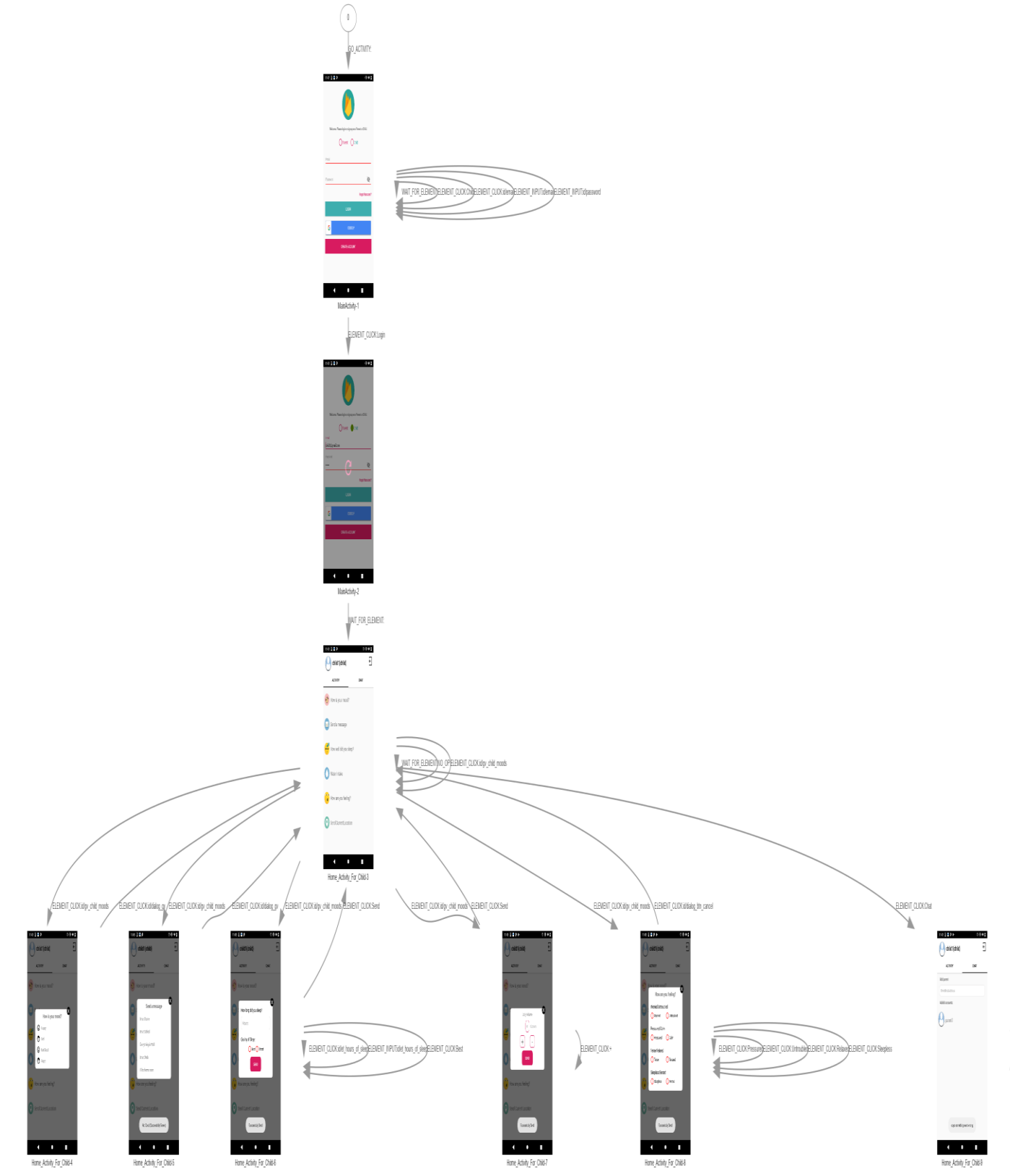


Figure 4.72: Child test cases

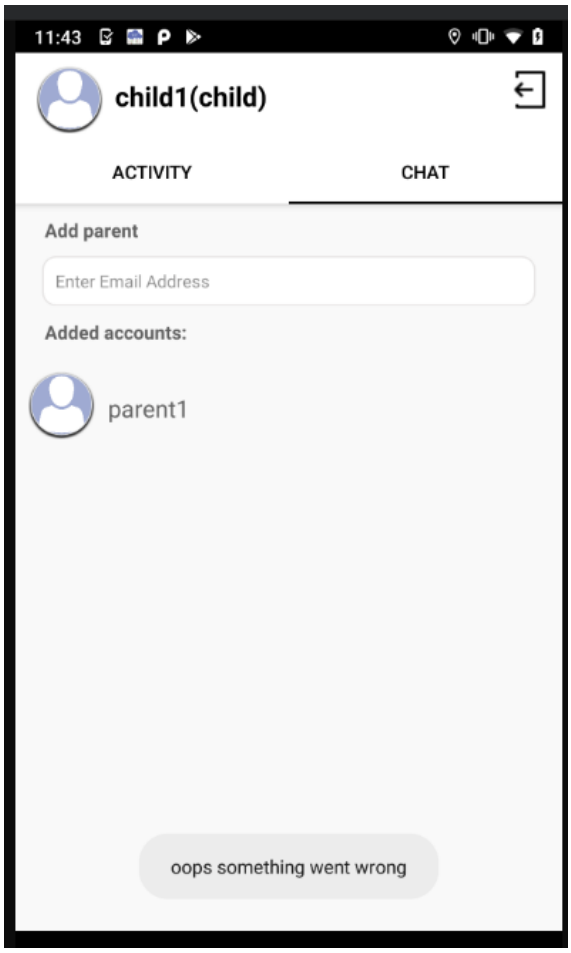


Figure 4.73: Add user toast message

Robo crashed test no.12 when presses on chat test and tries to check for email addresses, as there are no email addresses to choose. We have to add more users.

FATAL EXCEPTION: Thread-2

Process: androidx.test.tools.crawler:robov2crawldriver, PID: 13948java.lang.IllegalArgumentException: The text for # androidx.test.tools.crawler.proto.ElementDescription@e25df84class_name: "android.support.v7.widget.AppCompatAutoCompleteTextView"component_type: androidx.test.tools.crawler.proto.ElementDescription$ComponentType@98460a6 number=5 name=EDITABLE_TEXT

4.3.3 Parent test cases

1. **login manually as a Parent (PASS)**
2. **view mood (PASS)**
3. **View predefined message (PASS)**
4. **view sleep? (PASS)**
5. **view water intake (PASS)**
6. **view feeling (PASS)**
7. **view location (PASS)**
8. **chat message (PASS)**
9. **Delete statistics (PASS)**
10. **logout (PASS)**

