

**Seventh Framework Programme
CallFP7-ICT-2013-10**

Project Acronym: **SPLendid**
 Grant Agreement N°: **610746**
 Project Type: **COLLABORATIVE PROJECT: Small or medium scale focused research project (STREP)**
 Project Full Title: **Personalized Guide for Eating and Activity Behaviour for the Prevention of Obesity and Eating Disorders**

D4.1 Mock-up version of User Interaction Modules

Nature:	D(R: Report, P: Prototype, D: Demonstrator, O: Other)
Dissemination Level:	PU (CO: Confidential, PU: Public)
Version #:	1.0
Date:	29/09/2014
WP number and Title:	WP4 Personalized Guidance System
Deliverable Leader:	TSB
Author(s)/Contributor(s):	Authors: Isabel Martí, Juan Pablo Lázaro (TSB), Christos Maramis (AUTH) Contributors: Ioannis Moulos, Alexandros Mourouzis (AUTH), Juan Bautista Mocholí (TSB)
Status:	Submitted (Draft, Peer-Reviewed, Submitted, Approved)

Document History

Version ¹	Issue Date	Status ²	Content and changes
0.1	25/7/2014	DRAFT	Table of contents and some contents in section 1.
0.2	03/09/2014	DRAFT	Contents of section 1
0.3	08/09/2014	DRAFT	Contents of section 2 and Annex A
0.4	10/09/2014	DRAFT	Revisions of section 1 from AUTH
0.6	22/09/2014	DRAFT	AUTH contributions
0.7	23/09/2014	DRAFT	Sections 4, 5 and Executive Summary
0.8	25/09/2014	DRAFT	Peer review based corrections
1.0	29/09/2014	Submitted	Final version after peer review corrections

Peer Review History³

Version	Peer Review Date	Reviewed By
0.7	24/09/2014	Anna Anvret

¹ Please use a new number for each new version of the deliverable. Use "0.#" for Draft and Peer-Reviewed. "x.#" for Submitted and Approved", where x>=1. Add the date when this version was issued and list the items that have been added or changed.

² A deliverable can be in one of these stages: Draft, Peer-Reviewed, Submitted and Approved.

³ Only for deliverables that have to be peer-reviewed

Table of contents

DOCUMENT HISTORY	2
TABLE OF CONTENTS	3
ABBREVIATIONS, ACRONYMS AND NOMENCLATURE	4
EXECUTIVE SUMMARY	5
1 INTRODUCTION	6
1.1 PURPOSE OF THE DOCUMENT	6
1.2 METHODOLOGY.....	6
1.3 INTENDED AUDIENCE AND READING SUGGESTIONS.....	9
1.4 WRITING CONVENTIONS.....	10
1.5 OVERVIEW OF THE DOCUMENT	10
2 MOCK-UP'S OF PROFESSIONAL'S WEB TOOL (PWT)	11
2.1 USE CASES IMPLEMENTED IN MOCK-UPS.....	11
2.2 MOCK-UPS.....	13
3 MOCK-UPS OF MOBILE PERSONAL GUIDANCE APPLICATION	30
3.1 USE CASES IMPLEMENTED IN MOCK-UPS.....	30
3.2 MOCK-UPS.....	32
4 RUNNING INSTANCES	47
4.1 PROFESSIONAL'S WEB TOOL (PWT).....	47
4.2 MOBILE PERSONAL GUIDANCE APPLICATION	51
5 NEXT STEPS	52
ANNEX A. CHANGE IN BASE TECHNOLOGY FOR PWT	53
CONTENT MANAGEMENT SYSTEMS.....	53
WEB FRAMEWORK	55
CMS VS. FRAMEWORK.....	57
CHOOSING TECHNOLOGY FOR SPLENDID.....	58
ANNEX B	61

Abbreviations, Acronyms and Nomenclature

List of Abbreviations and Acronyms

Primary Screening Stage - PSS

Behavioural Assessment Stage – BAS

Personalized Guidance Stage – PGS

Primary Screening Event - PSE

Behavioural Assessment Event – BAE

Personalized Guidance Event – PGE

Professional Web Tool – PWT

Personal Guidance Application (end user’s mobile app) – PGA

Virtual Guidance Agent – VGA

Document of work – DoW

Health Professional – HP

Decision Support System - DSS

Executive Summary

SPLENDID aims at providing personalized lifestyle guidance services helping individuals to adopt healthy eating behaviours and activity. In order to provide that service, SPLENDID will develop a set of applications, aimed at helping end-users achieve those healthy behaviours and the health professionals to provide guidance to the users.

The *Mock-up version of User Interaction Modules* deliverable covers the work done in WP4 during the first 12 months of the project developing the prototypes of the applications. More concretely, a mock-up is *a prototype of the designed system used by the designer to test the product with the final users before it's in production*. This document provides information about how those mock-ups have been produced in tasks 4.1, “Professional Application”, and 4.2 “Personal Guidance Application”.

During this first period, the main purpose of those tasks is to provide a set of mock-ups presenting the visualization of the different use cases defined in deliverable 1.2. Section 3 of this document presents the set of mock-ups for the Professional Application and the Personal Guidance Application.

The work performed by the development groups in both T4.1 and T4.2 as gone beyond the definition of the mock-up screen. Both groups have already developed functional versions of some of the use cases. These functional versions are presented in section 3 of this document.

The partners of SPLENDID can use this document to know the current state of the applications. To the external reader, is a sample of what the SPLENDID applications will offer in the near future.

1 Introduction

The chapter aims to introduce the reader to the content of this deliverable, the way the content has been prepared and the intended use of the included material. We start by stating the purpose of this document and then present the methodology that has been followed to produce the herein provided material. Next, we specify the intended audience of this document and provide reading suggestions. Writing conventions follow next and the chapter concludes with an overview of the subsequent chapters.

1.1 Purpose of the document

This document presents deliverable D4.1, which is the first deliverable of Work Package 4 (WP4). This work package, entitled “Personalised Guidance System” is charged with the design and development of the software applications that constitute the Personalized Guidance System of SPLENDID. More specifically, the target software applications include the Professional Web Tool (PWT), the Personal Guidance Application (PGA) and the “Virtual Guidance Agent” (VGA), each undertaken by a separate task of the work package (T4.1 to T4.3, respectively). All tasks were initiated in Month 5 and will be complete at the end of the project.

As indicated by its title (i.e., “Mock-up version of user interaction modules”), the main purpose of this deliverable is to present the design of the user interfaces to be developed in the form of mock-up versions of the user interfaces of the Personal Guidance System. More specifically, the scope of this document is to:

1. Present the mock-ups corresponding to the use cases of the PWT.
2. Present the mock-ups corresponding to the use cases of the PGA.

In accordance with the DoW of SPLENDID, deliverable D4.1 will not deal with the mock-ups of the VGA. Elaborating on this, the user interface of VGA aims to enhance that of PGA by adding a short list of elements with underlying intelligence to optimize the guidance efficiency of the guidance process. Given the above statement, it is easy to conclude that i) the user interface of VGA will heavily rely on that of PGA, and ii) some of the aforementioned enhancements may not be reflected in the - inevitable static – presentation of the mock-ups that is attempted by this deliverable. Thus, it becomes clear that an attempt to present mock-ups for the VGA at this point would be premature.

It is worth pointing out that, although the scope of this deliverable only covers mock-up versions of the UI (this means, “drawings” of the screens), the deliverable also describes the progress made on software implemented so far for both the PGA and the PWT. This has been decided by the involved partners of SPLENDID due to the proximity of the release of the first prototype (Month 14), in order to be used as development milestone.

1.2 Methodology

The methodology used in order to design the user interfaces has been based on mock-ups. A mock-up is a *prototype of the designed system used by the designer to test the product with the final users before it's in production*. This way, the designers are able to acquire feedback from the final users in the early stages of the design process.

The user, aided by the designer, may test the mock-up (imagining that it works) and thus provide valuable feedback about functionality/usability/understanding of the basic design idea, etc.

According to the Interaction Design Foundation⁴, there are several advantages in the use of mock-ups when designing a new product:

- Mock-ups are supposed to be low-cost and not very elaborate. This way it's easier for the user to criticize your system, since some users can feel reluctant to criticize a very elaborate work.
- Mock-ups are a discussion medium and a discussion facilitator between the designer and the user.
- Mock-ups make it possible to do usability testing early in the development process.
- Mock-ups incite and legalize experimentation as they are inexpensive to alter.
- Mock-ups focus on content and functionality and turn attention away from details of graphic design.
- Mock-ups address the idea captured in a popular engineering one-liner: "You can fix it now on the drafting board with an eraser or you can fix it later on the construction site with a sledge hammer"

For all these reasons, a series of mock-ups were designed based on the use cases described in deliverable 1.2⁵, for both the PWT and the PGA. Then, the mock-ups refined in several iterations based on the opinion of the SPLENDID consortium and the participating or affiliated health professionals (comprising the potential future end users of PWT).

Although PWT and PGA are being developed by different partners/teams (TSB and AUTH respectively), it has been deemed beneficial for both teams to follow the same design process. This common design process is described below.

1. Initially, the use cases of the system, described in deliverable D1.2 ("Use Cases Specifications"), were carefully studied.
2. Based on the use cases, and on the behavioral goal setting meeting that took place on January 2014, each team prepared a first version of the screens for each use case. It should be mentioned that there is not a one-to-one mapping between use cases and screens; use cases describe a process, and one use case normally involves several screens, in the same way that some screens appear in several use cases. The result of this step was two sets of screens (one for the PWT and one for the PGA) that describe the possible ways the user can interact with the applications in order to follow different work flows described in the use cases.
3. The first versions of the mock-ups were presented to the SPLENDID consortium during the 1st plenary meeting of the project (April 2014, Neuchâtel, Switzerland) and

⁴ Interaction Design Foundation - <http://www.interaction-design.org/encyclopedia/mock-ups.html>

⁵ D1.2 Use Cases Specifications is a confidential deliverable for the consortium, submitted to the commission on February 2014

-
- feedback on how well the screens fitted the use cases and the professionals' expectations was collected.
4. Based on the aforementioned feedback, a second version of the screens was designed and presented in a workshop with health professionals at Mando (Stockholm, Sweden). Given the amount of information to be processed and the need of receiving meaningful feedback from the professionals, two different workshops were set up:
 - a. Workshop with health professionals to evaluate the PWT:
 - i. **Dates:** June 9th of 2014
 - ii. **Information about attendees:** 12 people attended. The participants were case managers, head clinicians, dieticians, designers and researchers and post-doc students.
 - iii. **Main methodology:** A short introduction about the project and the different stages followed by a revision of the different pages of the use cases of Behavioral Assessment Stage. During the revision of each screen, the conductor induces a discussion of the screen.
 - iv. **Summary of most relevant feedback:** In general lines, the professionals were happy with the screens and the flows between them. There was some discussion about the granularity of the graphs and more visible way to show the indicators. There were no major changes in layouts, schemas or components suggested.
 - b. Workshop with health professionals to evaluate the PGA:
 - i. **Dates:** June 16th, 2014
 - ii. **Information about attendees:** 14 people attended. The participants included clinicians, dieticians, case managers, graphic & web designers, and neuroendocrinology researchers.
 - iii. **Main methodology:** First, the participants were shortly introduced into the project and its different stages. Then, the sequences of mock-up screens that complete 4 PGA use cases of the Behavioral Assessment Stage of the PGA were presented one by one. During the mock-up screen presentation, the attendees were asked specific questions and encouraged to express their opinion on various aspects of the mock-ups in order to facilitate the feedback collection process.
 - iv. **Summary of most relevant feedback:** In general, the mock-up screens of the PGA were well received; the information flow and screen transition was considered satisfactory. However, some concerns were raised about the feedback provided by the application to the end user (it was considered more than needed for the Behavioral Assessment Stage); moreover, it was pointed out that icons used by the application should be revised.
 5. A third version of the mock-up screens was developed based on the feedback from the health professionals. This version is the one presented in this deliverable. However, this is not the final step of the design process: The revised mock-up screens will be reviewed by another group of health professionals (for the PWT) and individual end

users (for the PGA). This event involves interviews with health professionals and a focus group of individual end users and planned to take place at Wageningen during October 2014. Given that the mock-ups have already been examined by health professionals, we do not expect important changes resulting from this event.

6. The final version of the mock-ups, which will be the one implemented in the software applications, will be designed considering the findings from the Wageningen review events.

The described design process ensures that the final solution will satisfy the two main targeted groups of users: Students in a school environment (Sweden) and young overweight adults (The Netherlands). Between the third and the final version only minor adjustments are expected. As already mentioned, this deliverable presents the third version which is considered to be almost final.

The following picture shows a diagram of the described process. The events from where the designs were extracted are represented in blue, while the different versions of the mock-ups are represented in orange.

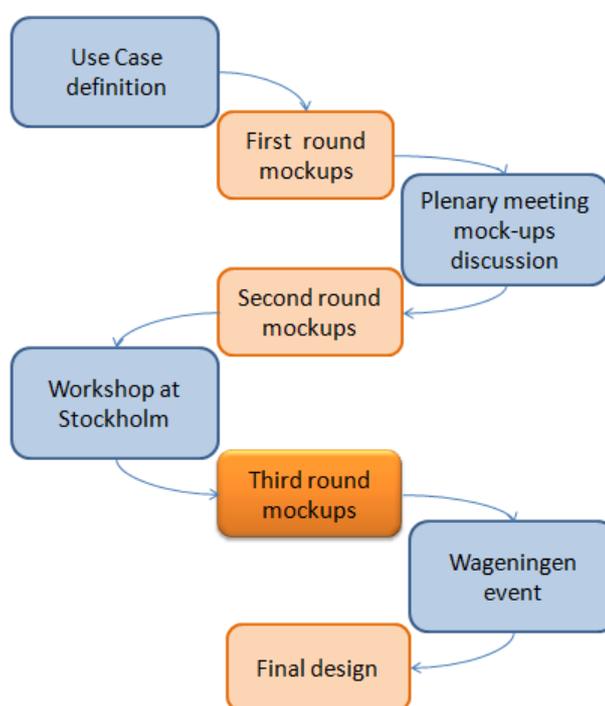


Figure 1 Flow diagram outlining the step of the design process for the mock-ups of the SPLendid applications. The third round of mock-ups appears emphasize since is the one presented in this document.

1.3 Intended audience and reading suggestions

The present deliverable has been classified as public, so it will be published beyond the consortium for anyone who wants to read it. However, the main audience of this deliverable are developers involved in Work Package 4 who need to take the information of this deliverable in order to develop the user interfaces as expected.

This is not a technical document, so it will be easily readable by non-technical readers. Also, although the applications are related and some use cases are better understood by watching both the professional and end user applications, the PWT screens and the PGA screens can be reviewed separately.

1.4 Writing Conventions

Since this deliverable is mostly visual, with the mock-ups of the different use cases, we agreed to provide a short description with each mock-up. Long descriptions with all fields of each mock-up have been avoided since they can be consulted in D1.2, where the use cases are described in detail.

1.5 Overview of the document

Chapters 2 and 3 constitute the main body of the document and include the mock-ups of PWT and PGA respectively. Chapter 4 presents the current working versions of the software applications under development and their scope. Chapter 5 describes the future work to be done in order to finalize the user interfaces. Annex A explains the decision to change the initially chosen development technology for PWT. Finally, Annex B provides additional information about the workshops that took place at Stockholm.

2 Mock-up's of Professional's Web Tool (PWT)

According to the DoW, the PWT is based in the Nomhad web application own by TSB.

The mock-ups have been developed based on the look and feel of Nomhad, using Adobe Illustrator and Adobe Photoshop. That way, the Nomhad elements have been arranged in a convenient way in order to fill the need of the different use cases.

In order to implement the mock-ups, the first step was to elaborate the site map of the application; this is the different screens that compose the web with the flows between them. Figure 3 shows the site map with all the web components.

During month 10 of the project (July), TSB proposed a change the base technology used in the development of the PWT, because of a number of reasons related to strategic decisions of the company TSB and as a consequence of the analysis of SPLENDID features to be implemented. The rest of the consortium understood the proposal and agreed in implementing such a change, given that the impact was perceived as low. Annex A explains the decision of changing the technology used for the development of the PWT.

In the new solution screens will maintain the layouts, components, and their position in the web browser, schematics, etc, however they will probably have a slightly different look and feel in the final implementation compared to the mock ups presented in the current document.

2.1 Use cases implemented in mock-ups

In order to accomplish Milestone 4 (First prototype) due for month 14, the mock-ups for every Use Case have been created. Although all of them were reviewed by the consortium in the plenary meeting, only the use cases corresponding to the BAS were reviewed at the Stockholm workshop due to lack of time to review them all. The BAS use cases were selected because we consider that the most important screens are those showing the information about user's meals and activities and these pages will look alike in all stages.

It is worth mentioning that there is not one to one mapping of screens with use cases, so we will point out how use cases and the different screens are related during the description of the mock-ups.

