

# *TapatApp*

*Final Project: TapatApp  
Institut Provençana - Course 2018/2019  
Development of multiplatform applications*

**Students:**

*Araceli Díaz Muñoz  
Laia S. Quichua Cisneros  
A. Marianna Torrealba D' Aquaro*

# INDEX

Title and purpose	4
User Story	5
Functional and Nonfunctional Requirements:	5
Used Programming Technologies:	11
Mockups	14
Use Case diagram and their textual description	15
Entity-Relation diagram and Relational model of Database	25
Description of the project methodology used	26
Project planning	27
Class Diagram	30
Test Cases	31

## Title and purpose

**Title :** TapatApp

**Purpose :**

### The Problem:

Around the world there's more than 200000 kids that will suffer or experience a visual disease calls Congenital Cataract, that is the most frequent cause of tractable blindness in childhood. The **congenital cataract** is a lens opacity present at birth and doesn't let the child see. One of the causes can be by genetic predisposition, metabolic disorders, or trauma to the eye that damage the lens. Occasionally, they occur spontaneously.

It's very important to detect this disease in the child's first days of life so can be operated and treated as soon as possible, if is not corrected in time it may cause long-term vision problems or even a permanent loss of vision on the child. Once operated, a strict rehabilitation guideline has to be followed to avoid amblyopia, which is commonly known as "lazy eye".

Apart from the other treatments they need, one of the most effective treatments is the application of an patch on the dominant eye, forcing the operated eye to develop as much as possible. However, it has to help the lazy eye to develop without harming the dominant eye, since it also has to learn to see in the plastic stage of the brain, which ends between the ages of 6 to 8 or 10.

One of the complications of the eye patch treatment is that depends on the age of the child. If it's on the first months of life the treatment begins by applying the patch half as long as the child is awake in the first months of life, and continues with a fixed daily time established by the ophthalmologist. Another complication is that the child does not sleep on a fixed schedule and there are no fixed hours of how long the patch should be worn, so we will assume the probability of the hours it will take, according to the hours that the child is usually awake, as it will vary and will not always be the same, as an example we can put a case of a family with your child who suffers from this disease, taking him to the ophthalmologist this prescribes that the patch should be placed for 6 hours in the day; It is here where the problem is because if the child takes a nap then the treatment of 6 hours in the day will stop and until he does not wake up will not continue, then the patch will no longer be placed 6 hours but less and would no longer be complying with the treatment. And as a last problem found is that the primary caregiver cannot take care of the child at the moment and has to take care of another person, and that the latter doesn't know how to carry out the treatment of the child (forget how long the child has worn the

patch, forget if the same has fallen asleep, among other things), and because of this delays or worsens the healing of the child's eye .

### **The Solution:**

At this time there is no company or individual that has created an application or product that can satisfy these exposed problems. That is why it has been proposed to create a mobile application that can be used to facilitate the time that the child must wear the patch, taking into consideration the times that may be asleep and that the patch can be removed during the day. And that mobile app will be called TapatApp.

TapatApp will keep a record of the hours the child has been asleep and how many hours the child has been using the patch. Then, based on the treatment they must follow (whether the child must use the patch for X hours or percentage per day), when the child must stop using the patch will be calculated, and the user will be notified with an alarm.

A user can share totally or partially the access to their child profiles with another user. This is useful, for example, when parents leave their child at school. They can share access to the profile with the teacher (who must also install the application and register) so that the teacher can monitor the child's treatment (awake/sleeping, using eye patch/no patch) and also make use of the alarm.

### **User Story**

this is what the client/user wants in the application to be created:

“I want to be able to control how long the child will be sleeping in order to know or predict how long the patch should be worn, since in the first few months of life the child should wear the patch half of the time he/she is awake. In addition, I want to know when to remove the patch from the moment I put it on. Finally, when the child stays with people other than the parents, I want the application to inform me of how long the patch has been worn, when the patch has been removed, when and how much sleep you have had, and when you have woken up, so that I can more effectively carry out the treatment and see that the treatment has an effect.”

## Functional and Nonfunctional Requirements:

Priority level				
1	2	3	4	5

### Functional Requirements:

<b>Requirement Code</b>	R - 1.1	<b>Requirement Type</b>	Functional
<b>Description</b>	<u>REGISTRATION:</u> An user that is not registered can create a profile so can maintain the session i can access to the functionalities of the application.		
<b>Priority</b>	5		

<b>Requirement Code</b>	R-1.2	<b>Requirement Type</b>	Functional
<b>Description</b>	<u>AUTHENTICATION:</u> An user that is already registered can login with its username and password, and access to the functionalities of the app, so can obtain created data and/or create and share new ones.		
<b>Priority</b>	5		

<b>Requirement Code</b>	R-1.3	<b>Requirement Type</b>	Functional
<b>Description</b>	<u>LOG OUT:</u> When the user is already logged in the application the user can log out from the app.		
<b>Priority</b>	4		

<b>Requirement Code</b>	R-1.4	<b>Requirement Type</b>	Functional
<b>Description</b>	<u>GET REFERENCE OF CHILD PROFILE CREATED BY ANOTHER USER:</u> An user that is already registered and logged in can share a child profile already created with another existent user.		
<b>Priority</b>	5		

<b>Requirement Code</b>	R-1.5	<b>Requirement Type</b>	Functional
<b>Description</b>	<p style="text-align: center;"><b>MAIN EMPTY PAGE:</b></p> <p>An user that is already been registered, logged in and doesn't have any associated child profile, the app will display this page so that the user can create a child profile when the latter seems fit.</p>		
<b>Priority</b>	<b>2</b>		

<b>Requirement Code</b>	R-2.1	<b>Requirement Type</b>	Functional
<b>Description</b>	<p style="text-align: center;"><b>MAIN LIST OF CHILD PAGE:</b></p> <p>Everytime that the user logged into the app and has more than 1 child profile associated, it'll show a page with a whole list of this child's profiles.</p>		
<b>Priority</b>	<b>3</b>		

<b>Requirement Code</b>	R-2.2	<b>Requirement Type</b>	Functional
<b>Description</b>	<p style="text-align: center;"><b>DISPLAY CHILD PROFILE:</b></p> <p>An user that has logged into the app, will show the main page of the child profile directly. This will only happen if the user has only 1 child profile associated or touching 1 profile on the list of child's profile.</p>		
<b>Priority</b>	<b>5</b>		

<b>Requirement Code</b>	R-2.3	<b>Requirement Type</b>	Functional
<b>Description</b>	<p style="text-align: center;"><b>CREATE CHILD PROFILE:</b></p> <p>When the user logged is creating the profile and adding the data, it will be necessary to enter the hour or the percentage of day that the child needs to wear the eyepatch. And also the amount of hours the child usually sleeps during the day (to have an average from the start for the calculations).</p>		
<b>Priority</b>	<b>5</b>		

<b>Requirement Code</b>	R-2.4	<b>Requirement Type</b>	Functional
<b>Description</b>	<p style="text-align: center;"><b>MODIFY CHILD PROFILE:</b></p> <p>The logged in user that has a child profile associated can modify its data.</p>		
<b>Priority</b>	<b>4</b>		

<b>Requirement Code</b>	R-2.5	<b>Requirement Type</b>	Functional
<b>Description</b>	<p align="center"><u>DELETE CHILD PROFILE:</u>  The logged user that has a child profile associated can delete its profile. the relation that has that child profile with other users will be deleted as well.</p>		
<b>Priority</b>	2		

<b>Requirement Code</b>	R-3.1	<b>Requirement Type</b>	Functional
<b>Description</b>	<p align="center">CREATION OF 'TAPS':  The 'taps' are the child treatment events.  Each time the user clicks on the buttons, creates a 'tap' with actual data.</p>		
<b>Priority</b>	5		

<b>Requirement Code</b>	R-3.2	<b>Requirement Type</b>	Functional
<b>Description</b>	<p align="center">DISPLAY LIST OF 'TAPS':  The user can see the list of all the 'taps' created, also can choose between the "awake/sleep" and 'wearing patch / not wearing patch' 'taps'.</p>		
<b>Priority</b>	4		

<b>Requirement Code</b>	R-3.4	<b>Requirement Type</b>	Functional
<b>Description</b>	<p align="center">ALARM CALCULATION:  The application must keep track by calculating when the eyepatch should be removed because the processing time of the current day has ended. If the processing is percentage-based, the application must perform calculations to update the processing time to make it more accurate.</p>		
<b>Priority</b>	5		

<b>Requirement Code</b>	R-3.5	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>MONITORIZE WHEN CHILD IS WEARING / NOT WEARING EYE PATCH:</p> <p>When the user reports that the child is wearing the eye patch, the constant calculations begin to check, whether he has reached the treatment time he should have reached on the current day. It will stop the remaining time if the user informs that the child stops wearing the patch and will reactivate it when it is put back on.</p>		
<b>Priority</b>	<b>5</b>		

<b>Requirement Code</b>	R-3.6	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>MONITORIZE WHEN CHILD IS AWAKE / ASLEEP:</p> <p>When the user reports that the child is sleeping and was wearing an eye patch, the variable that controls the application of the treatment will be automatically stopped, it will be reactivated when the user indicates, first that he has woken up and second when he puts the patch back on.</p> <p>The user provides a record of the start and end of the 'tap' in order to be able to calculate with greater precision the time of application of the current day's treatment.</p>		
<b>Priority</b>	<b>5</b>		

<b>Requirement Code</b>	R-3.7	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>MODIFY 'TAPS':</p> <p>The logged user that has relation to the child profile and has 'taps' created can modify its data</p>		
<b>Priority</b>	<b>5</b>		

<b>Requirement Code</b>	R-3.8	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>DELETE 'TAPS':</p> <p>The logged user that has relation to the child profile and has 'taps' created can delete it.</p>		
<b>Priority</b>	<b>5</b>		



<b>Requirement Code</b>	R-4.1	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>DISPLAY USER'S PROFILE: The user can access to its profile and see their data and do some change.</p>		
<b>Priority</b>	2		

<b>Requirement Code</b>	R-4.2	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>MODIFY THE USER'S PROFILE: When the user is in their profile it can change the email and/or password and save it into the database.</p>		
<b>Priority</b>	4		

<b>Requirement Code</b>	R-4.3	<b>Requirement Type</b>	Functional
<b>Description</b>	<p>DELETE THE USER'S PROFILE: The user can delete it's own profile. If the user had created it's own child profile this profile gets deleted to..</p>		
<b>Priority</b>	4		