Figure 4.74: Performance

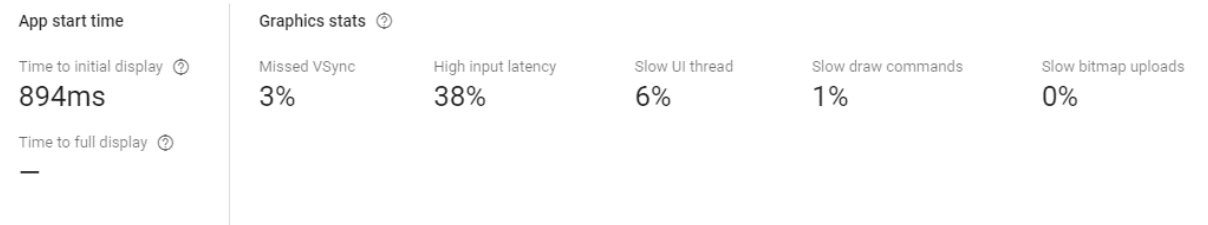


Figure 4.75: Test stats

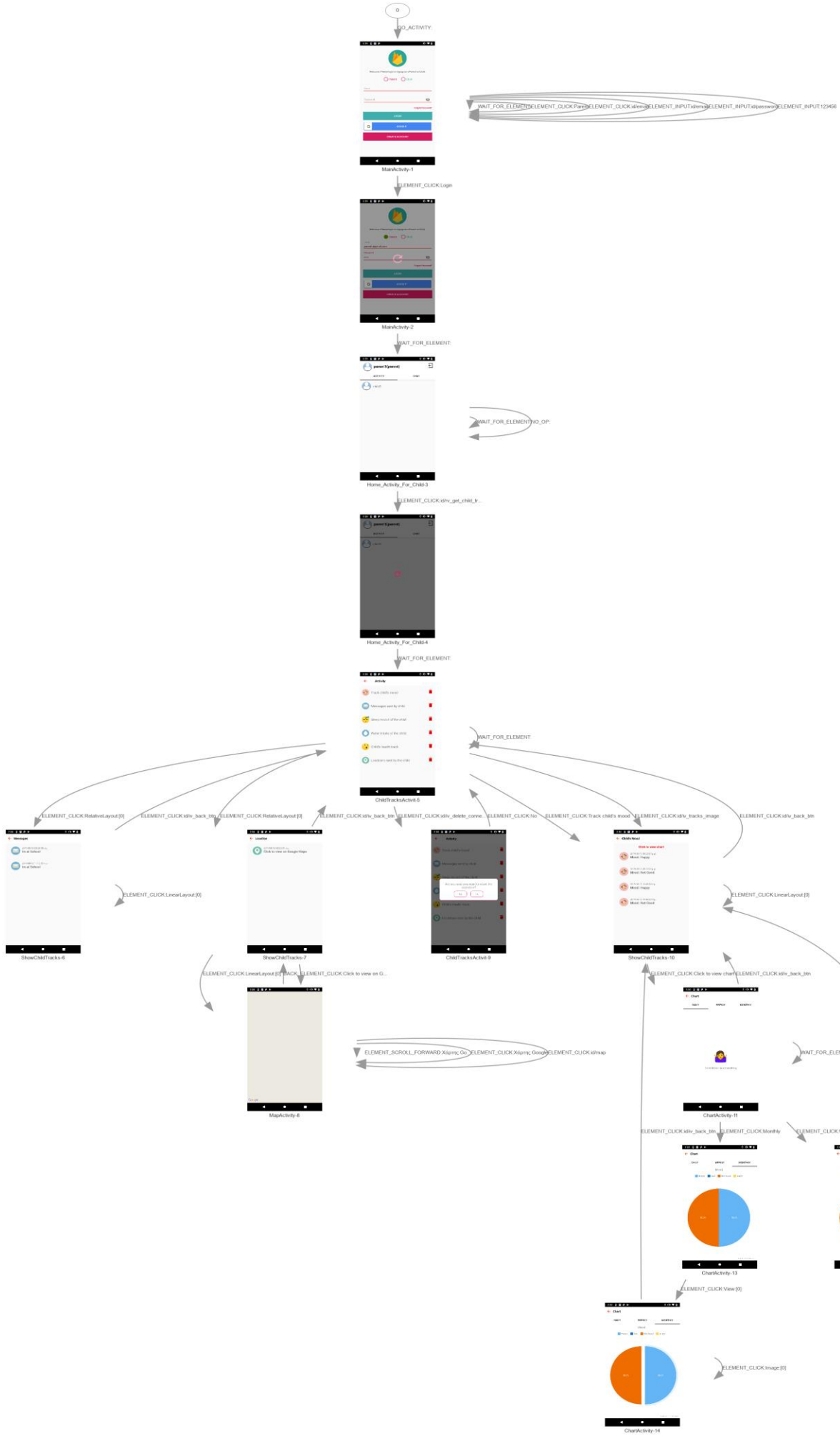

0
DO_ACTIVITY:
MainActivity-1
WAIT_FOR_ELEMENT:ELEMENT_CLICK:Parent;ELEMENT_CLICK:id/emai;ELEMENT_INPUT:id/email;ELEMENT_INPUT:id/password;ELEMENT_INPUT:123456
ELEMENT_CLICK:Login
MainActivity-2
WAIT_FOR_ELEMENT:
Home_Activity_For_Child-3
WAIT_FOR_ELEMENT:NO_OP:
ELEMENT_CLICK:id/rv_get_child_lr...
Home_Activity_For_Child-4
WAIT_FOR_ELEMENT:
ChildTracksActivit-5
WAIT_FOR_ELEMENT:
ELEMENT_CLICK:RelativeLayout:[0]
ELEMENT_CLICK:id/iv_back_btn
ELEMENT_CLICK:RelativeLayout:[0]
ELEMENT_CLICK:id/iv_back_btn
ELEMENT_CLICK:id/iv_delete_conne...
ELEMENT_CLICK:No
ELEMENT_CLICK:Track child's mood
ELEMENT_CLICK:id/iv_tracks_image
ELEMENT_CLICK:id/iv_back_btn
ShowChildTracks-6
ELEMENT_CLICK:LinearLayout:[0]
ShowChildTracks-7
ChildTracksActivit-9
ShowChildTracks-10
ELEMENT_CLICK:LinearLayout:[0]
ELEMENT_CLICK:LinearLayout:[0]
ELEMENT_CLICK:BACK
ELEMENT_CLICK:Click to view on G...
ELEMENT_CLICK:Click to view chart
ELEMENT_CLICK:id/iv_back_btn
MapActivity-8
ELEMENT_SCROLL_FORWARD:Χάρτης Go...
ELEMENT_CLICK:Χάρτης Google
ELEMENT_CLICK:id/map
ChartActivity-11
WAIT_FOR_ELEMENT:
ELEMENT_CLICK:id/iv_back_btn
ELEMENT_CLICK:Monthly
ELEMENT_CLICK:Weekly
ChartActivity-13
ChartActivity-12
ELEMENT_CLICK:View:[0]
ELEMENT_CLICK:Image:[0]
ChartActivity-14

Figure 4.76: Parent Test cases

4.4 Evaluation

The heuristic evaluation method has been selected as it is quicker as well as simpler method to execute, especially for expert from accessibility computing. It is an expert-based summative evaluation method, which also facilitates app redesigning and versioning.

