SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF BIRMINGHAM

Web-based Interactive Intervention Programme for Family Members of Addicts

Miel Zozaya Garcia

MSC in Internet Software Systems

Supervisor: Dr Behzad Bordbar

13/09/2010

ABSTRACT

This project aims to develop a web application for the Alcohol, Drugs and the Family (ADF) group. The ADF group already holds a website that implements the 5-step self-help programme that the group developed. The developed web application aims to improve the support the support that nowadays is provided to the family/network members of addicts. The aim is fulfilled by improving the 5-step self-help programme, by insertion of multimedia content, and building extra functionalities in top of the programme that provide better support to the users. These functionalities include, among others, a forum for the users, the supervision from part of a professional, downloadable content for the professionals.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF CODES

1. INTRODUCTION

"Web-based Interactive Intervention Programme for Family Members of Addicts" is a software project that has been developed for the Alcohol, Drugs and the Family (ADF) group. The group's contact person has been, for the lifetime of this project, Mr Akan Ibanga from the Psychology Department of the University of Birmingham. This project consists of a web-application that, as its central selling point, implements the 5-step self-help programme that was created by the ADF group. The special thing about this programme is that it does not help those with alcohol or drugs problem. Instead this program supports family and network members that are concerned and/or affected by its relative's alcohol or drugs misuse.

At present, ADF group holds a website that implements the 5-step self-help programme. The problem of this website is that it has little functionality as it only implements the 5-step programme. The programme by itself is proven not to be so effective because the communication with other people it is for most of the patients essential. Due to this the project client desires a web-application where the 5-step self-help programme is implemented and surrounded by further functionalities that will better support the clients of the website.

In the following sections of this document the project is further analysed and explained. In section 1.1 further information on the ADF group and the 5-step programme is given. In section 1.2 the current website is analysed for later, in section 1.3, analyse the problematic that conducted to this project and objectives of it, in section 1.3.1. In section 1.4 the achievements of this project are stated and in section 1.5 the planning of the project is presented. Section 2 explains the project in detail with its requirements, solution design and implementation among others. Finally, in section 3 the reader can find the conclusion of the project. The file structure of the provided CD together with guidelines to run the prototype can be found in APPENDIX D.

1.1. PRELIMIRARIES

Referring the official website of ADF (Alcohol, Drugs and the Family Self-Help Manual n.d.), the Alcohol, Drugs and the Family group is formed by clinicians and researchers and it was funded in 1980s. The original funders of the group are Richard Velleman and Jim Orford, who are still working members of the group. Although the group has been based in different places during its existence, nowadays its core is divided between the cities of Bath and Birmingham, including surrounding National Health Service (NHS) and academic institutions such as the University of Birmingham.

The ADF group is focused on; underlining the impact that alcohol or drugs problem of a family member can have on other members of the family, creating methods of helping those family members and increasing the visibility of the research and good practices in the area. In this order ADF group created the 5-step method. The method is composed, as it name says, by 5 steps:

1. *About you and the problem you are having:* This step of the programme helps the family members to think more about their situation and the problems that they are having. Family members, at the end of this step, should be able to identify the stress and strains relevant to their situation and the health problems that they may have.
2. *Increasing your knowledge and understanding:* This step aims to increase the family members' knowledge and understanding on drugs. This information helps patients to feel clarify their own fears, solve misunderstandings that they may have and reduce the stress that they are having.
3. *Ways of Responding:* This step of the method makes the patient to think how he responds and copes with his relative's behaviour. By taking this step the family member should reach to identify which is the way to cope that helps them and their family most.
4. *Getting help from the others:* On this step the client thinks about the persons that may help them to better deal with their relative's problem. The amount of help that a client might have can largely influence the way that he responds to his relative's behaviour and the amount of stress that they are experiencing.
5. *Getting further help:* This step takes a look on further help that the family members may need in related issues, such as, domestic violence or the impact that the situation may have on their children.

The 5-step method has been putted intro practice by many health professionals mainly in UK but also in other countries such as Italy (Velleman, et al. 2008). Results of the made evaluations on (Orford, Patel, et al. 2007) and (Orford, Templeton, et al. 2007) show that the majority of the patients that take the 5-step method experience positive transformations of their situations. The most common transformations experienced by the patients are the following:

1. *Focus on own life and needs:* The most frequent transformation is that patients increase the independence from their relative's problem. The patients start to think more on themselves and their needs that consequently translate to having a better life condition.
2. *Being assertive:* Family members express that they start being more assertive with their relatives. They become more assertive by communicating more directly with their relatives, being firmer in maintaining a course of action or by being more active in organizing alternative activities.
3. *Calming down:* Patients explain that thanks to the programme they calm down and it helps them to be less emotive when interacting with their relative.

4. *Seeing the links:* Patients gain understanding about their relative's alcohol or drug problem and realize about the links between the relative's problem and their physical or mental health.

The studies made in (Orford, Patel, et al. 2007) also say that the majority of the patients expressed a preference for professional contact. The reason for this preference is that it results very helpful for the patients to talk to "someone who cares" or "simply talking" about their problem. In this sense it is shown that the 5-step method is much more useful when having contact with a professional or someone when using it as it makes patients "feel worth something", "not alone", "not a failure".

1.2. BACKGROUND

At present, as previously mentioned, the ADF group holds a website that implements the 5-step self-help manual. The website uses PHP as implementing language. A snapshot of the mentioned website is show in Figure 1.

The manual implements the steps of the 5-step method presented on section 1.1. Each of the steps contains relevant information and one or more exercises for the patient. These exercises have the form of questions that are answered by the patient. Answering the questions gives him more insight on specific aspects of his situation. Additionally, users can take the self-help programme in order from the first step to the last or can choose to take steps that interest them most.

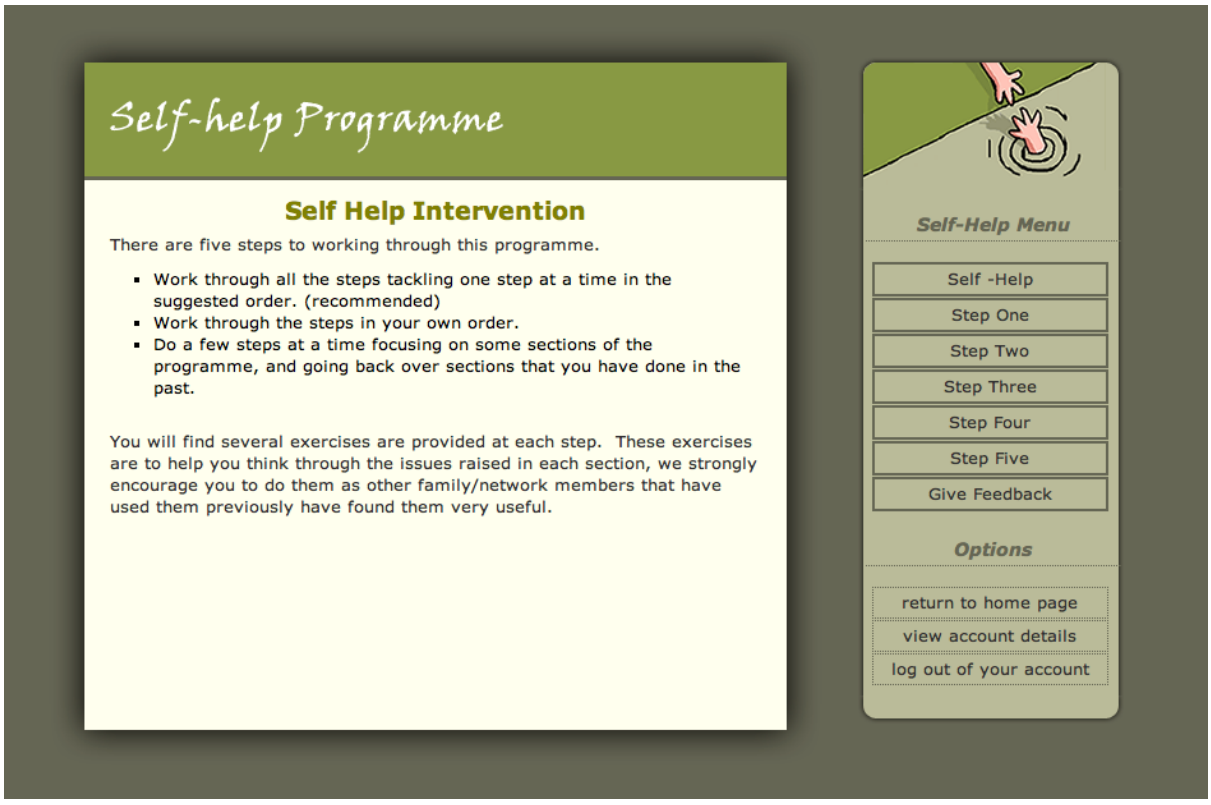


Figure 1: Snapshot of ADF group's website (at present)

However, the website does not provide the user any extra functionality apart from the commented one. Users do not get any response from the website. In that sense the website is limited as the user only reads and answers the questions that he is prompted with no other interaction with the website or participants of it.

1.3. PROBLEMATIC

The actual version of the website is a properly working system that has a few hundred registered users. The feedback provided by users generally suggests that the system is useful and that it gives them the help they need. However, a further research made by the project client on the activity of the users showed that:

- Most of the user activity was created by a very short percentage of the users.
- Most of the users quit using the system in few months and never use it again.

What this research showed to the project client that most of the users do not find the website interesting enough to keep using the self-help support system. A discussion followed this research in order to find the most plausible reasons for the problem. First of the main reasons is that the website has few functionalities as it only provides the user with the self-help support system. This support system is a stand alone system that can create a sensation of loneliness. In this self-help support program the user reads information and answers does exercises that consist on answering questions. In this sense, the user could feel that his answers and progress are going nowhere and that it does not have further support.

The second main reason which is tightly related to the first one is that the system lacks of interactive sensation. As commented the user does not get any communication from the system so that he is the only one participating. Moreover, the self-help support program is linear and it does not give any option to the user. This lack of interactivity does any system less engaging.

In third position, the self-help support is at the same says self-help. This means that the person having a problem helps themselves. Actually, many people in a problem of such kind may be able to help themselves with the help of a support program but there are people that require further direct help from another person.

The forth and last reason is the attractiveness of the user interface. Is a fact that the human been gets engaged by what is considered as ‘nice’ user interface and that likes more multimedia content than text one. In this sense the actual system should be made more attractive for the users.

1.3.1. OBJECTIVES

The objective of the project is to provide the project client with a web application that solves the problems stated in section 1.3. For doing so this project aims, first, to improve the self-help support program of the previous website and, second, to surround this main feature of the website with new features. The goal of the improvement and the extension is to increase the website’s interest, interactivity, attraction and, mainly, level of support.

In order to improve the self-help support programme two sub-objectives are defined:

- Insertion of multimedia content.
- Provide the programme with a 'game' sensation by giving options and level end rewards.

The second main objective of the project is to surround the self-help programme with new features:

- The first of these features is to provide the clients with the option to communicate one with the other facilitating for them a discussion forum. This functionality will allow users to share their experiences and to ask questions and get answers for them. By this is believed that users will not tend to fill alone and that will find that someone cares for them.
- The second is the insertion of a new user; the Professional. This user allows a professional in psychology to have a list of clients that he can supervise in their progress in the 5-step self-help programme as well as being able to communicate with them.
- The third is to give the mentioned Professional user the ability of downloading specialised tools provided by the ADF group. By the use of specialised tools, the professionals can be more effective on supporting their clients.
- The fourth is to redesign and improve the visual interface used at present.

It is believed that all this functionalities once implemented will give the clients better support since, in addition to the more engaging self-help manual, the surrounding functionalities will help them not to feel lonely, to have contact with someone who cares and to receive better help from their professional. In conclusion, the new system will provide them a better surrounding environment.

1.4. ACHIEVEMENTS

The achievements of the project are here stated so the reader can have a fast reference to them:

- The user interface has been redesigned and improved; section 2.4.1.1, page 32.
- The 5-step self-help step has been improved; section 2.4.1.3, page 34.
- A Client's discussion forum have been implemented; section 2.4.1.5, page 37.
- Clients can now be supervised by psychology professionals; section 2.4.1.6, page 37.
- Professionals can now download tools for further helping his clients; section 2.4.1.7, page 38.
- Administrators are provided with an automated generation of Microsoft's style sheets files from database information; section 2.4.1.9, page 40.

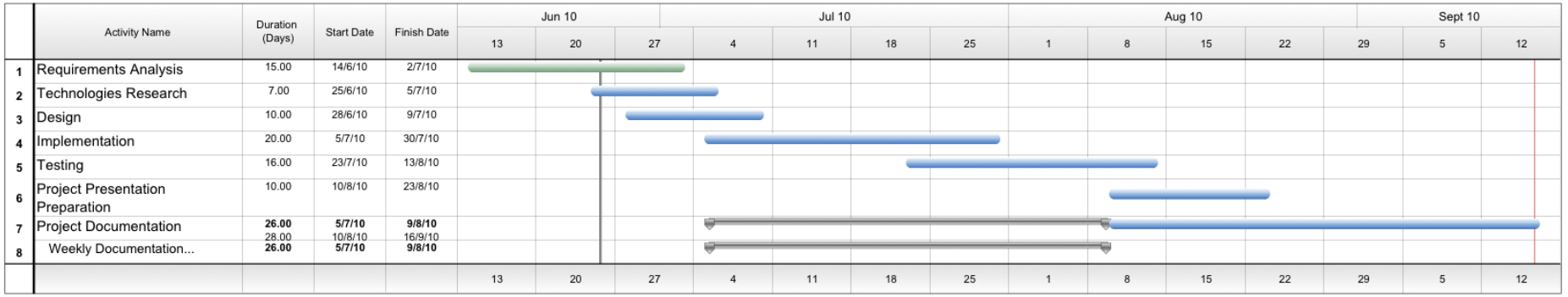
1.5. PLANNING

In this section the planning of the project is presented. In Figure 2 it is shown the original planning made at the beginning of the project. The schedule is composed by 7 main tasks:

- *Analysis of the requirements of the system:* To compose the system requirement document the project client was interviewed twice. More details on requirements can be found in section 2.1.
- *Technological research:* Research that intends to find the better technology to accomplish the requirements. More details on the research can be found in section 2.2.
- *Design:* A task that is much related to the technology or technologies selected on the previous task. In this task the solution design is developed. More details on the solution design can be found in section 2.3.
- *Implementation:* This task involves the coding of the design making use of the technology or technologies selected. More details on the implementation can be found in section 2.4.
- *Testing:* This task comprises testing the implemented code, in order to find faults on it and make the code robust by solving them.
- *Project Presentation Preparation:* This task involves the preparation of presentation slides, pitch and demonstration demo. The slides of the presentation can be found in the cd that accompanies this report.
- *Project Documentation:* This task comprises documenting the progress of the project and final report writing.