## Nonfunctional Requirements:

<b>Requirement Code</b>	R- 5.1	<b>Requirement Type</b>	Nonfunctional
<b>Description</b>	<p style="text-align: center;"><b>USER ROLES:</b></p> <p>The user can access to delete &amp; modify the children profiles that were created from that account (admin permission). The user can also partially access the children profiles that were referenced to that account via external link (caretaker permission).</p>		
<b>Priority</b>	<b>4</b>		

<b>Requirement Code</b>	R-3.3	<b>Requirement Type</b>	Functional
<b>Description</b>	<p style="text-align: center;"><b>ALARM NOTIFICATION:</b></p> <p>The logged user that has launch the application foreground or has leave it in the background the user will receive a notification/alarm when the eyepatch has to be removed, i.e. when the treatment time ha finished that day.</p>		
<b>Priority</b>	<b>3</b>		

<b>Requirement Code</b>	R- 5.2	<b>Requirement Type</b>	Nonfunctional
<b>Description</b>	<p style="text-align: center;"><b>OFFLINE MODE:</b></p> <p>Each time the user changes the child's profile, if the phone is offline the change is saved in an internal file (along with the date and time when it was changed) so that it can be sent to the database once the connection is back.</p>		
<b>Priority</b>	<b>3</b>		

<b>Requirement Code</b>	R- 5.3	<b>Requirement Type</b>	Nonfunctional
<b>Description</b>	<p style="text-align: center;"><b>THE "ABOUT" BUTTON:</b> The system will have a manual or a short information about the application and how it works</p>		
<b>Priority</b>	<b>1</b>		

<b>Requirement Code</b>	R- 5.4	<b>Requirement Type</b>	Nonfunctional
<b>Description</b>	<p style="text-align: center;">REDIRECTION AFTER LOGIN:</p> <p>If the user is waiting for the child's profile to be shared, he will be notified when the administrator User of that profile has shared it, after the notification he will already be able to view the child's profile, with the caretaker user permissions.</p>		
<b>Priority</b>	1		

<b>Requirement Code</b>	R- 5.5	<b>Requirement Type</b>	Nonfunctional
<b>Description</b>	<p style="text-align: center;">"CARETAKER" ROLE SPECIFICATION:</p> <p>A user who is related to the child's profile as a caregiver can view but not modify the profile of that shared child, but can view, modify and delete 'taps'.</p>		
<b>Priority</b>	4		

### **Used Programming Technologies:**

The group that will create the application, after having observed the advantages and disadvantages of some technologies that are suitable for their creation, has come to the conclusion of using these technologies below:

Java: Apart from the fact that the group has good knowledge of this technology, it is independent of the platform (multiplatform), there are no problems with memory release in the system and it is also the same language with which it is used to create both the server and the client app and is the most used programming language around the world.

MySQL: Is one of the most used and recognize database manager, allows multiple storage engines to be selected for each table, available on most platforms or systems, and provides a system of secure host based verification passwords and privileges and encrypted password traffic when connecting to a server.

Android Studio: Apart from that it is currently one of the most used operating systems in many mobile devices and one of the most affordable and intuitive for the

end user, is suitable for different devices using the said operating system (Android Watch, Tablets, ...), allows the import of projects made in Netbeans or Eclipse, does frequent updates and use Java as a programming language.

### **Problems and Solutions found:**

**Database:** In terms of the connection with the database, there's a lot of options to do the connections but the most mentioned were JDBC (more know as 'Java Database Connection', the standard connection from Java to MySQL) and JPA (Java Persistence Architecture). After debating what to use, the group has chosen the standard connection to the database for the next reasons: JDBC directly connects with the database, send queries as normally would do when making a query directly to the database. A very important thing in particular with the JDBC connection is that the programmer is who creates the tables, the programmer is very much more in contact with the database, that means, modify the tables of the database directly. For the contrary, the JPA creates the tables based on the classes created into the project, which in some way is very useful and optimize the code, but maintains the SQL "hide" from the programmer and can not interact directly.

**Server:** When it comes to the server, there were several ways on doing it but the group chose working with servlets due for this reasons: The servlet is a Java specific and also an API, the authentication is through username and password and works with cookies, which they intent to maintain a new session active. Each time a request is generated the cookie will store the information and maintain the data in session until a established time pass or the session is out manually.

Another

**Client:** Deciding for how to do it was a little bit difficult due to there's a lot of ways and libraries that could be very used in the creation of the client. But, in the end, the Volley library was the chose option.

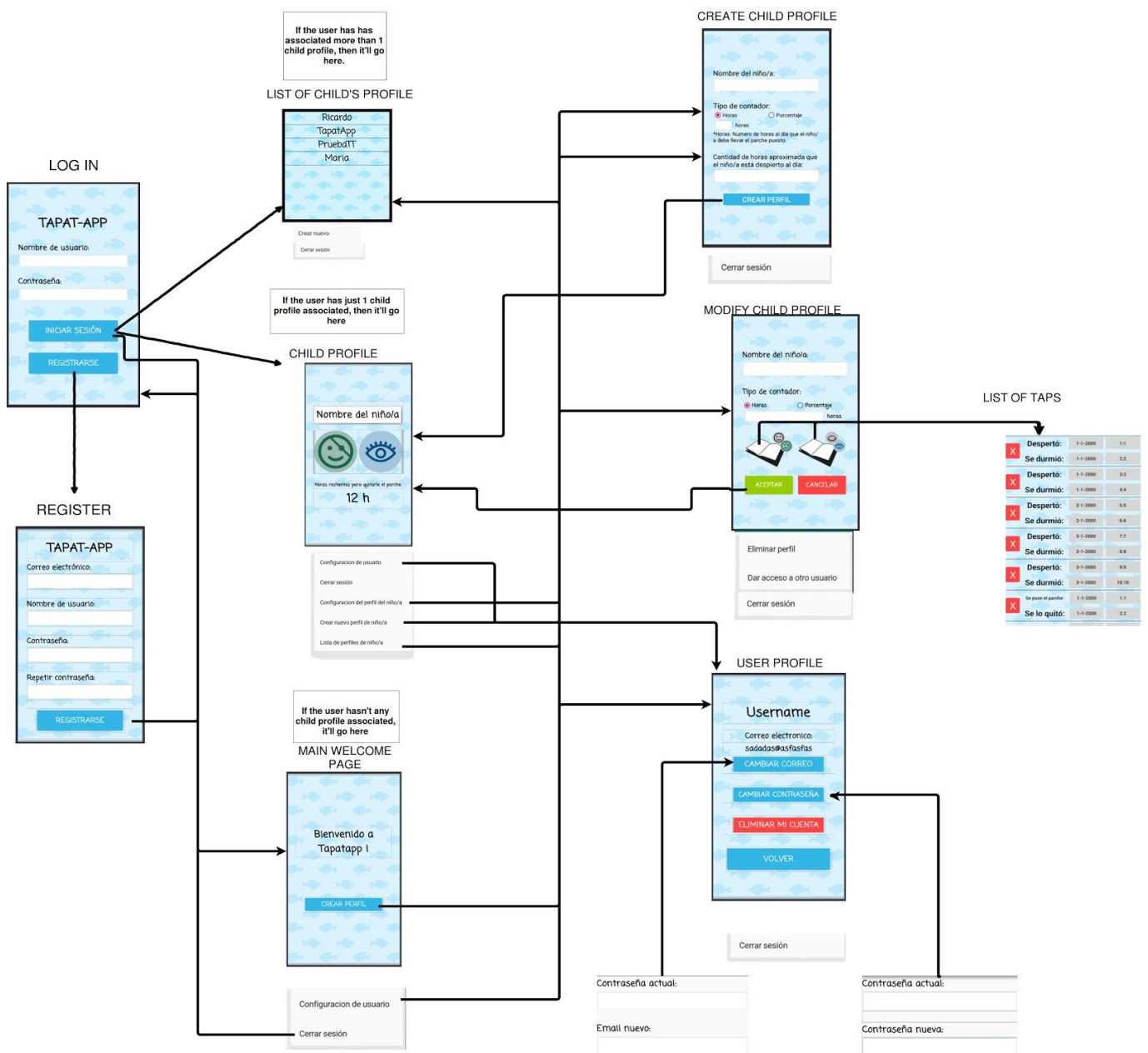
Volley is an android library created by Google itself and its purpose is to make request and take response from an easy and fast way. A very important thing to remark is Volley replace the way some other libraries, like the AsyncTask, execute the request store them into a RequestQueue. The AsyncTask doesn't control which request has to be executed first and the others to wait. Another important detail is that the library HttpURLConnection and HttpClient ( that is from Apache) has some errors and some problems and some of them are deprecated, the Volley library will replace this two libraries as time goes by.

Another issue is that the initial project has, as a minimum API the 16 but, because it's not 100% compatible with a lot of mobiles, and the API was changed. But the new minimum API, which is the 14, doesn't allow using the LocalDateTime class. The solution was the next one: Use the Calendar class instead. This class is quite

heavy to use (it uses many intermediate methods and global static variables for anything), so, to make things easier: Create an intermediate class (MyTimeStamp) that uses Calendar and has only the methods that are really needed.

