3.1 External interfaces requirements

3.1.2 Hardware Interface

This software application needs at least a smartphone or a computer using Internet that can use GPS Service and GSM Service.
Bluetooth and/or Wifi connections are also required to collect the health data generated by other devices (smartwatches, smart connected clothes and other wearables)

3.1.3 Software Interface

The application is using external services that are suitable for embedding in order to simplify the overall architecture and make it more lightweight.

- Collecting the health data

This application requires the collect of some health data measured by one or many devices (smartphone, smartwatches, smart connected clothes, …). There are several solutions to this requirement. One of them is using the Google Fit API that collects the data from different devices working with Android or Wear OS (Version of Google’s Android operating system designed for smartwatches and other wearables).

- Geolocation

Data4Help needs to know at anytime the GPS location of the users especially for the *AutomatedSOS*’s users. There exists several application to collect the GPS location such as Google Android Device Manager API.

- Mapping and displaying of the run

Track4Run requires a map of the run planned. This map is useful for defining the path of the run but also for showing the situation of the run to spectators. A option would be the using of the service Google Maps that allows us the realization of these functions in real-time.

3.1.3 Communication Interface

This network-connected Android application uses HTTPS to send and receive data. The sending of HTTPS requests (encrypted requests with TLS) is important for the protection and the confidentiality of the data of the users. The Android platform includes the HttpURLConnection client, which supports TLS, streaming uploads and downloads, configurable timeouts, IPv6, and connection pooling and can be used for the development of the application.

One way to simplify the overall architecture and make it more lightweight is to use the Google APIs (see section 3.1.2). The communication with the different google services (such as Google Maps, Google Fit) is carried out by the using of GoogleApiClient ("Google API Client") object. The Google API Client provides a common entry point to Google Play services and manages the network connection between the user's device and each Google service.

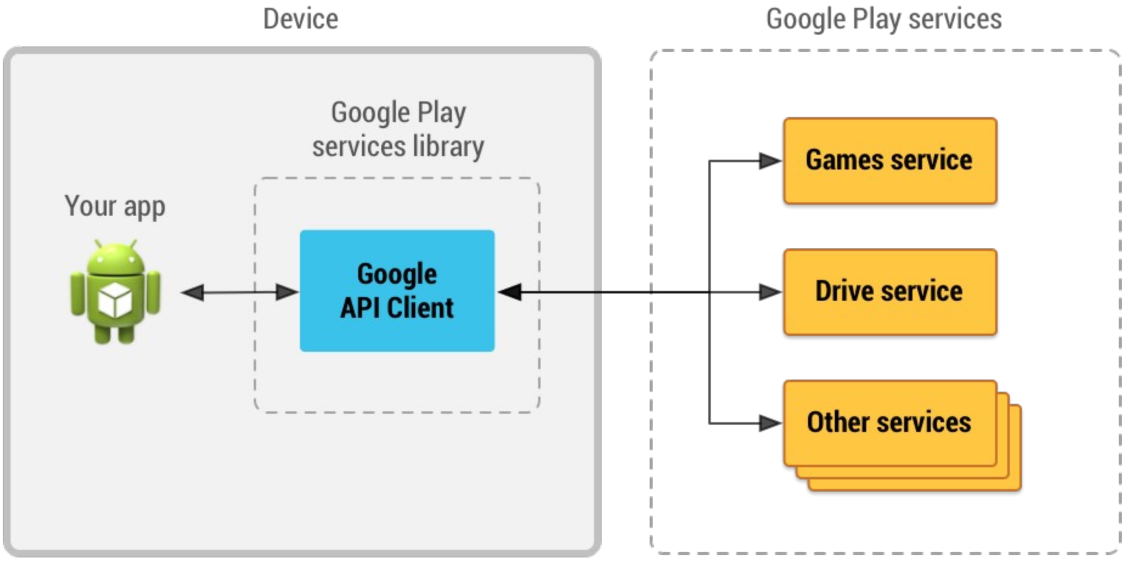

Figure 1: An illustration showing how the Google API Client provides an interface for connecting and making calls to any of the available Google Play services such as Google Play Games and Google Drive.