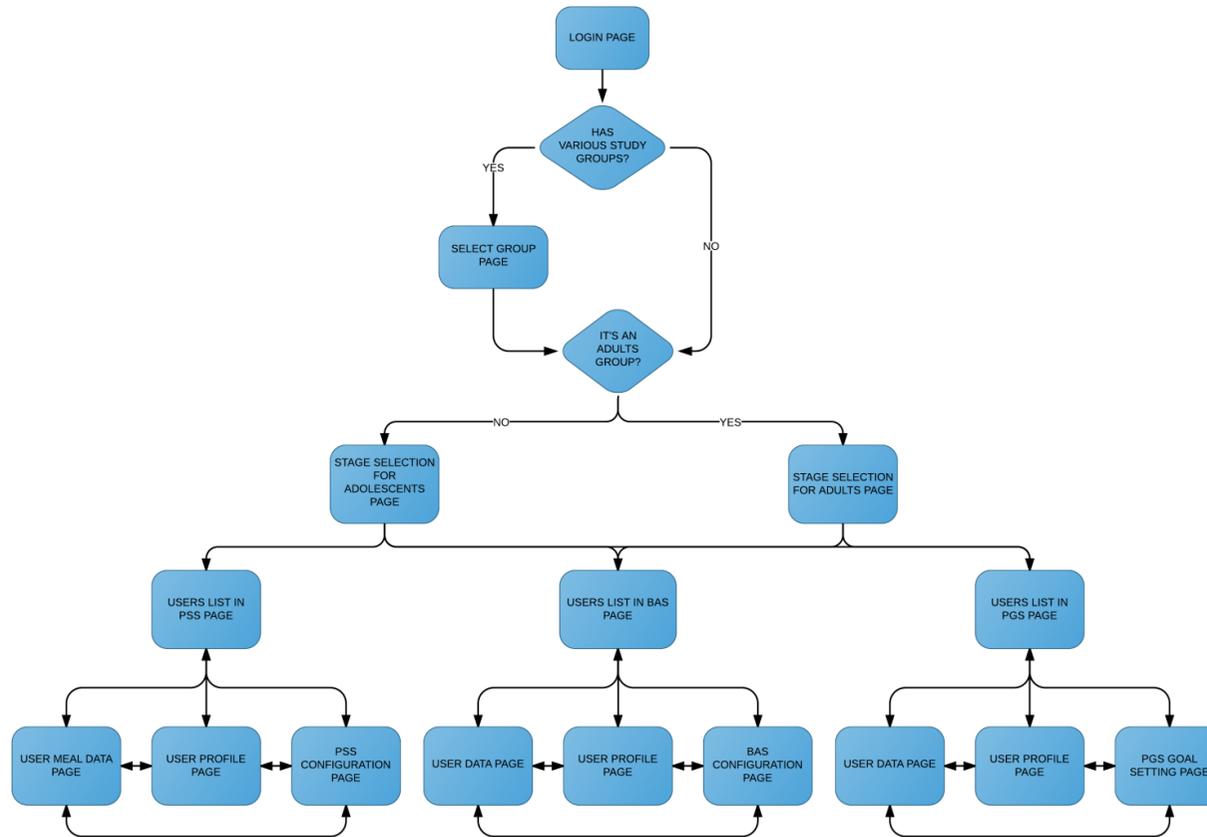


Figure 2 Site map for the PWT

2.2 Mock-ups

2.2.1 Use Case 1: Login to Web Interface

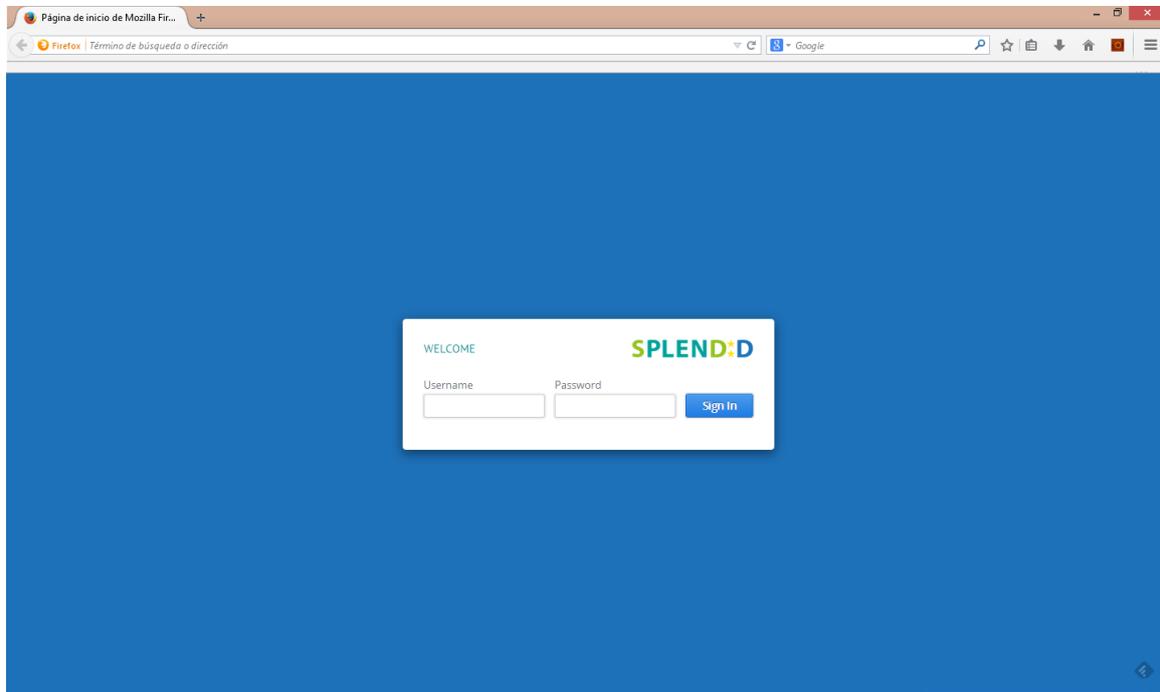


Figure 3 Login screen

When a HP wants to access the PWT, she has to go to the login page and introduce username and password. The next screen visible will depend on how many groups of user's belong to that HP or the HP has view privileges.

If she only has more than one group, she will have to choose first the group to be reviewed and then select the stage. Otherwise she'll be directly taken to the stage selection screen (Figures 4, 5 and 6). In the case the HP doesn't have any groups; she can create a new Primary Screening Stage or a new Behavioural Assessment by clicking on the buttons shown in Figure 4.

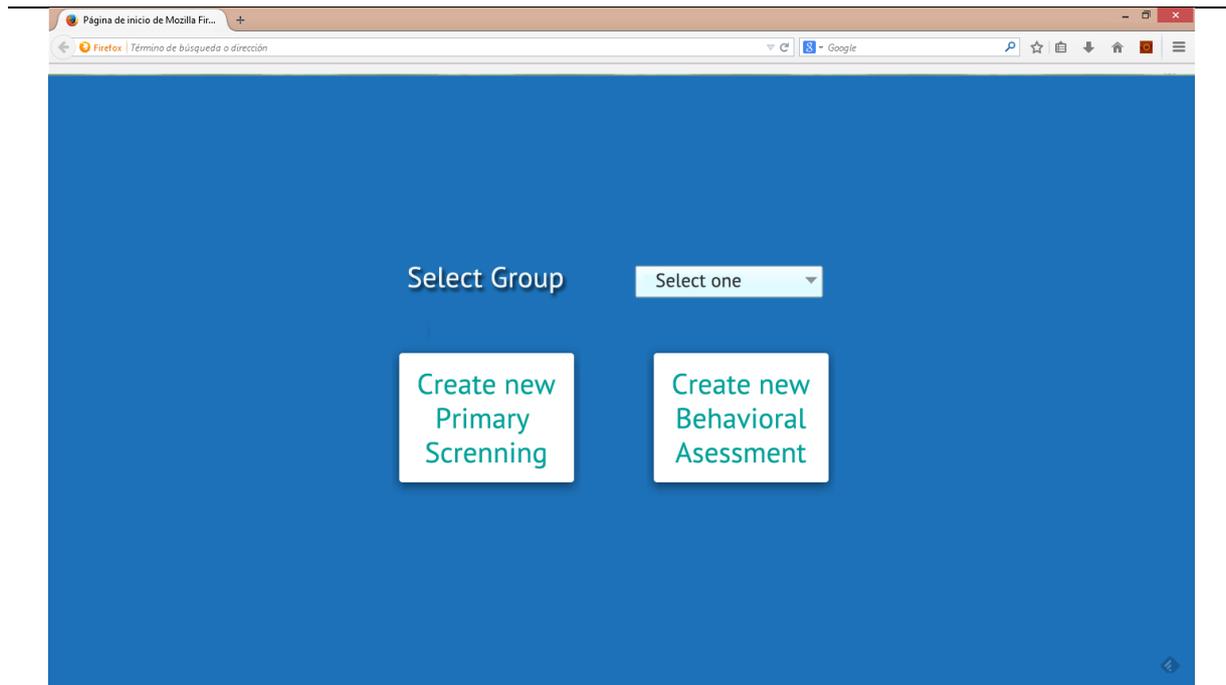


Figure 4 Selection of study group

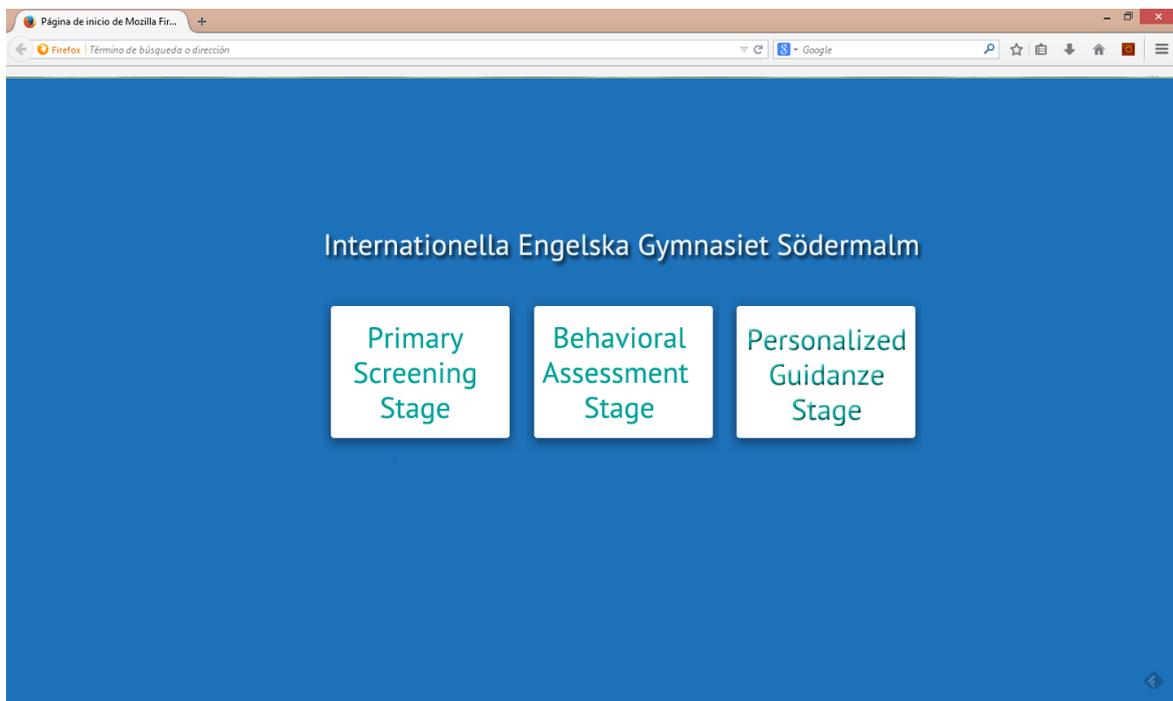


Figure 5 Stage selection screen for adolescents study group



Figure 6 Stage selection screen for adults study group

2.2.2 Use Case 2: Register Assistant Account

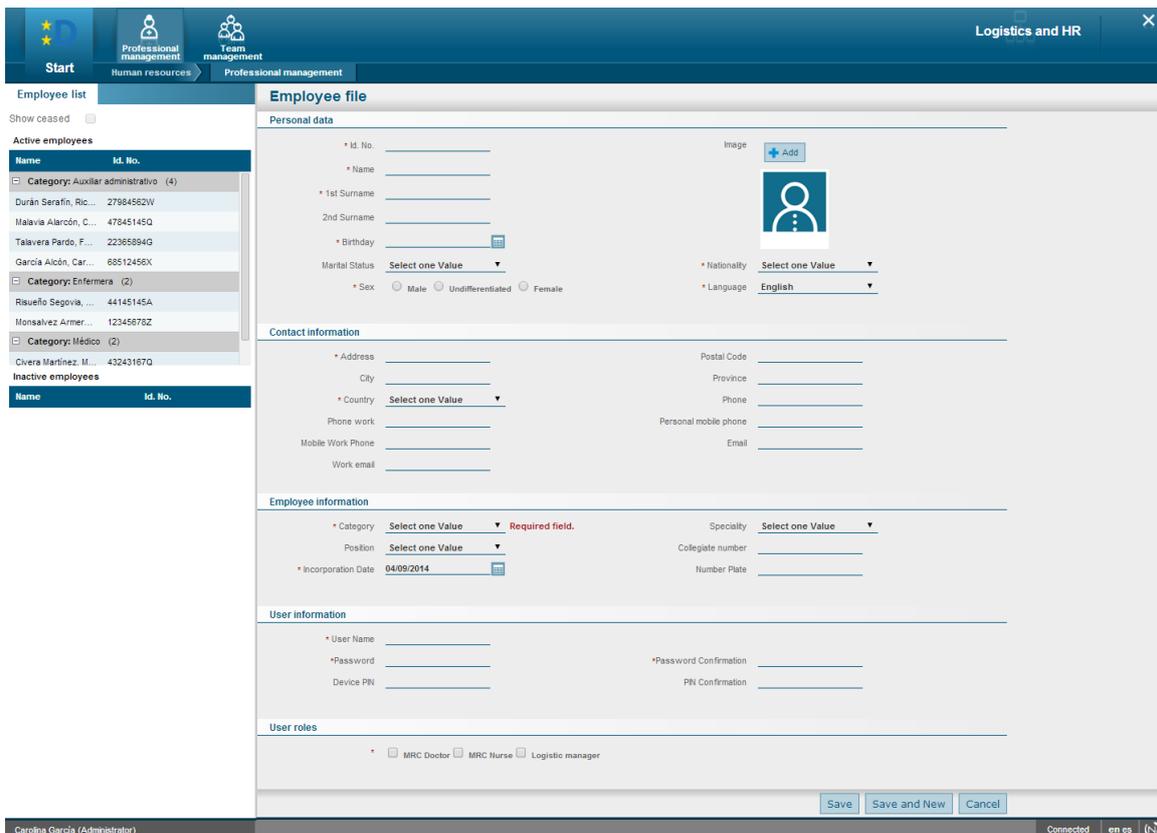


Figure 7 Register Assistant Account screen

This screen shows the entire input fields necessary to register a new assistant account. When an assistant is registered the necessary personal and contact data are stored in the database.

2.2.3 Use Case 3: View Assistant Account

The screenshot shows the 'Employee file' interface. On the left, there is a sidebar with 'Employee list' and 'Active employees' (4) and 'Inactive employees'. The main area is titled 'Employee file' and contains the following sections:

- Personal data:** Id. No. 27984562W, Name Ricardo, 1st Surname Durán, 2nd Surname Serafin, Birthday 03/16/1974, Marital Status, Sex Male, Nationality Spanish, Language Spanish.
- Contact information:** Address C/San Vicente Mártir, City Valencia, Country Spain, Postal Code 49250, Province Valencia, Phone, Personal mobile phone 667915822, Email rduran@admin.es.
- Employee information:** Category Administrative Assistant, Position, Incorporation Date 02/05/2000, Speciality, Collegiate number 4060, Number Plate 77777777.
- User information:** User Name rduran.
- User roles:** MRC Doctor, MRC Nurse, Logistic manager (checked).

At the bottom right, there are 'Disable' and 'Modify' buttons. The bottom status bar shows 'Carolina García (Administrator)', 'Connected', and system icons.

Figure 8 View assistant account screens

The HP can review the information stored for one assistant in this screens. Compared with the screen of the previous use case, it can be easily observed that it has the same scheme; the only difference is that in this view the fields are not editable.

2.2.4 Use Case 4: Update/Delete Assistant Account

There is no specific mock-up for this use case, since the account can be deleted by clicking in the “Delete” button from the view screen.

The same way, when the HP wants to edit an account, she clicks in “Modify” and the different fields just become editable, as in Figure 7.

2.2.5 Use Case 5: Register Primary Screening Stage

The screenshot shows the 'Primary Screening Stage' configuration page. The top navigation bar includes 'Start', 'Students List', 'Student Summary', 'Training Data', and 'Configuration'. The 'Configuration' menu is active, and the 'Primary Screening Stage' sub-menu is selected. The main content area is divided into two sections: 'PSP list' and 'Primary Screening Stage'.

PSP list: A table with columns 'Name', 'Location', and 'HP'. It contains one row: 'Example', 'Somewhere', 'John Doe'.

Primary Screening Stage: A form with the following fields:

- * Name:
- * Start Date:
- * End Date:
- * Location:
- * Place Weigh:
- * Team:
- Description:

At the bottom of the form are buttons for 'Save', 'Save and New', and 'Cancel'. The footer shows the user 'Carolina Garcia (Administrator)' and a 'Connected' status.

Figure 9 Create a new Primary Screening screen

The HP can access the page to create a new PSS through the Configuration menu. There, a page to introduce all required data is shown.

2.2.6 Use Case 6: View Primary Screening Stage Information

The screenshot shows the 'Primary Screening Stage' information page. The navigation bar includes 'Start', 'Students List', 'Student Summary', 'Training Data', and 'Configuration'. The 'Primary Screening Stage' section is active. On the left, there is a 'PSP list' table with columns 'Name', 'Location', and 'HP'. The main area displays 'PSS Information' for a specific stage named 'IEGS Trial', including details like Start Date (03/05/2015), End Date (06/05/2015), Location (Stockholm), Plates Weight (125g), Team (Team IEGD), Number of students (35), and Description (Primary Screening Phase II IEGS). At the bottom, there are 'Disable' and 'Modify' buttons.

Figure 10 Visualization of PSS data

Same as with the assistant accounts, the visualization of the data is the same page as in Figure 9 but with no editable fields.

2.2.7 Use Case 7: Update Primary Screening Stage

In order to edit a PSS, from the visualization screen the HP can click on “Modify” and the fields will become editable as in Figure 9.

2.2.8 Use Case 8: View Primary Screening Event Data

When the HP accesses the Primary Screening study, the first screen is the list of users participating in that study.

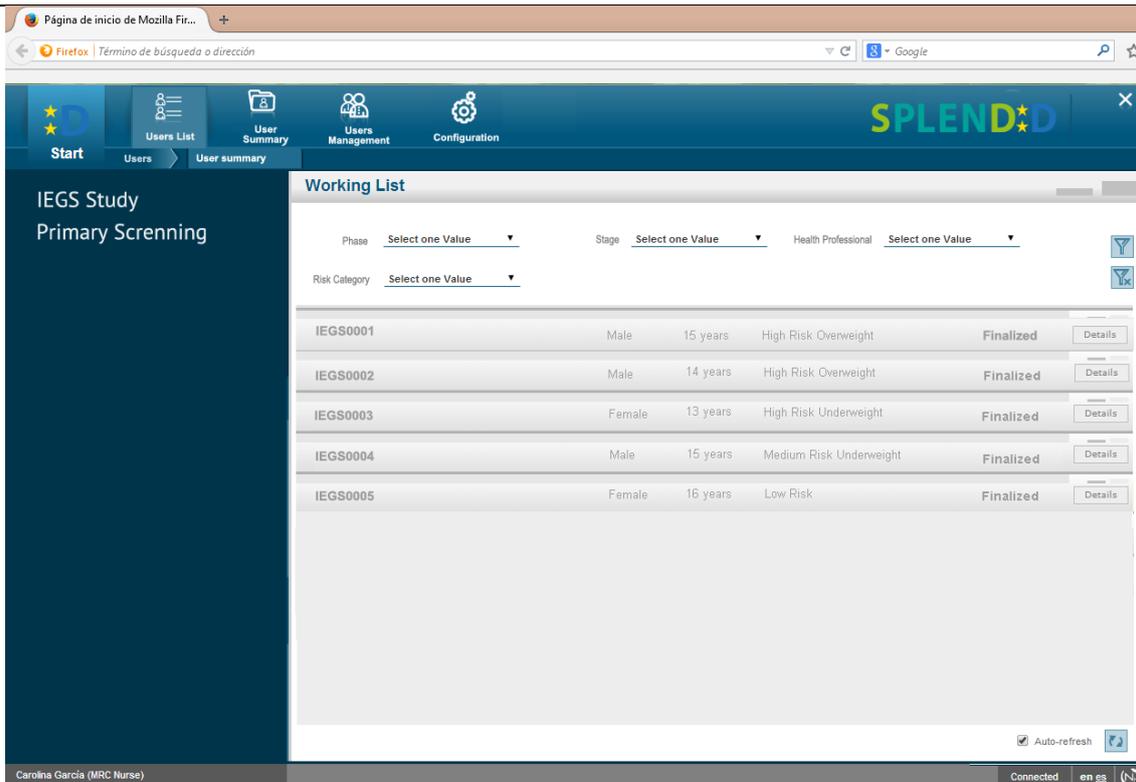


Figure 11 User list for a PSS

From that list, the HP can select a user to review the data extracted from her meal.