The schedule it has being designed with tight deadlines. This type of scheduling has resulted very convenient since it has allowed rescheduling deadlines to accommodate unexpected events. Specifically, 3 unexpected events have occurred during the project. The most significant unexpected event occurred on 22nd of July 2010. On that date, during a meeting with the Project Client, new requirements where added to the project. This required rewriting the requirement document, redesigning the solution design and adding extra functionalities to the implementation. Due to this event it was decided to merge the implementation and testing together and, in addition, to extend the deadline until the 25th of August 2010.

The other two unexpected events can be found in APPENDIX A. In this appendix, reports of the meetings that the Project Developer had with the Project Client and the Project Supervisor can be found.



	Activity Name	Duration (Days)	Start Date	Finish Date
1	Requirements Analysis	15.00	14/6/10	2/7/10
2	Technologies Research	7.00	25/6/10	5/7/10
3	Design	10.00	28/6/10	9/7/10
4	Implementation	20.00	5/7/10	30/7/10
5	Testing	16.00	23/7/10	13/8/10
6	Project Presentation Preparation	10.00	10/8/10	23/8/10
7	Project Documentation	**26.00** 28.00	**5/7/10** 10/8/10	**9/8/10** 16/9/10
8	Weekly Documentation...	**26.00**	**5/7/10**	**9/8/10**

Figure 2: Project Schedule

2. PROJECT DEVELOPMENT

2.1. REQUIREMENTS

In this section the system requirements are presented. In the process of specifying those requirements the Project Client has been interviewed. In a first place two meetings took place; one for writing a first draft of the requirements and the second for presenting the written document and correct it. Later, as explained in section 1.5 and reported in "Unexpected event 2" in APPENDIX A, the system requirements were extended. Following the system users and the last version of the requirements is presented.

2.1.1. System Users:

- *Browser:* The Browser is the user of the system that does not have further permissions than browsing.
- *Client:* The Client is the user of the system that needs or is seeking for help/support/advice. It is the main user in which the system is focused on.
- *Professional:* The Professional is a professional in psychology that is a separate entity of the system.
- *Administrator:* The Administrator is the user that will administrate the website. Its administrative rights go from content modification to user access privileges control.

2.1.2. Users Requirements:

2.1.2.1. Browser requirements:

1. The Browser must be able to see information pages (such as home, about ADF and users feedbacks).
2. The Browser must be able login on the system, in the case that it has correct pair of username and password, and gain permissions of the referred user.
3. The Browser must be able to register on the system in order to gain Client permissions. In the registration process he must provide a username, an email, a password and a telephone number, give demographic information (see Demographics.doc in CD) agree with the 'Terms & Conditions' and answer to three questionnaires (see Coping questionnaire.doc, FMI.doc and Symptom Rating Test.doc in CD).
4. The Browser must be able to request Professional permissions.
5. In the request he must have to provide his identity (name, surname), a contact email, further contact details, the identity of the organization that is working for, the position that holds in the organization, affiliation information and a statement of reasons for the request.
6. In the case that the request is accepted he must be able to register as a Professional. During this registration process he must provide a username, an email, a password

and a telephone number, give demographic information (see Demographics.doc in CD) and agree with the 'Terms & Conditions'.

2.1.2.2. Client requirements:

7. The Client must be able to use the 5-step self-help program.
8. The Client must be able to enter feedback on the 5-step self-help program.
9. The Client must be able to see its answers to the exercises of the 5-step program.
10. The Client must be able to prints its answers to the exercises.
11. The Client must be able to read and post messages and respond to them on a clients' discussion forum.
12. The Client must be able to use a nickname of his choice to post messages on the clients' discussion forum.
13. The Client must be able to communicate with his Professional, in case that is linked to one.
14. The Client must be able to change its password.
15. The Client must be able to request a new password if he forgets the previous one.
16. The Client must be able to choose how he wants to be contacted (email or post). In the case that he chooses to be contacted by post it must enter his address.
17. The Client must be able to deny its professional to see his exercise information, in the case that is linked to one.
18. The Client must be able to enter feedback of the website.

2.1.2.3. Professional requirements:

19. The Professional must be able to use the 5-step self-help program.
20. The Professional must be able to enter feedback on the 5-step self-help program.
21. The Professional must be able to see its answers to the exercises of the 5-step program.
22. The Professional must be able to prints its answers to the exercises.
23. The Professional must be able to invite his clients to use the web-application in a way both of them link together.
24. The Professional must be able to see and print the answers of the Clients that are linked to him.
25. The Professional must be able to see the questionnaire score of a client of him.
26. The Professional must be able to communicate with Clients that are linked to him.
27. The Professional must be able to read and post messages and respond to them on a professionals' discussion forum (optional).
28. The Professional must be able to use a nickname of his choice to post messages on the professionals' discussion forum.

29. The Professional must be able to download content uploaded by an Administrator.
30. The Professional must be able to change its password.
31. The Professional must be able to request a new password in the case that has forgotten the previous one.
32. The Professional must be able to enter feedback of the website.

2.1.2.4. Administrator requirements:

33. The Administrator must be able to access clients' and professionals' discussion forums.
34. The Administrator must be able to edit messages from the clients' or professionals' discussion forum.
35. The Administrator must be able to add/remove a category from the messages in clients' and professionals' discussion forums.
36. The Administrator must be able to change access rights of the users.
37. The Administrator must be able to accept Professional registration requests.
38. The Administrator must be able to upload and remove uploaded content from the website.
39. The Administrator must be able to see the 5-step self-help feedback of the users.
40. The Administrator must be able to see the website feedback of the users.
41. The Administrator must be able to edit the 5-steop self-help content and exercises.
42. The Administrator must be able to download in a .xls document:
    - Demographic information of the users.
    - Questionnaire answers of the users.
    - Self-help Program activity history of the users.
    - Exercise answers of the users.

2.1.2.5. System requirements:

43. The System must encourage the user to think more after answering the exercises. It is suggested this to be in form of pop-ups. As extra functionality sound could be used.
44. The System must verify that the email provided by a user is a working email.
45. The System must show a score to the questionnaire filled by clients and compare it with the average score of all the clients on the system. After the second time that a client answers the questionnaire the system must also compare the score with the client's previous scores and give an advice.
46. The System must give clients the possibility of retaking the questionnaire every three months. Until this is not retaken the system must prevent the use of its privileges to the client.

47. The System must advice clients on using 'private browsing', and gives tips on how to configure the major browsers (Internet Explorer, Mozilla Firefox, Safari, Google Chrome) to do so, when using the website.
48. The System must enforce that for the use of a discussion forum the user has agreed with a certain consensus.
49. The System must notify a user if a message of his is edited.
50. The System must notify the participants of a message thread if the category of the message is added/removed.
51. The System, after a client finishes one step, must show the options of 'take a brake', 'do it again' and 'next step'.
52. The System should ask the users to fill a captcha in the registration form. (Optional)
53. The three questionnaires must be modifiable. The questions, answers, punctuation of them and evaluation of the questionnaires can be modified. The system must provide an easy way of modifying it.

2.1.2.6. Visual requirements:

54. The website should be visually attractive.
55. At the beginning of each of the 5-steps of the program a video must be displayed.
56. The exercises should be enriched with visual effects at the beginning of steps and after accomplishing exercises or steps that would engage and reward the user. (Optional)

2.2. TECHNOLOGICAL RESEARCH

Nowadays, the market offers many languages and frameworks for the development of web applications. Most used programming languages are PHP, Java and ASP.NET. It can not be said that one is better that the other as all have their strengths and weaknesses.

Currently, the ADF group's website is programmed using PHP. Nevertheless, for the development of this project Java language together with Apache Wicket framework have been chosen. The main reason that pushed to this decision is that Apache Wicket provides a framework that allows to program web applications by completely following object orientated (OO) java programming. In addition, for the management of the database Hibernate framework is used and for putting all the components that compose the system together Spring Framework is used.

Following sub-sections give details on why these technologies are used and what benefits do they provide.

2.2.1. Apache Wicket VS PHP

PHP is widely used in the Internet. As said on (The PHP Group n.d.) PHP is a scripting language that was developed entirely focused on web programming. This programming language implements the application logic into the same pages that contain the

presentation of the document. Apache Wicket provides an improvement on this aspect. Referencing (Dashorst and Hillenius 2008) in Wicket the web application is just Java and just HTML using meaningful abstractions.

In Wicket, pages and components are created by using regular Java constructs. For creating a page a common java class is created. Inside this class components can be created using the *new* keyword and hierarchies can be created by adding child components any component. This allows treating the pages and components as desktop SWT frames and components. This gives the web pages the same flexibility that Java provides, such as inheriting classes.

The presentation of the pages is written in clear HTML templates. These templates do not include any kind of scripting intro the code. They use plain HTML. For placing the components defined in the Java classes *wicket:id* attributes are used in the mark-up. This provides clean HTML documents that can be designed by separated web designers that do not need to understand or know any other programming language.

Finally, wicket provides abstractions of all the widgets that can be found in a web page, such as buttons, links, drop-down choices etc. It also provides abstractions for web sessions, validators, web requests etc. This makes even easier to develop web pages as Java UI's would be. Moreover, the developer can also create its own components and simply instantiate them in the web pages classes.

2.2.2. Hibernate

There is well know in the area of Computer Science the mismatch problem between the object model used for implementing applications and relational database model used for the data storage. These two models do not correlate perfectly together and in common scenarios there is very much time expended on coding the transformation from the objects to relational tables and back.

In order to skip this mismatch problem Hibernate is used. Hibernate, (JBoss Community n.d.), provides object/relational mapping. It lets to develop persistence classes using natural OO idioms. The managing of persistence objects is done in a transparent way once the developer has provided the mapping of a class and its relational representation. This means that from that moment on the developer will only make use of objects; he will save objects, load objects, find objects with that have certain attribute values etc. So using this framework significantly reduces the time spent on database management.

2.2.3. Spring

Referencing (SringSource community n.d.), Spring provides "centralized, automated configuration and wiring of your application objects". It improves applications testability and scalability as components can be independently tested and then wired to the application through Springs configuration file. Moreover it also provides a common abstraction layer for transaction management, integration with Hibernate and AOP functionality among others.

All this features simplify the work of the developer as from a single file it can control how the components are configured, how those components wire together the transaction management of the database etc.

2.3. SOLUTION DESIGN

This section presents the solution design of the project. First, the requirements stated on section 2.1 are analysed for identifying the system's use cases. The identified use cases respond to the functionalities needed to be implemented in the project's resulting product. Use cases diagrams and use cases descriptions can be found in section 2.3.1 (some of the descriptions can also be found in APPENDIX B).

Second, a high level architecture of the system is presented in section 2.3.2. The architecture presented on this section aims to design a modular and scalar system by dividing the different responsibilities of the system in components. Section 2.3.2 also presents the system's most relevant components and the class diagrams of their internal composition. In addition, APPENDIX C contains more specific class diagrams that are not included in the reports main body.

2.3.1. Use Cases

In this section the use cases derived from the requirements are presented. The general use case diagram is displayed in Figure 3. The actors that appear in the schema correspond to the users described in section 2.1.1.

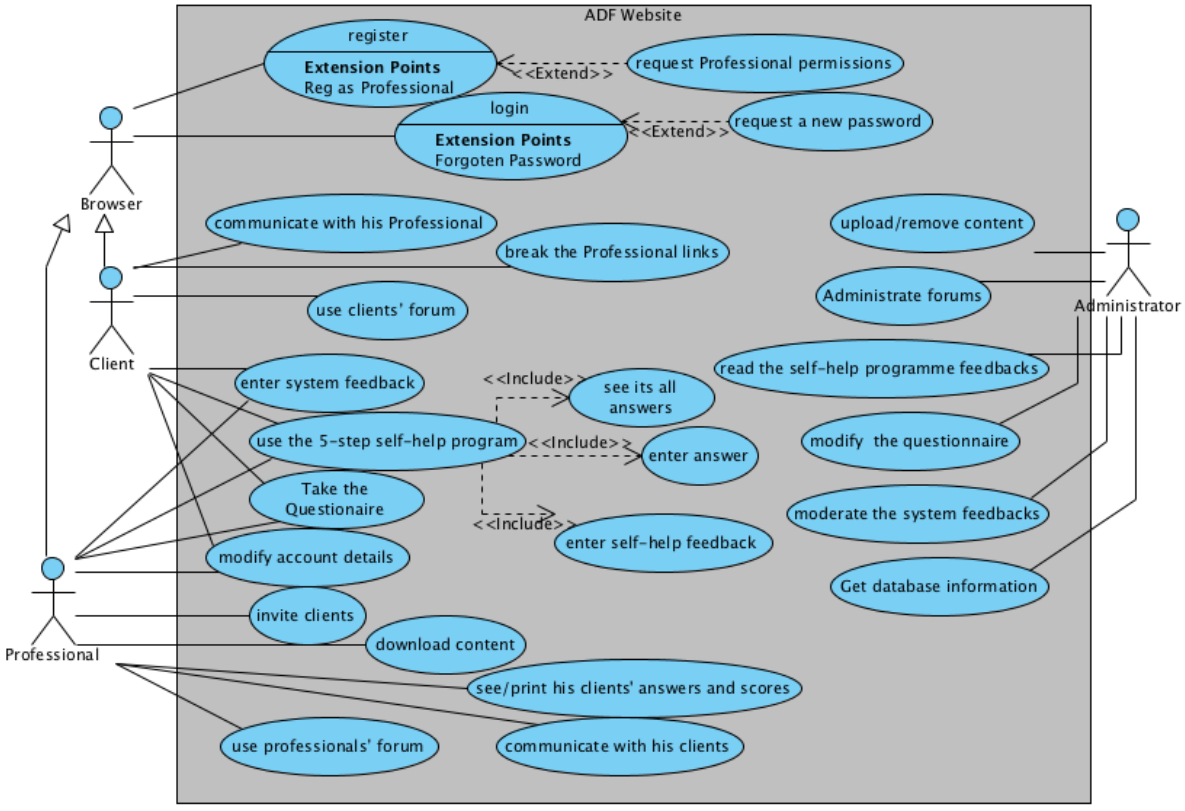


Figure 3: Use Case Diagram

In the next sub sections the most relevant use cases are presented (the rest of use cases can be found in APPENDIX B). The use cases are organized in different sections depending who is the actor that makes use of them: in section 2.3.1.1 Browsers' use cases; in section 2.3.1.2 Clients' and Professionals' use cases; in section 2.3.1.3 Clients' use cases; in section 2.3.1.4 Professionals' use cases; and in section 2.3.1.5 Administrators' use cases.

2.3.1.1. Browsers' Use Cases:

In this section are presented the most relevant use cases available for a Browser user.

