**UserAuthService:**

This interface is implemented by the UserProfileMng and provide the functionalities to authenticate a user.

- register(email: String, username: String, password: String, SSN: String): String

This method register a new user into the system, it also validate the fields which can’t be used for other users. it returns the token to keep the user logged in.

* addPersonalInformation (token:String firstName: String. lastName: String, birthDate: Date, gender: GenderType, weight: int, height:int , phoneNumber: String):void

This method uses the registration token to add the mandatory data of the user into the system.

The last three fields can be update later at any time.

* login(username: String, password: String): String

This method is used to log in a user with his credentials, It returns the token to keep the user logged in

* getUser(token:String): User

This method is used to get the User object from the corresponding token.

**TpAuthService:**

This interface is implemented by the TpProfileMng and provide the functionalities to authenticate a ThirdParty.

* register(email: String, username: String, password: String, companyName: String): String

This method register a new ThirdParty into the system, it also validate the fields which can’t be used for other ThirdParties. it returns the token to keep the ThirdParty logged in.

* login(username: String, password: String): String

This method is used to log in a ThirdParty with his credentials, It returns the token to keep the ThirdParty logged in.

- getTp(token:int): ThirdParty

This method is used to get the ThirdParty object from the corresponding token.

**UserDataService:**

This interface is implemented by the UserDataMng and provide the functionalities to update users data.

* updateHealthStatus (user: User, max\_pressure: int, min\_pressure: int, bpm: int) : void

This method is used to update the Health Status of the user (1st parameter). It notify the other components about the new data arrived.

* updateLocation (user: User, lat: int. long: int): void

This method is used to update the location of the user. It is used to update the location of a user once new data arrives.

* updatePersonalData(user: User, height: int, weight: int,  phoneNum: String, SOSsubscription: bool): User

This method is used to update the personal data of the user.

**RequestService**:

This interface is implemented by the RequestMng and provide the functionalities to interact with the requests.

* addSingleReq (SSN: String, description: String, tp: ThirdParty) : void

This method is used to add a new single request from a thirdParty (last parameter).

- addGroupReq(gender:GenderGroupReq, sqa:SquareArea, ageR;ageRange, tp: ThirdParty) : bool

This method is used to add a group request from a thirdParty (last parameter). It return the acceptance of the request.

- getAllReq(tp: ThirdParty): List<Request>

This method is used to get all the requests made by a specific ThirdParty

* refuse (user: User, idReq: int): List<SingleReq>

This method is used by a user to refuse a specific request. It return the list of all the pendant request of the user.

* accept (user: User, idReq: int): List<SingleReq>

This method is used by a user to accept a specific request. It return the list of all the accepted request of the user.

* withdrawAuth(user: User, idReq: int): List<SingleReq>

This method is used by a user to withdraw a specific accepted request. It return the list of all the accepted request of the user.

* getAcceptedReqList(user: User): List<SingleReq>

This method is used to get the list of all the accepted request of a specific user.

* getPendantReqList(user: User): List<SingleReq>

This method is used to get the list of all the pendant request of a specific user.

**SubscribedRequestService**:

This interface is implemented by the RequestMng and provide the functionalities to get informations about subscribed requests.

* getTpSubscribed (u: User): List <ThirdParty>

This method is used to get the list of all the Third Parties subscribed to the new data of a specific user.

- getValidGroupSub (u: User): List <GroupRequest>

This method is used to get the list of all the GroupRequests which a specific user is in the limits, the number of users is > 1000 and the corresponding third party is subscribed.

* getGroupData (gr: GroupRequest): List <User>

This method is used to get the list of data from a group requests.

**SubscribedDataService**:

This interface is implemented by the SubscriptionMng and provide the functionalities to notify the arrive of new data.

* sendData (u: User, hs: HealthStatus): void

Notify the arrive of new health status data of the specific user (1st parameter).

**ApiSubscribedService**:

This interface is implemented by the APIsMng and provide the functionalities to send new data to a third party;

* sendSingleData (u: User, hs: HealthStatus,tp: ThirdParty): void

This method is used to send a user data and health status to a third party subscribed to it.

* sendGroupData (tp: ThirdParty, listHs: List<HealthStatus> listWeight: List<Integer>, listHeight: List<Integer>, gp: GroupRequest): void

This method is used to send a set of data and health status to a third party subscribed to the corresponding group request.

**SosService**:

This interface is implemented by the SosMng and provide the functionalities to notify the arrive of new data of an user subscribed to the AutomatedSos service.

* checkHelthStatus (u: User): void

This method is used to check the health status of a specific user.