UNIVERSITÀ DEGLI STUDI
DI TRENTO

Life Coach

Project Report

Team Members: Elvis Sotir Koci , Kemele Muhamed Endris

Laboratory to Service Design and Engineering.

Academic Year: 2013-2014

Contents

Figures

1. Introduction

The goal of this project is to offer a solution that could help user monitor their life style and health measures. This application is thought for both users who are facing health issues and for those who want to improve their lifestyle and health. The application allows the user to manage their measurements, define goals and monitor them, manage schedules and log everyday meal consumption. The users get immediate feedback about their progress towards the goals, about their measurements and meal logging. . Another feature is the motivational quotes, which tries to give some kind of support to the users to fulfill their goals. Also the application reminds the users about their schedules. Currently the application is supporting reminders only through the User Interface.

2. Analysis

1.1. Scenarios

The most usual scenarios about application similar to the LifeCoach are about senior citizens. They are commonly facing health issues and need to frequently monitor their health measurement, such as blood pressure, heart rate and other. They also have to take medicines and regulate their diet. Also, another important factor for the elderly is physical activity. By performing physical activities such as walking and jogging they might reduce the risk of facing in the future other health problems and also improve their current health condition. Physical activities, food consumption and other activities can be grouped under lifestyle measure, which shall help monitor the lifestyle of senior citizens. However just monitoring is not enough, the elderly need support and motivation to maintain the right lifestyle and follow the suggestions of the doctors.

Furthermore other type of users could find the application helpful. Life coach could support younger people that are facing health issues and they need to monitor their health and lifestyle measures. Also, it can support people who have much more simpler goals, such as losing weight, become more physically active, regulate their sleeping habit , etc.

All these scenarios were considered for defining the requirements the next section of the document.

1.2. Requirements

As mentioned in the introduction LifeCoach is meant to support and motivate people who want to improve their health condition and their lifestyle. In order for the application to achieve this goal, it has to fulfill certain requirements:

- User registration and profile creation.
- User authentication. The data registered in the application has to be protected because they are extremely sensible and could be misused by unauthorized individuals.
- The system has to support registering the lifestyle and health measures of users. These measures can be either manual entered by the user or automatically retrieved from sensors.
- The user shall be able to create goals. These goals have to be related to the values of health and life style measures that the user want to achieve. For example "loosing 5kg weight in two months".
- The system has to provide awareness to the user about their goals. The user should be able to see his current goal situation (progress) and estimation for the future. In this way the user will be able to improve and/or adjust his plan.
- The system shall warn the users when they try to define goals that are too risky and or not good for his health.
- Motivations shall be provided to give some kind of moral support to the users, in their pursuit to achieve the goals.
- Users could be supported with suggestion about daily activities in relation with the goals that they have specified.
- The application should provide the functionality to schedule activities and provide reminders when the time comes about these activities.
- The application should provide meal logging functionality. Registering meal consumption is very important as it an activity related with many of the measures, so by consequence also related to the goals.
- The data entered by the user shall be validated. The application has to do validation for wrong format, and "logical" validation. For example, the user can not create two controversial goals for the same measure. Another case could be that the start time of the scheduled activity is not after the end time, etc.
- The application shall have an easy to use and highly visual interface. The application will be used by elderly, so it is very important to minimize the effort of learning the user interface. Also the application shall use graphs, progress bars to make it easier for the user to understand his/her current progress.
- Also system should support all CRUD (Create, read, update, delete) operation about all the conceptual objects used in the application, such as measures, goals, schedules, etc.

Note that the list above is not exhaustive. Only some of the main requirements are mention in the list. Also, not all of the requirements mentioned above are implemented in the application presented in this report. The main focus is given on the service design part.

2. Design

2.1. Conceptual Model

Internal/local storage classes are classes used in LifeCoach schema to manage stored information. The relation between the classes is described in figure 1 below. For each one