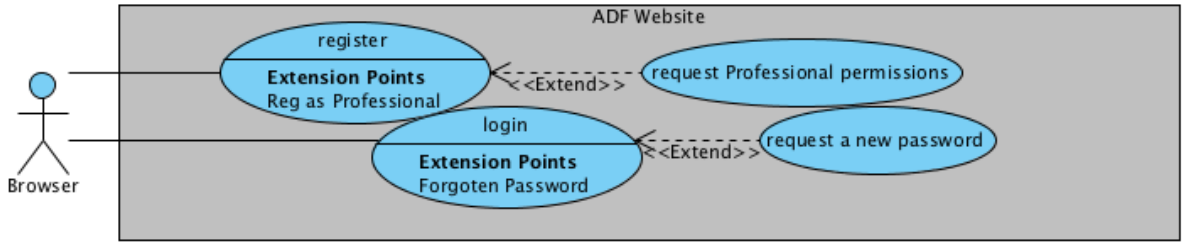


Figure 4: Browsers' Use Cases diagram

Name: Register
Number: UC-1
Preconditions: • The user selects to register, it is invited by a professional or it is registering as a professional after his petition has been accepted.

Flow of events:
• The user enters username, email, confirmation of email, a telephone number and the text of the captcha image that the system has show to him.
• The system validates the data:
  ○ If the username already exist the user is asked to enter a new one.
  ○ If the email or the confirmation of the email do not correspond to email pattern the system asks to enter them again.
  ○ If the email and the confirmation email are not the same the system asks the user to re-enter them.
  ○ If everything is fine the user proceeds.
• The user enters his demographic information. (see demographics.doc in the CD)
• The user uses Takes the Questionnaire. (see the use case for details).
• The registration ends and an email with the username and a generated password are sent to the provided email address.

Postconditions: • The user is now registered in the system. • If it was invited by a Professional the Client and the Professional users will be linked together. • If it was registering following the link of a petition acceptance the user will be registered with Professional rights. • Otherwise it will be registered with common Client rights.

Name: Request Professional permissions
Number: UC-2
Preconditions: None

Flow of events:
• The user enters name, middle name, surname, email, organization that is working for, the position that holds in the organization and a statement of why it wants to have Professional permissions.
• The system validates the data:
  ○ If the email does not correspond to email pattern the system asks to enter them again.
  ○ If the email is already used in another petition the system asks to enter another email.
  ○ If everything is fine the request is sent.

Postconditions: • The request is saved and yet unaccepted in the system.

2.3.1.2. Clients' and Professionals' Use Cases:

In this section are presented the most relevant use cases available for both Clients and Professionals.

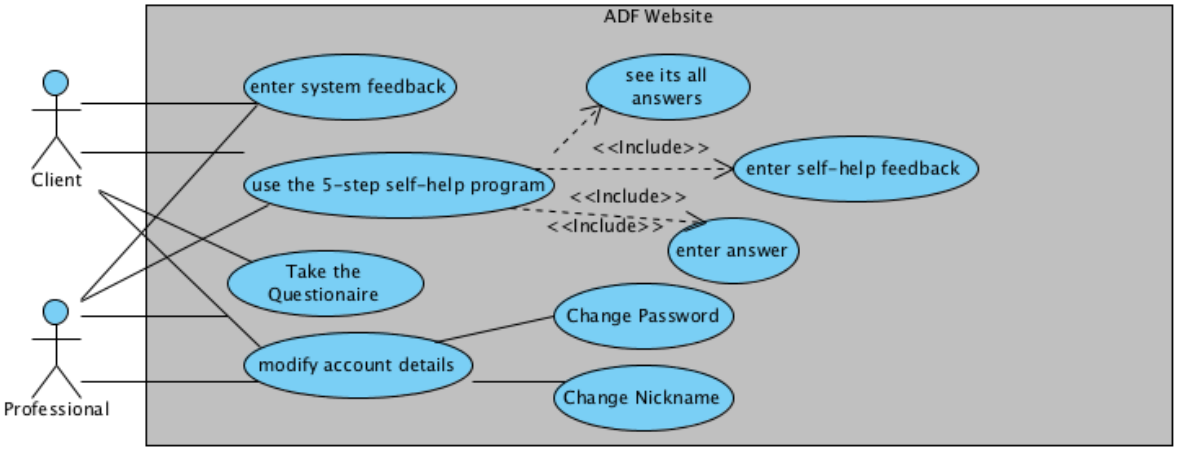


Figure 5: Clients' and Professionals' Use Cases diagram

Name: Use the 5-step self-help program
Number: UC-6
Preconditions: • The user must be a Client or Professional.
Flow of events: • The user selects one of the steps of the program. • The user goes through the program: reads and sees information, does answers to exercises (see "enter answer" use case). • At the end of the program the user can also enter feedback on the program (see "enter self-help feedback)
Postconditions: • The system has records of the activity of the users through his steps on the manual.

Name: See its all answers
Number: UC-7
Preconditions: • The user must be a Client or Professional.
Flow of events:

• The user selects to see its answers. If the user is a professional it can also select to see its Clients' answers. • The system shows all the answers to the user. • If the user prints the page that shows the exercises the print should be readable and well formatted.
Postconditions: None

Name: Enter answer
Number: UC-8
Preconditions: • The user must be a Client or Professional. • The user is in an exercise of the self-help program.
Flow of events: • The user is answers the questions on the exercise. • The user submits the exercises. • The user selects to continue with the program. • The system asks the user if he has thought enough or not. • If the user answers "no" he will continue in the same exercise. Else if the user selects "yes" he will continue with the program.
Postconditions: • All the answers that the user has entered are saved in the system.

Name: Take the questionnaire
Number: UC-10
Preconditions: • The user must be a Client or Professional. • The user is registering or it has past a multiple of 3 months (3, 6, 9… months) since he registered.
Flow of events: • The user answers to Coping Questionnaire. (see COPING QUESTIONNAIRE.doc) • The system calculates the evaluation scores of the questionnaire and presents them to the user. The user is also presented the mean score in of all the users in the same questionnaire. • If the user has a previous score this is also presented, compared with the actual one and a progress comment that depends on the amount of improvement or declination. • The user answers to Family Member Impact Questionnaire. (see FMI.doc) • The system calculates the evaluation scores of the questionnaire and presents them to the user. The user is also presented the mean score in of all the users in the same questionnaire. • If the user has a previous score this is also presented, compared with the actual one and a progress comment that depends on the amount of improvement or declination. • The user answers to Symptom Rating Test. (see SYMPTOM RATING TEST.doc) • The system calculates the evaluation scores of the questionnaire and presents them to the user. The user is also presented the mean score in of all the users in the same questionnaire. • If the user has a previous score this is also presented, compared with the actual one and a progress comment that depends on the amount of improvement or declination.
Postconditions: • The questionnaire answers and evaluations are now saved on the system.

2.3.1.3. Clients' Use Cases:

In this section are presented the most relevant use cases available only for Clients.

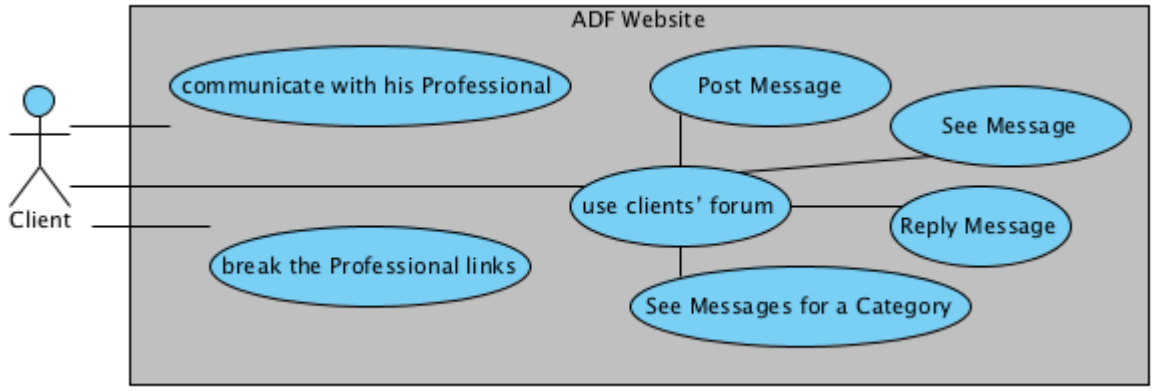


Figure 6: Clients' Use Cases diagram

Name: Communicate with his Professional
Number: UC-13
Preconditions: • The user must be a Client and have a link to a Professional.
Flow of events: • The user writes a message body and sends it. • The system adds the message to the inbox of his Professional
Postconditions: • The Professional of the user has a new message in his inbox.

Name: Use clients' forum
Number: UC-15
Preconditions: • The user must be a Client.
Flow of events: • The user selects to use the clients' forum. • The system allows the user to post, see, reply messages and see messages for a certain category in the clients' forum.

Name: See messages for a category
Number: UC-16
Preconditions: • The user must be using a forum (Client of Professional)
Flow of events: • The user selects a category from the list of categories for the forum that they are using. • The system shows the user the messages on that category

Name: See Message
Number: UC-17
Preconditions: • The user must be using a forum (Client of Professional) • The user must select certain category to see messages of.
Flow of events: • The user selects a message to see. • The system shows the body him the entire message.

Name: Post Message
Number: UC-18
Preconditions: • The user must be using a forum (Client of Professional) • The user must select certain category to see messages of.
Flow of events: • The user writes a message body and title and submits the message. • The system saves the message in the forum that the user is using and adds to it the category selected by the user.
Postconditions: • The system has a new message on the forum and category selected.

Name: Reply Message
Number: UC-19
Preconditions: • The user must be using a forum (Client of Professional) • The user must have selected a message to see.
Flow of events: • The user writes a message body and title and submits the message. • The system saves the message as a reply to the message that the user is seeing and notifies the users that had relation with the message about the change.
Postconditions: • The parent message has a new reply and the users that had relation with the message have an inbox message that notifies the change.

2.3.1.4. Professionals' Use Cases:

In this section are presented the most relevant use cases available only for Professionals. Post Message, See Message, Reply Message and See Messages for Category are the same as the ones presented on previous section.

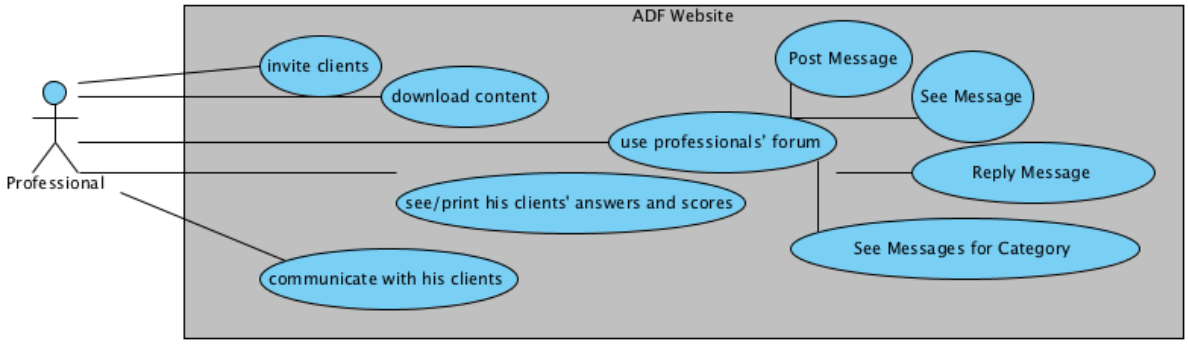


Figure 7: Professionals' Use Cases diagram

Name: Invite clients
Number: UC-20
Preconditions: • The user must be a Professional.
Flow of events: • The professional enters an email address to send the invitation to.

• The system sends a email to the entered email address with a link to proceed to the registration procedure that will link the new registered Client to the Professional

Name: Download content
Number: UC-21
Preconditions: • The user must be a Professional.
Flow of events: • The user selects a content (tool) to download from the content (tool) list. • The system sends the user the content (tool).

Name: See/print his clients' answers
Number: UC-22
Preconditions: • The user must be a Professional and he must have at least one Client.
Flow of events: • The user selects a Client from the client list to see ADF program answers from. • The system shows the user his Client's answers.

Name: Communicate with his Clients
Number: UC-23
Preconditions: • The user must be a Professional and he must have at least one Client.
Flow of events: • The user selects a Client from its Client list to send a message to. • The user writes a message body and sends it. • The system adds the message to the inbox of the selected Client.
Postconditions: • The selected Client of the user has a new message in his inbox.

2.3.1.5. Administrators' Use Cases:

In this section are presented the most relevant use cases available only for Administrators.

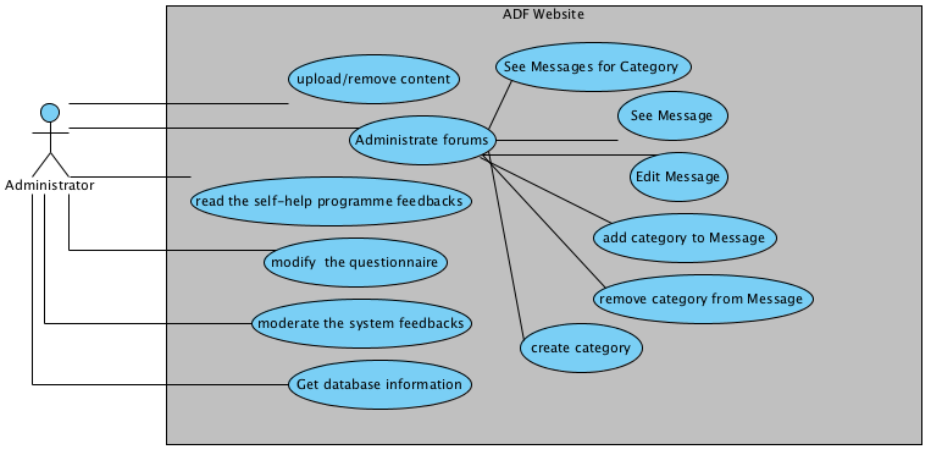


Figure 8: Administrators'' Use Cases diagram

Name: Upload/remove content
Number: UC-25
Preconditions: • The user must be an Administrator.

Flow of events:
• The system shows the user with the list of content (tool) available for Professionals.
• The user can choose to remove one of the tools.
  o The system will remove the selected tool from the system.
• The user can also upload a new tool. For this purpose the user will enter a name and a description for the new tool and will choose a file (the tool) to upload.
  o The system will save the new tool in the system.

Postconditions: • If the user has decided to remove a tool this tool will be no longer available for the Professionals. • If the user has decided to upload a new tool it will be available for the Professionals.