4.4.1 Severity ratings

No	**Severity ratings**
0	No usability problem
1	cosmetic problem
2	insignificant usability problem
3	significant usability problem; need to be fixed
4	Usability disaster; Must be fixed

Figure 4.77: Severity ratings

4.4.2 Usability problems identified

Sr. No.	**Heuristics**	**Usability problems Identified**	**Severity rating**
1	Visibility of system status	No messenger notifies	1

2	Match between system & the real world	No problem found	0
3	User control & freedom	No problem found	0
4	Consistency & standards	No problem found	0
5	Error prevention	Sleep entry (24hour)	3
6	Recognition rather than recall	No problem found	0
7	Flexibility & efficiency of use	No settings options, or modifications on the user interface available	3
8	8 Aesthetic & minimalist design	Add user screen	1
9	Help users recognize, diagnose & recover from errors	No problem found	0

10	Help and documentation	No help or documentation	2

Figure 4.78: Usability problems identified

4.4.3 Heuristic evaluation

Using heuristic evaluation

1. **Status**

The system provides appropriate feedback to the users. The system informs users what is happening. For example, when logging the progress bar is loading. The progress bar is used in many functions like login. When there are no data in the database shows a screen informing the user for the specific situation using images.

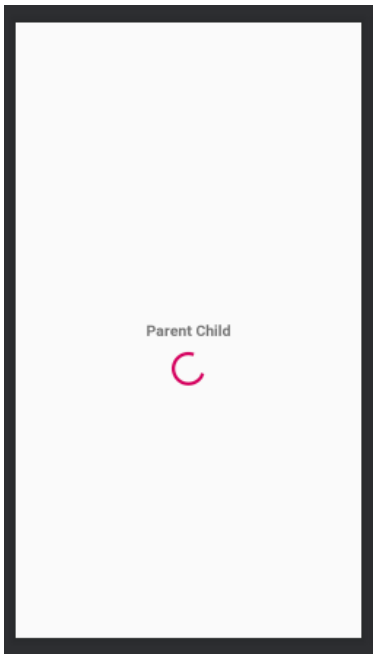


Figure 4.79: Load screen

Problems found

No messenger progress for message progress sent.