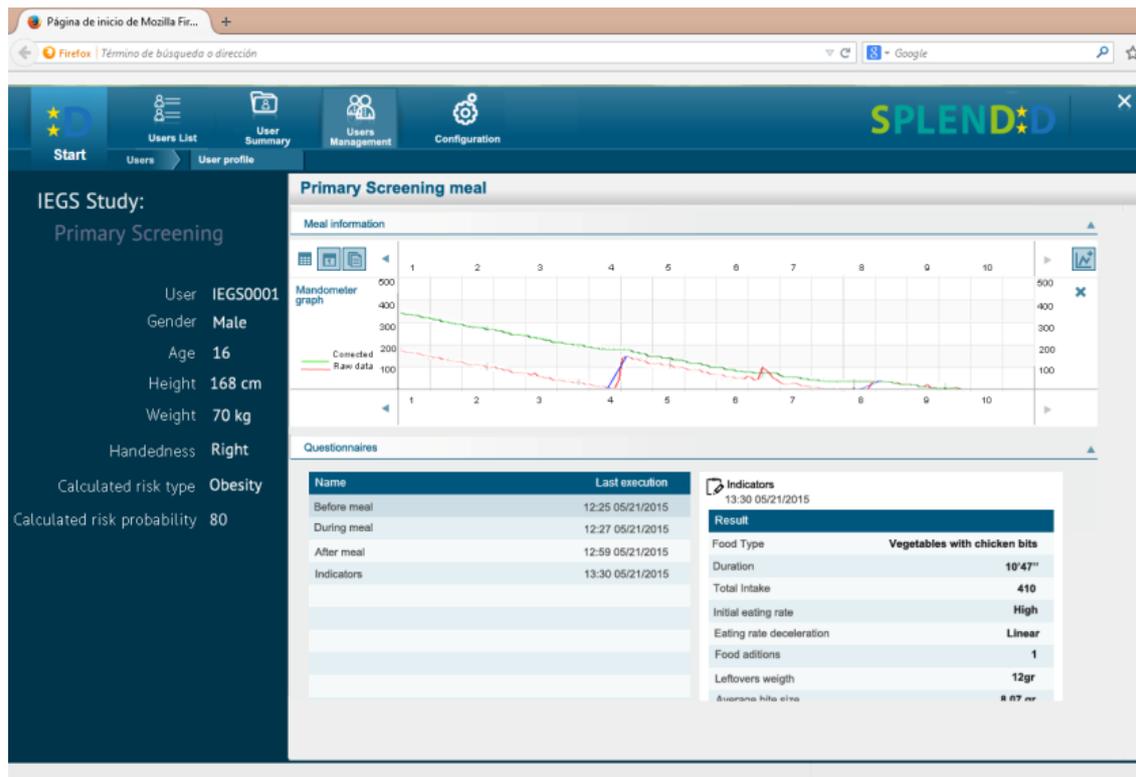


Figure 12 Primary screening event data screen

In this screen the HP can check all the information about the Primary Screening meal of a user. At the left side of the screen the information about current used is visible. Meanwhile, at

the right side all the information of the meal: the Mandometer graph, the questionnaires and the list of indicators extracted by the DSS from the meal.

2.2.9 Use Case 9: Confirm/Modify Primary Screening Risk Category

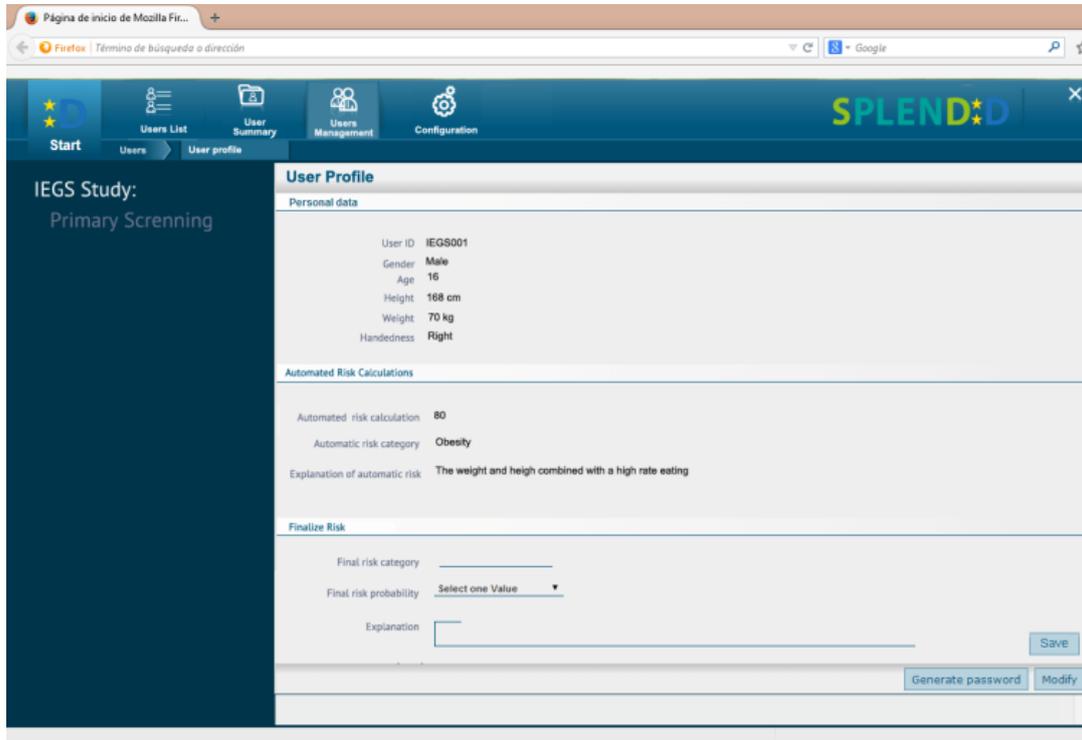


Figure 13 User Profile Screen

Figure 13 represents the User Profile screen. In this screen the HP can check the user data like age, gender, height, weight and handedness. Also, this profile presents the automatic risk calculated by the DSS, given the probability, the category and the explanation.

The last section of the screen is for the HP to finalize the risk. Once the risk has been finalized that section will have the final values. The HP can always modify information of the user by clicking on the “Modify” button.

2.2.10 Use Case 10: Export Primary Screening Data to File

The use case to export data doesn't have a concrete screen. In the screen where the HP can review all data and in the user's profile screen, there's a button that allows the HP to export all data.

2.2.11 Use Case 11: Logout from Web Interface

The HP can log out from the web application by clicking the X button in the top right side of the web application, which is visible from every screen.

2.2.12 Use Case 12: Register Adult User

Figure 14 Register adult user screen

For the adult's study, the users have to be registered when entering the study, so the HP need a section to register a new user. In that file, the HP can introduce the characteristics of the user like age, gender, height, weight, and handedness. The HP can access this screen when creates a new BAE accessing from the adults part of the login process.

2.2.13 Use Case 13: View Student/Adult User

Figure 15 View adult user information

This screens shows the information contained in the previous screen.

2.2.14 Use Case 14: Update Student/Adult User

There is no specific mock-up for this use case, since the account can be updated in “Modify” in the view screen and the different fields just become editable, as in Figure 14.

2.2.15 Use Case 15: Register Behavioural Assessment Event

Figure 16 Create a BAE screen

There are two possibilities to create a BAE: create it for a user coming from a PSS or create it directly for an adult.

In the first case, the HP will select a user to move it from PSS to BAS from the user’s profile screen. Then, the BAE screen will appear. In this screen, the HP annotates the date and time of the meeting with the user, the location, and the start and end dates of the BAS. Also, from this screen the HP can select other professionals related with the user, such as assistants or other HP with view permissions. There is also a field in order to register the devices that are handled to the user.

In the adults’ case, the HP will access this screen from the Configuration menu. There, the same information is asked. In this case the HP will have to create the person to whom the study is address.

2.2.16 Use Case 16: View Behavioural Assessment Event Information

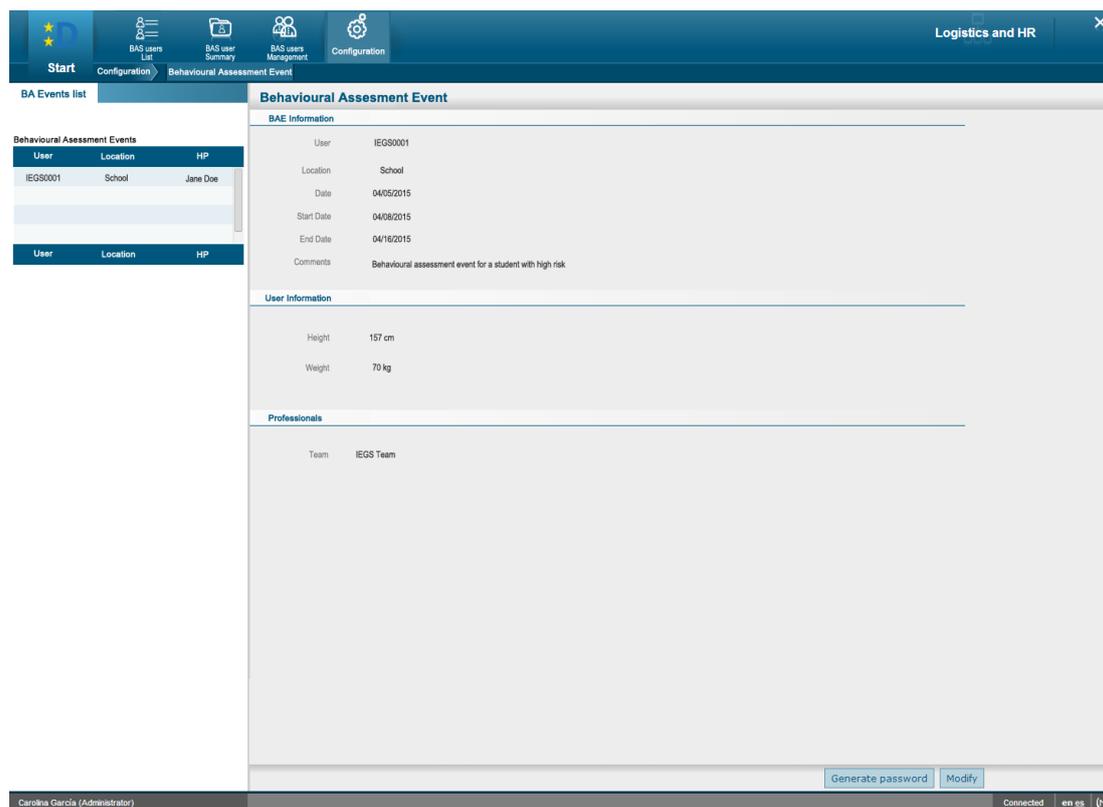


Figure 17 View BAE screen

This view shows the information of a BAE. It is the same screen as in Figure 15, but with no editable fields.

2.2.17 Use Case 17: Update Behavioural Assessment Event

In order to edit a BAE, from the visualization screen the HP can click on “Modify” and the fields will become editable as in Figure 15.

2.2.18 Use Case 18: View Behavioural Assessment Data

This use cases, together with use case 26, is the most complete and complex use cases to translate to screens. The HP has to be able to see all the information collected about meals and physical activity during the period in a comprehensive way.

In order to do so, the User Summary section in the navigation panel will have a tab panel inside, with different panels showing the information about meals, activity and global indicators for the period.

In this document, we show the designed screens to show meals (Fig. 18, 19 and 20) and activity (Fig. 21 and 22), since there has been some unsolved discussion about the way to show the global indicators.

Also, there has been some discussion to include a tab with mixed information about meals and activity, which has not yet been designed.

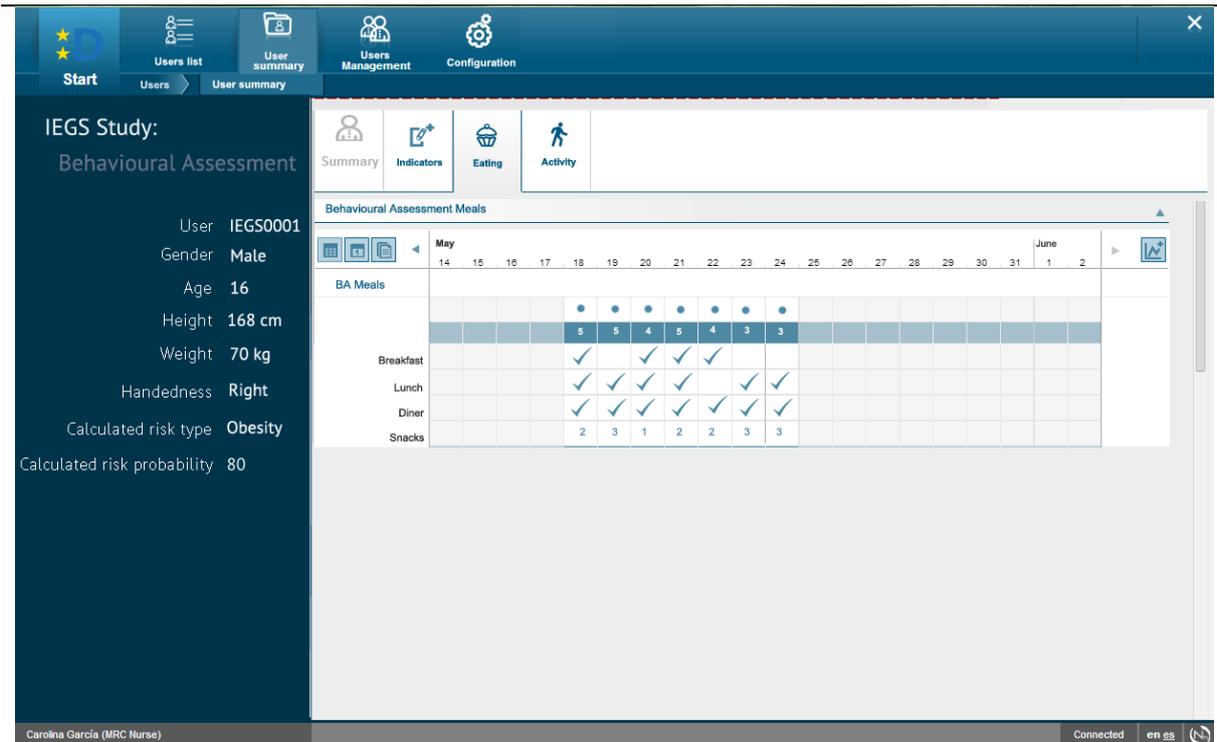


Figure 18 First screen of the meal visualization. This screen shows the days where the BAS took place and the meals of each day

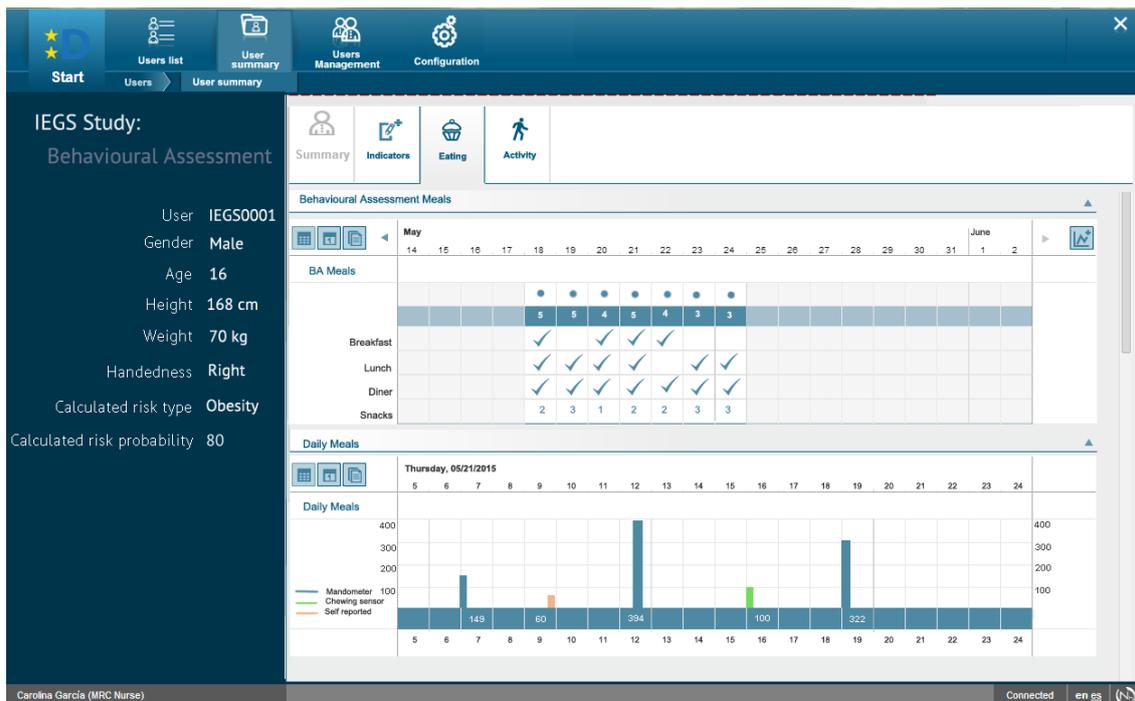


Figure 19 Second screen of the meals visualization. After clicking on a day, a graph with the meals of that day, indicating quantity eaten and how the meal was recorded is shown.