Name: Modify the questionnaire
Number: UC-27
Preconditions: • The user must be an Administrator.
Flow of events: • The user edits the questions, answers and/or evaluation of a given questionnaire. • The system will from then on use the modified questionnaire when the users "Take the questionnaire".
Postconditions: • The questionnaire is modified.

Name: Get Database Information
Number: UC-29
Preconditions: • The user must be an Administrator.
Flow of events: • The system shows the user the list of database information that he can view. • The user selects to view certain information. • The system generates a Microsoft style sheet (.xls) file from the information selected and sends it to the user.

Name: Use professionals' forum
Number: UC-30
Preconditions: • The user must be an Administrator.
Flow of events: • The user selects to administrate the clients' or professionals' forum. • The system allows the user to edit a message, see a message, add a category to a message, remove a category from a message, see messages for a certain category in the selected forum and create a category in the selected forum.

2.3.2. System Architecture

The system architecture, as most of web architectures, is composed by three tiers; Client, Application Server and Database Server. Figure 9 shows the high-level architecture of the system.

In the Client tier the WebBrowser component is used to represent any web browser that a client's computer could run to access the services provided by the web application. This component communicates with the Application Server using for it the Internet, so that, TCP/IP and HTTP request and response standards are used.

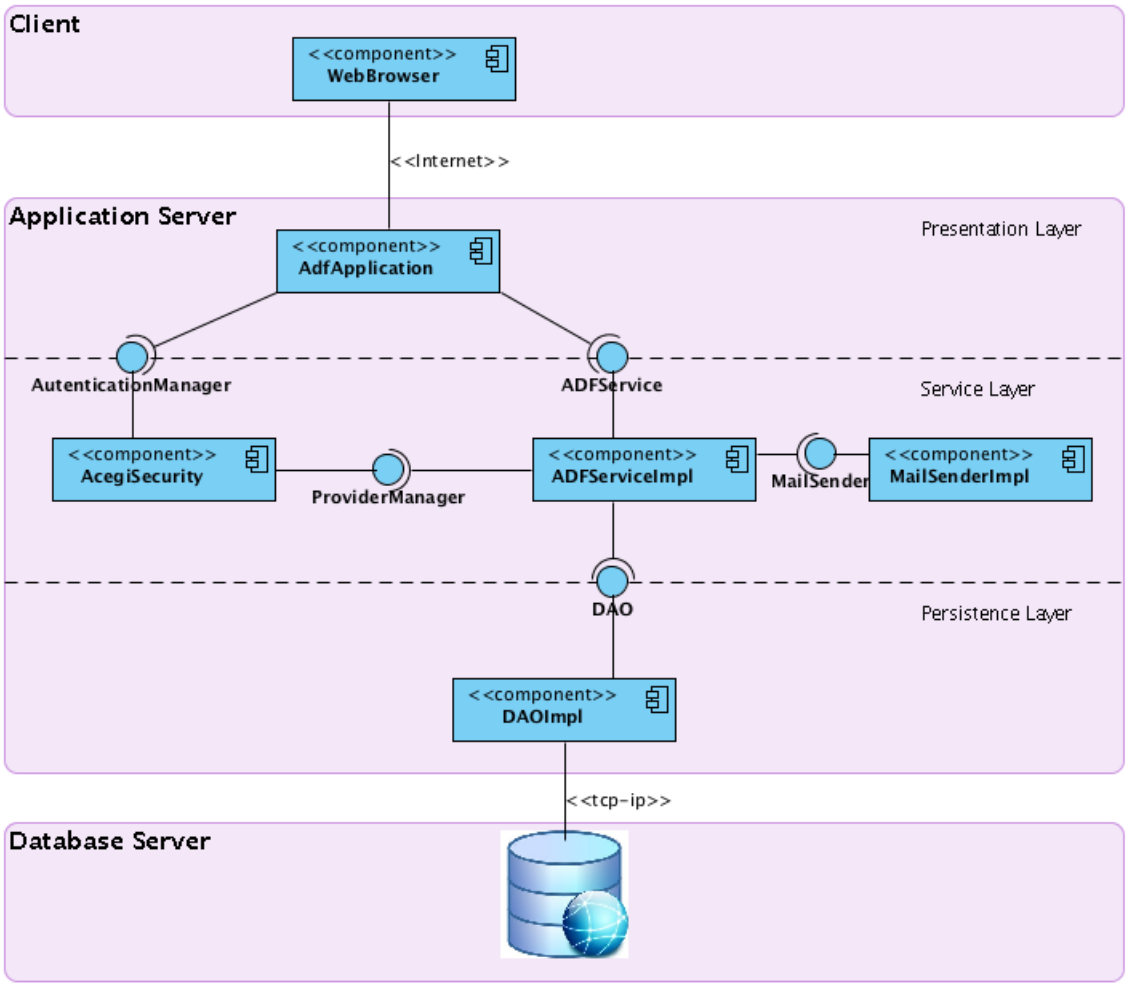


Figure 9: Main Architecture

The Application Server is composed by three layers; Presentation Layer, Service Layer and Persistence Layer. In the Presentation Layer the AdfApplication component if found. This component is a web application that deals with the presentation and user input. This application requires AuthenticationManager interface, for dealing with logged user permissions and rights, and ADFService interface that provides the component with the services that it presents to the clients, such as, change password or get the list of messages in a forum. Further details on this component are found in section 2.3.2.2.

In the Service Layer, the ADFService interface is provided by the ADFServiceImpl component. This component is the centre part of the application where the business logic is stored. It requires several interfaces but the main ones are: ProviderManager, MailSender and DAO. First, the Provider Manager interface is provided by the AcegiSecurity. This is really not a component but a library that in use with Spring Framework provides the required interfaces to the components that is linked to. Second, the MailSender interface is provided by MailSenderImpl. Third, DAO interface is provided by DAOImpl in the Persistence Layer. This is composed by a compound of interfaces and components that are responsible of saving and returning persistent objects. More details on DAOImpl are presented in section 2.3.2.4. The ADFServiceImpl is also further examined in section 2.3.2.3.

The Database Server tier contains the database that preserves the persistent objects provided by DAOImpl. The second and third tiers communicate one with the other by TPC/IP.

Finally, the AdfApplication, ADFServiceImpl, DAOImpl components and the interfaces that connect them use the same data representation. A diagram that shows the classes used to represent the data upon which the application works can be found in section 2.3.2.1.

The reasons for composing the system as shown in Figure 9 are modularity and scalability. The responsibilities of the system are clearly divided in its different components. Moreover, the only dependencies between the components are the interfaces that they require and provide. Due to this, components can be taken out and replaced by others as long as the new component provides the same interfaces. This allows the system to be updated by sections providing, for example, a more efficient DAOImpl component that could improve the overall performance of the system without changing the remaining components. Furthermore, it does not only allow replacing components but also inserting components that make use of the already working system. In example, a mobile (iPhone) application could be developed and directly connect it to the system by only connecting this application component to the already existing ADFServiceImpl component, the same way the AdfApplication is connected.

2.3.2.1. Model

This section presents the classes that represent the data upon which the system works. Figure 10 shows the Model diagram. Before going into explanations on the classes, the reader can notice that each of the classes on the diagram is linked to a *.hbm.xml artefact. These artefacts provide a mapping of the classes to their database representation that is used by Hibernate for constructing the database and saving and retrieving objects from/to it.

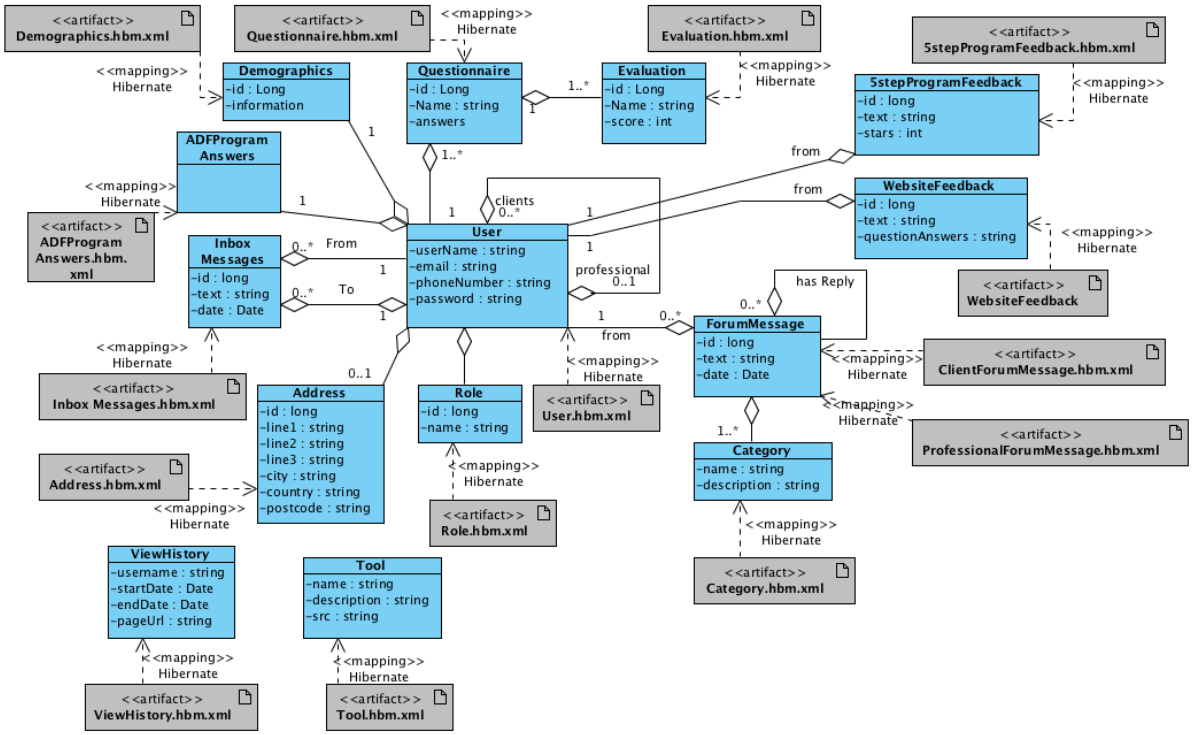


Figure 10: Model diagram

The central part of the Model is the user class. This class represents the user in the system. Each user has a Role that will classify the user as a Client, Professional or Administrator. The user can have a set of Client users in the case that the user is a Professional or a Professional user in case that the user is a Client. Although the User class allows having both relations at the same time the business logic should only allow having only one of those relations depending on the Role of the User. Moreover, the user also has a set of received InboxMessage objects, an Address object, Demographic information object and ADFProgramAnswers object (see APPENDIX C ADFProgramAnswers diagram for more details).

At the right of the User class the ForumMessage class is found. This class represents a forum message. The ForumMessage class has a set of ForumMessage objects that represent the reply messages of it and a set of Category objects that represent the category upon which the message is classified. With the intention of classifying the forum messages into different forums the ForumMessage and Category classes are inherited by ClientForumMessage, ProfessionalForumMessage, ClientCategory and ProfessionalCategory. The forum classification is designed like that, and not as a property of the ForumMessage class, because this allows Hibernate to separate the messages and categories into different tables for different forums. Separating the messages in different tables makes the retrieval of messages faster as the database manager has fewer conditions to compare.

At the top of the diagram the Questionnaire class is found. This “answers” attribute of this class represent other 30 int type attributes. This amount is the maximum amount of questions that a questionnaire can have. This boundary was established by the Project Client

as future modifications of the questionnaires will have fewer questions and never more. In addition, each questionnaire has one or more Evaluation objects that represent the score that the questionnaire took in certain evaluation.

Finally, other remarkable classes are the ViewHistory and Tool. The first represents the user activity history information that will be recorded for the ADF self-help manual. The second represents a tool that is downloadable by the Professionals.

2.3.2.2. AdfApplication

This section presents the design of the web application. The web application has been designed having in mind that Apache Wicket is the technology that will be used to encode the project. This affects the web application's diagrams[1].

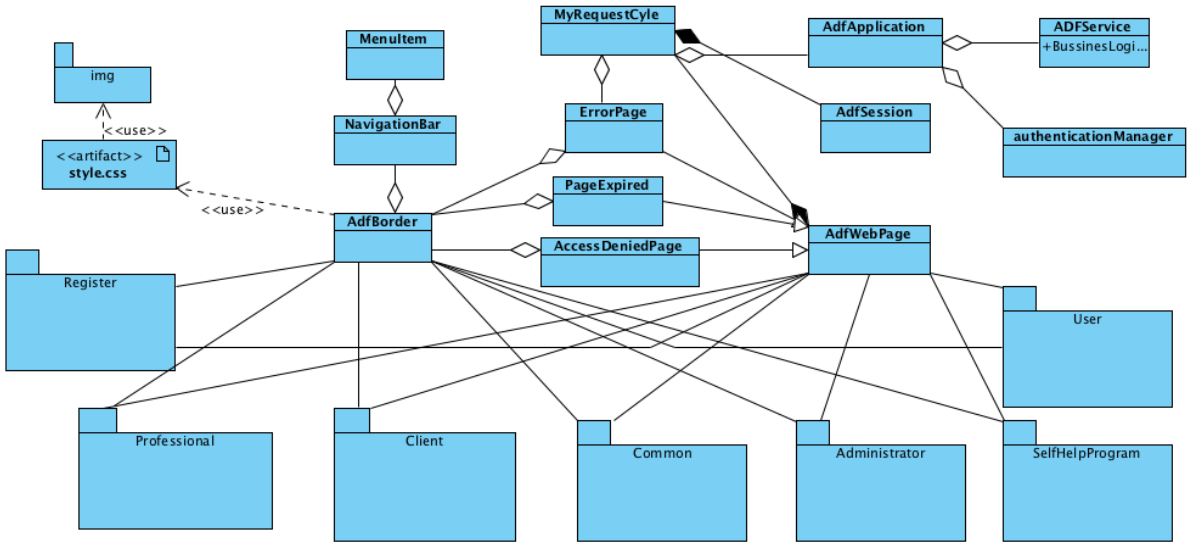


Figure 11: Web application (AdfApplication) diagram

Figure 11 shows the main class diagram of the web application. In the figure can be seen the AdfApplication class that builds the web application. This class extends Wicket's AuthenticatedWebApplication class. It aggregates the authenticationManager and ADFService classes for the security service and website's service. It also aggregates MyRequestCycle, ErrorPage, PageExpired, AccessDeniedPage and Common package for handling exceptions and setting the webpages home page and login page that are located on Common package.

