

Life-coach application

Introduction

The Life-coach is a simple console application which helps the user to track his/her current life status. The application automatically sets up goals for each day and checks if the user achieved that goal. If the user achieved a goal the application congratulates him/her and sends a pretty picture as a reward. In the current stage of the application the user can track his/her sleeping hours and steps made that day and achieve rewards if he/she meets the daily goal.

Features

When the application starts the user has to login. Currently it is possible to login with 4 different users. After login the user can see his/her:

- **profile information** - choose this option to list personal information about the user like first name, last name, email address, user name, etc.
- **current life status** - choose this option to see the last update of life status information.
- **achieved goals** - in this option the user can see what goals have he or she achieved so far.
- **register new life status** - in this option the user can update his/her life status regarding to a chosen measure type. When the user inputs the data the application updates his/her life status information and checks if she/he have already met the goals for that day. If the user met the goal the application sends a greetings.

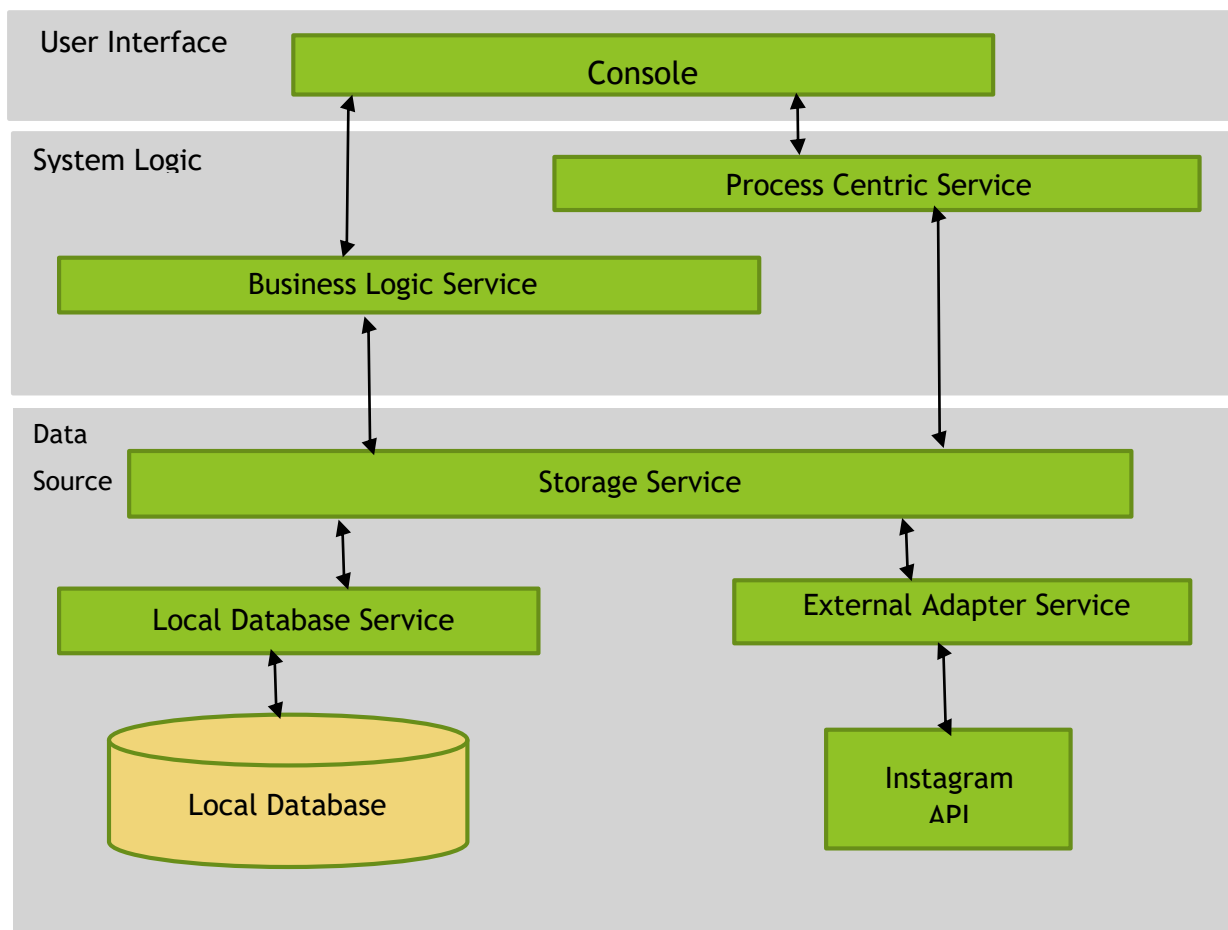
The user then can logout and login again if she/he wants or close the application.

Architecture

The application consists of five different services.

- **External Adapter Service** - this service provides the access to the external API which is in this case the *Instagram API*. According to preset tags it connects to the Instagram and downloads a motivational picture which can be sent to the user later on. The communication between Instagram and the External Adapter happens through **REST** and uses **JSON** objects.
- **Local Database Service** - this layer provides access to the local database which is an SQLite database. The service takes care about persistency and data retrieval. The layer can be reached through **SOAP**.

- **Storage Service** - this layer sits upon the External Adapter Service and the Local Database Service layers. The Storage Service provides every data to the upper layers, and forwards every information to the lower ones. It combines all the possible data retrieval features. The upper layers can only get data through this layer. The layer uses **REST** to communicate to the **upper layers**, also to the **External Adapter**, but uses **SOAP** to connect to the **Local Database Service**.
- **Business Logic Service** - this service contains the main logic of the application. The Business Logic is responsible to check the goals of the user and compute if they are reached or not. The layer uses **REST** and **JSON** to communicate. It can be reached from the user interface but only to get data.
- **Process Centric Service** - the layer acts like a gateway between the user interface and the rest of the application. During complicated tasks, like the computation of a goal calls the appropriate services and forwards the necessary data to the user. It does not do computations, only delegates tasks and coordinates the flow of the data. The layer uses **REST** to communicate.



Architecture of the application

The format of the messages between the layers is JSON. I decided to use this format because it is getting popular due to its simplicity and compact structure. Also the Instagram API uses JSON as a message format which shows that current applications in use are based on JSON.

Further development

The application can be easily improved and can handle more goal types or measurements. It can be expanded only by modifying the business logic layer since in the lower layers the handling of multiple users and multiple measurements are implemented.

Also this application might use a simple console but it can be changed for example to smart phone application interface which would be logical regarding to the functionality of the application. By adding more goals and more measure types the application could keep track of all the activities of the user and provide motivation.

The goals could adjust to the intensity of the activities what the users provide, and increase the amount of work needed to achieve a goal. Now the application get its input data from the user but by adding other external APIs the activities could be tracked automatically.

The types of notification could be email, push-notification or other and the data could be stored online in some cloud service.

In conclusion there are many ways to improve the application and help people to be more fit and healthy.

Zsanett Orosz

2016.01.19