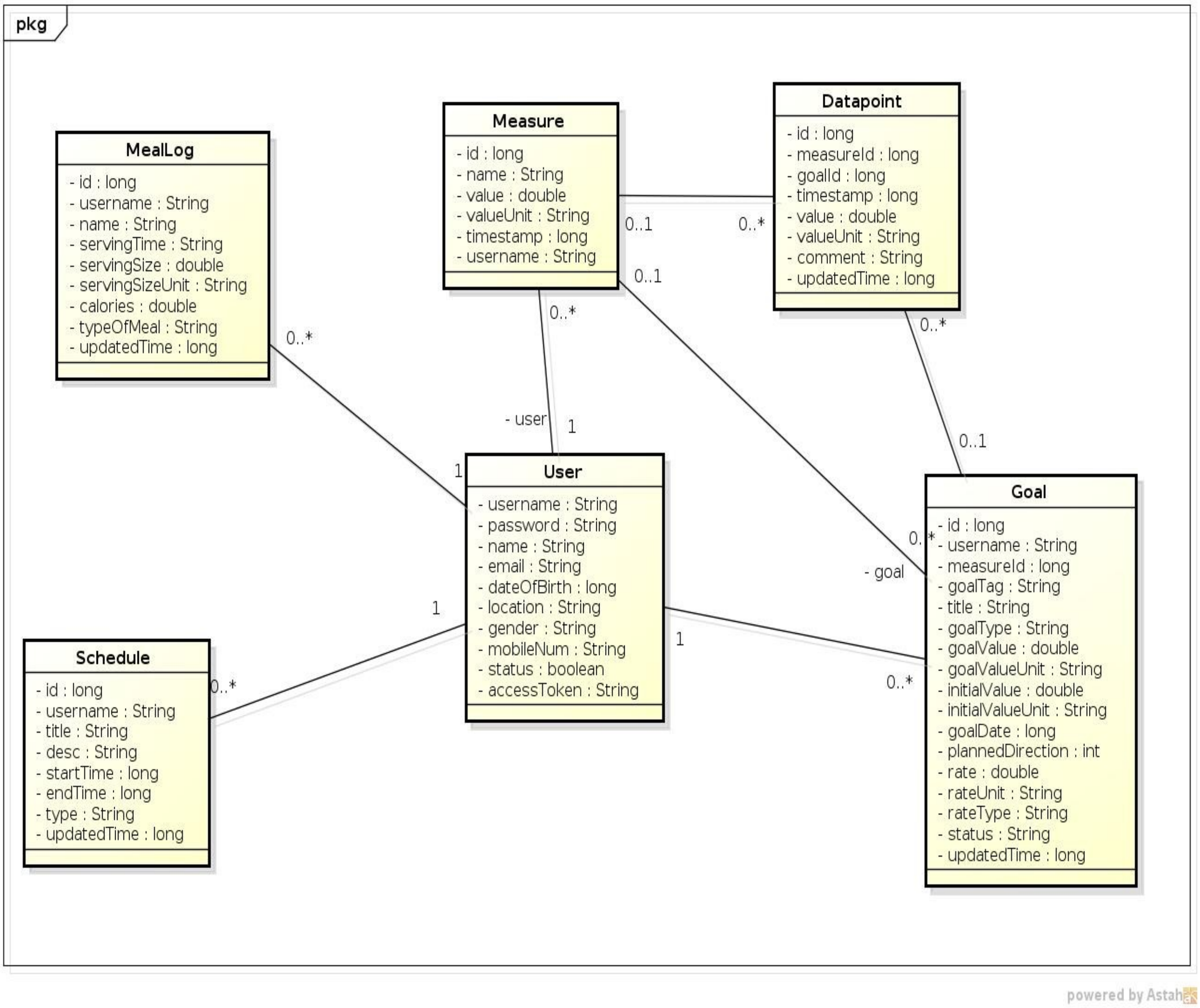


Figure 1 - Internal Storage Classes

these conceptual objects there exists a table in the local storage. A User object is holding the data about a user of the application. Masures represents either a lifestyle or health measure, such as number of steps, weight, hart rate, etc. Goal object contains information about a desired result (value of a measure) that the user want to achieve. Both Measures and Goals are related to Datapoint class. A datapoint represents the value of a measure at a given time. Measures can exist independently from Goals. Object of MealLog class hold information about a meal the user has consumed during the day. Application uses schedule objects to represent planned activities that the user want or has to do in the future.

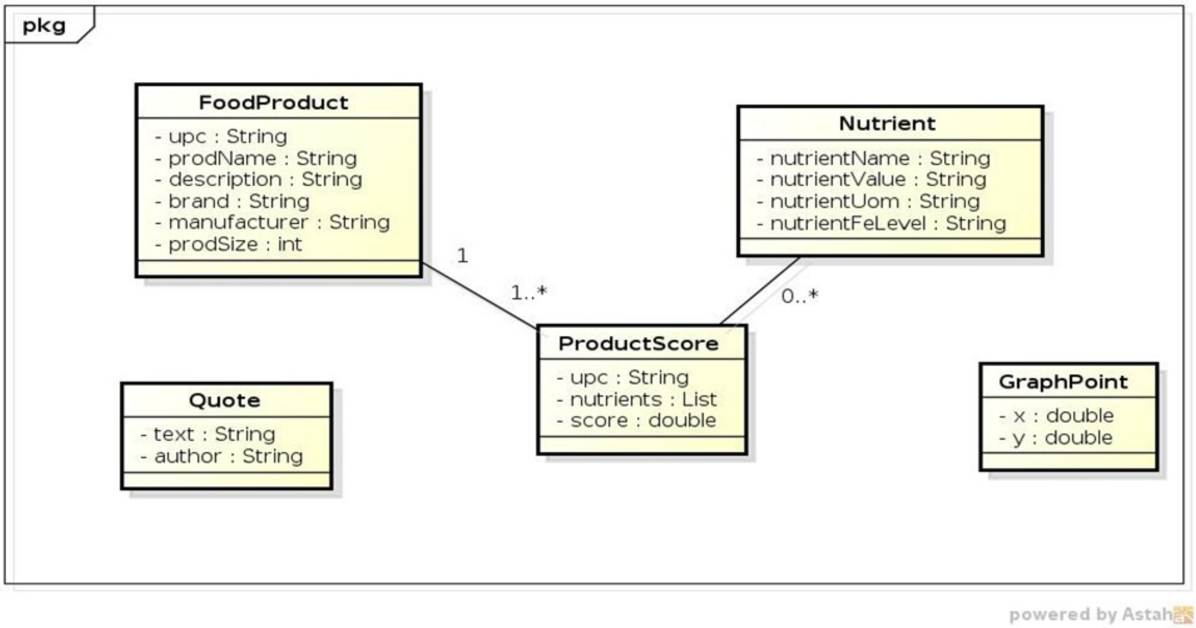


Figure 2 - External Storage Classes

Except GraphPoint , the concepts in the class diagram above are coming from external data sources. FoodProduct , ProductScore and Nutrient are used to hold data coming from FoodEssentials API. Quote is used for the motivational quotes retrieved from the STANDS 4 Quotes API. Finally, objects of the GraphPoint class are used to create list of point, which are later displayed in the graph, using the values of x and y as Cartesian coordinates.

2.2. Architecture

This section describes the architecture of the LifeCoach application. The architecture is divided into four layers. Starting from top down in the figure 3 below there are displayed the User Interface, Process Centric, Logic Centric and Data Centric Layer.

The communication between layers is done using adapters. Adapters are clients to the services on the immediate lower layer. Each service on each layer has its own adapter. An upper layer service uses the corresponding adapter to make a request to a service in the immediate lower layer. This design makes it very easy to use the same adapter multiple times. More importantly it hides the unnecessary details of the implementation of the services. The service in the layer below might be implemented as REST or as SOAP, but this not important for the service making the request as everything is handled by the adapter. If later a service implementation changes, then there is only the need to update the adapter. Otherwise without adapter there would have been the need to update the code in any function the service was called.

In diagram above the Schedule Logic and Food Info Service are marked with a star *. These services are implemented as SOAP RPC services. All the other services are implemented as REST.