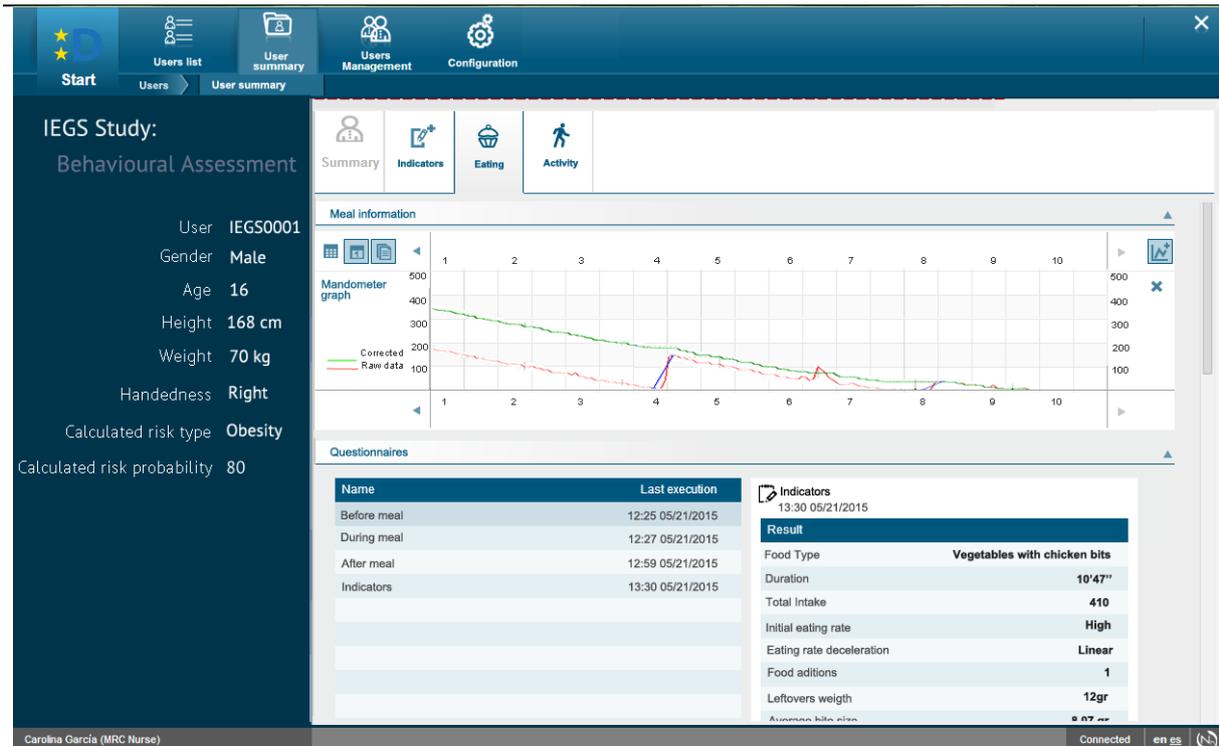


Figure 20 Third screen of meals. By clicking in a meal in the second screen, it shows the graph of the meal (if it's available) and the questionnaires and related indicators

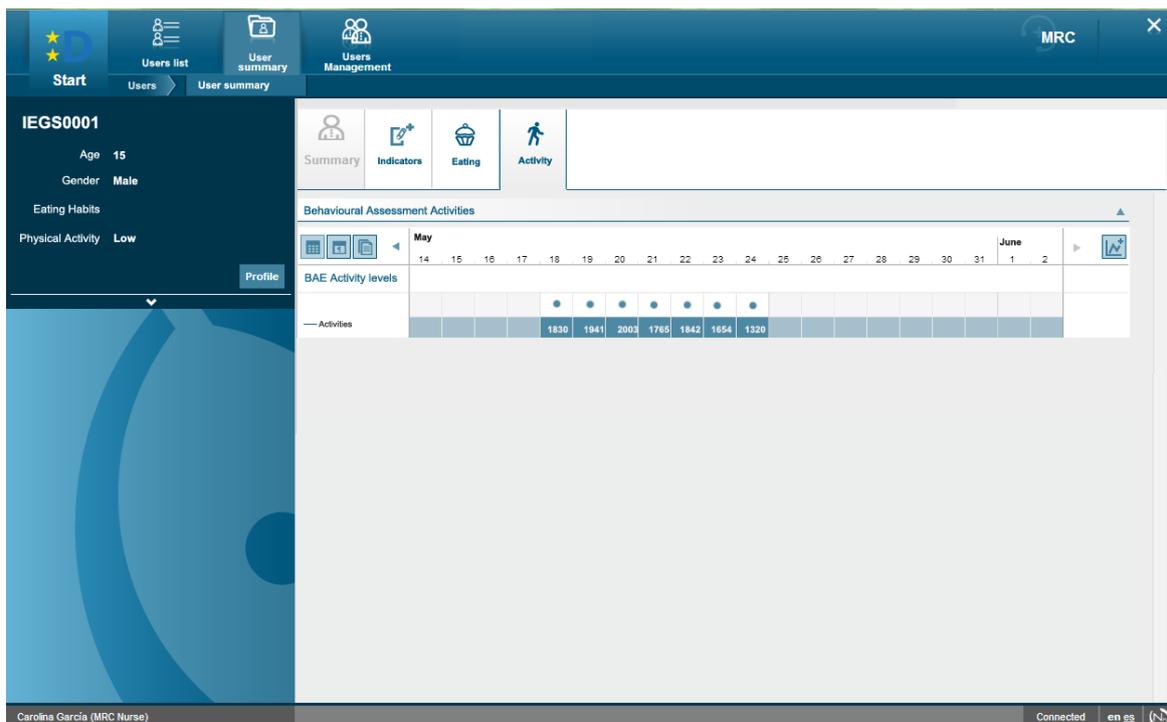


Figure 21 First screen to see the activity information. It contains information about which days the study took place and the calories burn each day

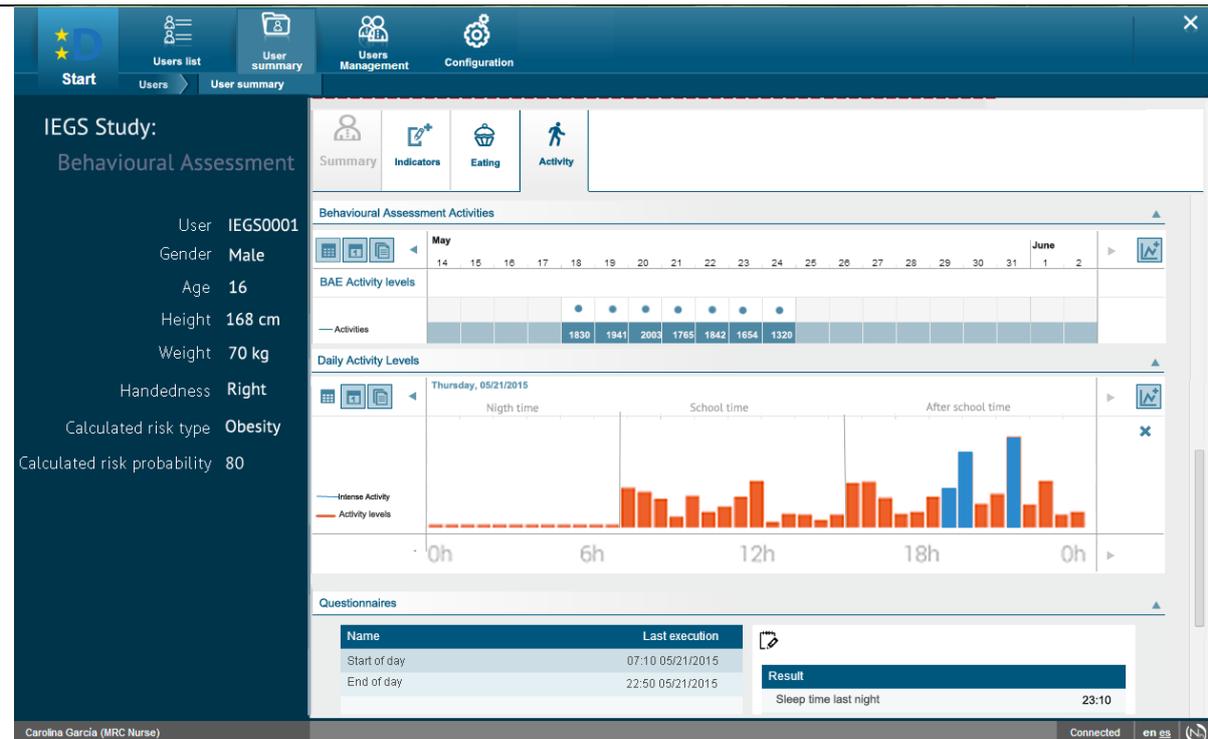


Figure 22 Second screen of activity. It is shown by clicking in a day. Shows information about activity levels during the day and intense periods. Also about questionnaires and indicators extracted from the activity

2.2.19 Use Case 19: Confirm/Modify Behavioural Assessment Risk Category

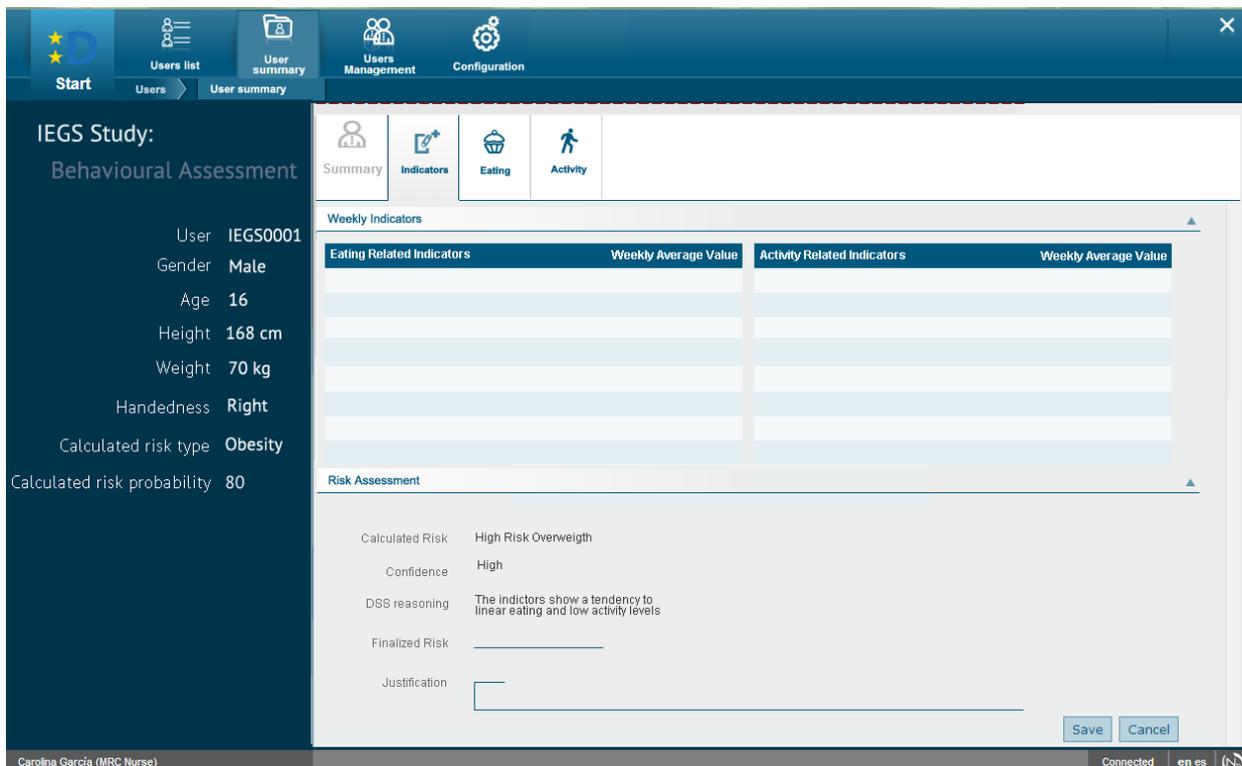


Figure 23 Risk assignment screen for BAS

During the BAS, the information collected in order to define the risk is bigger than in the PSS. That's why, instead of assigning a final risk in the user's profile, it is assigned in the same screen where the HP can check all the indicators extracted from the collected information, and so she easily decides the risk. The risk section will have the same format as for the PSS.

2.2.20 Use Case 20: Export Behavioural Assessment Data to File

The use case to export data doesn't have a concrete screen. In the screen where the HP can review all data and in the user's profile screen, there's a button that allows the HP to export all data.

2.2.21 Use Case 21: Register Personalized Guidance Event

In this case, the HP will access the registration screen the same way in both cases (adolescents and adults), from the profile of the user that has to go to PGS. Once the HP decides a user should go to PGS, she'll click on the button "Change Stage" from the user's profile and then fill all the information of the PGE. The screen to create a PGE is the same as the one to create a BAE (Figure 16).

2.2.22 Use Case 22: View Personalized Guidance Event Information

As in use Figure 17, this screen shows the information of the PGE.

2.2.23 Use Case 23: Update Personalized Guidance Event

In order to update the information of a PGE from the visualization screen the HP can click on "Modify" and the fields will become editable as in Figure 15.

2.2.24 Use Case 24: Set/Update Meal Goals

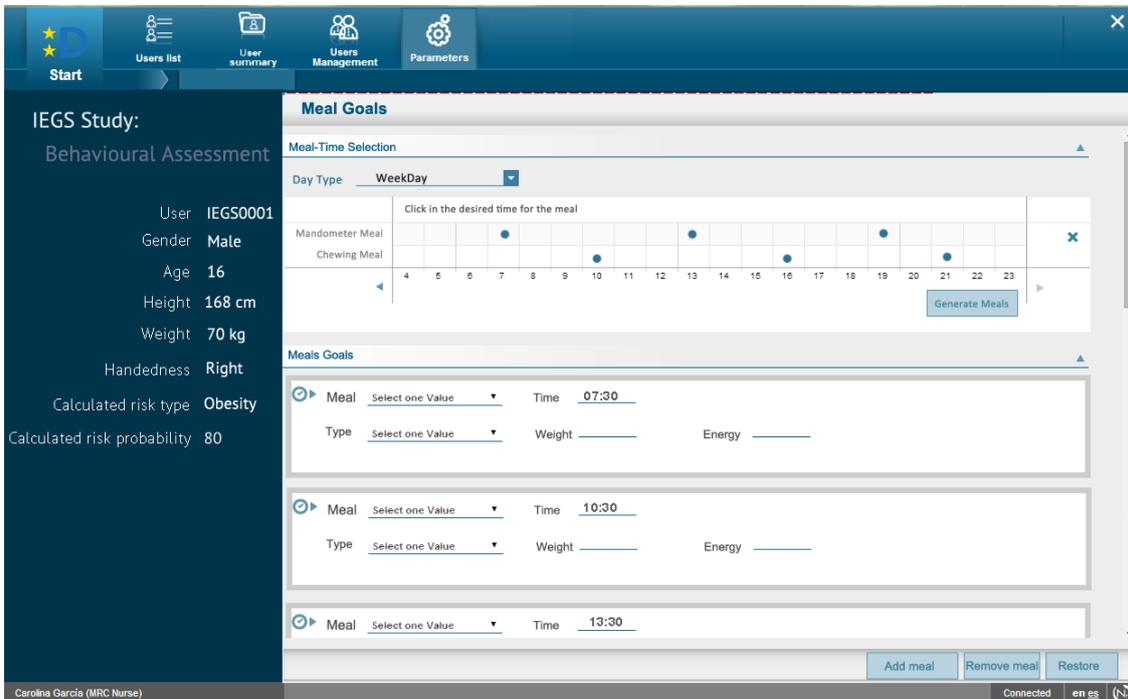


Figure 24 Set goals screen

In order to set the goals, the HP has to go to the Configuration menu. There, the HP can set meal goals for the PGS for the selected user, differentiating between work days and free days.

2.2.25 The exact types of goals have not been decided yet, so this screen can still suffer major changes. Use Case 25: Set/Update Activity Goals

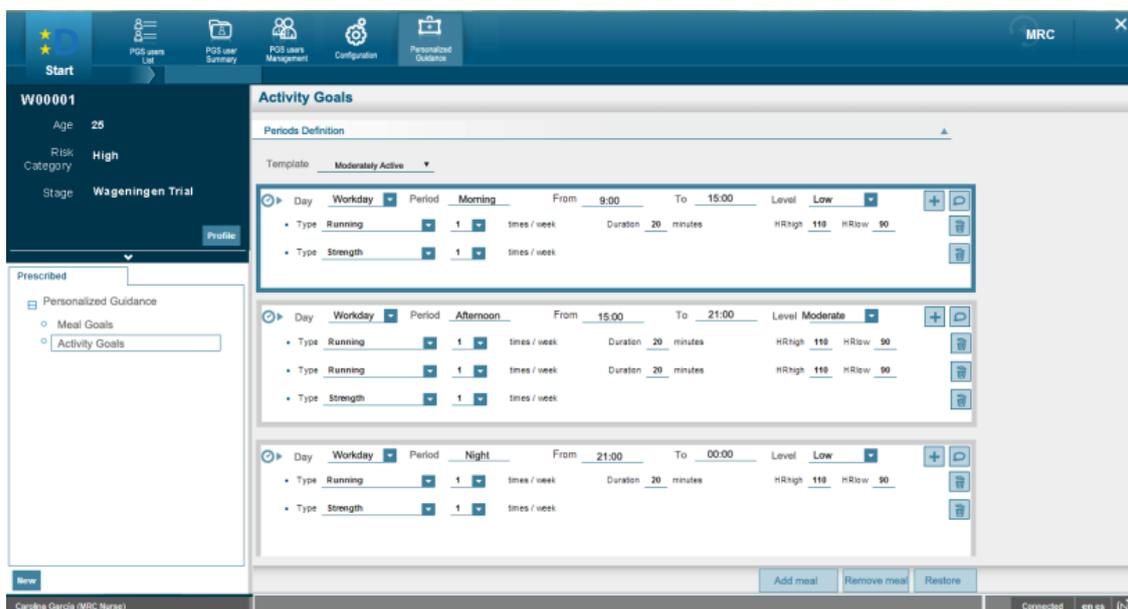


Figure 25 Activity goals screen

From the configuration menu, the HP can also set the activity goals. In a similar layout than the meal goals, the HP will be able to set different activity goals for the different periods of the day.

2.2.26 Use Case 26: View Personalized Guidance Data

The visualization of the collected data will be the same as in the BAS, the only difference will be that the compliance with the goals will be shown in the indicators list. There's no need to duplicate the screens here, since they can be checked in the Use Case 18: View Behavioural Assessment Data section.

2.2.27 Use Case 27: Export Personalized Guidance Event Data to File

The use case to export data doesn't have a concrete screen. In the screen where the HP can review all data and in the user's profile screen, there's a button that allows the HP to export all data.

3 Mock-ups of Mobile Personal Guidance Application

This section presents the mock-up screens that have been designed for the PGA. In the mock-up design process several tools have been employed, varying with the corresponding step of the design methodology that has been presented in Section 1.2.

In parallel with the study of the PGA use cases (step 1), the screens of the application were identified and their main building elements were sketched in a Microsoft Powerpoint document; the transitions between the screens (screen flow) were also documented. The first and the second version of the mock-up (steps 2 and 4 respectively) were prepared with the help of Fluid UI⁶, a web-based tool for UI prototyping. The more mature third version of the mock-up was designed with Adobe Photoshop⁷, so as to be able to extract high-resolution icons and graphic elements for the application implementation phase. Finally, the herein presented mock-up screens were implemented via Android programming with the help of the Eclipse IDE⁸.

3.1 Use cases implemented in mock-ups

According to deliverable D1.2 (“Use Cases Specifications”), the PGA has been associated with 16 use cases for the Behavioural Assessment Phase and 22 use cases for the Personalised Guidance Phase – however, most of them are similar or identical to the use cases of the former phase. Moreover, another 5 use cases are involved in the smartphone application to be developed in the context of the Primary Screening Phase. If we imagine that each use of the smartphone application requires on average 5 screens, it becomes clear that it would have been impractical to attempt to provide in the present deliverable mock-up screens for all the use cases of the PGA.

For the above reason, it was decided early in WP4 to focus on one phase of SPLendid and put our efforts in designing the most representative use cases of the phase under focus. For this purpose, the Behavioural Assessment Phase was selected. This selection can be easily justified by exploring the relation of the selected phase with the other two phases⁹. First of all, the main functionalities to be provided by the smartphone application in the Primary Screening Phase are a – small – subset of the functionalities of the Behavioural Assessment Phase. Indeed, the main purpose of the smartphone application in the Primary Screening phase is to allow the collection of some user input and the recording of a meal by means of the Mandometer®. Both these functions are included in the Behavioural Assessment Phase. Apart from that, since the Primary Screening Phase is carried out in an one-time, supervised event (a meal in a school environment), the corresponding smartphone application is not bound by the same demanding usability requirements that apply to the Behavioural Assessment Phase. For these reason, the mock-up designs of the latter phase can be easily reused for the first phase.

6 <https://www.fluidui.com/>

7 <http://www.photoshop.com/>

8 <https://www.eclipse.org/>

9 For a detailed description of the three phases of the system the reader is referred to deliverable D1.1 (“User Requirements”)

When it comes to the relation between the Behavioural Assessment Phase and the Personalized Guidance Phase, it can be stated that the functionalities of the latter phase constitute a superset of those of the former phase. Being more sophisticated and demanding, the Personalized Guidance Phase needs a more “informed” design process which is only possible only after considerable experience has been gained from the design of the Behavioural Assessment mock-ups. Even so, the Personalized Guidance mock-up will not be designed from scratch; instead, they will make heavy use of the mock-ups of the previous phase as their basis and they will offer only a short list of targeted enhancements. All the above facts establish the Behavioural Assessment Phase as the perfect starting point for the mock-up design task of PGA.