There are also important the AdfSession, AdfWebPage and AdfBorder classes. The first extends Wicket's AuthenticatedWebSession class. The object that creates from this class represents the session that a client creates when first when the first request to the web server

[1] Apache Wicket separates the logic and the presentation of web pages and components, the logic is located in a java file and the presentation in an html file both with the same name. In the developed web application's diagrams only the java file is present, excluding the html for having clearer diagrams, but it does not to be forgiven that any web page or component class presented has an associated html (in most of cases at least).

is done and will last until the end of its activity with the web server. The second, AdfWebPage, extends Wicket's WebPage class. This class encloses the basic attributes and operators of a web page on the system and therefore it will be extended by all the other web pages in the system. The last, AdfBorder, extends Wicket's Border component class. This class contains the website's style sheet and web page's surrounding elements, such as, the header or the navigation bar, and will be aggregated in all the system pages.

Finally, the packages shown at the bottom of the diagram contain the remaining webpages that compose the website. Each of them is separated by the permissions needed to access its contents or by the functionalities that the webpages inside made available. The access is controlled using Acegi Framework's facilities. On the following figures the access permissions of each type of user (Browser, Figure 12; Client, Figure 13; Professional, Figure 14; Administrator, Figure 15) to the packages is shown.

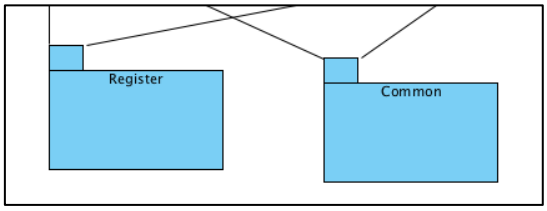


Figure 12: AdfApplication Browser access rights

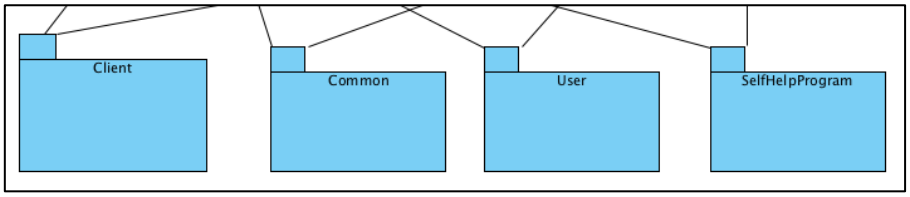


Figure 13: AdfApplication Client access rights

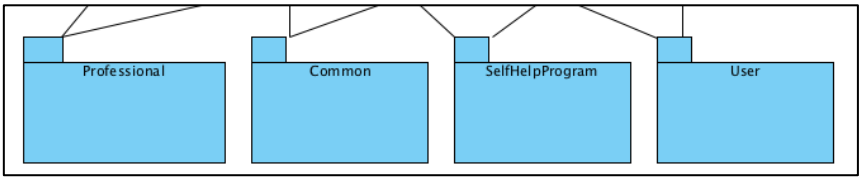


Figure 14: AdfApplication Professional access rights

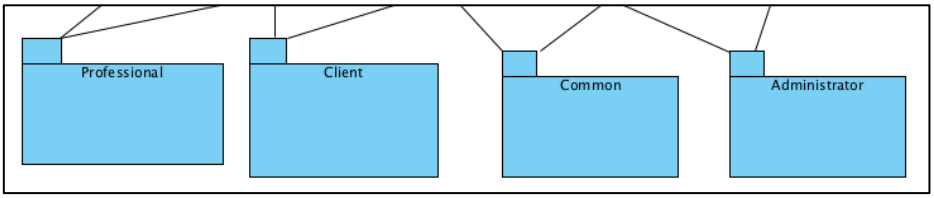


Figure 15: AdfApplication Administrator access rights

2.3.2.3. ADFServiceImpl

This section presents the design of the ADFServiceImpl component. This component is composed by one class that implements the ADFService interface. The interface is shown in Figure 16. The figure shows the methods that the AdfApplication has available. The

methods resemble the use cases of the system (section 2.3.1), although some of the use cases make use of more that one method.

Figure 16: ADFService interface

Security also plays a role in this component, as not all users should have rights for executing the same methods. Following lists of executable methods available for different type of users are displayed:

- *Methods available for all Users:*
  - Authenticate.
  - isUserRegistered.
  - registerClient.
  - registerProfessional.
- *Methods available for Administrator, Professional and Client:*
  - addADFProgramAnswer.
  - addInboxMessage.
  - changePassword.
  - deleteInboxMessage.

- getADFProgramAnswers.
- getAdfUser.
- getAllADFProgramAnswers.
- getInboxMessages.
- updateAdfUser.

- *Methods available for Administrator and Professionals:*
  - getTools.
- *Methods available for Administrator and Client:*
  - getClientCategories.
  - getClientForumMessage.
  - getClientForumMessagesByCategory.
- *Methods available for Professional and Client:*
  - addViewHistory.
- *Methods available only for Administrator:*
  - acceptProfessionalLoginPetition.
  - addCategoryToClientForumMessage.
  - addClientCategory.
  - addTool.
  - declineProfessionalLoginPetition.
  - deleteCategoryFromClientForumMessage
  - deleteClientCategory.
  - deleteTool.
  - editClientForumMessage.
  - getProfessionalLoginPetitionForms.
  - getViewHistories.
- *Methods available only for Professionals:*
  - getClientsForProfessional.
  - sendInvitation.
- *Methods available only for Clients:*
  - addClientForumMessage.
  - addClientForumMessageReply.

2.3.2.4. DAOImpl

This section presents the DAOImpl component. The diagram in Figure 9 presents a simplified vision of this component that, actually, provides more that one component. Figure 17 shows the internal class diagram of the DAOImpl component where the different interfaces that it provides are shown. The implementations of the interfaces will be done using Hibernate.

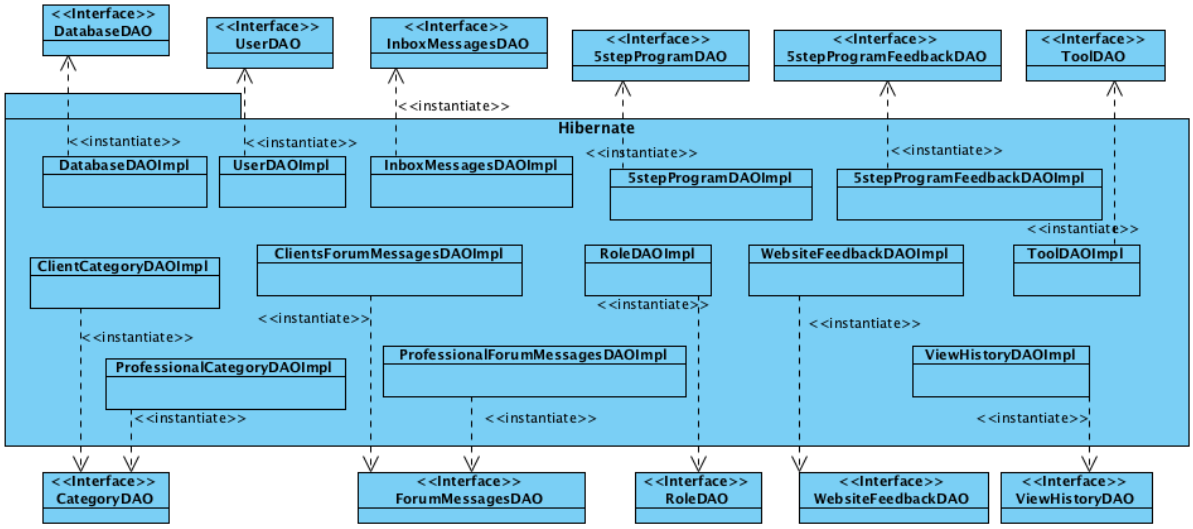


Figure 17: DAO diagram

2.4. IMPLEMENTATION

The project has been implemented using NetBeans 6.8 IDE. This IDE has been selected because it is easily configurable to run Apache Tomcat server and Apache Wicket, Spring and Hibernate frameworks and because the Project Developer is experienced on it. Although in the final version of the software another database may be used, for the development of the prototype PostgreSQL has been used because it is reliable and free. Using another database is not a big problem because thanks to Hibernate framework this change can be done by only changing the database configuration parameters.

In the following subsection, a detailed explanation of the implementation of the most relevant functionalities takes place. During the explanation of these functionalities prototype's snapshots and small Java code parts are inserted. If the reader is interested on more seeing the entire code, this is provided on the CD accompanying this report (see APPENDIX D for explanations on the CD structure).

After the explanatory section the testing of the prototype is presented on section 2.4.2. In this section includes the testing strategy used, the results of these tests and the solutions that have overcome the software faults.

2.4.1. Main features

The following sections present the implementation details of the user interface, system's access control, self-help programme, user's activity history, clients' discussion forum, Professional's supervision, Professional's tools upload and download, registration and automated generation of XLS files from database information. When presenting the implementation details references to system classes could be made, so for any doubts on them see section 2.3.2 or APPENDIX C.

2.4.1.1. User Interface

The design of the user interface aims to do the website more attractive for the users. For designing it the Project Developed interviews the Project Client on the 22nd of July as reported on APPENDIX A. During this meeting several websites are visited in order to get ideas that will make the appearance of the website better. This results in having the following directives the design of it (the resulting user interface of the directives is shown in Figure 18):

1. It must have a new colour scheme that feels more attractive and clear. Blue and white scheme is decided because the white colour gives clearness, the blue colour gives relaxing feeling and both combine very well together.
2. The webpage must have a big header as most of websites do because it takes more attention to it and makes the website more memorisable.
3. A nice navigation bar is desired. It is decided that Apples official webpages navigation bar will be replicated.
4. The pages on the website must accommodate to the size different window sizes.

Figure 18: ADF Website preview

In order to make a webpage that accommodates to the size of the browser window CSS liquid layout it is used. For implementing the liquid layout the elements of the HTML code must specify its width style attribute in percentages instead of in a fixed unit, such as, pixels or centimetres. But there are also boundaries that can not be exceeded because otherwise the elements will not display in the desired layout composition. An example of this is shown in Code 1, part of sys-style.css.

```
.exteriorcontainer{
    width: 83%;
    background: #FFF;
    max-width: 1260px;
    min-width: 1000px;
    margin: 0 auto;
}
```

Code 1: Liquid layout

For providing the webpage with a nicer appearance navigation bar elements show different background images when the item is inactive, active or the mouse is over it. This is done by, first, specifying different styles to 'a' element and its mouse over state 'a:hover' and, second, by creating a style class, with other style, and setting the class attribute of the link element that points to the actual page of that class. Code 2, part of sys-style.css, provides an example of this.

```
.menuitemInactive a{
    text-decoration: none;
    background-image:url(img/menuItemInactive.jpg);
    background-repeat:repeat-x;
    color: #000;
}

.menuitemInactive a:hover{
    text-decoration: none;
    background-color: #0066cc;
    background-image:url(img/menuItemSelected.jpg);
    background-repeat:repeat-x;
    color:#FFF;
}

.menuitemActive a{
    text-decoration: none;
    background-image:url(img/menuItemActive.jpg);
    background-repeat:repeat-x;
    color: #FFF;
}
```

Code 2: CSS dynamic highlighting

2.4.1.2. System's Access Control

The access control of the system is fundamental for the right working of the system. If the access is not controlled every user could play the role of the administrator, for example. The first step to control the access is to authenticate the users when logging in. For authenticating the credentials entered by a user authenticate method of the ADFService interface is used. The interfaces implementing component makes use of the ProviderManager service given by the Acegi Framework for verifying the credentials. If the method returns true the user is authenticated and his Role consequently his permissions are stored into his AdfSession.

Once the permissions of the user are saved in the session this can be used to check the access rights to a page and to an ADFService method and render web page components

depending on his role. For preventing users accessing web pages or component to which it has not access to Wicket's AuthorizeInstantiation annotation is used (see Code 3). In order to render components depending on the user's role MetaDataRoleAuthorizationStrategy is used as shown in Code 4. Finally for preventing usage of ADFServiceImpl methods without authorization Spring's Aspect's point cuts have been used in combination with Acegi. Point cuts are established before the methods are executed so that verification code can run and check that the user has the required permissions. On Code 5 a part of the Spring's configuration file where the described action is configured is shown.

```
@AuthorizeInstantiation({"ROLE_admin", "ROLE_client"})
```

Code 3: AuthorizeInstantiation

```
MetaDataRoleAuthorizationStrategy.authorize(deleteCategory, RENDER, "ROLE_admin");
```

Code 4: MetaDataRoleAuthorizationStrategy

```
<aop:pointcut id="adfServiceOperation" expression="execution(* adf.service.*.*(..))"/>
<aop:advisor id="security" pointcut-ref="adfServiceOperation" advice-ref="adfServiceSecurity" order="1"/>
<aop:advisor id="transactions" pointcut-ref="adfServiceOperation" advice-ref="txAdvice" order="3"/>
```

Code 5: AoP configuration in Spring

2.4.1.3. Self-help Program

Improvements on the self-help program include the insertion of multimedia content, step end rewards, step end options and the ability of seeing all the answers. The step end options are simple hyperlinks that point to home for the 'take a brake' option, to the beginning of the step for the 'do it again' option and to the next step for the 'continue' option that do not require further explanation. The other improvements are following explained.

The multimedia content it is placed at the beginning of each step. The insertion of this multimedia content has been done using html5 instead of commonly used flash. The reason for this that this is suits the new standards and avoids using company owned technologies. For insertion html5 video on our project Wicket Stuff's, (Wicket Stuff n.d.), Html5Video and MediaSource content are used. In addition, in the corresponding html file 'video' element is inserted with a 'wicket:id' attribute that references the html5Video component. Code 6, part of StepOneIntroductionVideo.java, and Code 7, part of StepOneIntroductionVideo.html, show an example of it.

As it can be seen in the code the references to the content have been made in a static way. This has been done like that because, as told by the Project Client, the content will not suffer from modifications and, at the first moment, it will not be placed anywhere else. Anyways, the content can be updated by replacing the source files directly (removing and inserting another with the same name) and the Project Client is capable of doing it like that.