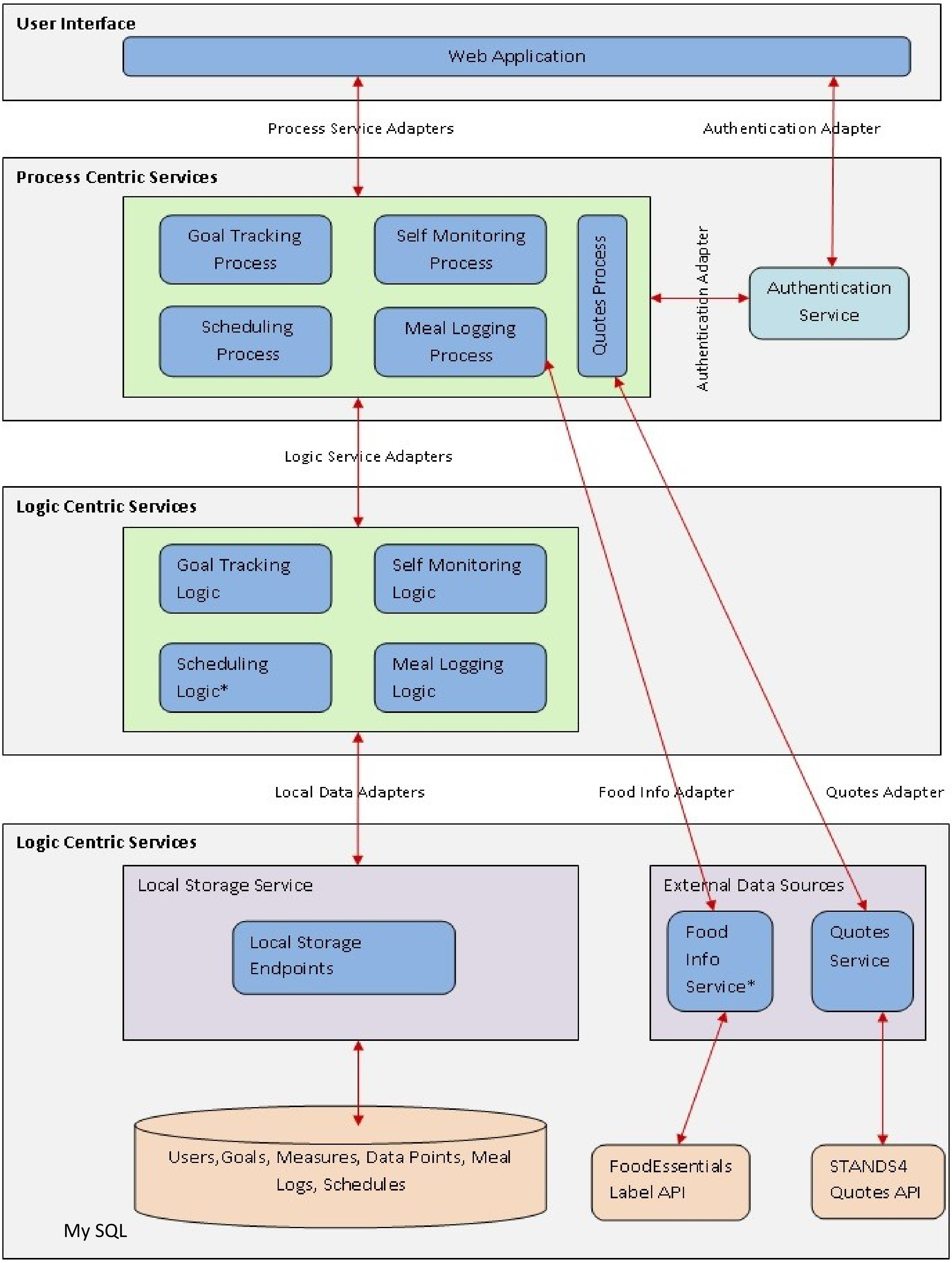


Figure 3 - System Architecture

2.2.1. User Interface layer

The user interface of application is thought as a web application. The interface guides the users through the processes that they want to accomplish, by consuming the endpoints from the services in the process layer. Moreover it provides the feedback from the goals, measure and meal logging to the user in an easy to understand and pleasant way, by using graphs and progress bars.

2.2.2. Process Centric layer

In the process centric layer there are placed all the services that support the process offered to the user, such as Goal Tracking, Self Monitoring, Scheduling, Motivational Quotes, Meal Logging and Authentication. Also in this layer there is an initial validation of the values send via request. All of these services are documented using apiary, an online documentation and mockup system. The reader could visit the documentation of all the process centric services of LifeCoach application by clicking here.

2.2.3. Logic Centric Layer

The logic centric layer is the intermediate layer between the process and the data centric layer. It performs pre-processing and validation of the data coming from the data layer to process layer and vice versa. There is an initial validation at the process centric layer of the values send via the request, but in this layer there is performed a more thorough validation in relation to the process that it is requested. For example, a user could not create a new goal related to some measure if already there exist an active goal about that measure. Another case could be that user cannot save two schedules within the same time interval. Also some of the values which are not provided by the user are generated here, such as updated time. Sometimes the full user object has to be retrieved from the local source and then is attached to the object that has to be persisted.

There is one to one correspondence between the services in the process layer and the ones in this logic layer. Goal Tracking process service calls only Goal Tracking logic, Meal Logging process service calls only Meal Logging logic service, and so on for the other services.

2.2.4. Data Centric Layer

The last layer, contain the data centric services. There are two types of services in this layer: local storage and the external adapters. The local storage service saves the data into the local data source (My SQL database). This is a single service but there are provided different endpoint for each of the local tables (conceptual objects) that used in the application. There are endpoints that support crud operations for User, Goal, Measure, DataPoint, Schedule and MealLog data.

The external adapters retrieve data from external data sources. From the FoodEssentials Label Api , there are retrieved information about food products and meals. For each of the product it is possible to retrieve information about the nutrients, such as calories and fat. Also FoodEssentials returns a food product score, which specifies how good a product is for the user, base on predefined user profile. The data retrieved from this source is first consumed by the Meal Logging process service at the process layer, and then are send to the User Interface. The goal of using this service was to help the users by providing some initial data about the meal they want to log. The user could search and retrieve food information from FoodEssential. The values about the nutrient and serving size of the meal are fetched, and the user could make modification if needed before adding the new log.

The other external adapter is used to retrieve motivational quotes. These quotes are provided by STANDS4 Quotes API. The adapter selects randomly one of the quotes retrieved from the external API and sends it to the Quotes process service, at the process centric layer. This quote is then displayed at the User Interface on most of the pages. The goal is to give some kind of motivation to the users to complete their goals, and continue tracking their measures, meals and schedules. The quote is updated every time the user logs in.