The next step was to select a list of use cases of the Behavioural Assessment Phase to work on. A thorough selection process was followed that took into account the significance of each use case for the SPLENDID system, their re-usability in the other two phases and, most importantly, the overlapping between the corresponding mock-ups. This selection process resulted in a list of 5 use cases, which are eventually presented in this deliverables. Table [1] provides the aforementioned list and also justifies the selection process.

Table 1: Outcome of the selection process for the use cases to be presented in D4.1. Selected use cases are highlighted in blue colour.

UC ID	UC Name	Explanation
UC.M.6	Initialize Smartphone Application	Used only once (by the assistant/health professional) in the application lifecycle
UC.M.7	Activate Smartphone Application	Used only once (by the assistant/health professional) in the application lifecycle
UC.M.8	Register Meal	Merged with UC.M.9. Included in UC.M.10 & UC.M.14
UC.M.9	Register Beverage	Merged with UC.M.9. Included in UC.M.10 & UC.M.14
UC.M.10	Record Meal with Mandometer	Primary recording function of the system
UC.M.11	Record Meal with Chewing Sensor	Almost identical to UC.M.10
UC.M.12	Present Sleep/Wake-up Time Questions	Typical input collection use case initiated by the system
UC.M.13	Notify User	Generic-Abstract
UC.M.14	Present In-day Questionnaire	Typical input collection use case initiated by the system
UC.M.15	Present End-of-day Questionnaire & Report	Almost covered by .14 & .12 apart form report (static version of .20)
UC.M.16	Ask User about Detected Meal	Partially covered by UC.M.10, UC.M..14 & UC.M.17
UC.M.17	Request User to Put Activity Meter Back on	Typical system notification
UC.M.18	Request User to Put Chewing Sensor	Almost identical to UC.M.17

	Back on	
UC.M.19	Remind User of Next Appointment	Partially covered by UC.M.17
UC.M.20	Get Help	Help material to be compile at the end of the application development
UC.M.21	View Usage Compliance Information	Primary reporting function of the system

As explained in Table 1, the mock-ups of the use cases that have been selected provide a very clear image of the screens of the PGA and also cover the majority of the design effort that has to be taken in the context of the Behavioural Assessment Phase. This is mainly owing to the overlapping between the functionality of various use cases. For instance, all the screens of UC.M.8 (Register Meal) are also employed by UC.M.10 (Record Meal with Mandometer). Another example is the pair of UC.M.10 and UC.M.11 (Record Meal with Mandometer) which are identical in terms of the involved screens with the single exception of the icons depicting the sensor which is used for the meal recording. To conclude, it is estimated that about 80% of the unique screens that are involved in the Behavioural Assessment Phase are included in the 5 presented use cases.

3.2 Mock-ups

3.2.1 Use Case UC.M.10: Record Meal with Mandometer®

This is probably one of the most important functions of the PGA. Here the users record a meal with the help of the Mandometer scale. First, the users bring the Main Screen of the application to foreground by clicking the icon of the app. on the desktop of their smartphone. From the Main Screen they can initiate the meal recording procedure, during which they are all the time guided by the PGA regarding any actions they need to take.

In the following, the flow of screens that have been designed for carrying out the functionality of this use case are presented; a short outline of each screen is provided along with the user action that leads to the subsequent screen. Only the basic flow of screens is presented – exceptions have not been included.

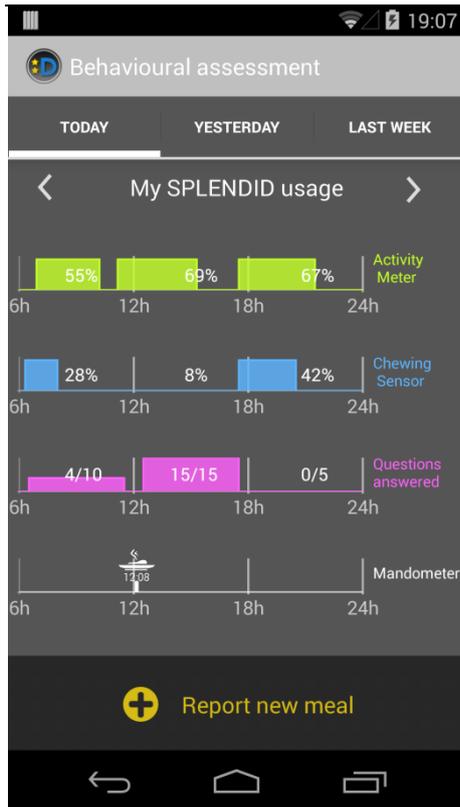


Figure 26: Main Screen of the PGA. Here, the users can be informed about the usage of the system on the present day, the day before and also take a look at a summary of the system usage for the entire last week.

The users are also provided with an option to “Report new meal”.



“Report new meal” button is pressed

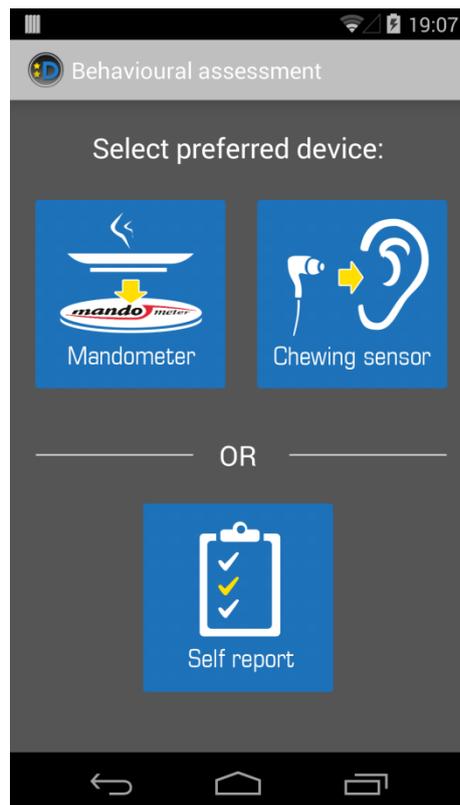


Figure 27: Recording/Reporting Method Selection Screen. In this screen the user can select either the sensor to be used for recording their meal or the option to report a – past – meal manually (without any sensors). The recording options are grouped together, while the reporting option is presented as an alternative.



“Mandometer” button is pressed

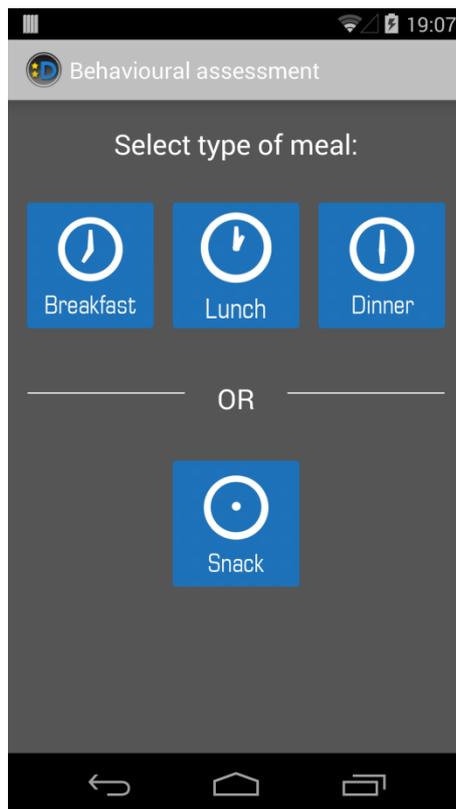


Figure 28: Meal Type Selection Screen. In this screen the users have to specify the type of the meal that is going to be recorded: Breakfast, Lunch, Dinner or Snack. The buttons for the main three meal types are grouped together, while the fourth button is positioned in some distance, separated from the aforementioned group, defining the type of unscheduled meals.



One of the meal type buttons is pressed

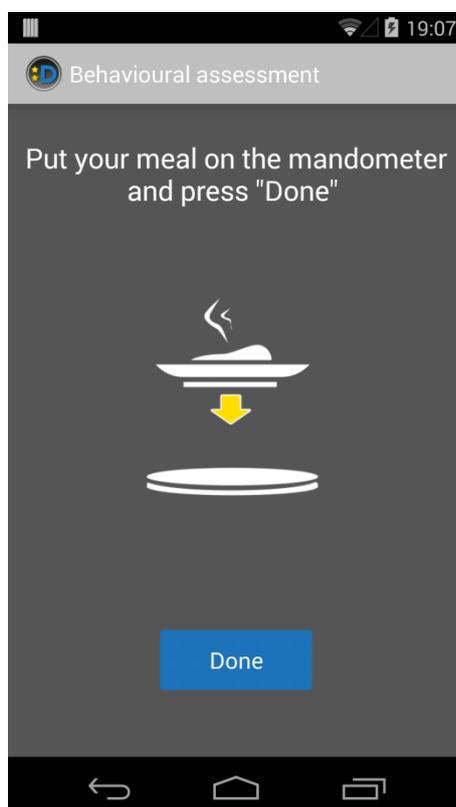


Figure 29: Plate Placing Instructions. Here the smartphone application provides instructions to the users for placing their plates (filled with food) on the Mandometer scale. The instructions are both textual and visual (with the help of an animated image). Once the users follow the application’s instructions, they press the “Done” button.



“Done” button is pressed

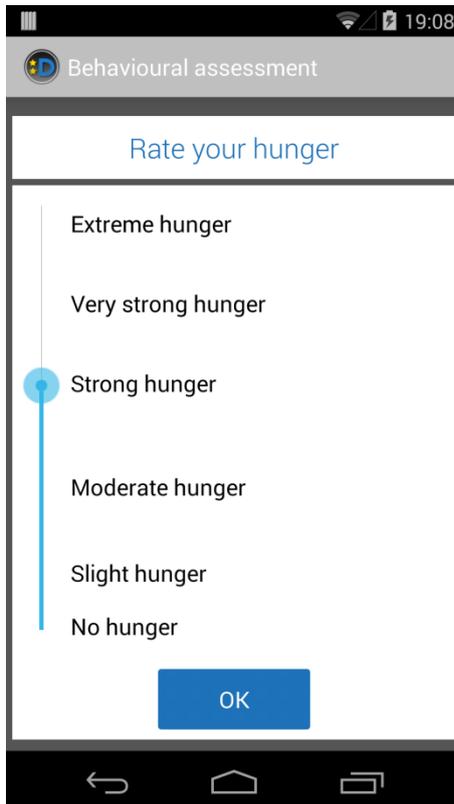


Figure 30: Hunger Rating Screen. This is screen allows the users to rate their hunger before they start their meal. A logarithmically scaled slider is used for rating hunger according to 6 categories. After finishing with the rating, the users press the “OK” button.



“OK” button is pressed

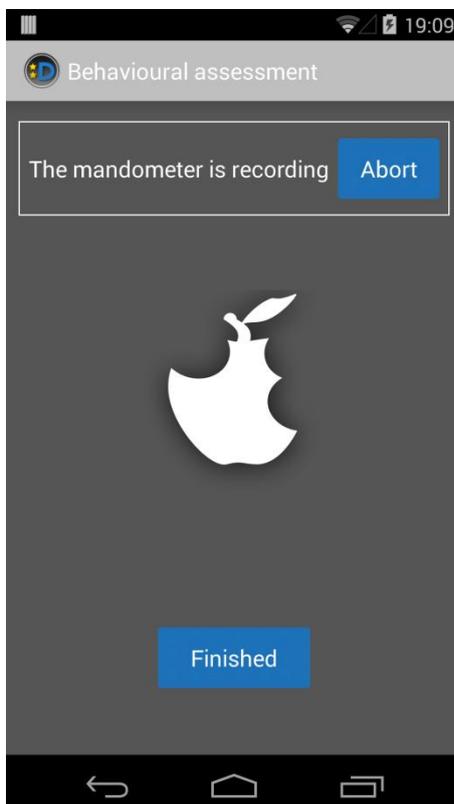


Figure 31: Meal in Progress Screen. This is the screen that appears while the meal is in progress. The users are informed that the Mandometer is recording the meal and they are provided with the option to abort the meal (“Abort” button). This option is meant to be used only in cases where due to force majeure the meal cannot be completed normally. An animated image in the middle of the screen (an apple that is being eaten) gives the effect of a progressing event. When the meal is completed, the users press the “Finished” button.

Note: Every time a predefined amount of eating time passes, a hunger rating dialogue emerges. This dialogue is identical in appearance and behaviour with the ones presented in the previous and the following figure. After the users’ hunger is rated, the smartphone application returns to the present screen.



“Finished” button is pressed

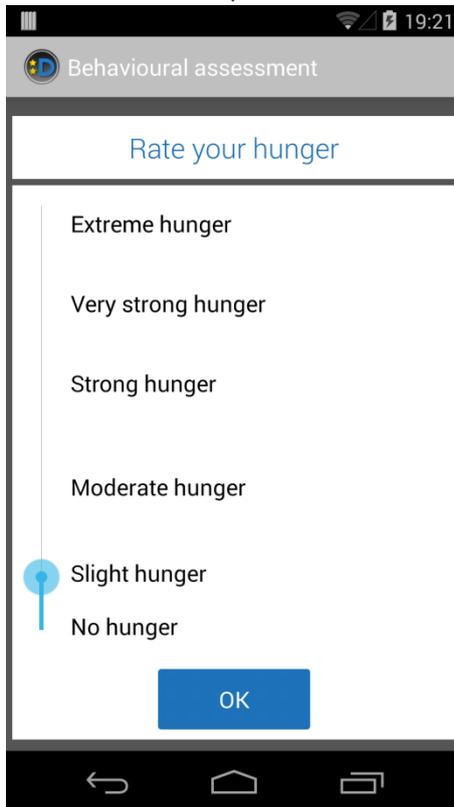


Figure 32: Hunger Rating Screen. This screen aims to capture the hunger state of the users at the end of the meal. What has already mentioned for the hunger rating screen applies also here. After finishing with the rating, the users press the “OK” button.



“OK” button is pressed

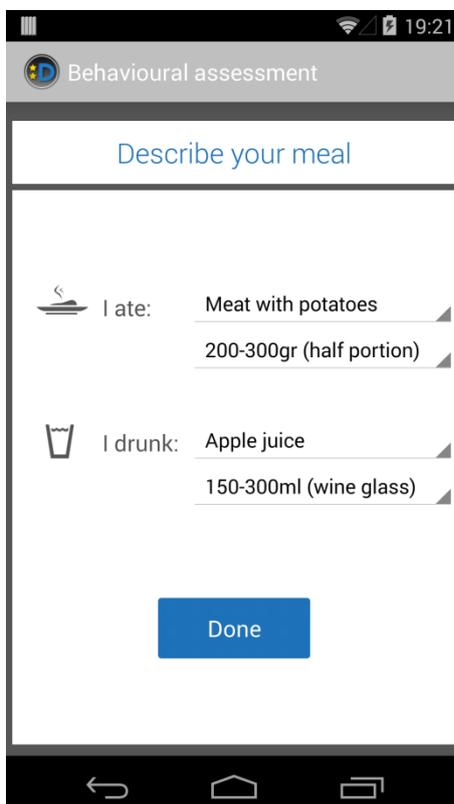


Figure 33: Food & Beverage Characterization Screen. With the help of a series of drop down lists, the users are allowed to characterize the food and the beverage they have just consumed. The information that is collected involves i) the food type, ii) the food quantity, iii) the beverage type, and iv) the beverage volume. Once the users provide all the requested information, they press the “Done” button.



“Done” button is pressed



Figure 34: Main Screen. The last action of the users brings the smartphone application back to its – already described – main screen. This concludes the current use case.

3.2.2 Use Case UC.M.12: Present Sleep/Wake-up Time Questions

An important portion of the PGA use cases aim at collecting subjective data (i.e., not sensor recorded data) concerning the user behaviour. This user input collection procedure can be initiated either by the user or by the system (i.e., the smartphone application). UC.M.12 is a prominent example of system-initiated user input collection. At some point in the morning, the users are asked by the smartphone application to answer a few questions related to the time they slept last night and woke up the present day along with their eating and physical activity behaviour in the meantime. The use case starts with a dialogue that is presented to the users asking them to complete the morning questionnaire. If they agree to take the questionnaire, the application guides them in a wizard-like fashion through the morning questions.

Same as in the previous use case, the screens of UC.M.12 are presented and outlined in a sequence that implements the basic flow of the use case.

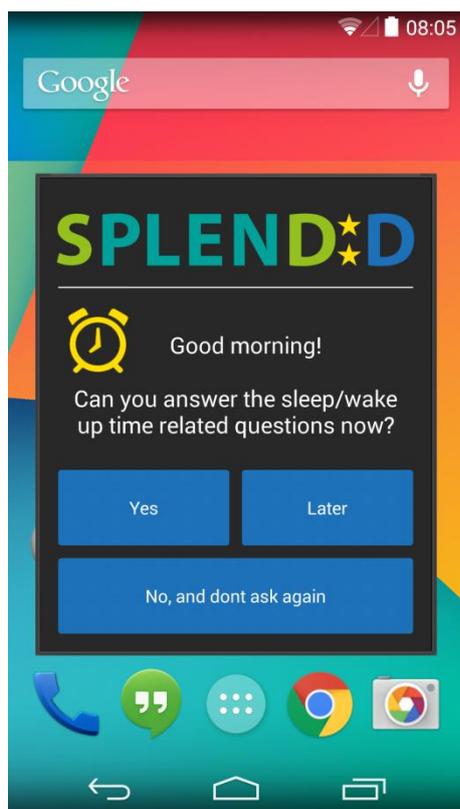


Figure 35: Morning Questionnaire Initiation Notification. This is the dialogue that is presented to the users asking them to take the morning questionnaire. The users have the option to start the questionnaire immediately (“Yes” button), postpone the questionnaire answering process (“Later” button) or dismiss the questionnaire altogether (“No, and don’t ask again” button).



“Yes” button is pressed

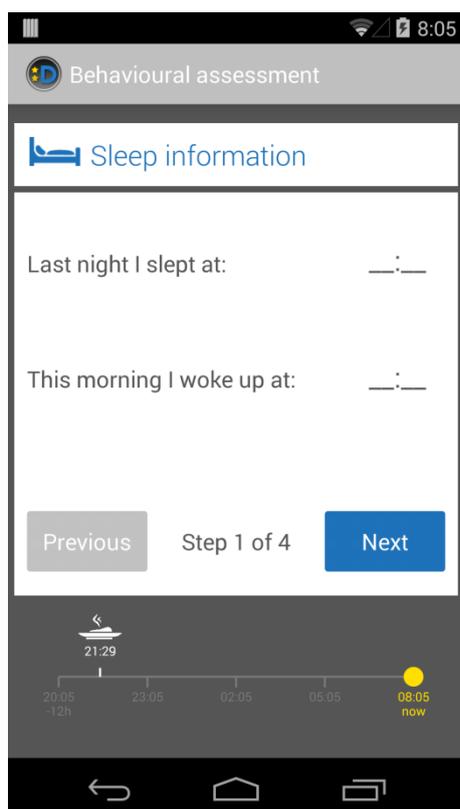


Figure 36: Sleep Information Screen. In this screen, the users can specify the time they went to sleep the previous night and woke up this morning by pressing the time setting buttons on the right side of the screen. To help them recall the requested information, a past timeline of related events (e.g., last interaction with the smartphone last night) is displayed at the lower part of the screen.