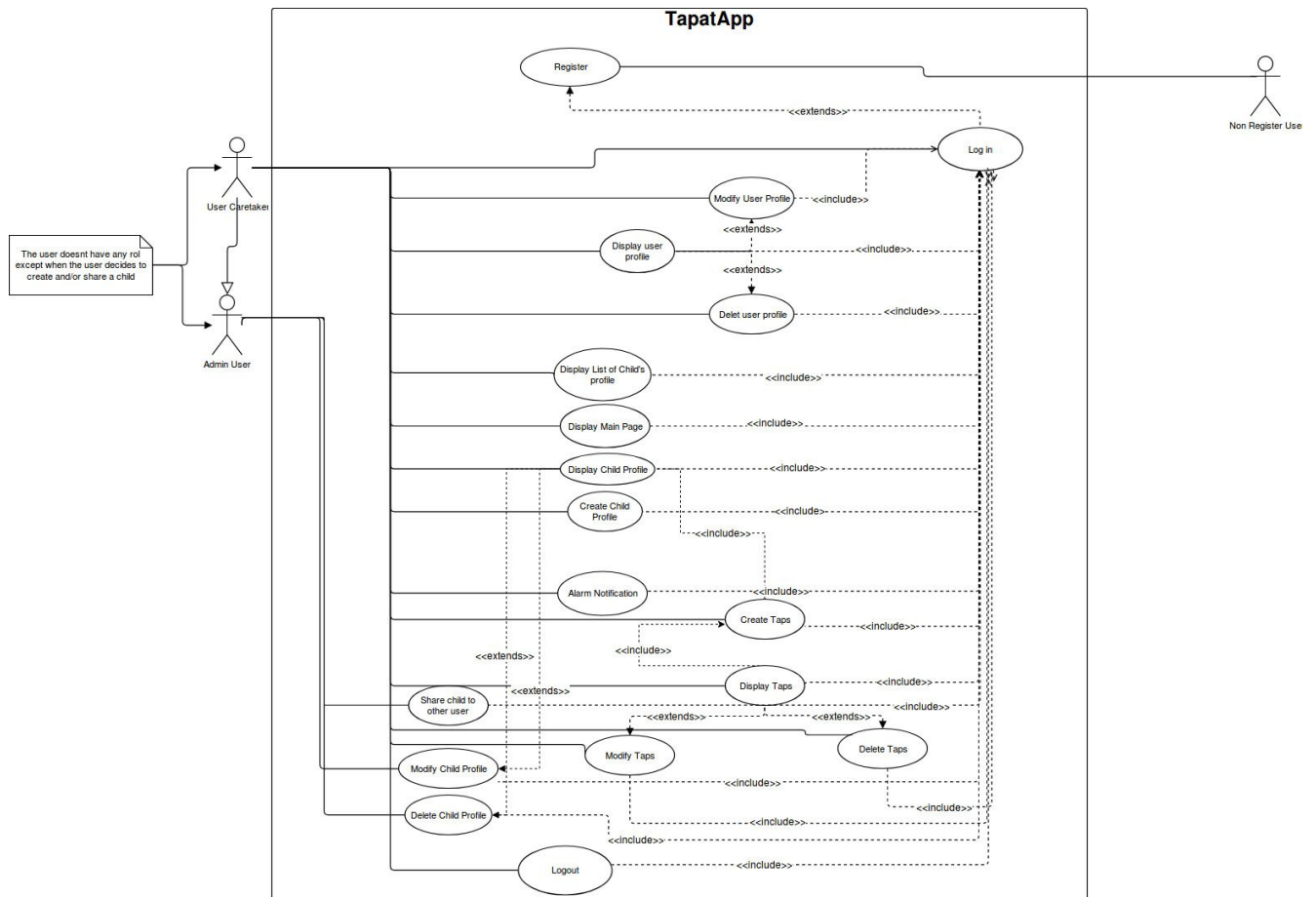
**Views:** Depending on which mobile the screen is very small, and doesn't allow to see all the options on the phone. The solution was to include a Scroll to all the layouts, so can scroll vertically through the screen.

## Mockups



## Use Case diagram and their textual description

Use Case diagram :



'Taps': Events saved by the application. Example: "Sleeping".

Actors	Description
Non Registered User	Actor who doesn't have a profile created on the system.
Admin User	User creator of the child's profile who has all the permissions to modify and delete the profile; adds, modifies and/or deletes 'taps' of the child.
User Caretaker	User who has limited permissions, visualizes the child's profile but cannot modify it; adds, modifies and removes 'taps' of the child.

Use Case textual description :

<b>Name:</b>	<b>Registering user</b>
<b>Priority:</b>	<b>5</b>
<b>Description:</b> <b>Let the actor Create a new user.</b>	
<b>Actors:</b> <b>Non Registered User</b>	
<b>Normal Flow:</b> <b>1.- The actor click on the button to register.</b> <b>2.- The application shows a form to enter the new user's data.</b> <b>3.- The actor enters the new user data.</b> <b>4.- The actor clicks on the [Register] button.</b> <b>5.- The system checks if the entered information is valid and stores them.</b>	
<b>Alternative Flow:</b> <b>5.A.- If the information are not valid, the system will notify the actor.</b> <b>5.B.- If the actor leaves the form empty, the system will notify the actor.</b>	

<b>Name:</b>	<b>Log in user</b>
<b>Prioridad:</b>	<b>5</b>
<b>Descripción:</b> <b>Let the actor login to the system as a registered user.</b>	
<b>Actores:</b> <b>User Caretaker or Admin User.</b>	
<b>Flujo Normal:</b> <b>1.- The app shows a form to enter the user information already created.</b> <b>2.- The actor enters the user information (Username and Password).</b> <b>3.- The actor clicks the [Login] button.</b> <b>4.- The app checks if the entered data exist and is valid.</b> <b>5.- The app let the user to logged in.</b>	
<b>Flujo Alternativo:</b> <b>4.A.- If the user information is not valid, the system will notify the actor.</b> <b>4.B.- If the user information doesn't exist, the system will notify the actor.</b>	

<b>Name:</b>	<b>Logout</b>
<b>Priority:</b>	<b>4</b>
<b>Description:</b> <b>The system allows the actor to logout.</b>	
<b>Actors:</b> <b>Caretaker User and Admin User.</b>	
<b>Normal Flow:</b> <b>1.- The actor clicks on the [Logout] button.</b> <b>2.- The system search for the session and close it.</b>	
<b>Alternative Flow:</b> <b>-</b>	

<b>Name:</b>	<b>Display user Profile</b>
<b>Priority:</b>	<b>4</b>
<b>Description:</b> <b>The system allows the actor to see the user profile</b>	
<b>Actors:</b> <b>Caretaker User and Admin User.</b>	
<b>Normal Flow:</b> <b>1.- The user clicks on the [Show User Profile]</b> <b>2.- The system will show the user page.</b>	
<b>Alternative Flow:</b> <b>-</b>	



<b>Name:</b>	<b>Modify User Profile</b>
<b>Priority:</b>	<b>4</b>
<b>Description:</b> <b>The system allows the actor to modify the information of the its user profile.</b>	
<b>Actors:</b> <b>Caretaker User or Admin User.</b>	
<b>Normal Flow:</b> <b>1.- The actor click on the data it want to change.</b> <b>2.- The system will show a emergent message to enter the actual password and the new data to change</b> <b>3.- The system validates the password and saves the new data.</b> <b>4.- The system updates the user profile automatically.</b>	
<b>Alternative Flow:</b> <b>3.A.- If the actor enters an incorrect actual password, the system will notify the actor.</b> <b>3.B.- If the actor doesn't enter any data, the system will notify the actor.</b>	

<b>Name:</b>	<b>Delete User Profile.</b>
<b>Priority:</b>	<b>4</b>
<b>Description:</b> <b>The system allows the actor to delete it's own user profile.</b>	
<b>Actors:</b> <b>Caretaker User or Admin User.</b>	
<b>Normal Flow:</b> <b>1.- The actor clicks on the [Delet Account] button.</b> <b>2.- The system will show an emergent message.</b> <b>3.- The actor clicks on [Yes].</b> <b>4.- The system checks if the user has child profiles associated and deletes the user and the child profiles.</b>	
<b>Alternative Flow:</b> <b>3.A.- If the actor clicks on the [No] button, the system will cancel the action and shows the user profile with its data.</b>	

4.A.- if the user of actor doesn't have any child profile associated, the system will only delete the user profile.

<b>Name:</b>	Display Main Page
<b>Priority:</b>	2
<b>Description:</b>	If the actor doesn't have any child profile associated, the system shows a main page.
<b>Actors:</b>	Caretaker User and Admin User.
<b>Normal Flow:</b>	<ol style="list-style-type: none"><li>1.- The actor logged into the system</li><li>2.- The system will check if the user has child profiles associated.</li><li>3.- If doesn't have any child profile associated, the system will shows the Main page so the actor can create a child profile or wait for other user to shared one.</li></ol>
<b>Alternative Flow:</b>	-

<b>Name:</b>	Display List of Child Profiles
<b>Priority:</b>	5
<b>Description:</b>	If the user have more than 1 child profile associated, the system shows a list of all child profiles that the user has associated.
<b>Actors:</b>	Caretaker User or Admin User.
<b>Normal Flow:</b>	<ol style="list-style-type: none"><li>1.- The actor logged into the system</li><li>2.- The system will check if the user has child profiles associated.</li><li>3.- If the user of actor has more than 1 child profile associated, the system will show a list of t;all the child profiles associated.</li></ol>
<b>Alternative Flow:</b>	-

<b>Name:</b>	<b>Display Child Profile</b>
<b>Priority:</b>	<b>5</b>
<b>Description:</b>	
The system will show the main page of the child profile when the actor has 1 child profile associated.	
<b>Actors:</b>	
Caretaker User and Admin User.	
<b>Normal Flow:</b>	
<ol style="list-style-type: none"> <li>1.- The actor logged into the system</li> <li>2.- The system will check if the user has child profiles associated.</li> <li>3.- If the user of actor has 1 child profile associated, the system will show directly the child profile.</li> </ol>	
<b>Alternative Flow:</b>	
-	

<b>Name:</b>	<b>Alarm Notification</b>
<b>Priority:</b>	<b>3</b>
<b>Description:</b>	
Each time the actor clicks to create the “wearing eye patch / not wearing eye patch” tap, the system will notify it to the actor when to take it of.	
<b>Actors:</b>	
Caretaker User or Admin User.	
<b>Normal Flow:</b>	
<ol style="list-style-type: none"> <li>1.- The actor create a ‘tap’.</li> <li>2.- The system will calculate and will be doing the countdown of the hours left.</li> <li>3.- When the system has reached the time will notify the actor.</li> <li>4.- The actor stops the alarm.</li> </ol>	
<b>Alternative Flow:</b>	
-	

<b>Name:</b>	<b>Creating Child Profile</b>
<b>Priority:</b>	<b>5</b>
<b>Description:</b> <b>Allow the actor to create a new Child profile.</b>	
<b>Actors:</b> <b>User Caretaker or Admin User.</b>	
<b>Normal Flow:</b> <ol style="list-style-type: none"> <li>1.- The actor clicks the [Create new Profile] to create show a form.</li> <li>2.- The system shows a form to enters the child's data.</li> <li>3.- The actor enters the child's data.</li> <li>4.- The actor clicks on the [Create Profile].</li> <li>5.- The system checks the data is valid and stores them.</li> </ol>	
<b>Alternative Flow:</b> <ol style="list-style-type: none"> <li>5.A.- If the information is not valid, the system will notify the actor.</li> <li>5.B.- If the actor leaves the form empty, the system will notify the actor.</li> </ol>	

<b>Name:</b>	<b>Modifying Child Profile.</b>
<b>Priority:</b>	<b>4</b>
<b>Description:</b> <b>Allows the actor to modify the child profile information.</b>	
<b>Actors:</b> <b>Admin User</b>	
<b>Normal Flow:</b> <ol style="list-style-type: none"> <li>1.- The actor click on the name of the child profile.</li> <li>2.- The system shows a form to enters the profile data.</li> <li>3.- The actor enters the child profile data.</li> <li>4.- The actor clicks on the [Accept] button.</li> <li>5.- The system checks if the information is valid and stores them.</li> </ol>	
<b>Alternative Flow:</b> <ol style="list-style-type: none"> <li>5.A.- If the information is not valid, the system will notify the actor.</li> <li>5.B.- If the actor leaves the form empty, the system will notify the actor.</li> </ol>	

<b>Name:</b>	Deleting child profile.
<b>Priority:</b>	2
<b>Description:</b>	Allows the actor to delete the child profile.
<b>Actors:</b>	Admin User
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1.- The actor clicks on the child's name.</li> <li>2.- The system shows the child profile.</li> <li>3.- The actor clicks on the [Delete] button.</li> <li>4.- The system show a warning message.</li> <li>5.- The user select the [Yes] button and deletes the child profile.</li> </ol>
<b>Alternative Flow:</b>	5.A.- The actor clicks on the [No] button, returns to the child profile information.