The last point about the html5 video is that it requires to be coded in three different formats. This happens because the web browsers that understand html5 do not use a unique codding method, so for the video to be supported by all the web browsers it has to be codded in mp4, orv and webm.

```
final List<MediaSource> mm = new ArrayList<MediaSource>();
mm.add(new MediaSource("../assets/sampleVideo.mp4", "video/mp4"));
mm.add(new MediaSource("../assets/sampleVideo.ogv", "video/ogg"));
mm.add(new MediaSource("../assets/sampleVideo.webm", "video/webm")); // Video added for IE9 support

IModel<List<MediaSource>> mediaSourceList = new AbstractReadOnlyModel<List<MediaSource>>() {
    private static final long serialVersionUID = 1L;
    public List<MediaSource> getObject() {
        return mm;
    }
};

border.add(new Html5Video("sampleVideo", mediaSourceList) {
```

Code 6: Html5 video java part

```
<video id="video" wicket:id="sampleVideo">
    Your browser doesn't support the HTML5 video tag. Download the latest version of it.
</video>
```

Code 7: Html5 video html part

For the final steps rewards the insertion of flash animations has been adopted as a solution. These animations must translate a feel of success as the ones showed when a stage of a game is passed. Flash animations have being selected as an implementing technology because it is easy to get this kind of animations free from the internet and because creating own ones is also easily done for anyone with little experience. For the insertion of this elements in Wicket the commonly used elements are used in the html part of the web pages and this are surrounded by <wicket:link> elements. Code 8 shows an example of this.

```
<wicket:link>
    <object classid="clsid:D27CDB6E-AE6D-11cf-96B8-444553540000"
        codebase="http://download.macromedia.com/pub/shockwave/cabs/flash/swflash.cab#version=6,0,29,0" width="700"
        height="500">
        <param name="movie" value="../assets/stars.swf"/>
        <param name="quality" value="high"/>
        <embed src="../assets/stars.swf" quality="high"
           pluginspage="http://www.macromedia.com/go/getflashplayer" type="application/x-shockwave-flash" width="700"
           height="500"></embed>
    </object>
</wicket:link>
```

Code 8: Wicket link example

The Presentation layers functionality of seeing all exercises is implemented on class ExerciseAnswersPage. For implementing the functionality the different exercises have been implemented extending Wicket's Panel component. This way each panels is not only inserted on its exercise page but also a page where we insert all of them. Reusing these components allows us to being able to only change one presentation component when updating the exercises. For restricting the panels, from allowing answering to exercises, a boolean parameter is passed to them. This results useful because it is not desired the user being able to answer the exercises out of the context of the self-help manual.

The panels are fed by exercise representing objects (see APPENDIX C; Package adf.model.exercises). Those objects are retrieved from the ADFService component by using getADFProgramAnswers, for individual, and getAllADFProgramAnswers, for getting all of

them, and passing a username and exercise class for the first one. For setting new answers the panels make use of the addADFProgramAnswer, passing as arguments the username, the exercise object and the class of the object.

2.4.1.4. User activity history

While the users use the 5-step self-help program the application records the activity of the user through the different pages of it. The user activity information results important to the Project Client as it will allow analysing how the users use the self-help program and improving it.

For tracking the user activity the time that enters a page and that lives it must be recorded. Unfortunately, there is no event triggered when a user lives a page so for setting the live time of a page the user's next web page request must be taken into account. In Wicket all the requests are handled by the WebRequestCycle. This component has been extended and its onEndRequest method overwritten to implement the code that will track the user activity. This code is presented on Code 9.

```
protected void onEndRequest() {
    String path = getWebRequest().getPath();
    // Filter request that are not pages
    if (this.getResponsePageClass() != null) {
        if (urlFor(this.getResponsePageClass(), PageParameters.NULL).toString().contains(path)) {

            AdfSession session = ((AdfSession) getSession());
            // end previous tracking
            session.endTracking();
            // set new traking depending on conditions
            boolean trackablePage = false;
            //Condition: page from path adf.page.SelfHelpSupport
            if (path.contains("SelfHelpSupport")) {
                trackablePage = true;
            }
            if (trackablePage) {
                session.startTracking(RequestUtils.toAbsolutePath(urlFor(this.getResponsePageClass(), PageParameters.NULL).toString()));
            }
        }
    }
    super.onEndRequest();
```

Code 9: User activity tracking

The algorithm first of all filters all the request that are not pages so that request for other content, such as video, does not interfere. If the request is a request for a page the previous tracking is ended. Then it is checked if the requested page is a page for which the activity must be tracked. In this case only the self-help supports activity must be tracked so it is checked if the path of the request contains "SelfHelpSupport" as all the pages on the self-help programme are mounted under SelfHelpSupport directory. If it is the case a new tracking is started.

The code makes use of two methods on the session: startTracking and endTracking. These methods have been located in the AdfSession because unlike the pages or the WebRequestCycle object it is not dropped each request and it can store data between requests.

2.4.1.5. Clients' Discussion Forum

The Client's discussion forum, ClientForum, is formed by three Wicket's Panel components: ClientCategoriesPanel, ClientThreadsPanel and ClientForumMessageDisplayPanel. The first displays the list of categories, the second displays the threads for a category and allows posting messages and the third displays a message and its replies allows posting a reply. The benefit of composing the client's forum this way is that the components can also be used in the ClientForumControl page of the Administrator. The administrator controls (such as delete category or edit message) are also coded in the very same panels but there are only rendered when an Administrator instantiates the panels. For this MetaDataRoleAuthorizationStrategy is used on the Administrator controls as explained on section 2.4.1.2.

An important part of the forum is how the threads are displayed. The Project Client desired to have a tree display of the messages where the replies of the main messages are shown. For doing this a TreeTablePanel has been implemented that makes use of the Wicket's TreeTable. Code 10, shows how the TreeTable component is used in order to compose the tree message structure. In the constructor a tree model filled by the messages list feds the TreeTable. For inserting a link on each of the message to the display of it onNodeLinkClicked is overwritten. For these links to work being the panels instantiated by different pages getPage().getPageClass() is used so that the page class where the panel is instantiated is dynamically gotten.

```
tree = new TreeTable("treeTable",convertForumMessageToTreeModel(messageList), columns){

    @Override
    protected void onNodeLinkClicked(AjaxRequestTarget target, TreeNode node) {
        super.onNodeLinkClicked(target, node);
        ModelBean modelbean = (ModelBean)((DefaultMutableTreeNode) tree.getNodeComponent(node)
                                          .getDefaultModel().getObject()).getUserObject();

        PageParameters param = new PageParameters();
        param.put("messageId", modelbean.getMessageId());
        setResponsePage(getPage().getPageClass(),param);
    }

};
```

Code 10: TreeTable

2.4.1.6. Professionals' Overview

The professional must had the possibility of seeing the exercises of their Clients and communicating with them. For allowing the professionals getting those functionalities, first, a list of his clients is loaded by making use of getClientsForProfessional method of the ADFService. This method provides a list of users that is used to construct a table with two links for each user. The first of the two links points to ExerciseAnswersPage to which the username of the Client is passed as an argument. The second link points to SendMessage to which also the username of the Client is passed.

Passing the username as a parameter opens a security breach because, although only Clients and Professionals are allowed to instantiate the page, another Professional could enter the pages' URLs using other Professionals' Client username. To prevent this from happening, the ADFService checks that the identity of the requester is a valid one.

Furthermore, the users can prevent their Professional, in case that they have one, from supervising their answers or communicating with them. For doing so denyProfessionalSupervising and denyProfessionalContact boolean attributes of the User class are checked when rendering the page. If any of this is set to true the link's visibility will be set to false, so that, the supervisor can not use them. Clients can change these parameters to their will on their account information.

2.4.1.7. Tools upload/download

The tools upload/download functionality is provided by two web pages; the Administrator's page that allows him to upload and remove content and the Professional's page that allows him to download the content. Both pages include the ToolListView component.

The ToolListView component makes use of the getTools method of the ADFService to get the list of Tools. The Tool class is composed by a name, a description and the tool's file name on the upload folder. The upload folder is a folder in the application's file system that stores all the tools provided to the Professionals. Code 11 shows how the list of tools is populated. It is remarkable how the DownloadLink is constructed with the file that is gotten from the source file reference of the Tool object.

```
final Tool tool = (Tool) li.getModelObject();
li.add(new Label("name", tool.getName()));
li.add(new MultiLineLabel("description", tool.getDescription()));

final File file = new File(getUploadFolder(), tool.getSrc());
li.add(new DownloadLink("download", file));

Link removeLink = new Link("remove") {

    @Override
    public void onClick() {
        ((AdfSession) getSession()).getAdfService().deleteTool(tool.getId());
        Files.remove(file);
        setResponsePage(getPage().getPageClass());
    }
};
removeLink.add(new JavaScriptEventConfirmation("onclick", "Are you sure that you want to delete the file?"));
li.add(removeLink);
MetaDataRoleAuthorizationStrategy.authorize(removeLink, RENDER, "ROLE_admin");
```

Code 11: Download List view

For the uploading the tools the algorithm shown in Code 12 is used. The code show is onSubmit method of the upload form.

```
protected void onSubmit() {
    final FileUpload upload =  (FileUpload) srcModel.getObject();
    if (upload != null) {
        // Create a new file
        File newFile = new File(getUploadFolder(), upload.getClientFileName());
        // Check new file, promt error if it does
        if(!checkFileExists(newFile, toolList)) return;
        try {
            // Save to new file
            newFile.createNewFile();
            upload.writeTo(newFile);
              UploadPage.this.info("saved file: " + upload.getClientFileName());
        } catch (Exception e) {
            throw new IllegalStateException("Unable to write file");
        }
        // Safe information to database
        Tool tool = new Tool();
        tool.setName((String)nameModel.getObject());
        tool.setDescription((String)descriptionModel.getObject());
        tool.setSrc((String)upload.getClientFileName());

        getAdfService().addTool(tool);
```

Code 12: Upload algorithm

2.4.1.8. Register Design

The registration process is composed by various pages. For keeping the registration information through the different pages, until the registration is complete, the session object is used. The registration information is contained by the RegistrationInformation class in the AdfSession.

The Registration first page takes to types of parameters: professional's username and professional petition id. If any of these values are set they are also stored on the registration information, see Code 13. Then once the registration is finished these values are checked, see Code 14. If the petition id is set a Professional user is created. If the professional's username is set a Client linked to the username owner Professional is created. If neither of the parameters is set a Client that does not have a Professional contact is created.

```
// If the clients is to be referenced to a professional
//  save the professional reference
if(professionalUsername != null){
    RegistrationInformation registrationInformation = getAdfSession().getRegistrationInformation();
    registrationInformation.setProfessionalUsername(professionalUsername);
    getAdfSession().setRegistrationInformation(registrationInformation);
// If the new user is going to be professional level registration
} else if(petitionId != null){
    RegistrationInformation registrationInformation = getAdfSession().getRegistrationInformation();
    registrationInformation.setProfessionalPetitionId(petitionId);
    getAdfSession().setRegistrationInformation(registrationInformation);
}
```

Code 13: Registration user differentiation 1

The registration is protected from auto registering bots by captcha verification. This type of verification displays an image that contains a text in a way that a computer programme can not read. The image is auto generated by CaptchaImageResource component that takes a randomly created string as a constructor argument. If the text introduced in the

captcha's textbox does not correspond to the generated string the registration procedure does not proceed.

```
// Check if it is a professional registration
if(registrationInformation.getProfessionalPetitionId() != null){
    if(!getAdfService().registerProfessional(registrationInformation)){
        return false;
    }
    return true;
}
// Check if the user is using an invitation from a professional
if(registrationInformation.getProfessionalUsername() != null){
    if(!getAdfService().registerClient(registrationInformation)){
        return false;
    }
    return true;
}

// If this two conditions fail, it is a normal registration for a Client permissions
if(!getAdfService().registerClient(registrationInformation)){
        return false;
    }
```

Code 14: Registration user differentiation 2

2.4.1.9. Automated Generation of XLS files from Database Information

The automated generation of XLS files from database information requires: first, to generate a XLS file from some data gotten from the database and second sending this file and then deleting it not to waste the server's storage space. All this has to be done when a link is pressed so that the algorithm has been inserted inside the onClick method of a Link. Code 15 shows the algorithm that does that.

```
public void onClick() {
    try {
        final String fileName = "View History.xls";
        final File file = File.createTempFile(fileName, null);

        ConvertToXls.convertViewHistory(file, getAdfService().getViewHistories());
        IResourceStream resourceStream = new FileResourceStream(file) {

            @Override
            public void close() throws IOException {
                super.close();
                Files.remove(file);
            }
        };
        getRequestCycle().setRequestTarget(
                new ResourceStreamRequestTarget(resourceStream) {

                    @Override
                    public String getFileName() {
                        return fileName;
                    }
                });
    } catch (IOException ex) {
        Logger.getLogger(DatabaseInformation.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

Code 15: XLS download link

The code shows the conversion of the user activity information. First creates a temporary file using a filename. Then the file and the list of ViewHistory objects are passed as an argument to convertViewHistory method of the ConvertToXls class. The code that does the transformation is shown in Code 16. Then the file is set as the request target of the action. Notice that the FileResourceStreams close method has been overwritten to remove the file after it is been sent.

Code 16 shows a part of the conversion of the ViewHistory object list to a XLS format document. For composing the XLS document Apache POI library has been used. In the code 'wb' is a XLS work book element, 's' is the spread sheet of the work book and 'r' and 'c' are rows and cells of the spread sheet. At the end of the algorithm when the work book has been completely filled it writes the files output stream (out).