2. **Match between system and the real world**.

The system provides a simplistic user interface, following real world conventions based on familiar ideas, which parents and child are familiar with. The information appears in a logical order. Icons like messages or emoticons are used to illustrate moods. The language that is used is aimed for teenagers' group.

3. **User control and freedom**.

The system has a very good implementation for undo, and an "emergency exit" for functions that be chosen by mistake.

4. **Consistency and standards**

The system follows consistency and standard. It is clear what each button does and the terminology that is used is enough. A back button is used in the toolbar. Design guidelines has been followed during the design. A problem is defined with the screen where a user should add child or parent. There is no description on what the user should do.

5. **Error prevention**.

The system supports error prevention. Inform the user in each screen especially using toast messages. One problem found in the sleep capture. The user can use any number in the hours of sleep, without prevent him. (more than 24 hours per day)

6. **Recognition rather than recall.**

The system supports recognition as makes options and actions visible. Each screen says where exactly the user is, from one screen to another. There is also the option to search for a user in the Add user activity, showing available users.

7. **Flexibility and efficiency of use**.

The system is not designed for expert users but for novice users. There are no options to modify or change settings. It has predefined functionality which can't be changed.

8. **Aesthetic and minimalist design**.

The design is minimalistic displaying only relevant information. The dialogues are not containing information that irrelevant. It's simple and effective. One problem found in the screen where the user should add the user. It's not so easy to understand what the system asks for.

9. **Help users recognize, diagnose and recover from errors**.

The errors speak to user language, suggesting the user what to do. The login page gives enough information about what the user needs to do, to recover from errors. The same applies to the signup screen.

10. **Help and documentation**.

There is no documentation at all.

Chapter 5: Conclusion and feature work

In this project we examined the nature of healthy communication in a parent child relationship and we tried to find ways to to implement specific features in the application in order to support healthy communication. This was managed using findings from the literature review, which provided us with functional and non-functional requirements using specific methodology. Now that the application is completed, we can conclude the dissertation objectives of dissertation and examine if they were achieved, summarizing the dissertation.

5.1 Summarize

During the literature review in chapter 2 we examined the nature of healthy communication and analyzed ways to surpass the obstacles which prevent an efficient communication between children and parents. We analysed the various forms of communication and the importance of the right skills that should have a parent in order to communicate successfully with his child. We examined the importance of communication in child parent relationship and how to improve it. We analysed the obstacles that prevent an efficient communication, analysing specific situations that the parents or the children face in their every day life. Finally, the nature of digital communication and how the digital era has changed the way we communicate as the technology had a huge explosion the last years, and how mobile phones are used by parents and children nowadays was examined. We found the importance of the body language during communication, or the tone of voice and examined specific examples. Using the literature review and various techniques methodology from the agile lifecycle approach that we adopted during the software lifecycle, we found ways to transform the needs of children and parents, into requirements in order to design our application which enhance healthy communication. We examined various applications on android market and examined their flaws and strengths, finding ideas for our application. We examined the parent child as users and based on their profile and their needs, we created a user interface that would be pleasant for them. We analysed the various techniques and methodology that was used in each phase (design, implementation, testing, evaluation) of the application development in each phase. We saw the architecture of the android application and examined the architectural principles that would be implemented during the development. We implemented the methodology and followed the plans that was described in methodology chapter in order to design implement test and evaluate the application. We analysed specific technologies we examined their role in the development, and how they were implemented. During the test phase we tested the functionality of the application and then evaluate it, finding specific issues in the user interface that affect it. All the objectives of the dissertation have been successfully accomplished creating an application which supports healthy

communication between child parent. Putting in practice the methodology and technologies that were analysed, we created an application which would be useful for parents and children. The agile approach helped us organize the software development and create a final version of our application. There are many technologies that can be used in order to create healthy communication. In this dissertation we examined many of them. However, there are not many applications available that currently support features to enable healthy communication in general, or in a child parent relationship.