Note: The present and the following screens that serve the collection of user input have been designed with a white background to give the users the impression of a questionnaire or a form. Moreover, these screens implement a wizard (i.e., setup assistant) of 4 steps. For this reason, the current wizard step is displayed and the Previous/Next button pair is employed.



One of the time setting buttons is pressed

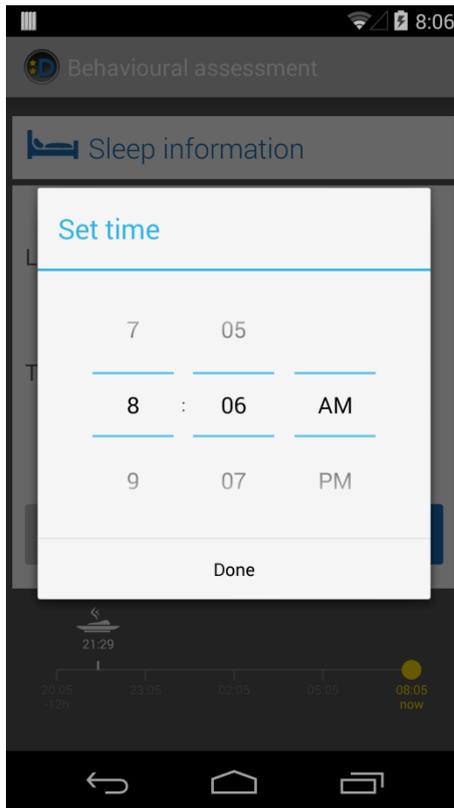


Figure 37: Time Picker. This is nothing but a native (build-in) Android dialogue for setting a time instance. Changing the hour, the minutes and the AM/PM indication is achieved via vertical sliding. When the time has been set the users press the “Done” button.



“Done” button is pressed

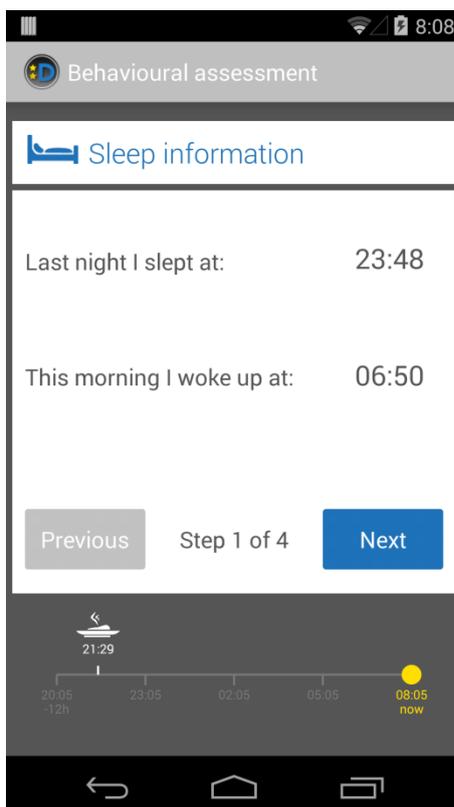


Figure 38: Sleep Information Screen (completed). This is the same with Figure 36 (two screens back) except for the fact that the sleep and wake-up times have been set by the users (with the help of the previous dialogue). The defined times are now displayed. After reviewing their selections, the users press the “Next” button.



“Next” button is pressed

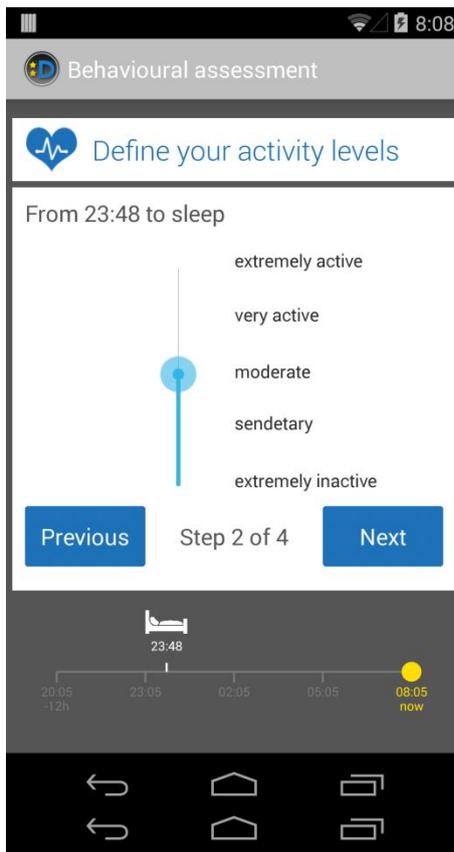


Figure 39: Physical Activity Rating Screen (previous night). In the second step of the morning questionnaire wizard, the users are asked to rate their physical activity concerning last night (considering 5 activity levels). A logarithmically scaled slider is used for this purpose, while a past timeline of related events (e.g., removal of the Activity Meter last night) helps users’ recollection. When the users are done with this screen, they press the “Next” button.



“Next” button is pressed



Figure 40: Physical Activity Rating Screen (this morning). This is the third step of the morning questionnaire wizard, where the users are asked to rate their physical activity concerning this morning. This screen heavily resembles the previous screen and their only difference is the query time period. When the users are done with this screen, they press the “Next” button.



“Next” button is pressed

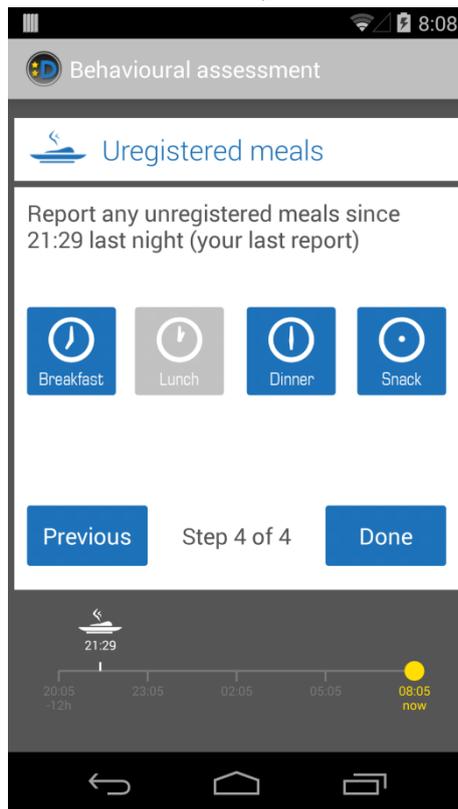


Figure 41: Report Meals Screen. This is the last step of the morning questionnaire wizard where the users are asked to report any meals that have taken place since the last registered meal but have not been registered into the system. Users’ recollection is assisted by a past timeline of related events (e.g., last registered meal). In case the users press one of the meal type buttons (in the middle of the screen), PGA will move the users to a sequence of screens for registering the meal (see next use case). When the users are done with this screen, they press the “Done” button.

Note: Pressing the “Done” button concludes the morning questionnaire wizard and, in the same time, the entire use case. PGA closes all its screens and dialogues and the smartphone displays whatever it was displaying before the initiation of the use case.

3.2.3 Use Case UC.M.14: Present In-day Questionnaire

According to the use case classification that was introduced in the previous use case (Section 3.2.2), UC.M.14 provides – just like UC.M.12 – a system-initiated user input functionality. As some of the input that needs to be collected by this use case coincides with the input collected in the context of UC.M.12, it follows that a portion of the screens of the present use case resemble or even identify with screens of UC.M.12.

At some point during the day, the smartphone application realizes that a significant amount of time has passed without the users recording or reporting any meals. At this point the application presents a notification to inform the users about the fact and asks them to report any unregistered meals. Once again, the application makes sure that the users can easily understand what actions they are supposed to take in order to carry through the functionality of the use case.

Same as in the previous use cases, the screens of UC.M.14 are presented and outlined in a sequence that implements the basic flow of the use case.

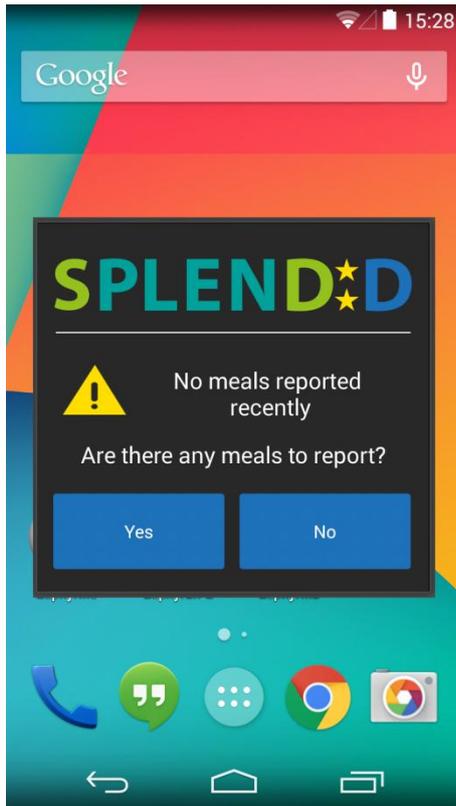


Figure 42: Meal Reporting Absence Dialogue. This is a dialogue that is presented to the users to inform them that a considerable amount of time has passed without them registering any meals into the system. The dialogue also asks whether any meals have taken place in the last hours but have not been registered. Possible answers are “Yes”, “No” – implemented as 2 buttons.

↓ “Yes” button is pressed

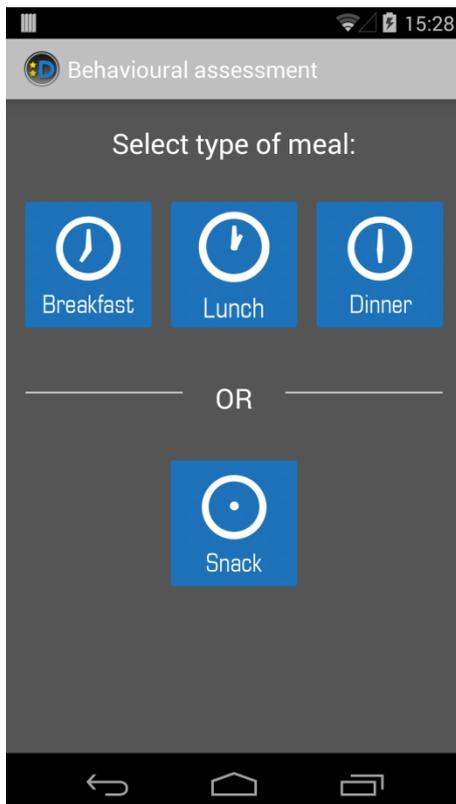


Figure 43: Meal Type Selection Screen. If the users admit the existence of unregistered meals, PGA presents the current screen, where they have to select the type of the unregistered meal – between Breakfast, Lunch, Dinner, and Snack. This screen looks a lot like the one in Fig. 28 (UC.M.10), which also asked the users to specify the meal type; however, in the latter use case, it was for recording purposes.



One of the meal type buttons is pressed

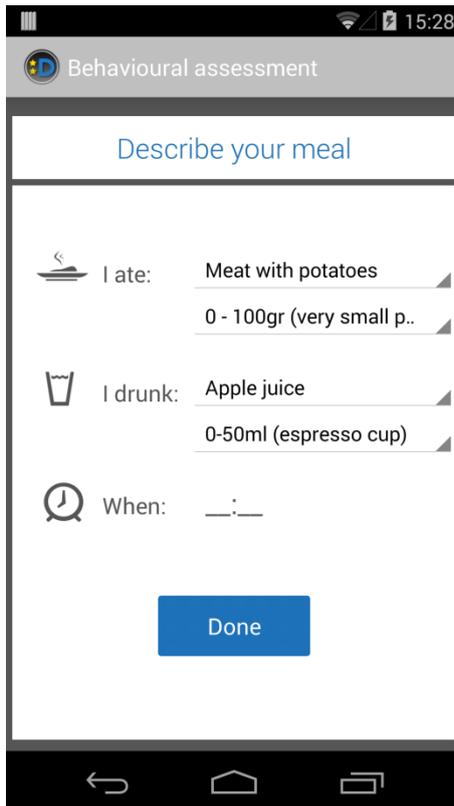


Figure 44: Food & Beverage Characterization Screen. This screen is similar to the one presented in Fig. 33 in the context of UC.M.10. With the help of a series of drop down lists, the users are allowed to characterize the food and the beverage they consumed. The only addition with respect to the screen of Fig. 33 is the time setting button for specifying the time the meal took place. Once the users provide all the required information, they press the “Done” button.



“Done” button is pressed

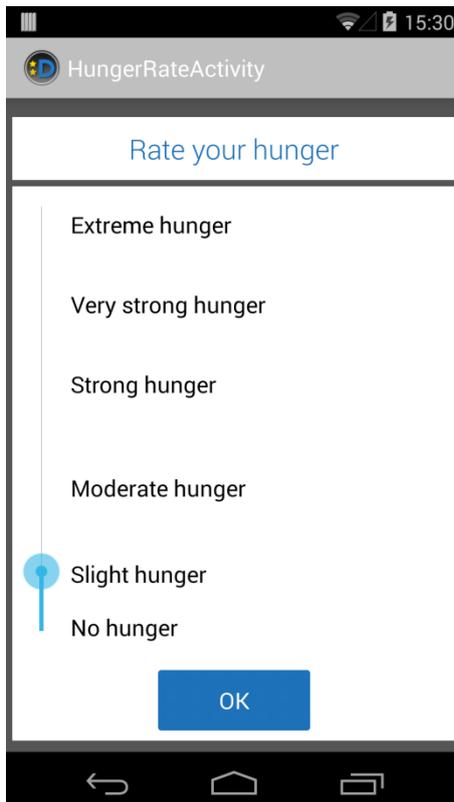


Figure 45: Hanger Rating Screen. This screen is identical with the one presented in Fig. 30 and 32 in the context of UC.M.10. The users move the slider to the appropriate hunger category and they press the “OK” button.

Note: This screen concludes the current use case. PGA closes all its screens and dialogues and the smartphone displays whatever it was displaying before the initiation of the use case.

3.2.4 Use Case UC.M.17: Request User to Put Activity Meter Back on

This is a very short use case, including – in its basic flow at least – only one screen. Although system-initiated, just like the previous two use cases, UC.M.17 deviates from what has already been presented until now in two ways. First, its purpose is to *notify* the users and not to also *collect input*. Secondly, it is event-triggered. This means that UC.M.17 is initiated whenever a specific event happens, which in our case is the diagnosis that the users have removed the Activity Meter. These two deviations from the previous use cases explain why UC.M.17 has been included in this presentation.

The single screen that constitutes the basic flow of UC.M.17 is presented and outlined below.

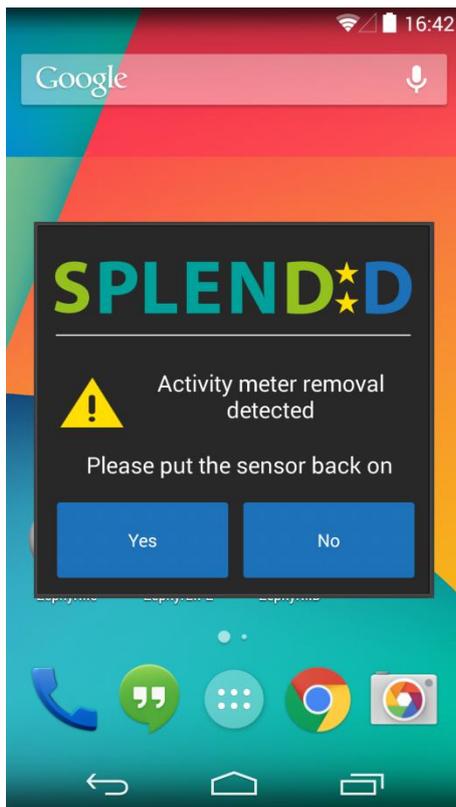


Figure 46: Activity Meter Removal Dialogue. This dialogue is presented to the user whenever the removal of the Activity Meter is detected. The dialogue informs the users about this fact and asks them to put the sensor back on. The users can either wear the sensor again (“Yes” button) or refuse to wear it (“No” button). Although not presented in this deliverable, it should be mentioned that if the “No” button is pressed, PGA will display a new screen that asks the users to specify the reason for not putting the sensor back on and the estimated absence time.

Note: If the users put the Activity Meter back on and then press the “Yes” button, the use case is concluded. PGA closes its dialogue and the smartphone displays whatever it was displaying before the initiation of the use case.

3.2.5 Use Case UC.M.21: View Usage Compliance Information

This use case implements the main reporting functionality of PGA. It revolves around the main screen of the application, which we have already seen, and provides two navigation directions: time and information type. Concerning the time dimension, the provided options are today, yesterday and last week (summarized information); navigation in time is implemented with the help of tabs or horizontal sliding. With respect to information type, the application provides summary of the system usage (sensors and question answering) and timeline of eating and physical activity events; navigation in this direction is implemented with the help of left and right buttons.

Same as in the previous use cases, the screens of UC.M.21 are presented and outlined in a sequence that implements the basic flow of the use case.



Figure 47: Main Screen of the PGA (system usage, today). Here, the users can be informed via a set of graphs about how much they have used the system the present day. This involves the usage of all three sensors and the question answering performance. Of course, the “Report new meal” button is always present at the bottom of the screen. In order to view the system usage information for previous day or the entire last week, the users can either press the corresponding tab or slide horizontally.



“YESTERDAY” button is pressed

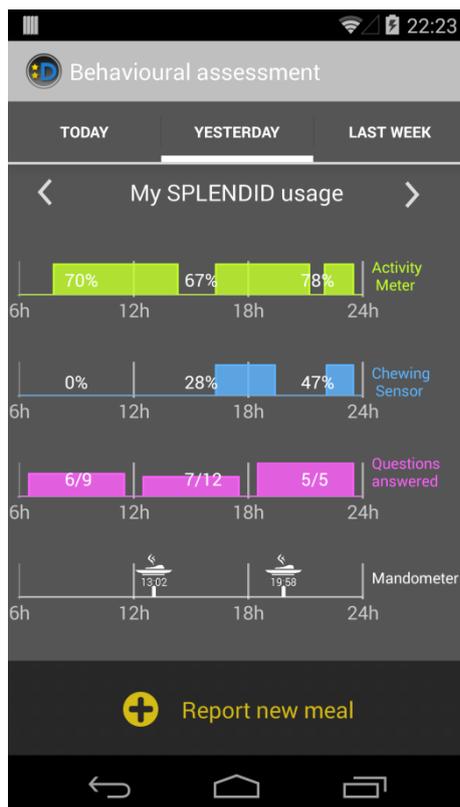


Figure 48: Main Screen of the PGA (system usage, yesterday). Clearly, this screen is identical in design with the previous one. In order to change the information type that is displayed, the users can employ the left/right (“<”, “>”) buttons.