Thanks to the service Google Fit, we can use the service Sensors API. Sensors API provides access to raw sensor data streams from sensors available on the Android device and from sensors available in companion devices, such as wearables (smartwatches, smart clothes, ...).

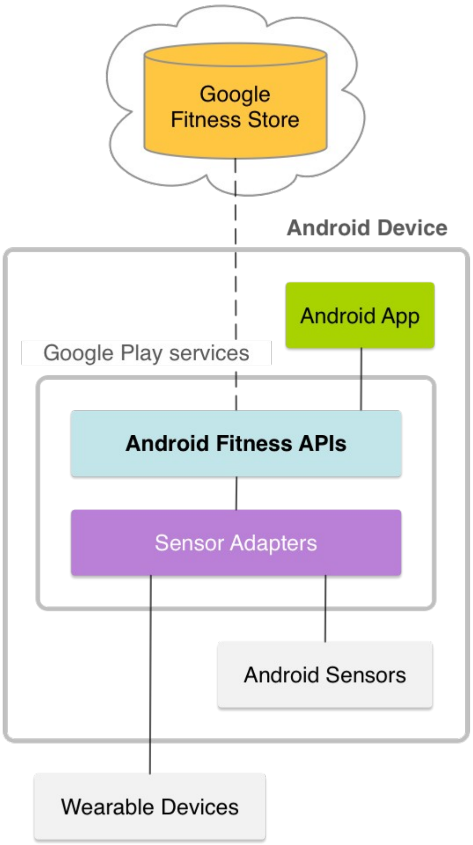

Figure 2: An illustration showing the communication between the application and the Google Fit Service.

In order to make an emergency call, *AutomatedSOS* uses the framework *Android Telecom.* The *Android Telecom* framework is responsible for managing calls on an Android device. This can include SIM-based calls using the Telephony framework, VOIP calls using SIP, or via a third-party VOIP.

3.2 Scenarios

3.2.1 Scenario 1

Luigi suffers of a serious fever. However, Dr Mario, the Luigi's doctor, still does not know the disease responsible of this fever and if the treatment given to Luigi will be efficient. Indeed, two treatments are possible : one if it is a food intoxication and the other one if it is viral disease. Both of them are subscribed to the service *Data4Help* and download the app on their phones. Luigi uses a smartwatch connected to his phone for monitoring of his temperature and Dr Mario can watch at anytime on his phone the collected data because Luigi approved the requests of his doctor. If after some hours, the treatment given to Luigi is still not efficient, Dr Mario can give the other treatment to his patient.

3.2.2 Scenario 2

A Research Institute *4TheEnvironment,* based in Milano, would like to study the effects of the pollution on the health of the young adults. For the realization of this study, *4TheEnvironment* decides to use the service *Data4Help,* when the pollution is above a threshold defined. With this service *4TheEnvironment* can collect some health data such as the temperature, the cardiac rhythm, the blood pressure, the dehydration state… measured by smartphones, smartwatches or other wearables on the young adults of 18-25 years old. *4TheEnvironment* collects and saves these data when they are above a threshold defined by the research institute. The number of people targeted is enough important (higher than 1000 in Milano) to guarantee the confidentiality of these people and therefore *4TheEnvironment* can realize its study.

3.2.3 Scenario 3

Chiara is 98 years old and still leaves alone in her home. She suffers of some cardiac problems and at this age the chance of heart attacks is really important. Her daughter, Lucia, has decided to register Chiara to the service *AutomatedSOS.* With a wearable (that she connected to the app), the cardiac rhythm is sent in real-time to *AutomatedSOS*. Yesterday, Chiara was victim of a heart attack and the service detected the attack some seconds after its beginning and sent an ambulance to the location of Chiara’s home few minutes after. Thanks to *AutomatedSOS* and its fast response time Chiara is still alive.