2.3. Modules

The documentation in apiary mentioned above is only about the process layer services and the endpoints they offer. Below it is provided a more general description about the services. This description is not based on the layer of the service but the reader might consider it as a logical horizontal grouping of services from process and logic layer.

Goal Tracking

This group includes Goal Tracking process and Goal Tracking logic services. These services offer all the functionality related to goal definition, monitoring and feedback. The processes handled by Goal Tracking services involve Goal, Measure and DataPoint conceptual objects. This relation it is also described in the section 3.1.

Self Monitoring

Self monitoring services allow the creation and management of measures. They are independent form the goal tracking services. Measures can exist without being part of a goal. The user can update the value of the measure and monitor how this value has changed over the time. These services involve in their operation Measure and DataPoint concepts.

Scheduling

Scheduling services allow defining schedules and getting reminders about them. User could see the schedules he/she has defined for a specific interval and the schedules for the current day. The system monitors and sends reminders to the user for all the schedules that are within the next hour.

Meal Logging

These services Provides the functionality to the users to record their meal consumption during the day. The user could get the meal logs he/she has recorded within an interval (a week) and the logs for the current day. Also, similarly the user could see the sum of calories he/she has consumed for every day within an interval , and the sum of calories for the current day.

2.4. Authentication

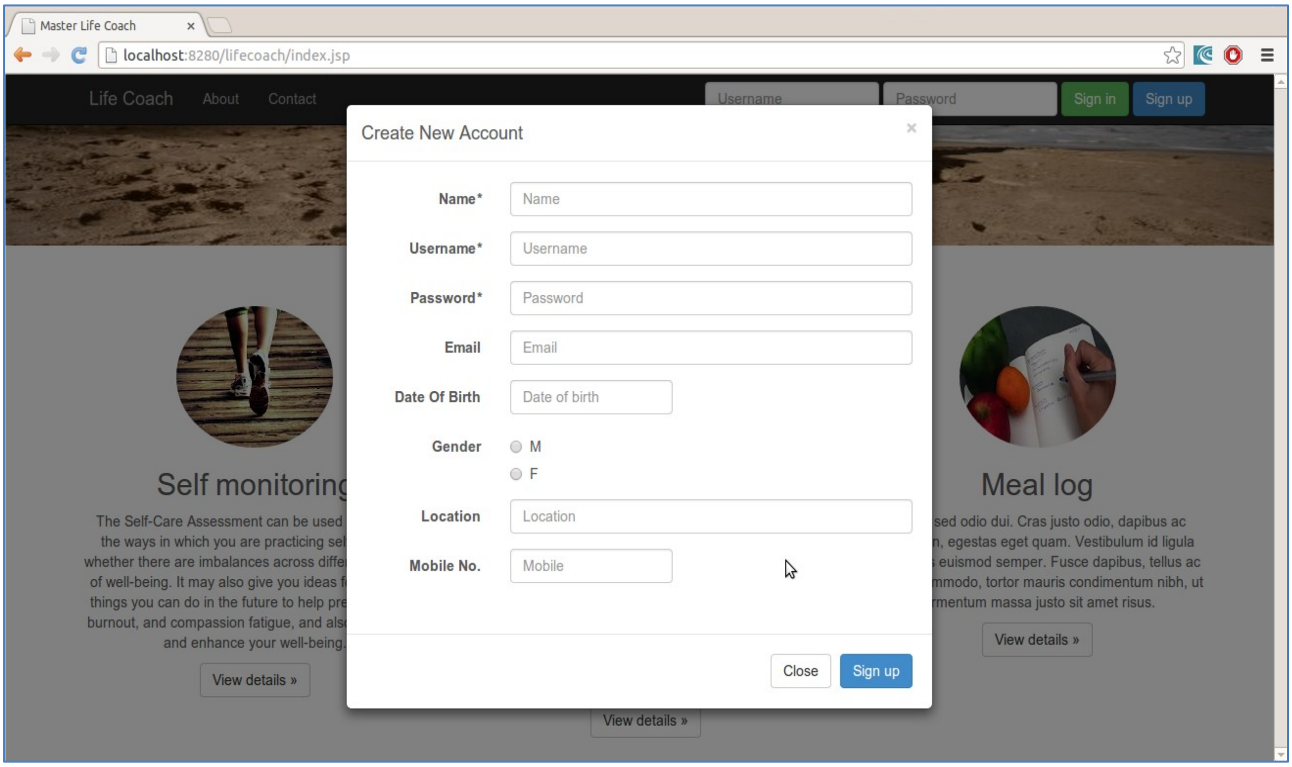
In the diagram figure 3 above the reader might had noticed that the Authentication process is not grouped together with the other process services. This is because this service is consumed both by the user interface and the other process services, in the same layer. After a successful authentication a session is created for the user. The session contains a unique token generated every time the user logs in. This token is contained in every call from the user interface to the endpoints of the process layer services. For each call the token is validated to the Authentication service before proceeding with the request.

4. Implementation

3.1. User interface

Below there are listed some screen shots from the implemented user interface of the LifeCoach application.

Sign Up



The Self-Care Assessment can be used
the ways in which you are practicing sel
whether there are imbalances across diffe
of well-being. It may also give you ideas f
things you can do in the future to help pre
burnout, and compassion fatigue, and als
and enhance your well-being.

sed odio dui. Cras justo odio, dapibus ac
n, egestas eget quam. Vestibulum id ligula
euismod semper. Fusce dapibus, tellus ac
mmodo, tortor mauris condimentum nibh, ut
rmentum massa justo sit amet risus.

Figure 4 - Sign Up Form

Sign up screen is used to create a new user. Name, username and password are the required fields during sign up.