“>” (right) button is pressed

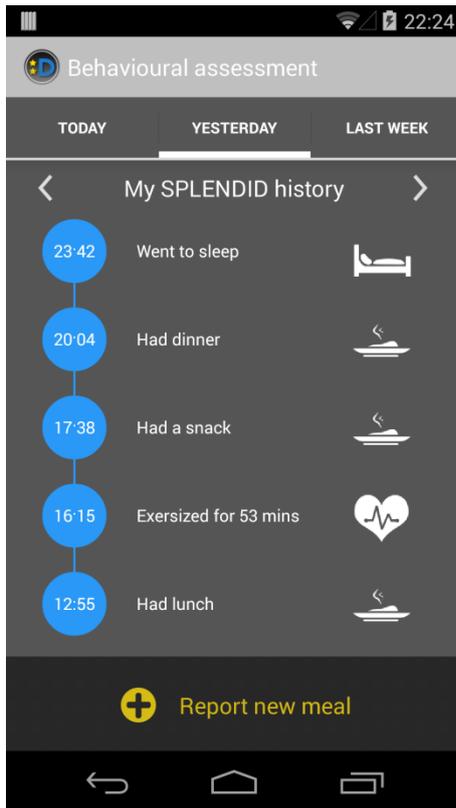


Figure 49: Main Screen of the PGA (timeline, yesterday). After using the “>” (right) button, the main screen displays a – vertical – timeline of eating and physical activity events – referring to the previous day in this particular figure. The events are indicated by an appropriate icon and a textual description, which includes – when applicable – the duration of the event. Scrolling down might be needed.



“TODAY” button is pressed

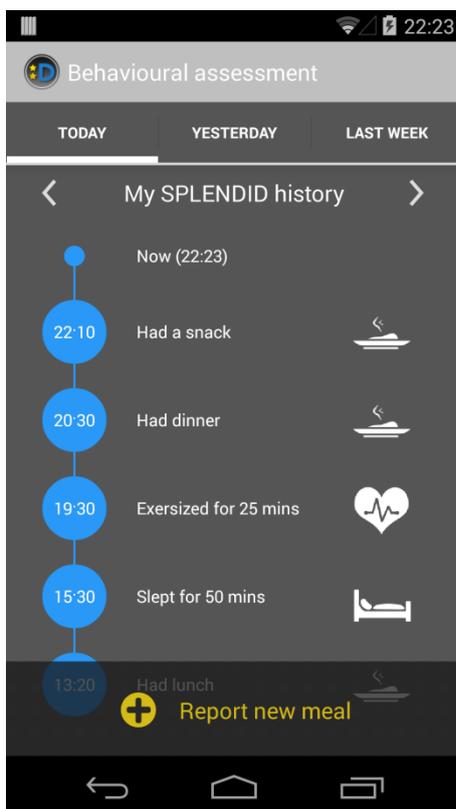


Figure 50: Main Screen of the PGA (timeline, today). This screen is obviously identical in design with the previous one. One deviation from yesterday’s timeline is the existence of the “now” (i.e., current time instance) event.

Note: This screen concludes our presentation of use case UC.M.21. Of course, the users can go on and navigate again and again along the available information. However, one of the “flavours” of the main screen will remain on the foreground until the users press the back button of the Android OS.

4 Running instances

4.1 Professional's Web Tool (PWT)

4.1.1 Scope of the running instance

As exposed in section 1.1, the aim of this document was not only to present the mock-up drawing versions of the different applications, but also show the progress in their development.

Regarding the PWT, given the decision to change the technology, only some of the use cases have been developed so far. In this case, most of the use cases related to the PSS have been developed. Also, some of the web services that will be necessary for the integration with the PGA and the DSS are available.

The use cases developed are:

- Use Case 1: Login to Web Interface
- Use Case 5: Register Primary Screening Stage
- Use Case 8: View Primary Screening Event Data
- Use Case 9: Confirm/Modify Primary Screening Risk Category
- Use Case 11: Logout from Web Interface

At the same time, the web services implemented are:

- Create new HP
- Create new Group
- Create new Adolescent
- Create new Primary Screening Stage
- Create new Primary Screening Event
- Create new Meal

4.1.2 Screenshots (remove if it does not apply)

Following, we present the screens of the developed use cases with the new framework. As can be seen in the pictures, the look and feel is different from the one in the mock-ups, but the schematics of the screens remains intact.

Use Case 1: Login to Web Interface

As presented in Figure 2, we have created a flow so when the HP logs in, she must choose the group and the stage to check. This has been defined this way in order to make easier for the HP to see the subjects in the user's list and avoid having a lot of unrelated subjects in the same screen. At the same time, from this screens the HP has the option to create a new PSS or a new adult user in BAS.

The following figures represent the first flow of screens.

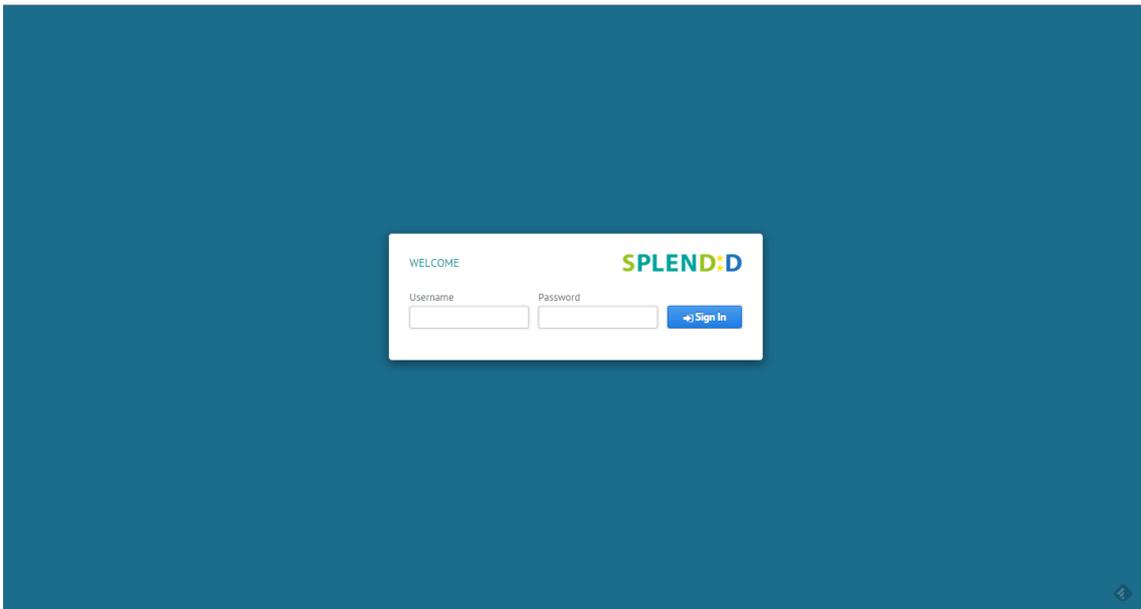


Figure 51 Login screen

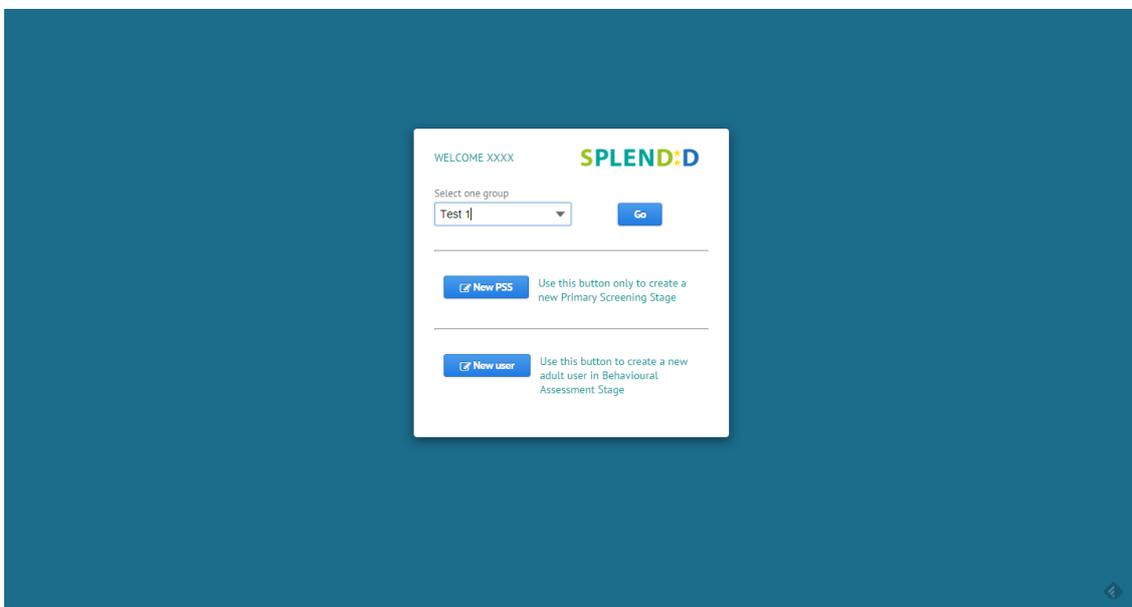


Figure 52 Group selection screen. Also the HP can create a new PSS or new adult rom here

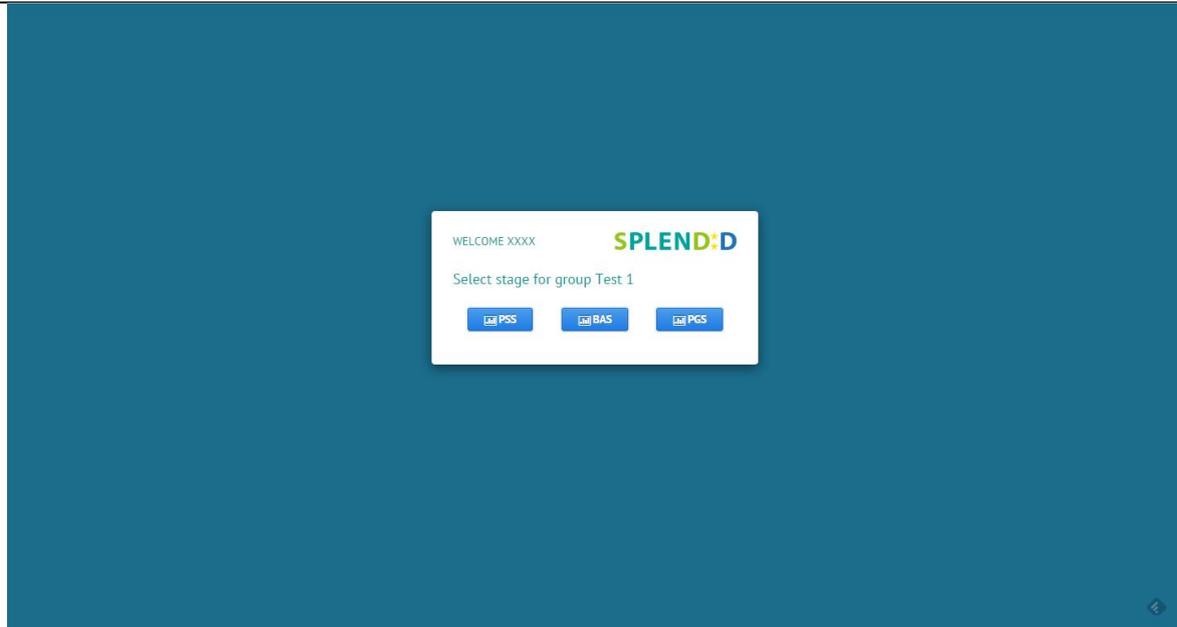


Figure 53 Once the group has been selected, a screen to choose stage is shown.

Use Case 5: Register Primary Screening Stage

As we said before, the HP can create a new PSS from the login set of screens. Also, once the HP is inside the application, the Configuration button offers the options to see the current PSS information or create a new one, so Use Cases 6 and 7 will have the same screen as this one.

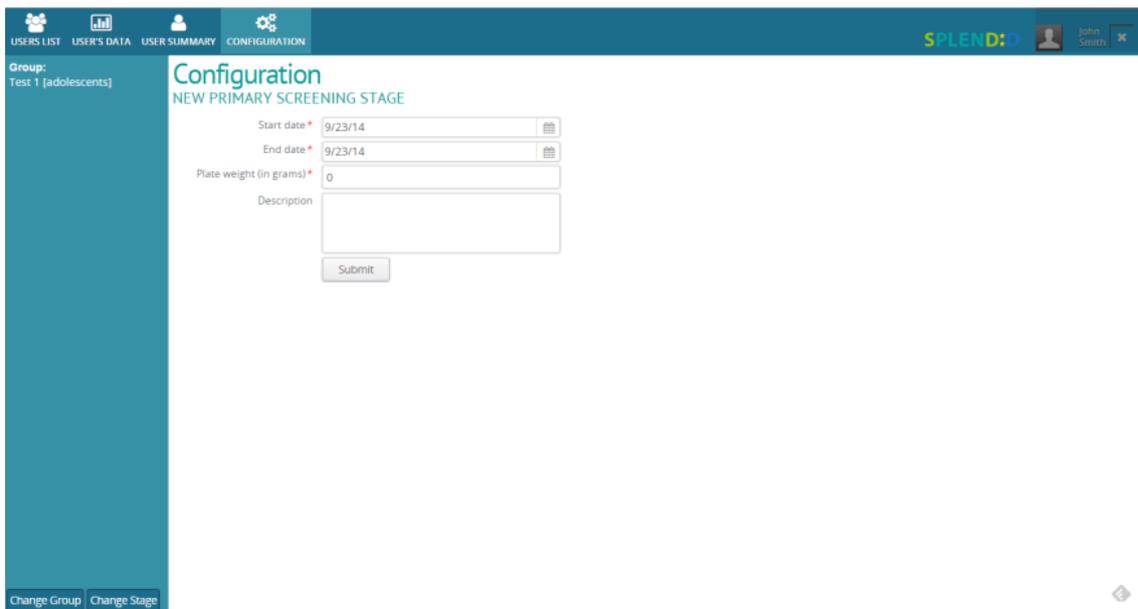


Figure 54 PSS creation screen

Use Case 8: View Primary Screening Event Data

Same as in the previous version of the PWT, in order to see the PSE data of a user, the HP has to select the user from the user's list. Then, by clicking on the list or in the Users Data tab, she will access to the data recorder in the PSE.

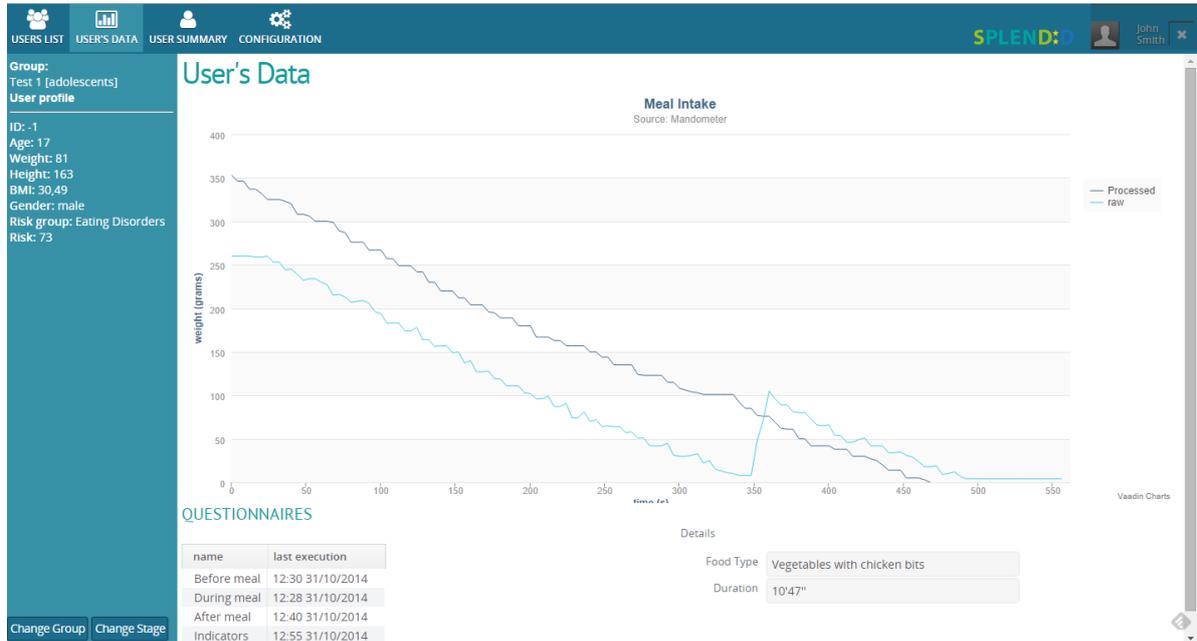


Figure 55 PSE Data screen. This screen shows the Mandometer graphic, together with the questionnaires and the indicators extracted from the meal.

Use Case 9: Confirm/Modify Primary Screening Risk Category

In the User's Profile, the HP is able to assess the risk category calculated by the DSS. To do so, the HP can consult all information about the user in order to do an informed assessment.

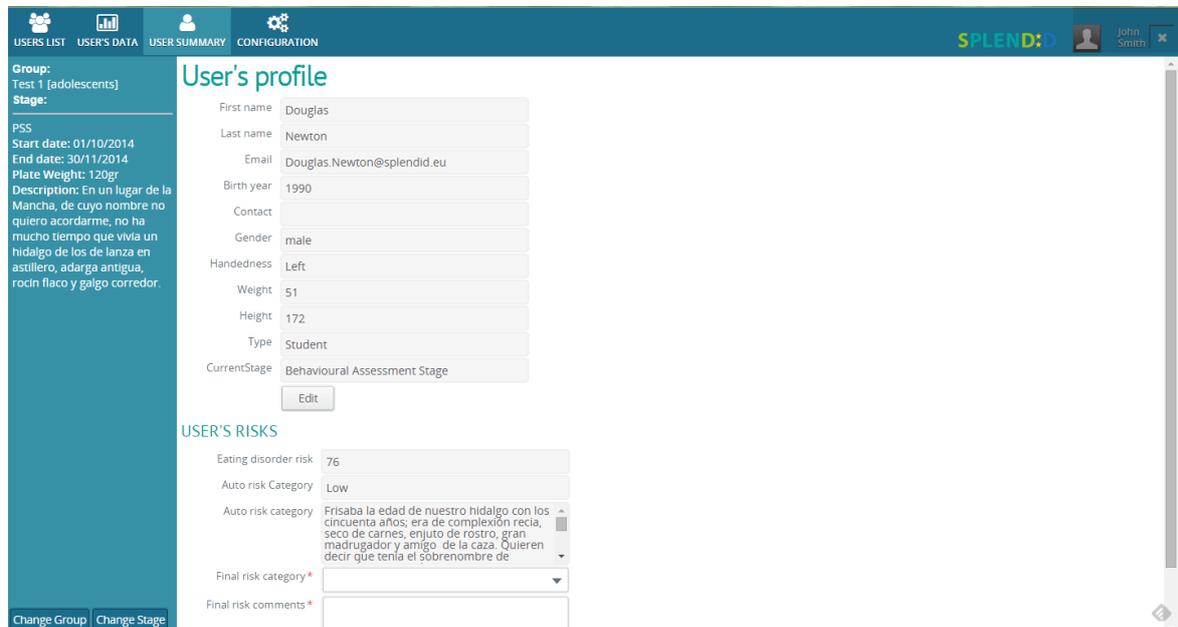


Figure 56 User's profile and risk assessment screen

Use Case 11: Logout from Web Interface

In order to log out from the PWT, the HP has to click in the X button at the top right of the screen.

4.1.3 Access to the instance

The V0.1 of the PWT is available at private area of the development site at this address <https://andromeda.ee.auth.gr:9090/webAppDemo/app#!login> .