3.2.4 Scenario 4

Stefania participates at Polirun that is a run organized by Politecnico di Milano. This year, the organizers have decided to use the service *Track4Run* to define the path of the run and invite the different participants. The organizers asked to all participant to measure their cardiac rhythm with a wearable. Like Stefania, each runner must subscribe to Track4Run, must be equipped of a wearable measuring the cardiac rhythm and must allow the access to these data by the organizers to participate at Polirun. For the run with this service, the Stefania’s family can access in real-time at her location and follow her at some key points to support her. At the middle of the run, Stefania has a loss of consciousness. Her cardiac rhythm is irregular and abnormally low, the problem is detected by the service and the organizers. A medical unit is immediately sent to her location.

3.3 Functional requirements

[G1] - Collect data (location, health status: cardiac rhythm, steps) from all registered users

R1: System should allow registration of users with codice fiscale and personal data (age, gender, blood type)

R2: System should collect and store date of the registered users

D1: Users have valid codice fiscale

D2: Location of users is obtained from GPS of the user's device

D3: Users all always connected to internet

[G2] Registered third parties can access data from individual users with permission

R3: System should allow registration of third party with valid ID

R4: System should pass requests from third party to individual based on codice fiscale

R5: System should make the requested data available to the third party only if the individual approves the request

D4: Third party knows the codice fiscale of the user

[G3] Registered third parties can request for anonymized data of groups of individuals

R6: System should be able to retrieve data based on category

R7: System should be able to accept or refuse a request based on the size of data

[G4] Registered first parties can subscribe and receive data

R8: System should allow subscription for data requests from third party

R9: System should send requested data as soon as they are produced

[G5] registered elderly users can subscribe for a personalised and non-intrusive SOS service

R10: System should allow subscribtion for elderly people by entering personal preferences (tresholds for health parameters)

[G6] Call an ambulance for subscribed elderly people if needed

R11: Monitor health parameters of subscribed elderly people continuously

R12: Call an ambulance only when health parameters go below threshold

D5: Facility to call an ambulance is available

[G7] Run organizers can set up a path for the run

R13: System should allow registration of run organisers

D6: System uses existing map application

[G8] run organizers can set up enrollment process

R14: System should allow creating an enrollment process

[G9] run participants can enroll for a run

R15: System should allow run participants to enroll for a run

D7: Run participants should be registered users of Data4Help service

[G10] showing the situation of the run to spectators

R16: system should create a view of the map with all run participants positions

R17: system should allow spectators to have access to the view

D8: spectators have internet access

3.3.1 Use case diagram

Name	**Sign Up**
Actor	User
Entry conditions	The user has installed the application on her/his device.
Events flow	1. The user clicks on the « Sign Up » button. 2. The user gives her/his codice fiscale. 3. The user fills all the mandatory fields and provides the necessary information. 4. The user clicks on « Confirm button ». 5. *Data4Help* saves the data.
Exit conditions	The user has successfully registered and now she/he is able to use the application.
Exceptions	1. The user is already signed up. 2. The user did not fill all of the mandatory fields with valid data. 3. The codice fiscale is already taken. 4. The e-mail is already registered. 5. All the exceptions are handled by notifying the user and taking him back to the sign up activity.

Name	**Log in**
Actor	User
Entry conditions	The user is previously successfully signed up and has the application installed on her/his device.
Events flow	1. The user opens the application on her/his device. 2. The user enters her/his codice fiscale in the “Codice fiscalle” and “Password” fields of the log in page of *Data4Help.* 3. The user clicks on the “Log in” button. 4. The user is successfully logged in her/his “Data4Help” and the system automatically redirects her/him to the home page.
Exit conditions	The user is successfully redirected to the home page.
Exceptions	1. The user enters an invalid codice fiscale.

	2. The user enters an invalid Password. 3. All the exceptions are handled by notifying the user and taking her/him back to the login activity.