Login

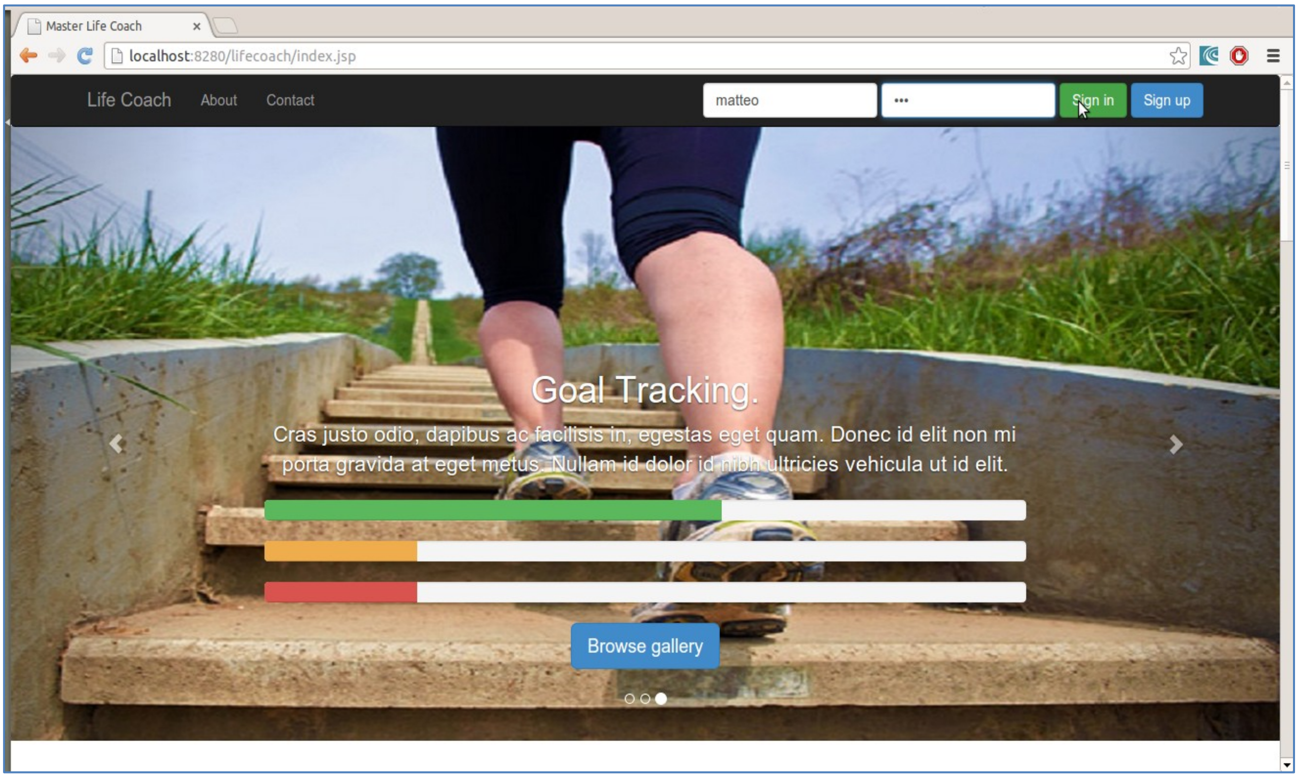
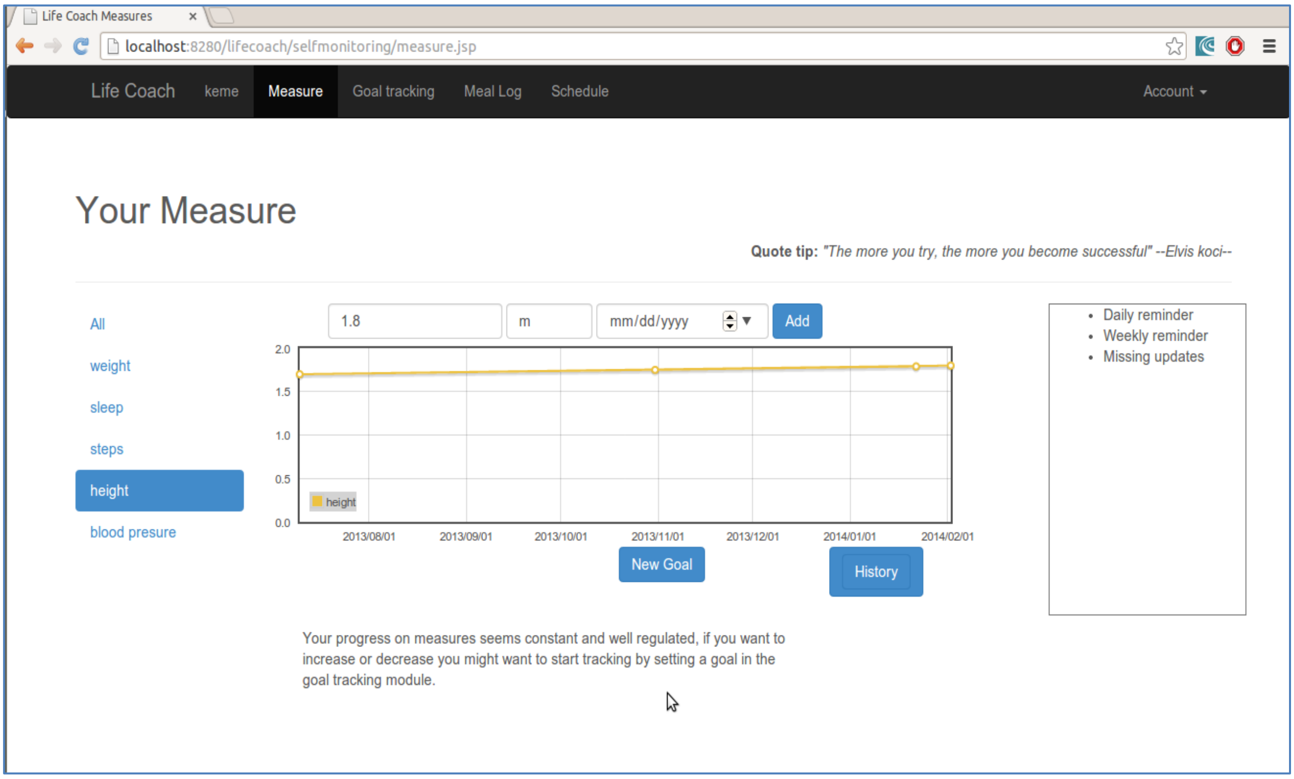

Master Life Coach
localhost:8280/lifecoach/index.jsp
Life Coach
About
Contact
matteo
•••
Sign in
Sign up
Goal Tracking.
Cras justo odio, dapibus ac facilisis in, egestas eget quam. Donec id elit non mi porta gravida at eget metus. Nullam id dolor id nibh ultricies vehicula ut id elit.
Browse gallery

Figure 5 - Login Form

Login screen is located in the main page (home page). While the user logs in the access token is generated and returned with user information. If successful with username and password, user access token and username will be stored to the http session which will be used for every call now on for this user.