4.2 Mobile personal guidance application

4.2.1 Scope of the running instance

As it has already been mentioned in Section 1.1, the intention of WP4 has been to attempt to *implement* the mock-up screens meant for presentation in this deliverable – instead of just generating them with the help of graphics editing or mock-up design software. The purpose of this goal is twofold. First, to speed up the development process of the software applications by working as an additional WP4-imposed milestone. Second, to be used as a demo of the work that has been done in the context of WP4.

Indeed, this goal has been achieved for all the herein presented screens of PGA. This means that the complete screens of all the use cases presented in Section 3 have been implemented in the Android/Java programming language and have been deployed as an Android application. In fact, all the screens that have been included in Section 3 are screen captures of a running instance of this application. The application demonstrates a fully functional version of the basic flow of use cases UC.M.10 – Record Meal with Mandometer, UC.M.12 – Present Sleep/Wake-up Time Questions, UC.M.14 – Present In-day Questionnaire, UC.M.17 – Request User to Put Activity Meter Back on, and UC.M.21 – View Usage Information.

4.2.2 Access to the instance

The demo application can be installed from the apk (Android package) file that is available at private area of the development site at this address <https://andromeda.ee.auth.gr/mobileAppDemo/> . Android 4.4 (KitKat) is required for installing the application. In brief, the installation process is as follows: First, allow the installation of applications from unknown sources by checking the respective box in Android's security settings. Then, use a web browser (e.g., Google Chrome) to download the apk file. Navigate to the “downloads” folder and open the apk file to install the application.

After installation, the application can be initiated by clicking on the app. icon. This opens a navigation screen with 5 buttons, one for each use. Pressing a button starts the demo of the corresponding use case. This is a fully functional version of the basic flow of this use case. This means that the users can use the available buttons, provide the requested input and browse through the provided information, i.e., a close-looking simulation of the real application for the particular use cases. Instructions on how to proceed from each screen are provided through the (i) button on the action bar (top bar) of the demo application.

5 Next steps

There's still a lot of work to be done in the different applications of the complete SPLENDID ecosystem.

The next step is to finish developing the prototypes for MS4, due in month 14. For this milestone, both development teams aim to have a mock-up version of all screens, together with a developed functional version of the applications, so they can be presented at the first year review and help in the implementation of the roadmap of WP4. The roadmap for this stage can be consulted at the current version of Deliverable 5.1. After the review, the development teams will define a development roadmap for the second prototype. This roadmap will be presented at version 2 of D5.1, since we defined that deliverable as a live one that will be adapted to the development stages.

Annex A. Change in base technology for PWT

In July 2014, M10 of SPLENDID Project, TSB proposed to make a change with regards to the original decision to use Nomhad technology in order to produce the Professional Web Tool front end and back end, as described in DoW and in Consortium Agreement.

At that point, all mock-ups and flows of the information designed so far followed the constraints of Nomhad features and capabilities. In fact, the workshops done with end users were based in mock-ups using the original Nomhad look and feel.

The reasons for proposing such a change in the strategy are the following:

1. TSB is currently (July 2014) involved in a corporative process were the access to Nomhad might be affected for products and research opportunities.
 - People with expertise in Nomhad will have strong limitations to participate as researchers to build SPLENDID.
 - If SPLENDID is based in Nomhad there is a risk that the new situation blocks the potential future exploitation opportunities based on that technology.
2. When TSB was still considering using Nomhad, after analysing the proposed features and workflows designed together with the end users (professionals), the estimation of efforts to be implemented showed that Nomhad was being used as a normal web platform instead of re-using its parts as a chronic condition management platform. Therefore, the effort necessary to use Nomhad is similar to the effort of re-using a pre-existing web platform. The risk is that TSB does not have so much experience in using other platforms than Nomhad.
3. Relying in a new framework allows better options for future commercialization in TSB and for the rest of the consortium and ensures that we will get support from the community if we select a good solution.

In summary, after evaluating the risks and effects of the two options (remain with Nomhad or changing to a different the web technology) the decision is to change the web technology given that the benefits are bigger than the original solution and the risks can be mitigated in short time once the appropriate platform is selected.

In the current appendix TSB provides an analysis of the web platforms available in the market. It is highlighted the requirements that Nomhad provided, so we can identify which is the most appropriate platform to be used for SPLENDID implementation of both front end and back end of Professional Web Tool with the minimum impact in the implementation process.

Content Management Systems

According to the 1998 patent, a CMS is a system, including web based systems that organizes the content of the information separately from the appearance of the presented information. When retrieving the information, an assembly process gathers the content in accordance with the document's outline and formats the content in accordance with the document's template. The data structure allows triggers to be associated with content such that automated maintenance procedure can be implemented based on the activation of the triggers.

Although nowadays the most extended definition of a Content Management System (CMS) is a computer application that allows publishing, editing and modifying content, organizing, deleting as well as maintenance from a central interface; a CMS is more a concept than a product itself.

The basic idea is to have a way of organizing content of the web application, no matter if it's data, information or knowledge, in a way that's easy to administer and that facilitates the creation and communication of this content. To sum up, the key goal of a CMS is to increase the integration and automation of the processes that support efficient and effective delivery of the content of a web application.

The main expected business benefits a using a CMS are:

1. Reuse of information by allowing the integration of data from different sources
2. Allow the efficient re-purposing of information, like for instance having the same information on a web page, on PDF, plain text, etc.
3. Allow information maintenance to be delegated but at the same time preserving central control.
4. Ensuring consistency between web pages by separating the design from the content.
5. Permitting non qualified personnel to fill the web application with information.
6. Facilitating good information management practice so the appropriate metadata are capture at the time the content is created or modified, thus allowing the search engines to index the appropriate information from the web.
7. Allow to restore the web application to a previous state.

There are several existing CMS frameworks based on Java technology, the most remarkable are:

- Hippo CMS has Apache 2.0 license. It provides a highly configurable and scalable architecture, since all content is stored as documents (not pages) and the HST mapping system allows that content to be routed and transformed to various domains, websites, published modes (published v preview), and languages.
- DotCMS has GPL3.0 license. DotCMS delivers responsive, theme based templates that automatically adapt and optimize your content, images and videos to fit mobile devices. Custom themes can be developed to leverage native device capabilities such as geo-location services based on a standards based database of device attributes (WURFL). It also provides easy and open APIs to query, retrieve (xml, json) and write content, binary assets and even remote widgets.
- Magnolia is focused on flexibility and ease-of-use for enterprise business users and developers. It has a Community Edition, with GPLv3 license, and two types of Enterprise editions.
- Jahia delivers web content integration software by combining enterprise web content management with document and portal management features. Jahia is content centric depending on the type of project you envision, this is a major difference under GPLv2 license
- Liferay provides a single presentation layer for integrating all enterprise systems into a single easy to use interface for end users. Liferay Portal is a central presentation layer platform that allows users, administrators and developers to integrate content and

services from backend or legacy applications. Liferay supports multiple methods to integrate, including SOAP, REST, RSS, as well as proprietary API's.

According to *Builtwith*¹⁰, this is the distribution of websites that use the analyzed CMS. One thing to remark is that Hippo CMS does not appear in *Builtwith*, so we don't have usage statistics of this CMS, although it appears in every list of the most popular java CMS we have found in the web.

Active Websites

The active sites within our system on a 2 week moving average.

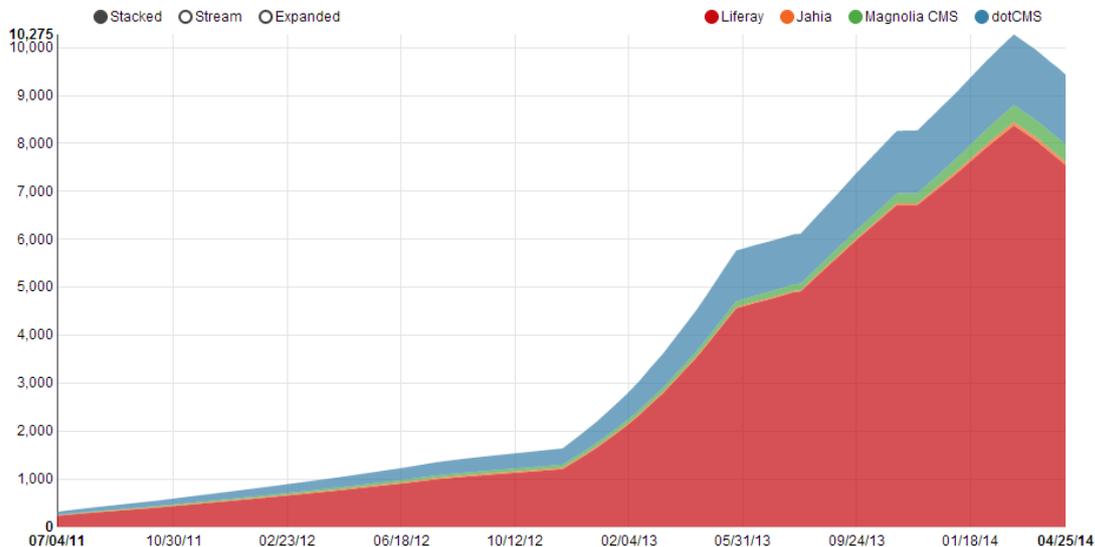


Figure 57 CMS usage comparison

Web Framework

A web application framework (WAF) is a software framework that is designed to support the development of dynamic websites, web applications, web services and web resources. The framework aims to alleviate the overhead associated with common activities performed in web development. For example, many frameworks provide libraries for database access, template frameworks and session management, and they often promote code reuse.

There are different types of framework architectures:

- **Model view Controller (MVC):** Most of the frameworks follow this pattern in which the data model and the business rules are separated from the user interface.
- **Push-based vs pull-based:** The first kind use actions that do the required processing, and then "push" the data to the view layer to render the results. The second one starts with the view layer, which can then "pull" results from multiple controllers as needed. In this architecture, multiple controllers can be involved with a single view.
- **Three-tier organization:** applications are structured around three physical tiers: client, application, and database. The application contains the business logic, running

¹⁰ <http://builtwith.com/>

on a server and communicates with the client using HTTP. The client, on web applications is a web browser that runs HTML generated by the application layer.

- The main features that different frameworks share are:
- **Web template System** as a way to combine different web templates to form a complete page possibly using some data source to customize the pages or present a large amount of content on similar-looking pages.
- **Caching** of web documents in order to reduce bandwidth, server load and perceived lag.
- **Security** with authentication and authorization methods to allow the server to identify users of the application.
- **Database access, mapping and configuration** usually through an API which allows the applications to work with different databases with no code changes.
- **Scaffolding** is a technique supported by some model-view-controller frameworks, in which the programmer can specify how the application database may be used. The compiler or framework uses this specification, together with pre-defined code templates, to generate the final code that the application can use to create, read, update and delete database entries, effectively treating the templates as a "scaffold" on which to build a more powerful application.
- **URL mapping** as a mechanism to interpret URLs. A URL mapping system that uses pattern matching or URL rewriting allows more "friendly URLs" to be used, increasing the simplicity of the site and allowing for better indexing by search engines.
- **Ajax** support
- Tools for creating and providing **Web services**
- **Resource-oriented Architecture (ROA)** is a style of software architecture and programming paradigm for designing and developing software in the form of resources with "RESTful" interfaces. These resources are software components (discrete pieces of code and/or data structures) which can be reused for different purposes.

Some of the most well known Java frameworks are, Google Web Toolkit, Spring MVC, Grails, Vaadin and Play.

In *ZeroTurnaround*¹¹ they have elaborated a report comparing different java frameworks.

As can be seen in the graphic, by looking at the features, the Grails frameworks obtains the best score, followed by Vaadin, which is the winner if we look at how good is each framework for each type of web app.

¹¹ <http://zeroturnaround.com/rebellabs/>

	Rapid Application Development	Framework Complexity	Ease of Use	Documentation & Community	Framework Ecosystem	Throughput/ Scalability	Code Maintenance/ Updates	UX/Look and Feel	Testability	Security	Totals
Spring MVC	2.5	3.5	3	4	4	4	3	2	4	5	35
Grails	5	3	4.5	5	4.5	4	4.5	4	5	2.5	42
Vaadin	4.5	4	4.5	5	3	4.5	4	5	3	4	41.5
GWT	4	4	4	4.5	3	4.5	4	5	2.5	1	36.5
Wicket	3.5	2.5	3.5	3	3	3	4.5	3.5	4	5	35.5
Play	5	2	3.5	4	4.5	5	4	3	5	4	40
Struts	2	4	3	2.5	3	3	3	2.5	4	2.5	29.5
JSF	3	3.5	4	4.5	4	4	4	4.5	2	2.5	36

Figure 58 Scores for the different categories analyzed

Framework scores for all features (raw) and averages (weighted) across app types

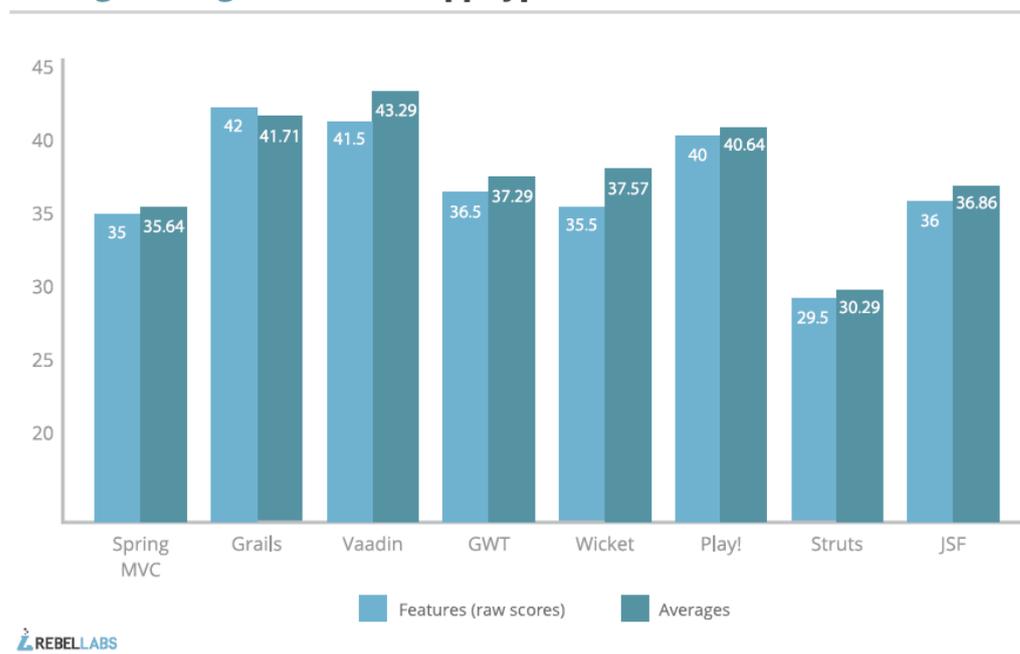


Figure 59. Scores compared by features and apps

CMS vs. Framework

Previous sections describe the CMs and frameworks as tools that can be used to develop web applications. This sections aims at exploring the difference between them and how to choose the best option for the development.

To sum up, a CMS is a program that allows you to create a support structure for creating and managing web content. It is a platform that contains a set of software tools that control one or more databases, where the content is hosted, allows manage and uniformly accessible

dynamic website, providing an environment to enable the submission, maintenance, expansion and periodic updates, with the collaboration of multiple users either through Internet or Intranet accessing the manager through the web browser.

A framework is a support structure defined object-oriented software, customizable and interchangeable composed for the development of application components. It can be considered as incomplete and generic configurable application that can add the final pieces to build a particular application. Contains a set of concrete and abstract classes that are generic and reusable application for a specific application domain; typically may include support programs, libraries and an interpreted language among other software to help develop and unite the different components of a project.

Using a framework can accelerate the development process, the ability to reuse code so increased productivity and time short market in application development is obtained, it also promotes good development practices as the use of patterns as is MVC.

As it is described, a framework and a CMS are very different things that can be used to develop web applications. A CMS allows you to create a web application in a short time, mostly based in configurations of users, roles, database access, workflows and information to be shown. Meanwhile, using a framework to build a web app will require more time, but it will allow you to create more custom and scalable web applications and connect more easily with third parties.

Choosing one or another depends mostly in what you need to develop and how much time you have to do it. One possible way to choose between using a framework or a CMS is to follow this advises:

Choose a CMS if:

- You know the site won't grow much
- They have very few complex requirements
- They are very low on budget

Choose a framework if

- You know the site will need custom functionality eventually
- They have mission critical needs
- They have to integrate with other services
- They need something custom from the very start.

Choosing technology for SPLENDID

Requisites

The aim of SPLENDID is to develop a web application for the health professionals, so they can consult the information of the end-users gathered from the devices (Mandometer, Activity Tracker and Chewing sensor). Also, the HP should be able to define goals for the latest stage of the pilot, the Personalized Guidance Stage.

In order to fill these requirements, the requisites that the chosen CMS or framework must fulfil are:

1. Easy database integration and persistence
2. Web services implementation
3. Security: authentication and authorization, different views depending on the user

4. Definition of workflows for pages, data and end-users (from PSS->BAS->PGS)
5. Administration for users/devices/stages, etc
6. Reutilization of information
7. Repurpose of information
8. Separation between content and design
9. Open source
10. Java based

Once the requisites have been define, it is possible to study different frameworks and CMS in order to decide which one is the best option to be used in SPLENDID for the development of the PWT.

Technology Selection

In this section we compare the requisites we have with the analyzed technologies, in order to decide which one to choose. Las two requisites are out of the table, since all technologies analyzed are java and open source.

Requisite	Hippo	DotCMS	Jahia	Magnolia	Lyferay	Spring	GWT	Grails	Vaadin	Play
1	X	X	X	X	X	X	X	X	X	X
2	X	X	X	X	X	X	X	X	X	X
3	X	X	X	X	X	X			X	X
4	X	X	X	X	X	X		X	X	X
5	users	Users	Users	Users	users					
6	X	X	X	X	X	X			X	
7	X	X	X	X	X	X	X	X	X	X
8	X	X	X	X	X	X	X	X	X	X

Table 2 Checklist of requisites

As can be seen in the table, all analyzed technologies comply with most of our requisites.

The CMS fill all our requisites, except the administration of devices and stages, but that's something we already guesses we would have to implement. The major drawback of using a CMS is that, although they look like they fill our needs, they are usually more focused for other types of web applications, like blogging, corporate or e-commerce. In our case, the web application is not only a way to show information. We will have a big amount of new data every day and we will have to do some processing over that data.

Also, the CMSs don't give you the control of the database, since they create their own tables, as a framework in which you are creating your own database from scratch. Given the amount of information we have, a well design database is important to maintain consistency and speed up the work with the database.

Regarding the frameworks, Spring and Vaadin are the ones that fill most of our requisites.

The drawbacks of choosing a framework are that we will have to develop all the business logic of the applications, while a CMS usually takes that away. Also, it is faster to create a UI and an administration site with a CMS than with a framework, in which we have to develop everything.

Given the advantages and drawbacks, and taken into account the advices in section 5, we believe the best option to develop the web application for SPLendid is to use Vaadin framework. This way, we will have absolute control over the database and the business logic, and at the same time we have a tool that facilitates the implementation of security, interfaces, web services, etc.

Annex B

This annex present pictures of the workshops that took place in Stockholm.