Name	**Accept requests**
Actor	User
Entry conditions	The user has already logged in.
Events flow	1. The user clicks on the button “Requests”. 2. The user clicks on the button “Approve” to accept each request sent. Otherwise, he clicks on the Button “Disapprove”. 3. The user clicks on the “Confirm” button. 4. The system saves the preferences.
Exit conditions	User preferences are saved to her/his profile and are considered.
Exceptions	There are no requests and the user is notified with an empty list of requests.

Name	**Add a wearable**
Actor	User
Entry conditions	The user has already logged in
Events flow	1. The user clicks on the button “Wearables”. 2. The user clicks the option “Add a wearable”. 3. The user chooses the wanted wearable in the list proposed by *Data4Help*. 4. The user confirms her/his choice in clicking one the button “Associate”. 5. The system saves the new wearable.
Exit conditions	The new wearable is saved to her/his profile and is considered.
Exceptions	1. The system find no wearables and notifies the user with an empty list of wearables.

	2. The system does not success to associate the wearable and notifies the user of its failure.

Name	**Set preferences**
Actor	User
Entry conditions	The user has already logged in.
Events flow	1. The user clicks on the button “Preferences”. 2. The user clicks on a button representing a wearable connected with the system. 3. The user chooses the set of health parameters measured by the wearable. 4. The system saves the preferences.
Exit conditions	The preferences are saved to her/his profile and are considered.
Exceptions	1. The system find no wearables connected to the service and notifies the user. 2. The system find any health parameters that can be measured by the wearable and notifies the user.

Name	**Collect the data**
Actor	*Data4Help*
Entry conditions	1. One or more wearables are associated with the system. 2. One or more wearables are allowed to collect some health data.
Events flow	1. *Data4Help* gathers all the data provided by external services and wearables. 2. *Data4Help* saves the data on the servers.
Exit conditions	The health data are saved on the servers.
Exceptions	1. There are problems in the collecting of health data and *Data4Help* notifies the user.

Name	**Sign Up**
Actor	Third party

Entry conditions	The third party has installed the application on its device
Events flow	1. The third party clicks on the « Sign Up » button. 2. The third party fills all the mandatory fields and provides the necessary information. 3. The third party clicks on « Confirm button ». 4. *Data4Help* saves the data.
Exit conditions	The third party has successfully registered and now it is able to use the application.
Exceptions	1. The third party is already signed up. 2. The third party did not fill all of the mandatory fields with valid data. 3. The username is already taken. 4. The e-mail is already registered. 5. All the exceptions are handled by notifying the third party and taking him back to the sign up activity.

Name	**Log in**
Actor	Third party
Entry conditions	The third party is previously successfully signed up and has the application installed on her/his device.
Events flow	1. The third party opens the application on its device. 2. The third party enters its credentials in the “Username” and “Password” fields of the log in page of “Data4Help”. 3. The third party clicks on the “Log in” button. 4. The third party is successfully logged in its *Data4Help* and the system automatically redirects its to the home page.
Exit conditions	The third party is successfully redirected to the home page.
Exceptions	1. The third party enters an invalid Username. 2. The third party enters an invalid Password. 3. All the exceptions are handled by notifying the third party and taking its back to the login activity.

Name	**Access of the data of a specific individual**
Actor	Third party
Entry conditions	The third party has already logged in.
Events flow	1. The third party clicks on the button “Search”. 2. The third party writes the codice fiscale of the targeted user. 3. The third party chooses the health data that it wants collect. 4. The system asks a request to the user if it is the first time that the third party wants collect these data. 5. If the user allows the request, the third party can collect the wanted health data.
Exit conditions	The health data of a specific individual are collected.
Exceptions	1. The codice fiscale is not valid, the third party is notified that the code does not work and the third party is taken back to the login activity. 2. The request is rejected, the third party is notified and is taken back to the login activity. 3. The request is accepted but there are no data, the user is notified and is taken back to the login activity.

Name	**Access of the data of a population**
Actor	Third party
Entry conditions	The third party has already logged in
Events flow	1. The third party clicks on the button “Search for a population”. 2. The third party chooses the criteria that define the targeted population (such as age, sex, locations…). 3. The third party selects the health data that it wants to collect on this population. 4. If the size of the targeted population giving these data is larger than 1000, the request is accepted. 5. The third party can collect the wanted health data of the targeted population.