<b>Name:</b>	Share Child Profile.
<b>Priority:</b>	5
<b>Description:</b>	Allows the actor to share a child profile to other actor.
<b>Actors:</b>	Admin User..
<b>Normal Flow:</b>	<ol style="list-style-type: none"> <li>1.- The actor clicks on the [Share Child Profile] button.</li> <li>2.- The system will show an emergent message to enter the username of other actor.</li> <li>3.- The actor enters the username of another actor and clicks on the [Share] button.</li> <li>4.- The system will validate the user and create the association.</li> </ol>
<b>Alternative Flow:</b>	3.A.- If the actor enters a non existent user, the system wouldn't create the association and will notify the actor.

<b>Name:</b>	Display List of 'Taps'
<b>Priority:</b>	4
<b>Description:</b> The system allows the actor to see all the 'taps' that the user has already created from a child profile associated.	
<b>Actors:</b> User Caretaker or Admin User.	
<b>Normal Flow:</b> 1.- The actor clicks on one of the books of 'taps'. 2.- The system shows a list of the 'taps'.	
<b>Alternative Flow:</b> 2.A.- If there isn't any 'taps' created, the system will notify the actor.	

<b>Name:</b>	Creating 'Taps'
<b>Priority:</b>	5
<b>Description:</b> The system allows to the actor create 'taps' when clicked.	
<b>Actors:</b> SuperAdmin y Cuidador.	
<b>Normal Flow:</b> 1.- The actor clicked on one of the 'taps' on the child profile. 2.- The system creates a 'tap' and stores it.	
<b>Alternative Flow:</b> 1.A.- The actor clicks on both 'taps', the system wouldn't allow create it 1.B.- The actor clicks first on the "wearing eyepatch / not wearing eyepatch" tap before clicking on the "awake/sleep" tap, the system wouldn't allow create it.	

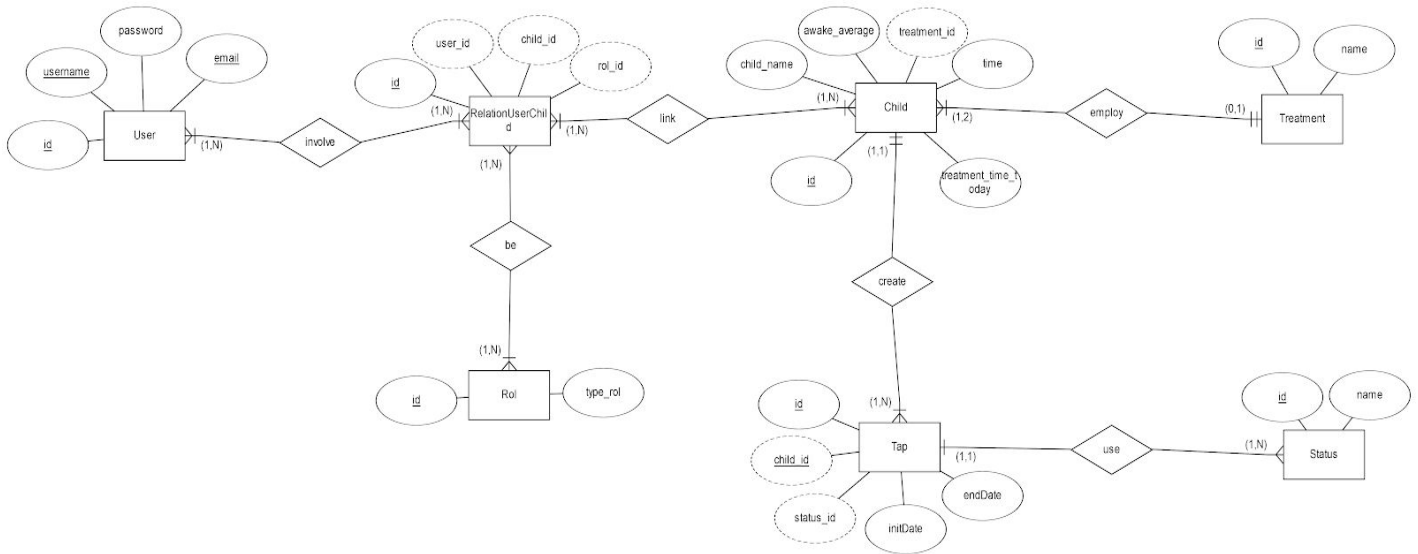
<b>Name:</b>	Modify Child Profile
<b>Priority:</b>	5

<b>Description:</b> The system allows the actor to modify the child profile information.
<b>Actors:</b> Admin User.
<b>Normal Flow:</b> 1.- The actor clicks on the name of the child. 2.- The system shows a form to be entered by the actor. 3.- The actor enters the new data and clicks on the [Accept] button. 4.- The system verifies if the information entered is valid and stores it.
<b>Alternative Flow:</b> 4.A.- If the actor enters an invalid information, the system will notify the actor. 4.B.- If the actor left the form empty, the system will notify the actor.

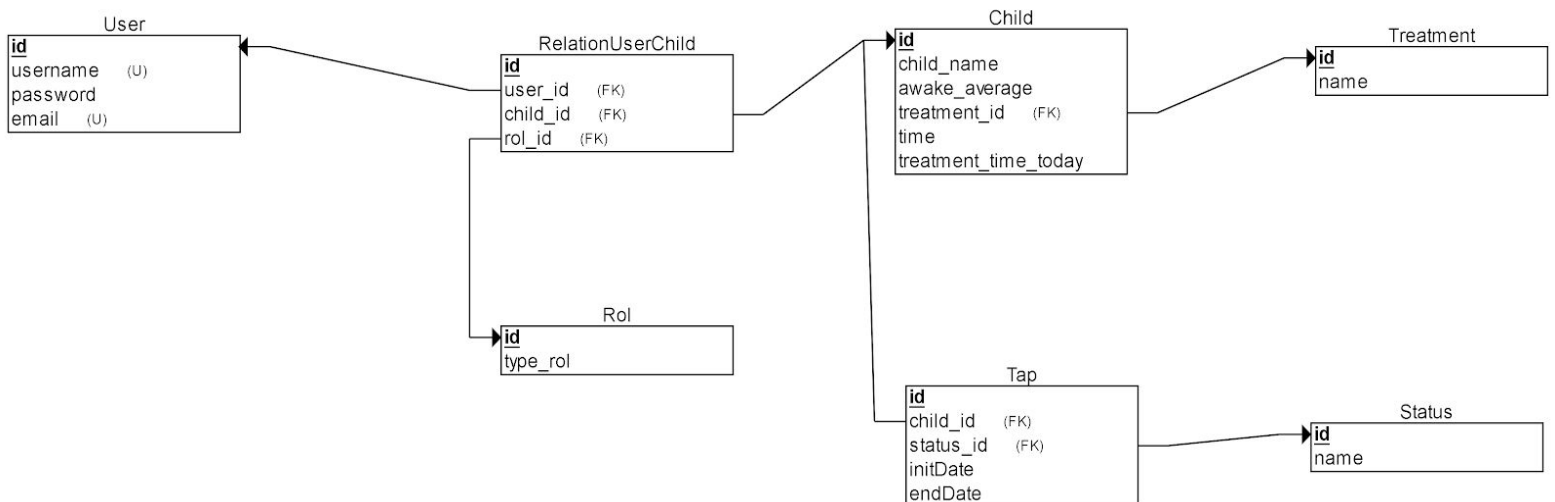
<b>Name:</b>	Delete Child Profile.
<b>Priority:</b>	5
<b>Description:</b> The system allows the actor to delete the child profile, even if the child profile is associated with other users.	
<b>Actors:</b> Admin User.	
<b>Normal Flow:</b> 1.- The actor clicks on the [Delete] button. 2.- The system will show a warning message. 3.- The actor click on the [Yes] button. 4.- The system will check the association between that child profile and the users and finally deletes it.	
<b>Alternative Flow:</b> 3.A.- The actor clicks on the [No] button, the system will cancel the action and shows the child profile information.	