5.2 Recommendation

During the implementation of the application we used Model View Presenter Pattern which is responsible for the presentation layer in order to separate the presentation from the logic and create a flexible and scalable app. If we didn't do that the code might would be much complex making sometimes even impossible to understand it or fix something. In the requirements section we have a functional requirement which would be implemented in the future. (settings requirement). Using this approach, it should be much easier to make fixes and changes to our application or add more features. Generally using MVP, we can create better apps, helping in complicated situations, creating extensible and maintainable applications.

5.3 Future Work

During the market analysis we saw many kinds of applications which supported the functionality that our application supports. There are many changes that should be made in the application in order to make it more effective. We saw for example in the reviews of various applications the need for an SOS button which would notify the user in an emergency. That would be a good feature for a future version. There are many things that can be changed lie the use of more moods, or a settings functionality on the child's menu where the predefined messages could be changed according to his needs. The current predefined messages are not representing a good sample of messages which enhance healthy communication. We can also support predefined voice messages that would give a pleasant effect in the interaction between child and parent. In order to have reminders when the child should add an entry in the appInstead of using alerts and alarms like other applications do we could implement a badge feature. For example, each time the child tracks something in the app a badge will be given. After a collusion with the parent a prize could be given in a discreet number of badges. This could make the application more appealing, change the behaviour of the user and make him be more responsible without using annoying reminders. The same can be applied for parents. Contextual triggers could be used to inform the parent for events that can be a good chance to come in communication with the child. using this way the application becomes more useful and more appealing. There are many

functionalities and technologies that can be implemented in the application which would make it a helpful tool for children and parents.

References

Android Developers. (2019). Fragments | Android Developers. [online] Available at: https://developer.android.com/guide/components/fragments [Accessed 18 Aug. 2019]. raywenderlich.com. (2019).

Android Developers. (2019). Layouts | Android Developers. [online] Available at: https://developer.android.com/guide/topics/ui/declaring-layout.html [Accessed 18 Apr. 2019].

Android Developers. (2019). Linear Layout | Android Developers. [online] Available at: https://developer.android.com/guide/topics/ui/layout/linear [Accessed 18 Apr. 2019].

Android Developers. (2019). Relative Layout | Android Developers. [online] Available at: https://developer.android.com/guide/topics/ui/layout/relative [Accessed 18 Aug. 2019].

Android Developers. (2019). Understand the Activity Lifecycle | Android Developers. [online] Available at: https://developer.android.com/guide/components/activities/activity-lifecycle [Accessed 18 Aug. 2019].

Benford, S., & Bowers, J. (1994). Managing mutual awareness in collaborative virtual environments. conference on Virtual, (August 1994).

Boseley, S. (2019). Mental illness costs UK £94bn a year, OECD report says. [online] the Guardian. Available at: https://www.theguardian.com/society/2018/nov/22/mental-illness-costs-uk-94bn-a-year-oecd-report-says-employment-economy-productivity [Accessed 10 Aug. 2019].

Bowlby, J. (1951). Maternal care and mental health. World Health Organization Monograph

Bowlby, J. (1969), Attachment and loss, Vol. 1: Attachment. New York: Basic Books.

Callie Thomson and Jane Nash (2016). Persuasive Design for Behaviour Change Apps: Issues for Designers. [online] Available at: https://dl.acm.org/citation.cfm?id=2987535 [Accessed 10 Aug. 2019

Communication is the key to healthy family relationships. (2017). Family Times.

EFFECTS OF NEW TECHNOLOGY ON PARENT-CHILD RELATIONSHIP IN DINA, JHELUM. (2017). The explorer, 1(4):106-109).

En.wikibooks.org. (2019). Introduction to Software Engineering/Process/Methodology - Wikibooks, open books for an open world. [online] Available at: https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Process/Methodology [Accessed 10 Aug. 2019].

En.wikipedia.org. (2019). Communication. [online] Available at: https://en.wikipedia.org/wiki/Communication [Accessed 10 Aug. 2019].