Exit conditions	The health data of a targeted population are collected.
Exceptions	1. The size of the population is less than 1000, the third party is notified and is taken back to the login activity.

Name	**Sign up for AutomatedSOS**
Actor	User
Entry conditions	The user has already logged in.
Events flow	1. The user clicks on the «AutomatedSOS» button. 2. The user confirms his/her subscription to *AutomatedSOS.* 3. The user fills all the mandatory fields and provides the necessary information. 4. The user provides her/his location and/or the device that can locate her/him. 5. The user specifies the health data to monitor. 6. The user defines a threshold for each health data. 7. *AutomatedSOS* saves the data.
Exit conditions	The user has successfully registered to *AutomatedSOS* and the thresholds are saved.
Exceptions	1. The user is already signed up. 2. The user did not fill all of the mandatory fields with valid data. 3. The user does not provide her/his location or a device that can locate her/him. 4. The user does not specify any health data. 5. The chosen health data can not be measured by any wearables. 6. The user does not specify any thresholds. 7. The chosen thresholds are not valid. 8. All the exceptions are handled by notifying the user and taking him back to the sign up activity.

Name	**Call an ambulance**
Actor	*AutomatedSOS*
Entry conditions	1. The user is registered to *AutomatedSOS.* 2. The health data collected are below the thresholds defined.
Events flow	1. The system monitors the health data for 5 seconds. 2. If these health data are still below the thresholds, the system gets the user’s location with the location given by the user or by a wearable connected to her/him. 3. A request with the user’s location and health data is sent to a medical service. 4. The request is treated by the medical service. 5. The system notifies the user that a medical service is called. 6. A request is sent by the medical service to confirm the ambulance. 7. The system notifies the user that an ambulance is coming to her/his location.
Exit conditions	A request sent by the medical service to confirm the ambulance.
Exceptions	1. The health data are already higher than the threshold for 5 seconds, *AutomatedSOS* stops its procedure and goes back to the entry conditions after notifying the user. 2. The system does not receive any request of the medical service after 1 minute. A new request with the user’s location and health data is sent to the medical service until having an answer.

Name	**Create a run**
Actor	Third party
Entry conditions	The third party has already logged in.
Events flow	1. The third party clicks on the “Track4Run” button. 2. The third party searches the location where the run will

	take place. 3. The third party defines the steps of the run on the map. 4. The third party chooses the criteria of the targeted population and participating to the run (age, location, sex, …). 5. The Third party chooses health data that can be collected. 6. The third party chooses if this collecting is mandatory. 7. The system sends an invitation to all people concerned by the criteria.
Exit conditions	The system sent the invitations.
Exceptions	1. The system does not find the location of the run. 2. The third party does not provide a correct path of the run. 3. No one was found according to the chosen criteria. 4. All the exceptions are handled by notifying the third party and taking him back to the *Track4Run* activity.

Name	**Participate to a run**
Actor	User
Entry conditions	The user has already logged in.
Events flow	1. The user clicks on the “Track4Run” button. 2. The user accepts the invitation(s). 3. The user fills all the mandatory fields and provides the necessary information. 4. The user allows the health data collected for the run.
Exit conditions	The user has successfully registered.

Exceptions	1. The user declines the invitation and the invitation is deleted and the user is taken back to the home page. 2. The user does not specify any health data. 3. The user does not allow the organizers to collect the health data, if the collecting is mandatory the user is notified and the registration is still incomplete so the user goes back to the step 4.

Name	**View the run**
Actor	Spectators
Entry conditions	The spectator has installed the application on her/his device.
Events flow	1. The spectator clicks on the “Track4Run” button. 2. The spectator searches a run. 3. The spectator visualizes the run with the location of all participants.
Exit conditions	The map is displayed.
Exceptions	1. The searched run does not exist, the user is notified and she/he is taken back to the Track4Run page.