## Entity-Relation diagram and Relational model of Database

### Relational Model



### Entity-Relation Model:

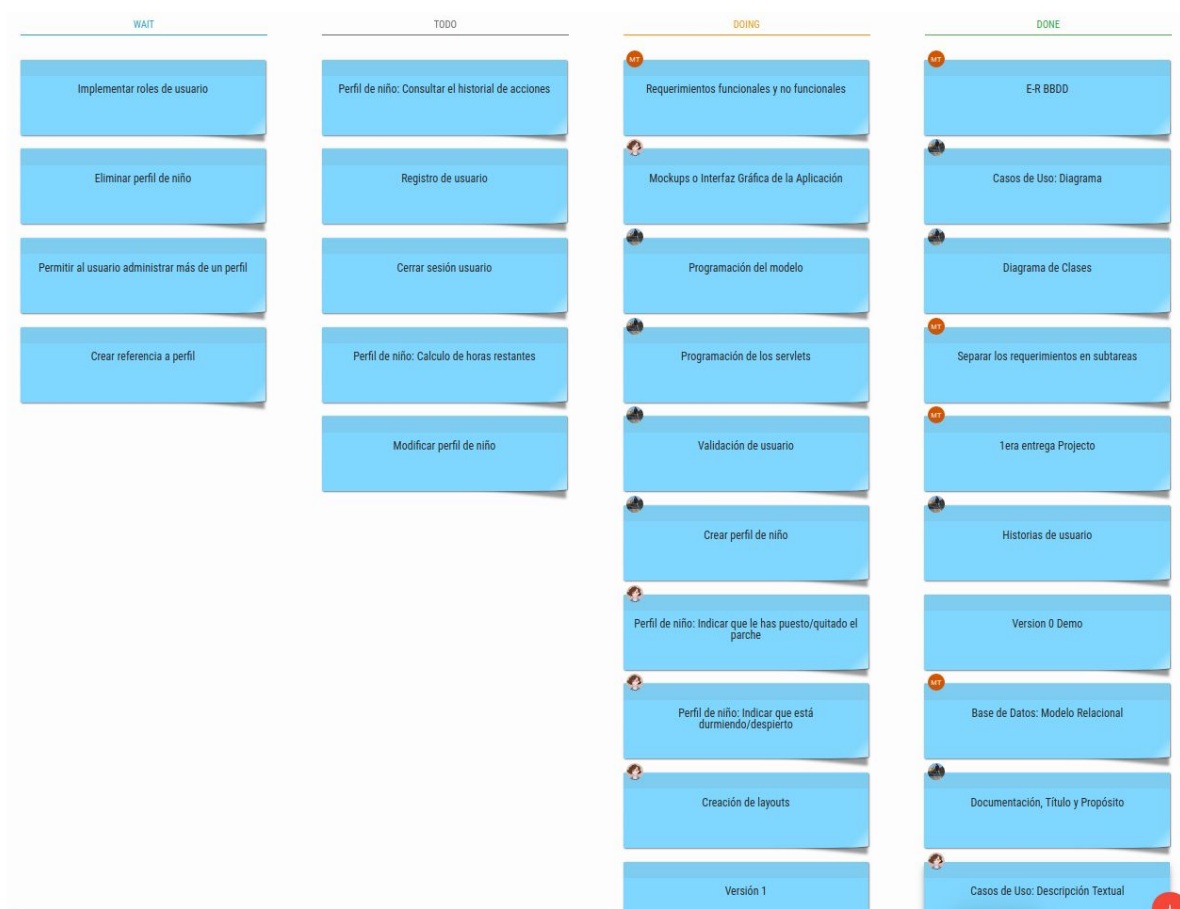




## Description of the project methodology used

The ScrumBan methodology consists in dividing the processes for the software development in stages. The first step to use this methodology is to identify these processes and organize them onto a workflow, which will be represented in the Kanban board. After that, the members of the project monitor the process continuously as to make sure that everybody is busy.

The workgroup used this method and also met once a week to update the kanban, take advice from the tutor and from the final client, making sure each reunion the client is satisfied and if is not what the client wants, some things of the project has to be modified and/or deleted. Every two weeks a demo is delivered and presented.



## Project planning

Dates:

### DEV 1

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
<b>1/04</b> DEMO 1	<b>2/04</b> Nuevo E-R	<b>3/04</b> Reunión cliente	<b>4/04</b>	<b>5/04</b> Reunión Team
<b>8/04</b>	<b>9/04</b> Conn Servlet-App	<b>10/04</b> Conn Servlet-App	<b>11/04</b>	<b>12/04</b> Conn Servlet-App Reunión Team
<b>15/04</b> Conn Servlet-App	<b>16/04</b>	<b>17/04</b> Reunión cliente	<b>18/04</b>	<b>19/04</b> Conn Servlet-App Reunión Team
<b>22/04</b> ACTIVITIES CLIENT	<b>23/04</b> ACTIVITIES CLIENT	<b>24/04</b> ACTIVITIES CLIENT Reunión cliente	<b>25/04</b> ACTIVITIES CLIENT	<b>26/04</b> ACTIVITIES CLIENT Reunión Team
<b>29/04</b> ACTIVITIES CLIENT DEMO 3	<b>30/04</b> ACTIVITIES CLIENT	<b>1/05</b> ACTIVITIES CLIENT Reunión cliente	<b>2/05</b> ACTIVITIES CLIENT	<b>3/05</b> ACTIVITIES CLIENT Reunión Team
<b>6/05</b> ACTIVITIES CLIENT	<b>7/05</b> ACTIVITIES CLIENT	<b>8/05</b> ACTIVITIES CLIENT Reunión cliente	<b>9/05</b> ACTIVITIES CLIENT	<b>10/05</b> ACTIVITIES CLIENT. Reunión Team
<b>13/05</b> DEMO 4	<b>14/05</b> Arreglos	<b>15/05</b> Reunión cliente	<b>16/05</b> Arreglos	<b>17/05</b> Reunión Team
<b>20/05</b> Arreglos	<b>21/05</b> Arreglos	<b>22/05</b> Revisión	<b>23/05</b> Revisión Final	<b>24/05</b> Reunión

				Team
<b>27/05</b> Aplicación FINAL	<b>28/05</b> Documentación FINAL	<b>29/05</b> Reunión cliente: Entrega	<b>30/05</b> Vacaciones	

## DEV2

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
<b>1/04</b> DEMO 1	<b>2/04</b> Mod E-R	<b>3/04</b> Insert BBDD Reunión cliente	<b>4/04</b>	<b>5/04</b> Reunión Team
<b>8/04</b>	<b>9/04</b> conn BBDD - Servlet	<b>10/04</b> conn BBDD - Servlet Reunión cliente	<b>11/04</b>	<b>12/04</b> conn BBDD - Servlet Reunión Team
<b>15/04</b> conn BBDD - Servlet DEMO 2	<b>16/04</b>	<b>17/04</b> conn BBDD - Servlet Reunión cliente	<b>18/04</b>	<b>19/04</b> Reunión Team
<b>22/04</b> conn Client -Server	<b>23/04</b> conn Client -Server	<b>24/04</b> conn Client -Server Reunión cliente	<b>25/04</b> conn Client -Server	<b>26/04</b> conn Client -Server Reunión Team
<b>29/04</b> DEMO 3	<b>30/04</b> conn Client -Server	<b>1/05</b> Reunión cliente	<b>2/05</b> Servlet	<b>3/05</b> Reunión Team
<b>6/05</b> C Servlet-App	<b>7/05</b> C Servlet-App	<b>8/05</b> Reunión cliente	<b>9/05</b> Conn Servlet-App	<b>10/05</b> Reunión Team
<b>13/05</b> DEMO 4	<b>14/05</b> Arreglos	<b>15/05</b> Reunión cliente	<b>16/05</b> Arreglos	<b>17/05</b> Reunión Team
<b>20/05</b>	<b>21/05</b>	<b>22/05</b>	<b>23/05</b>	<b>24/05</b>

Arreglos	Arreglos	Revisión	Revisión Final	Reunión Team
<b>27/05</b> Aplicación FINAL	<b>28/05</b> Documentación FINAL	<b>29/05</b> Reunión cliente: Entrega	<b>30/05</b> Vacaciones	

### DEV 3

LUNES	MARTES	MIÉRCOLES	JUEVES	VIERNES
<b>1/04</b> DEMO 1	<b>2/04</b> Layouts-Views	<b>3/04</b> Activities Client Reunión cliente	<b>4/04</b>	<b>5/04</b> Activities Client Reunión Team
<b>8/04</b>	<b>9/04</b> Layouts-Views	<b>10/04</b> Activities Client	<b>11/04</b>	<b>12/04</b> Layout Test Reunión Team
<b>15/04</b> DEMO 2	<b>16/04</b>	<b>17/04</b> Layouts-Views Reunión cliente	<b>18/04</b>	<b>19/04</b> Activities Client Reunión Team
<b>22/04</b> Activities Client	<b>23/04</b> Layouts-Views	<b>24/04</b> Activities Client Reunión cliente	<b>25/04</b> Activities Client	<b>26/04</b> Layout Test Reunión Team
<b>29/04</b> DEMO 3	<b>30/04</b> Layouts-Views	<b>1/05</b> Activities Client Reunión cliente	<b>2/05</b> Activities Client	<b>3/05</b> Activities Client Reunión Team
<b>6/05</b> Activities Client	<b>7/05</b> Layouts-Views	<b>8/05</b> Activities Client Reunión cliente	<b>9/05</b> Activities Client	<b>10/05</b> Layout Test Reunión Team

<b>13/05</b> DEMO 4	<b>14/05</b> Arreglos	<b>15/05</b> Reunión cliente	<b>16/05</b> Arreglos	<b>17/05</b> Reunión Team
<b>20/05</b> Arreglos	<b>21/05</b> Arreglos	<b>22/05</b> Revisión	<b>23/05</b> Revisión Final	<b>24/05</b> Reunión Team
<b>27/05</b> Aplicación FINAL	<b>28/05</b> Documentación FINAL	<b>29/05</b> Reunión cliente: Entrega	<b>30/05</b> Vacaciones	

### Kanban Board and Tasks

TO DO	DOING	DONE
<ul style="list-style-type: none"> <li>Android - Funcionalidades               <ul style="list-style-type: none"> <li>- Historial</li> <li>* Crear</li> <li>* Modificar</li> <li>* Eliminar</li> <li>* Validación del Estado</li> </ul> </li> <li>Servlet - Funcionalidades               <ul style="list-style-type: none"> <li>- Usuario</li> <li>* Registro</li> <li>* Eliminar cuenta</li> <li>* Mostrar niñxs del usuario</li> <li>* Compartir niñx / ROL</li> <li>* Crear niñx</li> <li>* Modificar niñx (SA)</li> <li>* Eliminar niñx (SA)</li> </ul> </li> <li>Servlet - Funcionalidades               <ul style="list-style-type: none"> <li>- Relación Usuario - Niñx</li> <li>* Crear relación</li> <li>* Adjudicar Rol a la relación</li> <li>* Modificar la relación/rol</li> </ul> </li> <li>Conexión Servlet con Android Studio               <ul style="list-style-type: none"> <li>- Aprendizaje</li> <li>- Adaptar código (Muy difícil)</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>Android - Funcionalidades               <ul style="list-style-type: none"> <li>- Niñx</li> <li>* Dormidx / Despiertx</li> <li>* Parche / Sin Parche</li> <li>* Historial Estados</li> <li>* Historial Media (Estados) de la semana/día</li> </ul> </li> <li>- Android - Funcionalidades               <ul style="list-style-type: none"> <li>- Relación Usuario - Niñx</li> <li>* Crear relación</li> <li>* Adjudicar Rol a la relación</li> <li>* Modificar la relación/rol</li> </ul> </li> <li>Servlet - Funcionalidades               <ul style="list-style-type: none"> <li>- Filtro / Autenticación</li> <li>- Mantener Sesión / Inicio sesión</li> </ul> </li> <li>Documentación Final</li> </ul>	<ul style="list-style-type: none"> <li>Android - Funcionalidades               <ul style="list-style-type: none"> <li>- Usuario</li> <li>* Registro</li> <li>* Inicio sesión</li> <li>* Eliminar cuenta</li> <li>* Mostrar niñxs del usuario</li> <li>* Compartir niñx / ROL</li> <li>* Crear niñx</li> <li>* Modificar niñx (SA)</li> <li>* Eliminar niñx (SA)</li> </ul> </li> <li>Android               <ul style="list-style-type: none"> <li>- Layout(2)</li> <li>* Diseño</li> <li>* Funcionalidad Componentes</li> </ul> </li> <li>Base de datos               <ul style="list-style-type: none"> <li>- E-R versión Final</li> <li>- Creación Script</li> </ul> </li> <li>Servlet - Funcionalidades               <ul style="list-style-type: none"> <li>- Niñx</li> <li>* Dormidx / Despiertx</li> <li>* Parche / Sin Parche</li> <li>* Historial Estados</li> <li>* Historial Media (Estados) de la semana/día</li> </ul> </li> <li>Documentación - Delivery 2</li> </ul>

