

Virtual LifeStyle Coach App

2016 / 2017 IntroSDE Final Project

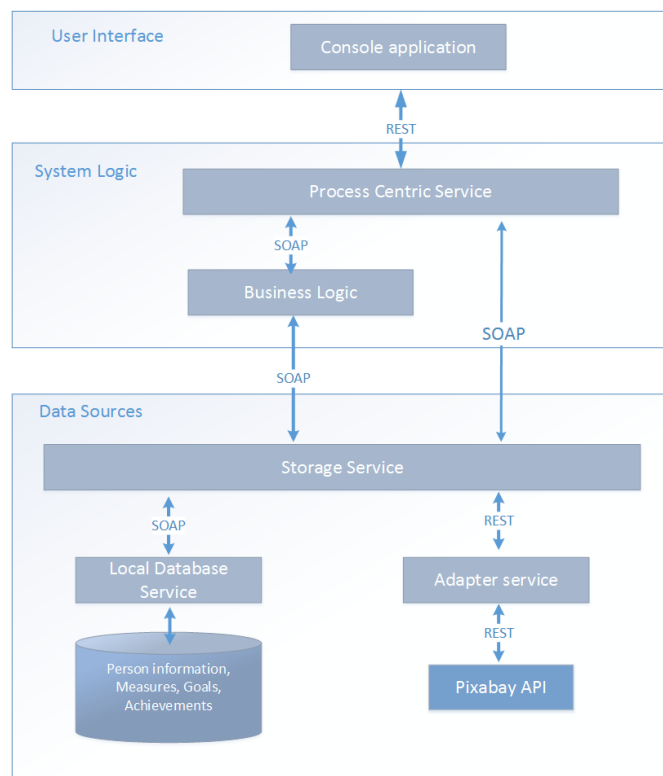
Liviu Bogdan 183121

Introduction

This implementation of the Virtual LifeCoach idea presents itself with a simple console user interface. Using this users can keep track their current health profile and goals. The user can add new measures as well as set personal goals. As the user updates the health profile he is given feedback on his goal progress, receiving encouragement if he strays from the goal or enjoyable pictures if he is moving closer to the goal. Once a goal is completed it is marked as an achievement and the user is congratulated. Goals are set based on available measure types, as such they can target weight, sleep, calories intake, etc.

Architecture

The application consists of five services, 2 REST-based and 3 through SOAP, as seen in the the following diagram:



Services:

- **Local Database Service:** this layer sits on top of an Sqlite database. It is used to persist and retrieve local data. It is exposed using *SOAP*.
- **Adapter Service:** this layer is used to interface with an External API, Pixabay. The API is accessed via REST and is used to retrieve pictures using a keyword. Internal services need to access is via REST.
- **Storage Service:** this layer is build on top of the previous two and it simply provides a single access point to all data to the system logic services. The service is exposed using SOAP and provides calls for every data handled by the lower layers.
- **Business Logic Service:** this service is used to handle computation and decision making. It decides whether a goal is completed and what kind of feedback to give the user. It is the third SOAP-based service in the application.
- **Process Centric Service:** the last service is used as a gateway by the user client via REST. It then forwards relevant requests to the Business Logic Service and Storage Service according to complexity. Simple operations such as readPerson go straight to the Storage Service as it consists in a simple retrieval while more complex operations that need to make decisions go to the Business Logic Service.

The operations described in the introduction can be initiated using the command line interface in the user client.

All user input is sent via REST to a Process Centric Service in charge of forwarding requests, either to the Business Logic Service or to the Storage Service for less complex operations. The Business Logic computes and informs the user of his goal progress and decides which kind of motivation to give the user. Feedback comes in the form of message + picture url which changes based on the previous decision (eg. moving further from the goal : try harder). The picture is retrieved from an external REST api, Pixabay. Both this external API and local data is only accessed through a Storage Service layer by the BusinessLogic. All others are forwarded directly to the StorageService.

All other exchanges are done via SOAP with the exception of Storage Service to Adapter Service (external data handler) which is again done via REST. REST requests and replies are handled in *application/json* format while SOAP uses *text/xml* format.