Measure

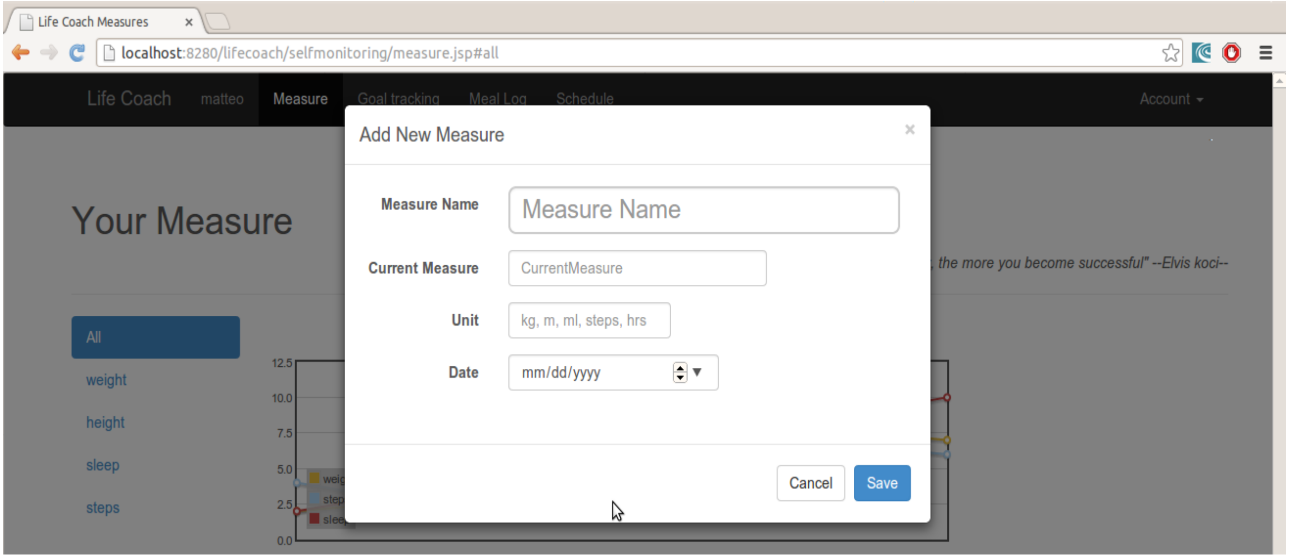


Quote tip: *"The more you try, the more you become successful" --Elvis koci--*

Your progress on measures seems constant and well regulated, if you want to increase or decrease you might want to start tracking by setting a goal in the goal tracking module.

Figure 6 – Measure module

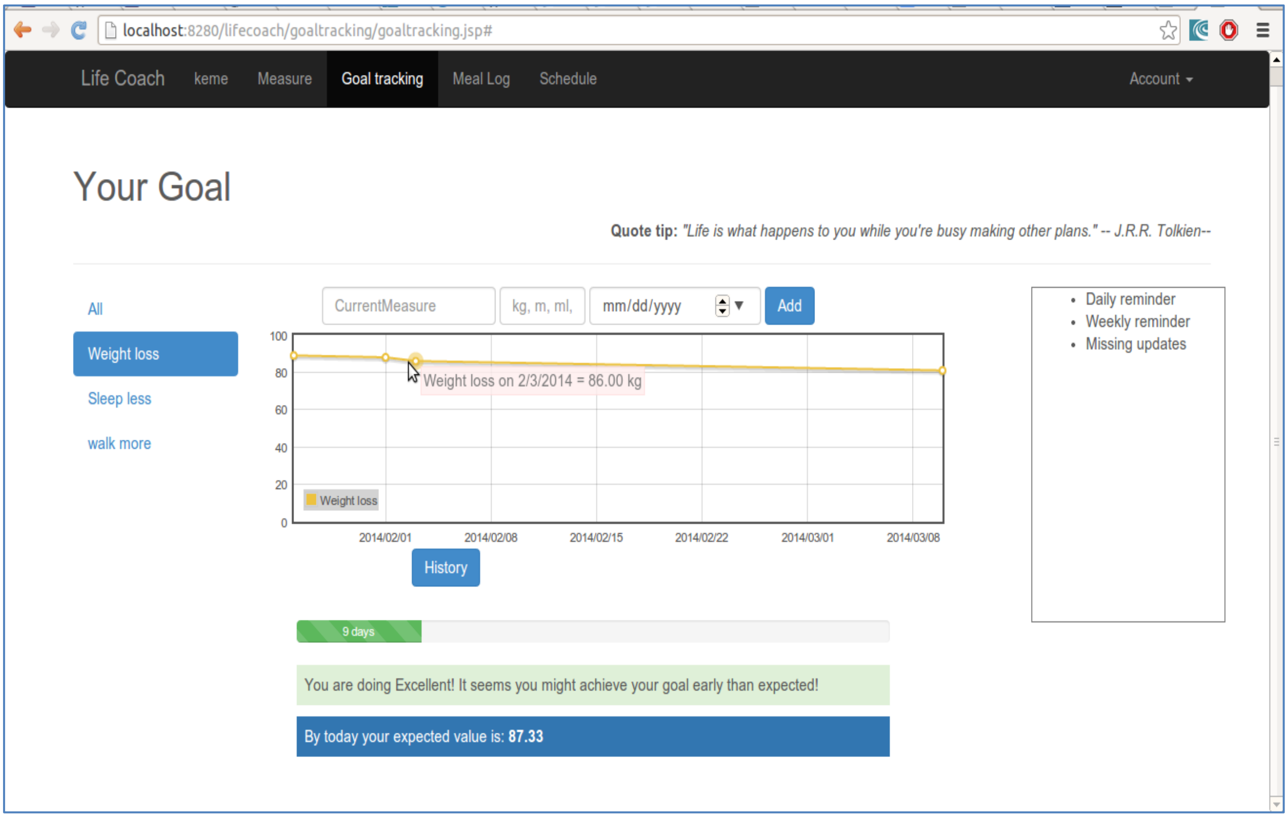
Measures created by the user whenever she/he wants. This means there is no predefined measures for each user, there are just suggestions. On the left panel user's measures are populated based on users' information on measures dynamically. The graph (time, measure value) displays the progress/history of measure from the first time measure is added to current measure. History of measure can also be viewed as raw table by clicking on History button.



the more you become successful" --Elvis koci--

Figure 7 - Add new measure

Goal tracking



Your Goal

Quote tip: *"Life is what happens to you while you're busy making other plans." -- J.R.R. Tolkien--*

You are doing Excellent! It seems you might achieve your goal early than expected!