Servlet - Funcionalidades  
- Usuario  
\* Registro  
\* Eliminar cuenta  
\* Mostrar niñx/xs del usuario  
\* Compartir niñx / ROL  
\* Crear niñx  
\* Modificar niñx (SA)  
\* Eliminar niñx (SA)

Android - Funcionalidades  
- Usuario  
\* Registro  
\* Inicio sesión  
\* Eliminar cuenta  
\* Mostrar niñx/xs del usuario  
\* Compartir niñx / ROL  
\* Crear niñx  
\* Modificar niñx (SA)  
\* Eliminar niñx (SA)

- Android - Funcionalidades  
- Relación Usuario - Niñx  
\* Crear relación  
\* Adjudicar Rol a la relación  
\* Modificar la relación/rol

Conexión Servlet con Android Studio  
- Aprendizaje  
- Adaptar código (Muy difícil)

Android - Funcionalidades  
- Niñx  
\* Dormidx / Despiertx  
\* Parche / Sin Parche  
\* Historial Estados  
\* Historial Media (Estados)  
de la semana/día

Android - Funcionalidades  
- Historial  
\* Crear  
\* Modificar  
\* Eliminar  
\* Validación del Estado

Android  
- Layout(2)  
\* Diseño  
\* Funcionalidad  
Componentes

Documentación - Delivery 2

Documentación Final

Servlet - Funcionalidades  
- Niñx  
\* Dormidx / Despiertx  
\* Parche / Sin Parche  
\* Historial Estados  
\* Historial Media (Estados)  
de la semana/día

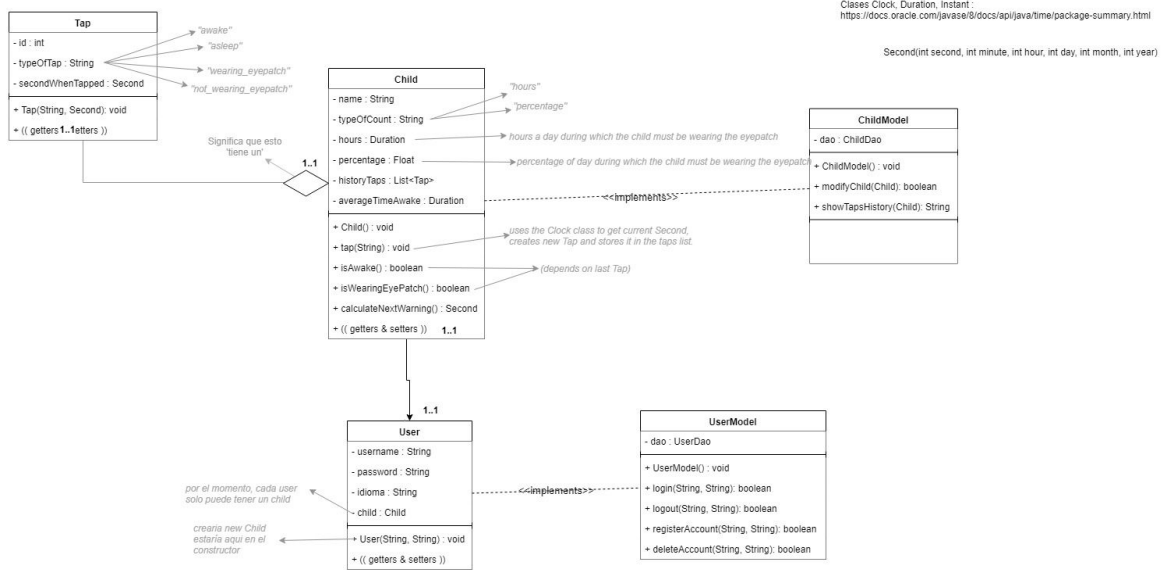
Servlet - Funcionalidades  
- Relación Usuario - Niñx  
\* Crear relación  
\* Adjudicar Rol a la relación  
\* Modificar la relación/rol

Servlet - Funcionalidades  
- Filtro / Autenticación  
- Mantener Sesión / Inicio sesión

Base de datos  
- E-R versión Final  
- Creación Script

TEMPLATE VERSIC	1.2	EN	<b>Tappat Proyecto</b>
PROJECT STATUS	inprogress	T0037	
PROJECT OWNER	marianna torrealba		
CREATED	29.01.2019		
<b>TASK OWNER</b>	<b>STATUS</b>	<b>ID</b>	<b>TASK NAME</b>
marianna torrealba	done	T0001	E-R BBDD
Laia Quichua	done	T0007	Casos de Uso: Diagrama
marianna torrealba	inprogress	T0008	Requerimientos funcionales y no funcionales
Laia Quichua	done	T0009	Diagrama de Clases
Araceli Diaz	inprogress	T0010	Mockups o Interfaz Gráfica de la Aplicación
Araceli Diaz	wait	T0011	
marianna torrealba	done	T0012	Separar los requerimientos en subareas
marianna torrealba	done	T0013	1era entrega Proyecto
Laia Quichua	done	T0014	Historias de usuario
Laia Quichua	inprogress	T0015	Programación del modelo
Laia Quichua	inprogress	T0016	Programación de los servlets
Laia Quichua	inprogress	T0017	Validación de usuario
Laia Quichua	inprogress	T0018	Crear perfil de niño
Araceli Diaz	inprogress	T0019	Perfil de niño: Indicar que le has puesto/quitado
Araceli Diaz	inprogress	T0020	Perfil de niño: Indicar que está durmiendo/despie
	planned	T0021	Perfil de niño: Consultar el historial de acciones
	wait	T0022	Implementar roles de usuario
	wait	T0023	Eliminar perfil de niño
	planned	T0024	Registro de usuario
	done	T0027	Version 0 Demo
Araceli Diaz	inprogress	T0028	Creación de layouts
	inprogress	T0029	Versión 1
	wait	T0028	Permitir al usuario administrar más de un perfil
marianna torrealba	done	T0031	Base de Datos: Modelo Relacional
Laia Quichua	done	T0032	Documentación, Título y Propósito
Araceli Diaz	done	T0033	Casos de Uso: Descripción Textual
	planned	T0032	Cerrar sesión usuario
	planned	T0033	Perfil de niño: Calculo de horas restantes
	wait	T0036	Crear referencia a perfil
	planned	T0035	Modificar perfil de niño

# Class Diagram





## Tester

# 1.- User Registration, Authentication and Logout of the app:

## 1.1.-Registration:

Test Code	R.-1.1
Short Description	The user can sign up into the application.
Tester	PASS

Pre-Condition	The user doesn't have an account.
---------------	-----------------------------------

Case	Action	Expected System Result	Pass/Fail
R.-1.1.1	Click on the [Register] button. The user enters in the username field an username that has already been created and click the [submit] button to send the data to the system.	The app clears the fields and shows the next message: "Username already taken /exist in the app"	PASS
R.-1.1.2	Click on the [Register] button. The user enters an email that is already taken and click the [submit] button to send the data to the system.	The app clears the fields and shows the next message: "The email you entered is already been taken, please try again".	PASS
R.-1.1.3	Click on the [Register] button. The user leaves 1 o various fields empty or with an space,... and click the [submit] button to send the data to the system.	The app clears the fields and shows the next message: "Some of the fields are empty, please enter some data".	PASS
R-1.1.4	The User click on the [Register] button and enters a password that has more than 10 characters	The app clears the fields and shows the next message: "The password is incorrect"	
R.-1.1.4	Click on the [Register] button. The user enters the fields with correct formats and correct values and click the [submit] button to send the data to the system.	The app save the user on the app system and database and let the user authenticate into the app.	PASS

Post-Condition	Saves the new user into the database.
----------------	---------------------------------------

## 1.2.-Authentication:

Test Code	R-1.2
Short Description	User can log in into the application.
Tester	PASS

Pre-Condition	The user has to be registered into the application
---------------	--

Case	Action	Expected System Result	Pass/Fail
R-1.2.1	The user enters an username that doesn't exist and click on the [Login] button so that the user can enter into the app functionalities.	The app clears the fields and shows the next message: "This username is not found, please try again".	PASS
R-1.2.2	The user enters an password that doesn't match with the user's profile and click on the [Login] button.	The app clears the fields and shows the next message: "The entered password is incorrect, please try again".	PASS
R-1.2.3	Click on the [Login] button. The user leave 1 or all the fields empty.	The app clears the fields and shows the next message: "The username and/or the password are empty".	PASS
R-1.2.4	The user enters all the fields in the correct formats and correct values. Click on the [Login] button.	The application let the user access to it's functionalities.	PASS

Post-Condition	The user is logged into the application and can use the main functionalities of the app.
----------------	--

### 1.3.- Logout:

Test Code	R.-1.3
Short Description	User can Log out the application.
Tester	PASS

Pre-Condition	The user has to be already logged into the app.
---------------	---

Case	Action	Expected System Result	Pass/Fail
R.-1.3.1	The user, inside the app, click on the [Logout]	The app close the sessions and goes to the login page	PASS

Post-Condition	The user is log out from the application so that can login againñ.
----------------	--