Getting Started with MVP (Model View Presenter) on Android. [online] Available at: https://www.raywenderlich.com/7026-getting-started-with-mvp-model-view-presenter-on-android [Accessed 18 Aug. 2019].

Huffpost.com. (2019). Parent-Child Relationship in The Digital Era. [online] Available at: https://www.huffpost.com/entry/parent-child-relationship-in-the-digital-era_b_58b43fb4e4b02f3f81e44a25 [Accessed 10 Aug. 2019].

Ian and Ian (2019). Peter Drucker 5 Rules for Successful Startup Founders and Entrepreneurs in 2019 Pressfarm. [online] Pressfarm. Available at: https://press.farm/peter-drucker-successful-startup-founders-entrepreneurs/). [Accessed 10 Aug. 2019].

Kumar, U. (2013). "A Sociological Study of the Effect of Communication in Industry: Especially in Context of BSP". IOSR Journal Of Humanities And Social Science, 18(6), pp.32-34.

Lenhart, Amanda, Rich ling, Scott Campbelln and Kristen Purcell. (2010). Teens and Mobile Phones. Pew Research Center http://www.pewinternet.org/2010/04/20/teens-and-mobile-phones

Liechti, O. (2000). A digital photography framework enabling affective awareness in home communication. Personal and Ubiquitous Computing, 6–24. Retrieved from http://www.springerlink.com/index/V7J760H0161R475R.pdf

Markopoulos, Panos, Bongers, B., Alphen, E. V., Dekker, J., Dijk, W. V., Messemaker, S., Poppel, J. V., et al. (2005). The Photo Mirror appliance: affective awareness in the hallway. Personal and Ubiquitous Computing, 10(2-3), 128–135.

Material Design. (2019). Develop for Android. [online] Available at: https://material.io/develop/android/ [Accessed 18 Aug. 2019].

Mattsoave.com. (2019). Usability Evaluation of IU.edu | 8. Aesthetic and Minimalist Design. [online] Available at: http://www.mattsoave.com/old/cogs187a/iu_site_eval/8aesthetics.html [Accessed 18 Aug. 2019].

Patricia Luciana Runcan a *, Corneliu Constantineanu b , Brigitta Ielics c , Dorin Popa d (2012). The Role of Communication in the Parent-Child Interaction.

Robinson, L. (2019). Effective Communication - HelpGuide.org. [online] HelpGuide.org. Available at: https://www.helpguide.org/articles/relationships-communication/effective-communication.htm [Accessed 10 Aug. 2019].

Runcan, P., Constantineanu, C., Ielics, B. and Popa, D. (2019). The Role of Communication in the Parent-Child Interaction.

Streetdirectory.com. (2019). The Importance Of Cell Phones In Modern Society. [online] Available at: https://www.streetdirectory.com/travel_guide/153893/cell_phones/the_importance_of_cell_phones_in_modern_society.html [Accessed 10 Aug. 2019].

Schmitz, M. (2019). Types of Communication - Types of Communication Styles | Conover Company. [online] The Conover Company. Available at: https://www.conovercompany.com/types-of-communication/ [Accessed 10 Aug. 2019].

Softwaretestinghelp.com. (2019). Functional Testing: A Complete Guide with Types and Example. [online] Available at: https://www.softwaretestinghelp.com/guide-to-functional-testing/ [Accessed 18 Aug. 2019].

Softwaretestinghelp.com. (2019). Functional Testing: A Complete Guide with Types and Example. [online] Available at: https://www.softwaretestinghelp.com/guide-to-functional-testing/ [Accessed 18 Aug. 2019].

Spyzie.com. (2019). 20 Best Apps for Parents of Teenagers: Monitor Teen Activities. [online] Available at: https://www.spyzie.com/parental-controls/best-apps-for-parents.html [Accessed 10 Aug. 2019].

The Importance of the Literature Search. (2008). Gastroenterology Nursing, 31(2), pp.151-153.

ToolsHero. (2019). MoSCoW Method: for setting Requirements by order of Priority | ToolsHero. [online] Available at: https://www.toolshero.com/project-management/moscow-method/ [Accessed 10 Aug. 2019].