By today your expected value is: **87.33**

Figure 8 - Goal Tracking Module

Goal tracking module is used to add new goal, manage and get feedback on goals. The same as measures, on the left panel the user interface shows links to all active goals which are currently active to be tracked. The system manages to check whether the goal is achieved, lost, canceled, or still active based on the goal date and final data point value. The goals can be associated with a measure or it can be without a measure. If it is linked with measure information, the goal graph and history shows only after the goal is set, but the measure history in the previous interface shows all history from measure first instantiated.

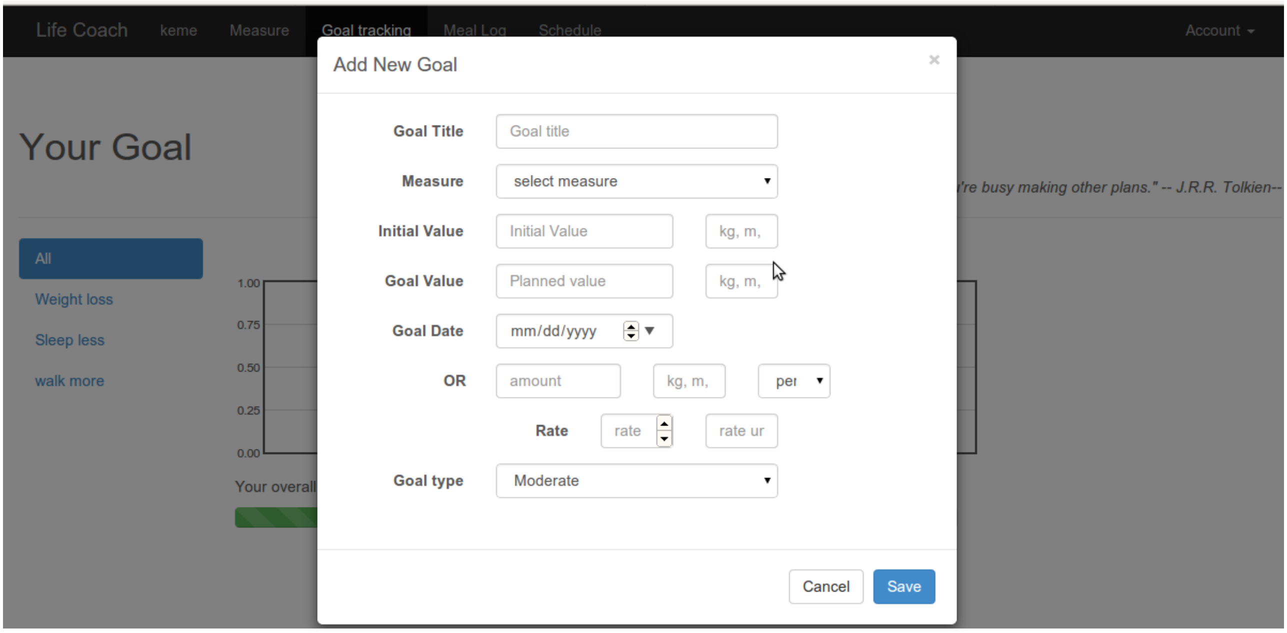
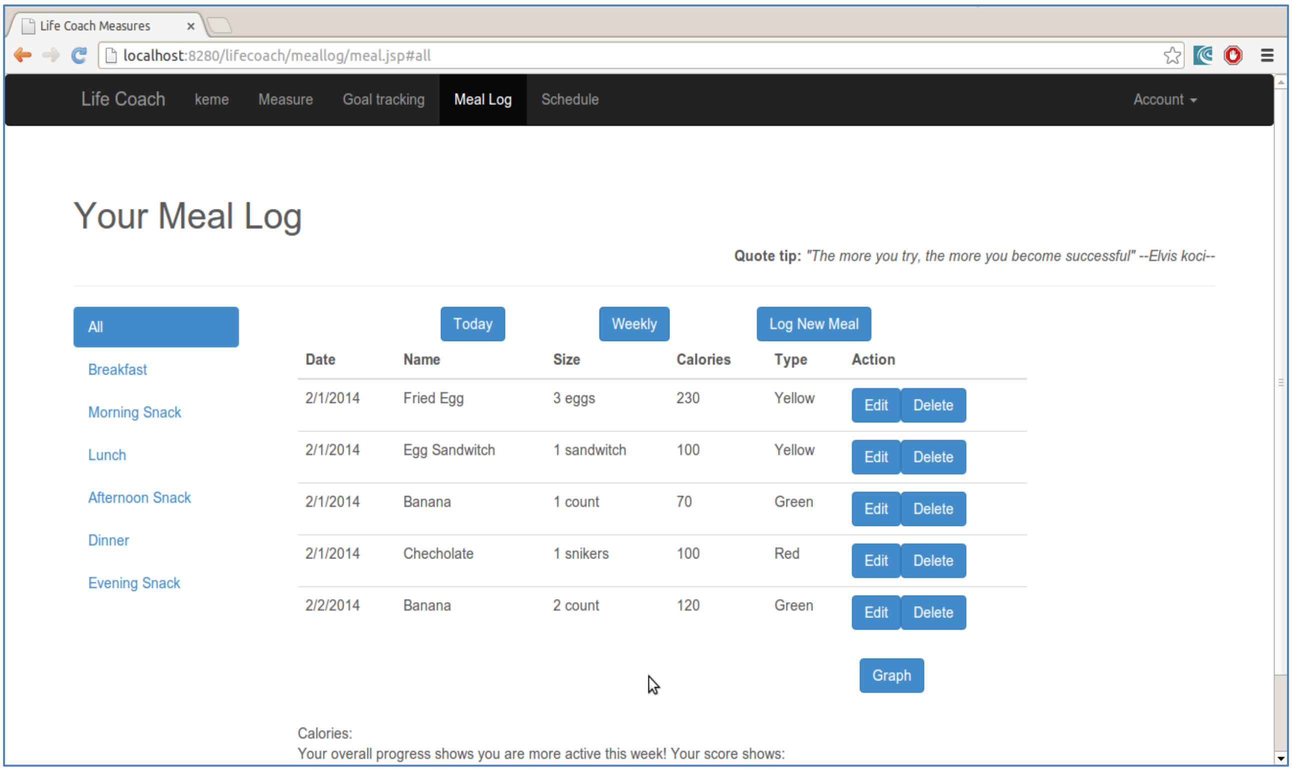