## 2.- Managing User Profile:

### 2.1.-Display User Profile:

Test Code	R-4.1
Short Description	It will show the profile related to the user.
Tester	PASS

Pre-Condition	The has to be logged into the app.
---------------	------------------------------------

Case	Action	Expected System Result	Pass/Fail
R-4.1.1	The user click on the [User Configuration] button	Shows a page with the user's data and option that the user can execute.	PASS

Post-Condition	The user can modify or delete its profile data.
----------------	---

## 2.1.-Modification/Update User Profile:

Test Code	R-4.2
Short Description	The user can modify it's profile.
Tester	PASS

Pre-Condition	-The user has to be logged into the app. -The user has to pressed the [User Configuration] button and display it's profile
---------------	---

Case	Action	Expected System Result	Pass/Fail
<b>R-4.2.1</b>	<b>checkpoint 1</b>		
R-4.2.1.1	The user select the [Change Email] button. Shows an emergent message to enters the actual password and the new email. The user enters an incorrect password and a correct email. Click on the [Save] button.	Shows another message indicating the next message: "Couldn't save the email: Incorrect password" and remains the previous emergent message.	PASS
R-4.2.1.2	The user select the [Change Email] button. Shows an emergent message to enters the actual password and the new email. The user enters a correct password but an existent email. Click on the [Save] button.	Shows another message indicating the next message: "Couldn't save the email: This email already exist on the system." and remains the previous emergent message	PASS
R-4.2.1.3	The user select the [Change Email] button. Shows an emergent message to enters the actual password and the new email. The user enters a correct password but an incorrect email. Click on the [Save] button.	Shows another message indicating the next message: "Couldn't save the email: Incorrect Email." and remains the previous emergent message	PASS
R-4.2.1.4	The user select the [Change Email] button. Shows an emergent message to enters the actual password and the new email. The user enters a correct password and a correct email. Click on the [Save] button.	The app takes the new email and saves it into its database. Updates the user's email with the new one and shows a message: "Email correctly save".	PASS

R-4.2.2	checkpoint 2		
R-4.2.2.1	The user select the [Change Password] button. hows an emergent message to entered the actual password and the new password. The user enters an incorrect actual password and a correct new password. Click on the [Save] button.	Shows another message indicating the next message: “Couldn’t save the new password: Incorrect actual password.” and remains the previous emergent message.	PASS
R-4.2.2.2	The user select the [Change Password] button. hows an emergent message to entered the actual password and the new password. The user enters a correct actual password but an empty new password. Click on the [Save] button.	Shows another message indicating the next message: “Couldn’t save the new password: Empty new password.” and remains the previous emergent message.	PASS
R-4.2.2.3	The user select the [Change Password] button. hows an emergent message to entered the actual password and the new password. The user enters a correct actual password and a correct new password. Click on the [Save] button.	The app takes the new password and saves it into its database. Updates the user’s password with the new one and shows a message: “Password correctly save”.	PASS

Post-Condition	-The app saves the new email and show it on the user’s profile. -The app saves the new password and show it on the user’s profile.
----------------	---

### 2.3.-Delete User Profile:

Test Code	R-4.3
Short Description	It will delete the user profile.
Tester	-

Pre-Condition	-The user has to be logged into the app. -The user has to pressed the [User Configuration] button and display it's profile
---------------	---

Case	Action	Expected System Result	Pass/Fail
R-4.3.1	The user clicks on the [Delete User]	The app shows a warning message.	PASS
R-4.3.2	The user clicks on the [Yes] button inside the warning message.	The app delete the user and all the child profiles associated	PASS
R-4.3.3	The user clicks on the [No] button inside the warning message.	The app cancels the operation.	PASS

Post-Condition	-Deletes the user.
----------------	--------------------

### **3.-Child Profile Management:**

#### **3.1.-Creation of the Child Profile:**

Test Code	R-2.2
Short Description	It creates a child profile.
Tester	PASS

Pre-Condition	-The user has to be logged into the app. -The user has to pressed the [Create new profile +] button and display the page to enters the child data.
---------------	---

Case	Action	Expected System Result	Pass/Fail
------	--------	------------------------	-----------

R-2.2.1	The user doesn't enters into any field and directly clicks on the [Create Profile] button.	Shows the next message: "Couldn't create the new profile: 1 or many fields are empty" and remains in the page.	PASS
R-2.2.2	The user enters an incorrect child name (includes symbols) and directly clicks on the [Create Profile] button.	Shows the next message: "Couldn't create the profile" and remains in the page.	PASS
R-2.2.3	The user only enters a correct child name but let the other fields empty and directly clicks on the [Create Profile] button.	Shows the next message: "Couldn't create the new profile: 1 or many fields are empty" and remains in the page.	PASS
<b>R-2.2.4</b>	<b><u>Check-point 1</u></b>		
R-2.2.4.1	The user enters an correct child name, choose the hour option and enters letters and special symbols.	The app doesn't allow writing letters and leave it empty.	PASS
R-2.2.4.2	(Assuming that the user has already entered a correct name and has already chose the hour option), the user enters a decimal number.	The app doesn't allow writing decimal number and leave the field empty.	PASS
R-2.2.4.3	(Assuming that the user has already entered a correct name and has already chose the hour option), the user enters a number value lower than 0.	The app doesn't allow writing number below 0 and leave the field empty.	PASS
R-2.2.4.4	(Assuming that the user has already entered a correct name and has already chose the hour option), the user enters a number value greater than 24 and clicks on the [Create Profile] button.	Below the field it'll show the next message: "Couldn't create the new profile: The hour can't be greater than 24h".	PASS
R-2.2.4.5	(Assuming that the user has already entered a correct name and has already chose the hour option), he user enters a number value greater than 0 and lower than 24 and clicks on the [Create Profile] button.	The app will register on the database this child and will associate it with this user. Then the app close the page and displays the new child profile.	PASS
<b>R-2.2.5</b>	<b><u>Check-point 2</u></b>		
R-2.2.5.1	The user enters an correct child name, choose the percentage	The app doesn't allow writing letters and symbols, and leave the field	PASS

	option and enters letters and special symbols.	empty.	
R-2.2.5.2	(Assuming that the user has already entered a correct name and has already chose the percentage option), the user enters a decimal number.	The app doesn't allow writing decimal number and leave the field empty.	PASS
R-2.2.5.3	(Assuming that the user has already entered a correct name and has already chose the percentage option), the user enters a number value lower than 0.	The app doesn't allow writing number below 0 and leave the field empty.	PASS
R-2.2.5.4	(Assuming that the user has already entered a correct name and has already chose the percentage option), the user enters a number value greater than 100 and clicks on the [Create Profile] button.	Below the field it'll show the next message: "Couldn't create the new profile: The hour can't be greater than 100%".	PASS
R-2.2.5.5	(Assuming that the user has already entered a correct name and has already chose the percentage option), he user enters a number value greater than 0 and lower than 100 and clicks on the [Create Profile] button.	The app will register on the database this child and will associate it with this user. Then the app close the page and displays the new child profile.	PASS

Post-Condition	<ul style="list-style-type: none"> <li>- Displays the new child profile with the hour entered.</li> <li>- Displays the new child profile with the percentage entered.</li> </ul>
----------------	--

### 3.2.-Share Child Profile to other User:

Test Code	R-1.4
Short Description	An user share the child profile to another user and give the later permissions.
Tester	PASS



Pre-Condition	-The user has to be logged into the app. - It has to display a child profile.
---------------	--

Case	Action	Expected System Result	Pass/Fail
R-1.4.1	The user clicks on the [Share Profile] button.	The system shows an emergent message so the user can enter the user information.	PASS
R-1.4.2	The user enters a nonexistent username and clicks on the [Share] button.	The system shows the next message: "The username doesn't exist"	PASS
R-1.4.3	The user enters an existent username and clicks on the [Share] button.	The app will show the next message: "Child Profile Shared" and creates the relation	

Post-Condition	-The system creates the relation of the user and child profile.
----------------	---

### 3.3.-Display List of Children's Profile:

Test Code	R-2.1
Short Description	Shows a list of child's profile associated to the user
Tester	PASS

Pre-Condition	-The user has to be logged into the app. -The user need to have more than 1 child profile associated.
---------------	--

Case	Action	Expected System Result	Pass/Fail
R-2.1.1	The user click on 2 or more child's profile on the list.	The app remains in the page doing nothing	Pass
R-2.1.2	The user clicks on 1 child profile on the list.	The app will redirect to that profile.	Pass

Post-Condition	The app will show he child profile.
----------------	-------------------------------------

### 3.4.-Display Child Profile:

Test Code	R-2.2
Short Description	Shows a list of child's profile associated to the user
Tester	PASS

Pre-Condition	-The user has to be logged into the app.
---------------	--

Case	Action	Expected System Result	Pass/Fail
R-2.2.1	<b>check.point 1</b>		
R-2.2.1.1	If the user has just 1 child profile associated	The system will show directly the main page of the child user	PASS
R-2.2.2	<b>check.point 2</b>		
R-2.2.2.1	The user is on the list of child profiles associated and clicks one of them.	The system will show that child profile.	PASS

Post-Condition	The app will show child profile.
----------------	----------------------------------

### 3.5.-Display Main Page:

Test Code	R-1.5
Short Description	Shows a main Page to user.
Tester	PASS

Pre-Condition	-The user has to be logged into the app.
---------------	--

Case	Action	Expected System Result	Pass/Fail
R-1.5.1	The user doesn't have any child profile associated.	The system will go to this page, so the user can create or being shared a child profile.	PASS

Post-Condition	The app will show child profile.
----------------	----------------------------------

### 3.6.-Modification/Update Child Profile:

Test Code	R-2.4
Short Description	The user can modify the child profile information
Tester	PASS

Pre-Condition	-The user has to be logged into the app. -The user needs to have the admin role.
---------------	---

Case	Action	Expected System Result	Pass/Fail
R-2.4.1	The user clicks on the [Configuration Child Profile]	The app shows a form with the actual information	PASS
R-2.4.2	The user modifies the name, the user leaves the field empty	The app cancels the action and shows this message: "The field can't be empty".	PASS
R-2.4.3	The user modifies the name, input an invalid name and clicks on the	The system will show a message: "The child name can only be letters"	PASS

	[Accept] button.		
R-2.4.4	The user modifies the name, enters a valid name and clicks on the [Accept] button	The system will store the name.	PASS
R-2.4.5	<b>check-point 1</b>		
R-2.4.5.1	(Assuming the user has entered a valid name and has already stored it) The user change the hour data, the user let the field empty and clicks on the [Accept] button	The system cancels the actions and shows this message: "The field can't be empty"	PASS
R-2.4.5.2	(Assuming the user has entered a valid name and has already stored it) The user changes the hour data, the user enters invalid data.	The system wouldn't let enter that data, will show a message: "invalid hour"	PASS
R-2.4.5.3	(Assuming the user has entered a valid name and has already stored it) The user enters on the hour more than 24 and clicks [Accept] button.	The system cancels the action and shows this message: "The hour can't be more than 24h"	PASS
R-2.4.5.4	(Assuming the user has entered a valid name and has already stored it) The user inputs a number below 0.	The system wouldn't let enter that data, will show a message: "can't be lower than 0".	PASS
R-2.4.5.5	(Assuming the user has entered a valid name and has already stored it) The user inputs a correct hour and clicks [Accept] button.	The system will store it.	PASS
R-2.4.6	<b>check-point 2</b>		
R-2.4.6.1	(Assuming the user has entered a valid name and has already stored it), the user change the percentage data, the user let the field empty and clicks on the [Accept] button	The system cancels the actions and shows this message: "The field can't be empty"	PASS
R-2.4.6.2	(Assuming the user has entered a valid name and has already stored it), the user changes the % data, the user enters invalid data.	The system wouldn't let enter that data, will show a message: "invalid percentage"	PASS
R-2.4.6.3	(Assuming the user has entered a valid name and has already stored it), the user enters on the % more than 100% and clicks on the	The system cancels the action and shows this message: "The percentage can't be more than 100%"	PASS

	[Accept] button		
R-2.4.6.4	(Assuming the user has entered a valid name and has already stored it), the user inputs a number below 0.	The system wouldn't let enter that data, will show a message: "can be lower than 0".	PASS
R-2.4.6.5	(Assuming the user has entered a valid name and has already stored it), the user inputs a correct % and clicks [Accept] button.	The system will store it.	PASS

Post-Condition	-The system has modifies the new data into child profile.
----------------	---

### 3.7.-Delete Child Profile:

Test Code	R-2.5
Short Description	The user can deletes the child profile.
Tester	PASS

Pre-Condition	-The user has to be logged into the app. -The user has to clicked on the [Configuration Child Profile]
---------------	---

Case	Action	Expected System Result	Pass/Fail
R-2.5.1	The user clicks on the [Deletes Child Profile].	The system will show a warning message	PASS
R-2.5.2	The user click on the [No].	The system will cancel the action.	PASS
R.2.5..3	The user clicks on the [Yes] button.	The system will deletes the child profile and it's relation	PASS

Post-Condition	-The system deletes the child profile.
----------------	--

## 4.-Taps Management:

### 4.1.-Creation of Taps:

Test Code	R-3.1
Short Description	Create the taps/historic of the child .
Tester	PASS

Pre-Condition	-The user has to be logged into the app. -The user has to be already into a child profile.
---------------	---

Case	Action	Expected System Result	Pass/Fail
R-3.3.1	The user click on both buttons	The app doesn't do any taps.	Pass
R-3.3.2	The user clicks on the [Eye] button	The tap creates indicating that the child is awake and adds an init time.	PASS
R-3.3.3	After clicked on the [Eye] button, the user click on the [Patch] button.	The system creates the tap that the is wearing the patch and add an init time	PASS
R-3.3.4	The user click on the [Patch] button for the second time.	The system saved the previous tap and add the end time (when the child is not wearing the patch)	PASS
R-3.3.5	The user clicked on the [Eye] button for the second time	The system saved the previous tap and add the end time (when the child has sleep)	PASS
R-3.3.6	If the user doesn't click on the [Eye] button first and click on the [Patch] button.	The system wouldn't allow create that tap.	PASS
R-3.3.7	If the user has clicked the [Eye] button for the first time, then clicked on the [Patch] button and then clicks on the [Eye] button without clicking on the [Patch] first.	The system will close and saves the both button with the same end time.	PASS
R-3.3.8	If the user clicks on whatever button and leaves the app on the background or does another thing	The system will keep active the taps until the user decides to.	PASS

	inside the app.		
--	-----------------	--	--

Post-Condition	-The system will create and saves the taps
----------------	--

## 4.2.-Display of All Created Taps:

Test Code	R-3.2
Short Description	shows the list of taps created.
Tester	-

Pre-Condition	-The user has to be logged into the app. -The user has to be already into a child profile.
---------------	---

Case	Action	Expected System Result	Pass/Fail
R-3.2.1	The user click on the [Configuration Child Profile] button.	The system display the child profile configuration page.	PASS
<b>R-3.2.2</b>	<b><u>Checkpoint 1</u></b>		
R-3.2.2.1	The user clicked on the [List of Taps "Sleep and Awake"] button	The system list the taps create as "Sleep/Awake"	PASS
<b>R-3.2.3</b>	<b><u>Checkpoint 2</u></b>		
R-3.2.3.1	The user clicked on the [List of Taps "Wearing / not wearing patch"] button	The system list the taps create as "Wearing / not wearing Patch"	PASS

Post-Condition	-Shows a list of the taps.
----------------	----------------------------

## 4.3-Modify Taps:

Test Code	R-3.7
Short Description	Modifies the tap information.

Tester	-
--------	---

Pre-Condition	<ul style="list-style-type: none"> <li>-The user has to be logged into the app.</li> <li>-The user has to be already into a child profile.</li> <li>-The user has to be already on the Configuration Child Profile</li> <li>-The user has to be already on the list of taps (either “awake/Sleep” or “wearing/not wearing patch”)</li> </ul>
---------------	--

Case	Action	Expected System Result	Pass/Fail
<b>R-3.7.1</b>	<b><u>Checkpoint 1</u></b>		
R-3.7.1.1	The user click on the hour (can be init hour, end hour or both)	The user displays a form to be completed	PASS
R-3.7.1.2	The user let the form empty.	The system cancels the action and shows an error message	PASS
R-3.7.1.3	The user complete the hour form with invalid values (initHour and endHour).	The app cancels the action and shows an error message.	PASS
R-3.7.1.4	The user complete the form with correct values (initHour and endHour).	The app saves the new data and show it.	PASS
<b>R-3.7.2</b>	<b><u>Checkpoint 2</u></b>		
R-3.7.2.1	The user click on the date (can be init date, end date or both)	The user displays a form to be completed	PASS
R-3.7.2.2	The user let the form empty.	The system cancels the action and shows an error message	PASS
R-3.7.2.3	The user complete the form with invalid values (initDate and endDate)	The app cancels the action and shows an error message.	PASS
R-3.7.2.4	The user complete the form with correct values (initDate and endDate)	The app saves the new data and show it.	PASS

Post-Condition	<ul style="list-style-type: none"> <li>-The system saves the new hour correctly.</li> <li>-The system saves the new date correctly.</li> </ul>
----------------	--



#### 4.4.-Delete Taps:

Test Code	R-3.8
Short Description	Delete a tap.
Tester	-

Pre-Condition	<ul style="list-style-type: none"> <li>-The user has to be logged into the app.</li> <li>-The user has to be already into a child profile.</li> <li>-The user has to be already on the Configuration Child Profile</li> <li>-The user has to be already on the list of taps (either “awake/Sleep” or “wearing/not wearing patch”)</li> </ul>
---------------	--

Case	Action	Expected System Result	Pass/Fail
R-3.8.1	The user click on the [X] button next to the tap description.	The system will show a warning message	PASS
R-3.8.2	The user click on the [No] button from the warning message	The system cancels the action and close the message.	PASS
R-3.8.3	The user click on the [Yes] button from the warning message.	The system will delete the tap from the database.	PASS

Post-Condition	-The tap is delete.
----------------	---------------------

DATA SET (DATOS DE PRUEBA):

TEST CODE	DATA #1	DATA #2	DATA #3	ACTION
R-1.1.1	rdiaz1	prueba	prueba@prueba.com	REGISTERING
R-1.1.2	prueba	prueba	hdjkhjfsfdsg	
R-1.1.3	-	-	-	
R-1.1.4	prueba	123456789000	prueba@prueba.com	
R-1.1.5	prueba	prueba	prueba@prueba.com	
R-1.2.1	gdhgdhgdghd	prueba	-	LOGIN
R-1.2.2	prueba	ghfjdfsgh	-	
R-1.2.3	-	-	-	
R-1.2.4	prueba	prueba	-	
R-4.2.1.1	FJSKGF	proves@proves.com		MODIFY USER PROFILE
R-4.2.1.2	prueba	hols@gmail.com		
R-4.2.1.3	prueba	hfjksgvfbkhsvnmxv		
R-4.2.1.4	prueba	proves@proves.com		
R-4.2.2.1	ggyghg	proves		
R-4.2.2.2	prueba			
R-4.2.2.3	prueba	proves		
R-2.2.1				CREATION OF CHILD PROFILE
R-2.2.2	niño@~\$prueba			
R-2.2.3	niñoPrueba			
R-2.2.4.1	niñoPrueba	(\$/-(%)?%)		
R-2.2.4.2	niñoPrueba	63,5		
R-2.2.4.3	niñoPrueba	-20		
R-2.2.4.4	niñoPrueba	60		
R-2.2.4.5	niñoPrueba	6		
R-2.2.5.1	niñoPrueba	\$.-%\$.		
R-2.2.5.2	niñoPrueba	45,2		
R-2.2.5.3	niñoPrueba	-60		
R-2.2.5.4	niñoPrueba	400		
R-2.2.5.5	niñoPrueba	25		

R-1.4.2	GHDJGFHFD			SHARE CHILD PROFILE
R-1.4.3	mtorres19			
R-2.4.2				MODIFY CHILD PROFILE
R-2.4.3	Jdhjf@jkid%			
R-2.4.4	José Manuel			
R-2.4.5.1				
R-2.4.5.2	jdk.#@			
R-2.4.5.3		78		
R-2.4.5.4		-4		
R-2.4.5.5		6		
R-2.4.6.1				
R-2.4.6.2	ghdds.\*+@#			
R-2.4.6.3		200		
R-2.4.6.4		-12		
R-2.4.6.5		43		
R-3.7.1.2				MODIFY TAPS
R-3.7.1.3	GM-1GF@	fdbshjfs456@		
R-3.7.1.4	13:56:36	16:45:12		
R-3.7.2.2				
R-3.7.2.3	hgfgfdghd@#	vfhdusgf456*/		
R-3.7.2.4	2019/04/13	2019/04/13		