Figure 9 - Add new goal

Meal Log



Quote tip: *"The more you try, the more you become successful" --Elvis koci--*

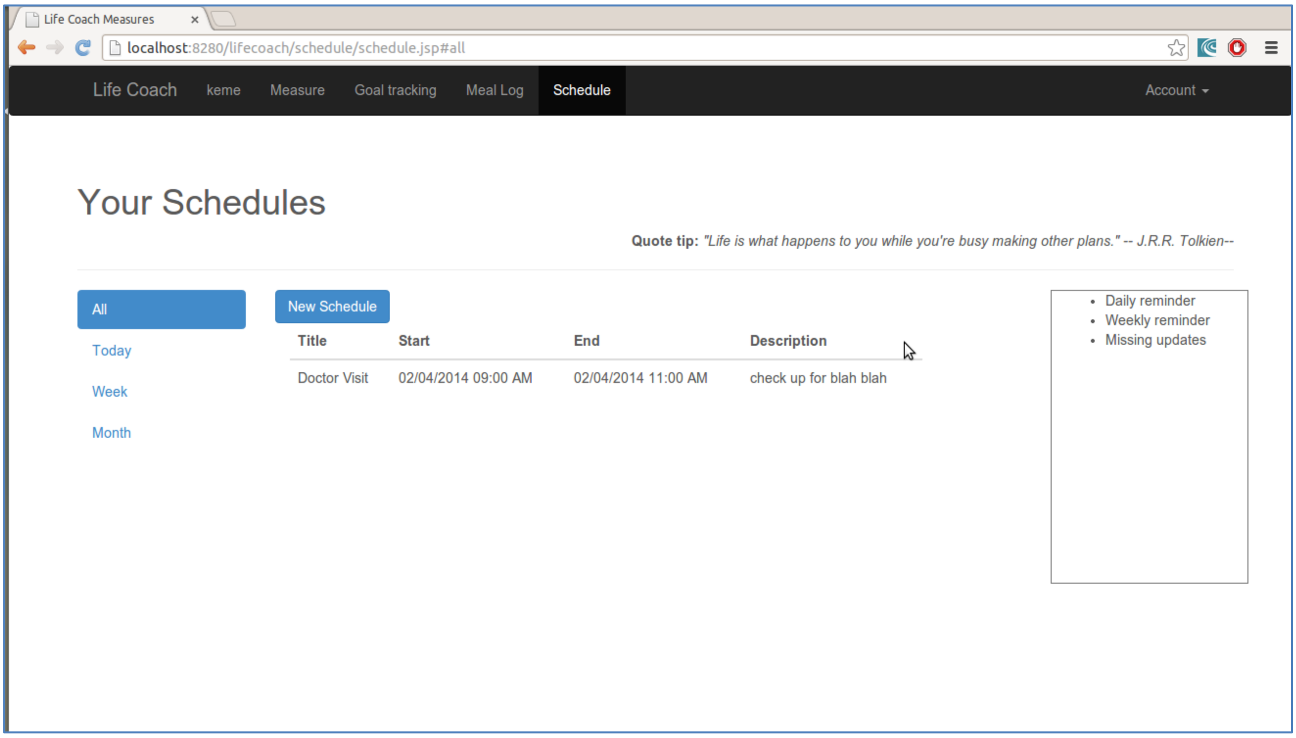
Date	Name	Size	Calories	Type	Action
2/1/2014	Fried Egg	3 eggs	230	Yellow	Edit Delete
2/1/2014	Egg Sandwitch	1 sandwitch	100	Yellow	Edit Delete
2/1/2014	Banana	1 count	70	Green	Edit Delete
2/1/2014	Checholate	1 snikers	100	Red	Edit Delete
2/2/2014	Banana	2 count	120	Green	Edit Delete

Calories:
Your overall progress shows you are more active this week! Your score shows:

Figure 10 - MealLog Module

Meal log interface is used to track daily meal calories and everyday meal information. While logging a new meal for a serving time, such as breakfast, morning snack, lunch, etc., the user interface helps the user to add a meal with calories information automatically from Food Essentials external adapter service.

Schedule



Your Schedules

Quote tip: *"Life is what happens to you while you're busy making other plans." -- J.R.R. Tolkien--*

Title	Start	End	Description
Doctor Visit	02/04/2014 09:00 AM	02/04/2014 11:00 AM	check up for blah blah

- Daily reminder
- Weekly reminder
- Missing updates

Figure 11 - Schedule

Schedule interface is used to add new schedules and view schedule information. For now it’s just a mock.

5. Conclusions

In conclusion, life coaching it is a complex problem to solve. Many elements have to be integrated in order to build a successful application. The user need store data and get feedback about those data. Also very important they user need support via suggestions and motivations. The application could include even more features than these.

As developers we learned that it is very difficult to build every part of this application from scratch. Reusing existing services that have covered some of the requirement makes the implementation potentially much easier and faster. On the other hand these services have to come from reliable sources. Finally, we tried to use these concepts in design of the LifeCoach application, by making the different parts of the application reusable. We tried to minimize the dependency by hiding the implementation details of each part and using adapter for the communication between them.