```
for (ViewHistory viewHistory : viewHistoryList) {
    rownum++;
    // create a row
    r = s.createRow(rownum);
    // Fill the cells of the rows
    c = r.createCell(0);
    c.setCellStyle(cs2);
    c.setCellValue(viewHistory.getUsername());
    c = r.createCell(1);
    c.setCellStyle(cs2);
    c.setCellValue(viewHistory.getStartDate().toString());
    c = r.createCell(2);
    c.setCellStyle(cs2);
    c.setCellValue(viewHistory.getEndDate().toString());
    c = r.createCell(3);
    c.setCellStyle(cs2);
    c.setCellValue((int)((viewHistory.getEndDate().getTime()-viewHistory.getStartDate().getTime())/1000));
    c = r.createCell(4);
    c.setCellStyle(cs2);
    c.setCellValue(viewHistory.getPageUrl());
}
wb.write(out);
```

Code 16: Convert to XLS

2.4.2. Testing

In this section the testing of the implemented software is presented. First, the testing strategy is presented. Second, the main incidents and their solutions are shown. And, third, the cross browser compatibility is discussed.

The testing strategy that has been carried out involves testing each functionality by doing a walkthrough of it. Making a walkthrough of a functionality means to, actually, run the entire system and test the functionality as a user of the system would do. The benefit of this type of test is that all the components of the system, the AdfApplication, the ADFServiceImpl and the DAOImpl, are tested in one go. This of course involves that, sometimes, the origin of the errors is more difficult to detect. Nevertheless, in a project with the time limitations that this project has, this type of test results beneficial as it saves the time of setting up each component's testing environment.

The errors that occur though the tests are detected by the logger information and the system consoles output. In the cases where those do not provide enough information about an

anomaly the NetBeans debugger is been used. In these cases, during the debugging, special attention is putted on the value of the variables.

Following this strategy many bugs have been found and solved. The most typical error has been caused due to incorrect component identification declarations between the Wicket page's java and html parts. When a component is inserted in a page it must have the same identification on both parts of the page. The solution of this type of error is straightforward as the console and logger provide enough information on which is the component that is failing. Following the main incidents for the different functionalities are presented:

1. Registration testing:

Incident Number: Inc-1
Condition for checking the existence of user does not work.
Cause: The check expects a null object if the user does not exist but an empty one is given instead.
Solution: Use Hibernates get method instead of load.

Incident Number: Inc-2
After accepting a Professional petition the petition is still in the petition list
Cause: There is not any clause to prevent that.
Solution: An 'accepted' attribute is added to the petition so it can be discarded from the petition list.

2. Professional Supervision testing:

Incident Number: Inc-3
See answer page gives an error.
Cause: The arguments of the ADFService method call where mistaken one with the other
Solution: Put the arguments in the right order.

Incident Number: Inc-4
See answer page gives an error again
Cause: Permission checking in ADFService is not properly done. If statements have the contrary clause
Solution: Correct if statements.

3. Inbox testing:

Incident Number: Inc-5
Delete gives an error
Cause: When deleting the message they are added again and this causes a reference error in Hibernate. This occurred because the reference to the users receiving and sending the message where done in their 'set' methods.
Solution: Create addReceiver and addSender methods InboxMessage class for adding the sender and referencing the message to the sender.

4. Clients' forum testing:

Incident Number: Inc-6
Browsers back button causes an error on the links of the TreeTable links.
Cause: When pressing the back button the cached page is loaded instead of making a request to the server. This creates an inconsistency between the state on the client and the state on the server.

Solution: Force the browser not to cache the page by setting the appropriate no-cache properties on the response header.

Cross Browser compatibility

The tests took place using Safari, Google Chrome and Mozilla Firefox browsers. The main reason for doing the tests using different browsers is to check mainly if the user interface is properly displayed among all the browsers. The tests result successful in all the cases. But when testing other functionalities two problems are encountered:

1. The solution for the problem stated on Inc-6 does not work with Safari browser. When searched for the reason it was discovered that safari does not respond properly to the no-cache headers.
2. The upload progress bar showed when uploading a file it only worked in Mozilla Firefox browser.

2.5. APPRISAL

The 5-step self-help program, the clients' forum, the professional supervision, the tool upload/download functionality, the registration procedure and the database information download functionalities have been implemented on this first prototype of the web application for Alcohol, Drugs and the Family group. Although, the requirements stated in section 2.1 are not fully covered, the implemented functionalities comprise the most relevant and challenging of the requirements for this project. Almost all the remaining functionalities can be copied and modified from the existing code. For example implementation of the Professionals' forum is identical to the implementation of the Client's forum. There is only one functionality, the modifiable questionnaire, that could not be implemented in time and that can not be directly copied from others. In this sense if the prototype is compared to the original requirement specification it has to be said that it has almost tick them all in terms of functionalities.

Taking a look to reliability, the developed software is reliable and robust. During the tests that the software has had several bugs where founded section 2.4.2. These bugs where then fixed and tested again until they resulted in success. The only issue that could not be resolved is occasioned by a cross browser test that showed that Safari browser does not respond as expected to the 'no-caching' headers and it still caches the forum pages making it to fail when the links are clicked. For the rest, the software has proved to be reliable and robust and to handle correctly different web browsers.

The performance of the software is the one expected from software of this kind. In this sense the implemented application has a normal response time if we take into account that it has been tested in a personal computer and not in a professional server.

Summarizing, the success of the software is remarkable for the time boundaries of the project although there is, of course, more work to do on it, in terms of requirements, to take it to the market.

3. CONCLUSION

The developed web application implements an improved version of the 5-step self-help programme implemented in the nowadays Alcohol, Drugs and the Family group's website. In addition, in top of this self-help manual the web application also provides functionalities that are believed to further support the family/ network members that are concerned about their relatives alcohol or drug addiction. The surrounding functionalities include a client's discussion forum, a client progress supervision facility for psychology professionals and a specialised tools download facility for those professionals that will help them to further help their clients. Furthermore, the system administrator is able to see how the users make use of the self-help manual and therefore analyse how to improve it so that will better help the users of it.

Although, the functionalities fit the main objectives imposed some of the functionalities could be improved. First, in the discussion forum messages are classified in a tree fashion. The replies for a message can be expanded and collapsed, which has the advantage of occupying little space. Nevertheless, if the forum receives lots of messages this will be all displayed in a unique page resulting in a too large web page. To prevent this pagination of the messages should be implemented. Second, the multimedia content provided on the self-help manual uses static references to the content. This means that for updating the content the content file must be done by replacing the file that is referenced by hand, without the assistance from the web application. To improve this, the multimedia content could be stored in the database and a new functionality provided to change this content. Furthermore, conditions, such as the age of the user, could be used to provide a more appropriate video to the user.

So far, the project lays the foundations for what will be the future site of the Alcohol, Drugs and the Family group. From this point on, the remaining unimplemented requirements that have to be implemented in order to make this project a final market product. However, as the project has focused on implementing the most challenging requirements, most of the remaining work can be completed by replicating and modifying the already implemented ones. In example, for a final version of the application it is required to have a discussion forum for professionals. As the prototype already has an implemented discussion forum for the implementation of the professional's one requires only copying the implemented discussion forum and replacing 'client' words by 'professional' ones.

4. REFERENCES AND BIBLIOGRAPHY

Alcohol, Drugs and the Family Self-Help Manual. n.d. http://www.alcoholdrugsandfamilies.nhs.uk (accessed September 11, 2010).

Bordbar, Behzad. "Internet Computing Workshop: Spring." n.d.

—. "Internet Computing Workshop: Spring security." n.d.

—. "Internet Computing Workshop: Wicket." n.d.

Bordbar, Bezhad. "Internet Computing Workshop: Hibernate." n.d.

Dashorst, Martin, and Eelco Hillenius. *Wicket in Action.* Manning, 2008.

JBoss Community. *Hibernate.* n.d. www.hibernate.org (accessed September 2010, 11).

Lloyd, Ian. *Web Site The Right Way: using HTML & CSS.* SitePoint Pty. Ltd., 2006.

Orford, Jim, Asmita Patel, Lorna Templeton, Alex Copello, and Richard Velleman. "The 5-Step family intervention in primary care: I. Strengths and limitations according to family members." *Drugs: education, prevention and policy* (Informa Healthcare) 14(1) (February 2007): 29–47.

Orford, Jim, Lorna Templeton, Asmita Patel, Richard Velleman, and Alex Copello. "The 5-Step family intervention in primary care: II. The views of primary healthcare professionals." *Drugs: education, prevention and* (Informa Healthcare) 14(2) (April 2007): 117–135.

SringSource community. *SpringSource.* n.d. www.springsource.org (accessed September 2010, 11).

The PHP Group. *PHP: Hypertext Processor.* n.d. www.php.net (accessed September 2010, 11).

Velleman, Richard, Caterina Arcidiacono, Fortuna Procentese, Alex Copello, and Pasquale Sarnacchiaro. "A 5-step intervention to help family members in Italy who live with substance misusers." *Journal of Mental Health* (Informa Healthcare) 17(6) (December 2008): 643 – 655.

Wicket Stuff. n.d. www.wicketstuff.org (accessed September 2010, 11).

Wium Lie, Hakon, and Bert Bos. *Cascading Style Sheets: designing for the web.* Essex: Addison Wesley Longman Limited, 1997.

APPENDIX

APPENDIX A

Timeline of events (meetings with the Supervisor and Client and events)

Date: 8 June 2010

Topic: 1st Meeting with Project Client (Akan Ibanga)

On Tuesday 8th I had the first meeting with the project client, Akan Ibanga.

Context:

Mr Ibanga is a phycology department member working in the Alcohol, Drugs and Addiction group. The group already has a web site that offers a self-help program for family members. This website is written in PHP and it is simple and static with no more features than the self-help content.

Meeting Content:

The meeting started with a brief presentation of the client (Mr Ibanga) and project developer (myself, Mr Zozaya). In the meeting Mr Ibanga, as representative of the Alcohol, Drugs and the Family group, presented what the project was about.

Summarising his wills, he expressed that he wanted the existing website to be more interactive with the user in order to make the program more helpful, and at the same time, more attractive to the end user. He was as well interested in introducing a forum for the use of the users where they could support and be supported by other users using the system. The third and last improvement in which he was interested, in was the introduction of another type of user in the system, the professional. Slightly introduced this "professional" would be a phycology professional that would have the possibility to monitor the progress and be in contact with a set of users.

To end the meeting, we shared our contact information and arranged a meeting to further analyse the requirements of the system.

Date: 10 June 2010

Topic: Meetings with Project Client: 2nd meeting with Mr Ibanga

On the Thursday 10th of June I meet with Mr Ibanga in other to further discuss the specification of the web site's features that the Alcohol, Drugs and Addiction group of the Phycology department of the University of Birmingham wants.

Duration of the meeting: 2 hours (5-7pm)

Assistants: Mr Ibanga and Mr Zozaya.

Meeting Content:

Mr Ibanga and I went further in the specification of the desired new features fro the Alcohol, Drugs and Family website. In the following attached document the discussed contents are displayed.

(Although I attempted to record the meeting the recording stopped in minute 1:24 without noticing it. Therefore it does not exist a consistent recording of the meeting)

Specification of improvement on ADF website:

Date: 17 June 2010

Topic: Meetings with Supervisor: 2nd Meeting

During of the meeting:

The supervisor was informed that the revision of the requirements was going to be delayed. The reason for this was that the project client was about to meet with the Alcohol, Drugs and Families members for the discussion of the requirements of the new website that they desired.

The project developer was informed that he could he could submit the project proposal, a week later, for the 25th of June.

Notes:

This is stated as second meeting as the "1st Meeting with the Project Client" is also considered 1st meeting with the project supervisor.

Officially, referring this as a meeting is not correct as it was really an email communication. However, for reasons of consistency and easier following of events it is reported as a meeting.

Date: 21 June 2010

Topic: Meetings with Project Client: 3rd Meeting

On the Monday 21st of June I meet with Mr Ibanga in other to present the requirements specification to him.

Assistants: Mr Ibanga and Mr Zozaya.

Meeting Content:

Mr Ibanga and I revised the requirements specification for the project. As a result of this meeting the specification was further defined and corrected. In the following attached document the second requirement specification is displayed.

Specification of ADF website:

Date: 22 June 2010

Topic: Unexpected event 1: National Identification Document Expiration

Due to the eminent expiration of the National Identification Document of the project developer, Mr Miel Zozaya Garcia, a travel to Spain was required in order to renew this document.

During the absence from England the first draft of the Solution design was developed:

Date: 25 June 2010

Topic: Submission of Project Proposal

The Project Proposal was sent to the supervisor of the project, Dr Behzad Bordbar. Here I attach the project proposal document:

Date: 8 July 2010

Topic: Meetings with Supervisor: 3rd Meeting

During the meeting:

- The supervisor was showed the first cut diagram of the solution design.

- The further developments of the design where discussed.

- Goals for next week:

    - Development of the entire Solution design:

        - Class diagrams.

        - Database relational diagrams.

Date: 15 July 2010

Topic: Meetings with Supervisor: 4r Meeting

During the meeting:

- The supervisor is shown the solution design that was requested the 3r meeting. The supervisor suggests some changes on the design.

- Goals for next week:

    - Correct the solution design.

    - Show the solution design to the client.

    - Start the implementation of the product.

Date: 22 July 2010

Topic: Meetings with Supervisor: 5th Meeting

During the meeting:

- The supervisor is shown the final solution design. Further questions of it, as, for example, the representation of frameworks on the design, are resolved.

- He is told that the meeting with the client wasn't possible during the week and that is programmed for the 22nd of July, same date of this meeting, at 4:30.

- Goals for next week:

  - Research on possible licensing requirements on Adobe Flash technologies.

  - Goal for the 2nd week of august: Finish the implementation of the product.

Date: 22 July 2010

Topic: Meetings with Project Client: 4th Meeting

During the meeting:

- The project client is shown the final solution design.

- The new structure of the exercises is discussed. This new structure aims making the users more engaged to the system. They were two basic ideas behind. The first is to provide users with video introductions to the steps that will define the purpose of them and show real life experiences of people. The second is provide the support program with a game sensation for the user by giving gratifications at the end of the steps (in the form of graphic animations) and options at the end of the step.

- The user interface is discussed with him:

During this talk the project developer proposes that the website should use other colours than the ones used in the previously developed website. This proposition is welcomed by the project client. Following this the project client is shown a brunch of websites, many of them related to problem supporting, in order to discuss the user interface of the project. The websites that where shown are the following:

- Welcome to the Apple Store - Apple Store (U.S.) - http://store.apple.com/us
- eCHECKUP TO GO :: San Diego State University Research Foundation - http://www.echeckuptogo.com/usa/research/
- Welcome to the Drinker's Check-up - http://www.drinkerscheckup.com/index.cfm?CFID=63492642&CFTOKEN=62966431
- Welcome to Moderate Drinking - http://www.moderatedrinking.com/home/default_home.aspx?p=register_login

- FearFighter - Panic & Phobia Treatment Online - http://www.fearfighter.com/patient_website/index_patient.html
- Patients - Beating the Blues® - http://www.beatingtheblues.co.uk/patients/
- e-couch: Welcome - http://ecouch.anu.edu.au/new_users/welcome01
- Living Life To The Full - http://www.livinglifetothefull.com/
- Smoking Cessation Service Research Network: SCSRN - http://www.scsrn.org/index.html

This discussion leaded to the user interface of the project.

- The project client presents the project developer new requirements that he has for the project. This leads to "Unexpected event 2".

- The project client is asked to provide the project developer with information. This information includes: specification of the questionnaires and scoring (evaluations) of them and information relating the new requirements (see "Unexpected event 2" for more details).

- The project client is told by the project developer that the new requirements will be discussed with the project supervisor and that the direction of the project from this point on may change due to the time limit of the summer project.

Date: 22 July 2010

Topic: Unexpected event 2: Requirement Amplification

During the 4th meeting with the project client, he presented the project developer with further requirements for the project.

The new requirements were the following:

- A new administration tool. This is essential for the interests the professionals working on the website as they gave them data to analyse. The new tool is required to provide them with information from the database in .xls format. Moreover, the information that they wanted aggregates the following requirements:

  - Demographic information of the users must be saved. The specification of this information that was asked to the client is attached bellow.
  - Answers of the questionnaires must be saved on the database so they can be displayed afterwards.
  - Information on the activity of the users, on the self-help support program, must be tracked and saved in the database. The interest of this is to know the time spent on each part.

- The self-help support program may have changes in a far future so it would be desirable if they could be editable from the administrator tools.

- In user administration of the administrator tools a user search functionality would be desirable.

- There are 3 questionnaires and they must have independent scoring.

Due to this a new requirements document is written. This document is attached.

Attachments:

Date: 26 July 2010

Topic: Meetings with Supervisor: 6th meeting

During this meeting:

- The supervisor is presented the contents of the meeting with the project client.

- The project developer and the supervisor discuss the direction that the project must take as there is no possibility of completing the entire website on the project time line.

The solution taken for this problem is to define a new scope for the project. The new scope states:

- The solution design must consider solutions for all the requirements

- All the functionalities will be implemented.

- Repetition of those functionalities will not be implemented. In example, a unique forum will be implemented.

- A guidelines section will be introduced on the dissertation document that will explain how to replicate those functionalities.

Goals for next meeting:

- Develop the new requirement document.

- Develop the new solution design.

Date: 5 August 2010

Topic: Meetings with Supervisor: 7th meeting

During the meeting:

- The supervisor is presented with the new requirement document and solution designs. He is satisfied with both.

- The supervisor is presented with the mini research about flash usage that said that the publication of .swf content can be done with any license restrictions and that the only payable license will be the one of the program used to create the content.

- He is also presented the progress in the implementation of the web application and the decision of the project developer to merge the implementation and the testing together.

- Goals for next week:

    - Continue developing and testing the web application.

Date: 6 August 2010

Topic: Unexpected event 3: Essay writing

The project developer is informed that he has de possibility of applying for a Fisher Scholarship for the 2010/2011 academic year. This application requires an essay writing. This essay writing stops the project development for a week as it required for the 13th of August.

Date: 10 August 2010

Topic: Meetings with Project Client: 5th meeting

During the meeting:

- The project client is presented with the advances on the progress of the web application. The project client is satisfied with it.

- The project developer proposes the idea of hosting the videos that the website will provide to their users on a video hosting website as YouTube. The reason for this proposition is that it could attract the new website clients that did not know about the website before the same time that reduced the websites host storage needs. But the project client was not attracted by the idea as he wonted the videos to be hosted locally to translate a more professional impression to the clients.

- The project client is presented and explained the new direction that the project could take if he is satisfied with that. The project client accepts the new direction of the project.

Date: 12 August 2010

Topic: Meetings with Supervisor: 8th meeting

During the meeting:

- The supervisor is informed about the unexpected event of the 6th of August (see "Unexpected event 3").

- The supervisor is informed with the contents of the meeting with the project client on the 10th of August.

- Goals for next week:

- Develop an outline for the dissertation document.

Date: 19 August 2010

Topic: Meetings with Supervisor: 9th meeting

During the meeting:

- The supervisor is presented with the dissertation outline. The outline is discussed and edited.

- Goals for next week:

  - Finish the implementation of the product.

  - Write two chapters of the dissertation.

Date: 26 August 2010

Topic: Meetings with Supervisor: 10th meeting

During the meeting:

- The project Supervisor was commented that only a small part of the implementation was left and that the project report was already started to be written.

Goals for next week:

- Prepare the project demonstration.

Date: 31 August 2010

Topic: Meetings with Supervisor: 11th Meeting

During this meeting:

- The project supervisor is shown the project demonstration slides. The slides are revised and corrected.

Goals for next week:

- Make a big advance on the writing of the project report.

APPENDIX B

Use Cases (not included on the report's main body)

1. *Browsers' Use Cases:*

Name: Login
Number: UC-3
Preconditions: • The user must hold an account in the system.

Flow of events:
• The user enters the username and the password.
• The system verifies the credentials:
  ○ If the credentials are not correct the system informs the user that the login has failed.
  ○ If the credentials are correct but the user is blocked the system informs the user that the login has failed.

Postconditions: • The user acquires the permissions that his type of account gives him.

Name: Request a new password
Number: UC-4
Preconditions: None
Flow of events: • The user enters the username of its account. • The system verifies if the username exists: ○ If the username does not exist the process ends without notifying the user. • If the username exists an email with a new generated password is sent to the email related to that username.
Postconditions: • The user with the specified username has a new password.

2. *Clients' and Professionals' Use Cases:*

Name: Enter system feedback
Number: UC-5
Preconditions: • The user must be a Client or Professional.
Flow of events: • The user enters an opinion on the system of him.
Postconditions: • The feedback entered is saved on the system.

Name: Enter self-help feedback
Number: UC-9
Preconditions: • The user must be a Client or Professional. • The user is in the self-help program.
Flow of events: • The user enters feedback about the self-help program. He enters information on how many time has been taking the program, which is the improvement that he has experienced and a feedback statement.
Postconditions: • The feedback is saved in the system.

Name: Change password
Number: UC-11
Preconditions: • The user must be a Client or Professional or Administrator.

Flow of events:
• The user enters the user's current password, the new password and the confirmation of the new password.
• The system validates the information:
  ○ If the new password and the confirmation of it are not equal the system informs the user and aborts the change.
  ○ If the current password entered does not match the password that the user currently has the system informs the user and aborts the change.
  ○ Else if everything is right the password is changed.

Postconditions: • The user has now new password as his password.

3. Clients' Use Cases:

Name: Break the Professional links
Number: UC-14
Preconditions: • The user must be a Client and have a link to a Professional.
Flow of events: • The user selects to deny Professional supervising and/or deny Professional contact.
Postconditions: • The Professional of the user can no supervise and/or contact his Client

4. Professionals' Use Cases:

Name: Use professionals' forum
Number: UC-24
Preconditions: • The user must be a Professional.
Flow of events: • The user selects to use the professionals' forum. • The system allows the user to post, see, reply messages and see messages for a certain category in the professionals' forum.

5. Administrators' Use Cases:

Name: Read the self-help program feedbacks
Number: UC-26
Preconditions: • The user must be an Administrator.
Flow of events: • The user selects to see the feedbacks that the users have entered. • The system shows the feedback to the user.

Name: Moderate system feedbacks
Number: UC-28
Preconditions:

• The user must be an Administrator.
Flow of events: • The system shows the user the system feedbacks that the users have entered. • The user can decide to delete or to publish a feedback. • If he decides to publish a feedback this feedback will be then shown in the Feedback webpage when users access it.
Postconditions: • If the user has decided to delete a feedback the feedback will no longer exist on the system. • If the user has decided to publish it, it will be loaded when Feedback webpage is requested.

Name: Edit Message
Number: UC-31
Preconditions: • The user must be using a forum (Client of Professional) • The user must have selected a message to see. • The user must be an Administrator.
Flow of events: • The user edits the message body and title and submits the message. • The system updates the message and notifies the user that posted it about the change.
Postconditions: • The selected message is edited and the user that posted the original message has a message in his inbox that notifies the change.

Name: Add category to a message
Number: UC-32
Preconditions: • The user must be using a forum (Client of Professional) • The user must have selected a message to see. • The user must be an Administrator.
Flow of events: • The user selects a category to add to the message from the list of categories for that forum. • The system adds the category to the message and notifies the users that had relation with the message about the change.
Postconditions: • The message now pertains also to the added category and the users that had relation with the message have an inbox message that notifies the change.

Name: Remove category to a message
Number: UC-33
Preconditions: • The user must be using a forum (Client of Professional) • The user must have selected a message to see. • The user must be an Administrator.
Flow of events: • The user selects a category to remove to the message from the list of categories of the message. • The system removes the category from the message and notifies the users that had relation with the message about the change.
Postconditions: • The message now does not pertain to that category and the users that had relation with the

message have an inbox message that notifies the change.

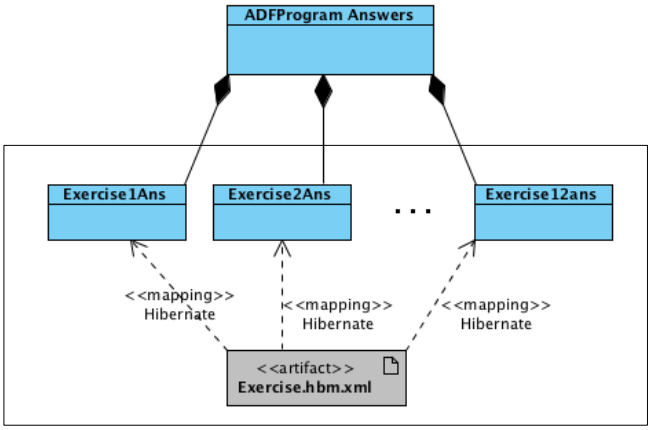
Name: Create a category
Number: UC-34
Preconditions: • The user must be using a forum (Client of Professional) • The user must be an Administrator.
Flow of events: • The user enters a name and description of a category to introduce and submits. • The system checks that the category does not already exist. • If the category does not exist the new category is added.
Postconditions: • The forum categories has now a new category

Name: Remove a category
Number: UC-35
Preconditions: • The user must be using a forum (Client of Professional) • The user must be an Administrator.
Flow of events: • The user selects a category to remove from the forum's category list. • The system checks if the category is not "All". If it is aborts. • The system removes that category from the list and from the messages that where classified into that category. This does not delete the messages as they all pertain to "All" category that couldn't be deleted.
Postconditions: • The category is deleted from the system.

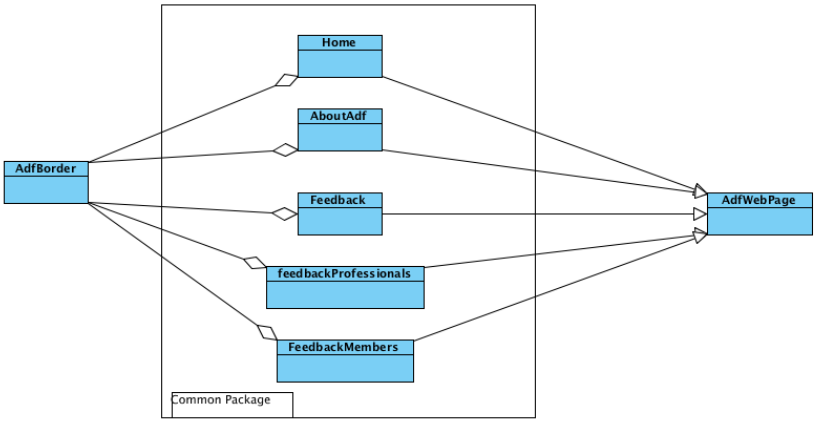
APPENDIX C

Class diagrams (not included on the report's main body)

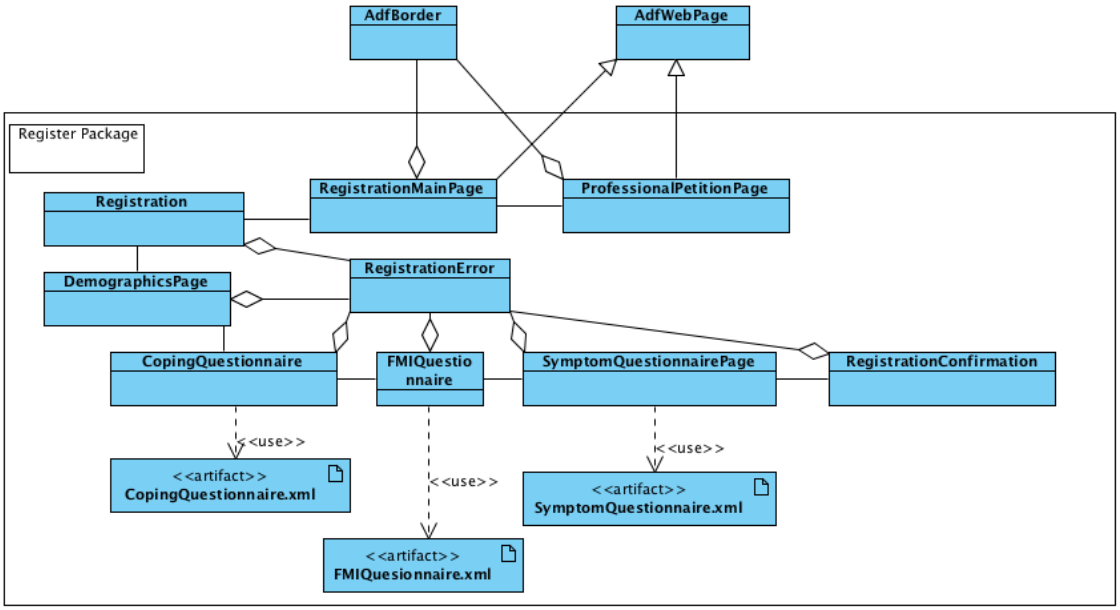
Package adf.model.exercises:



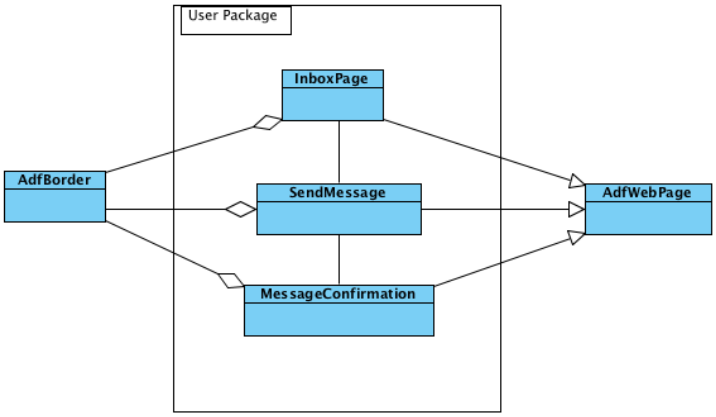
Package adf.page.common:



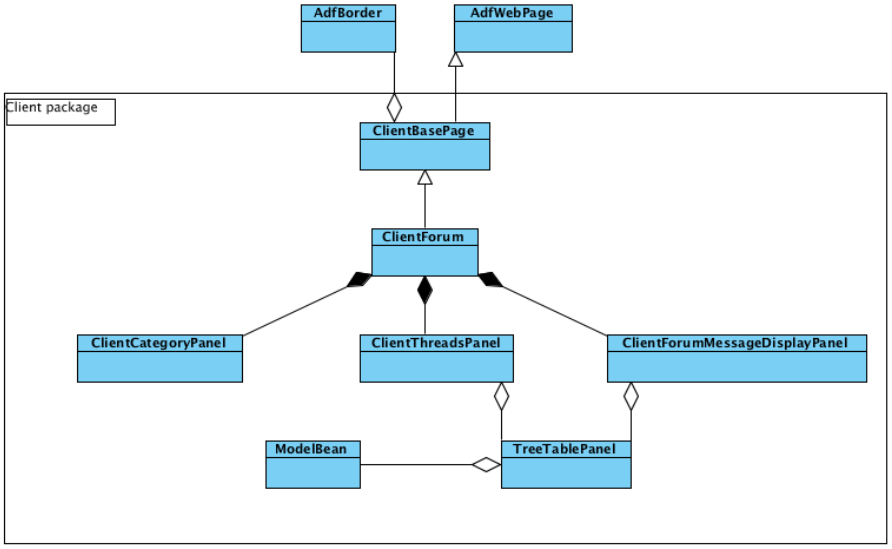
Package adf.page.register:



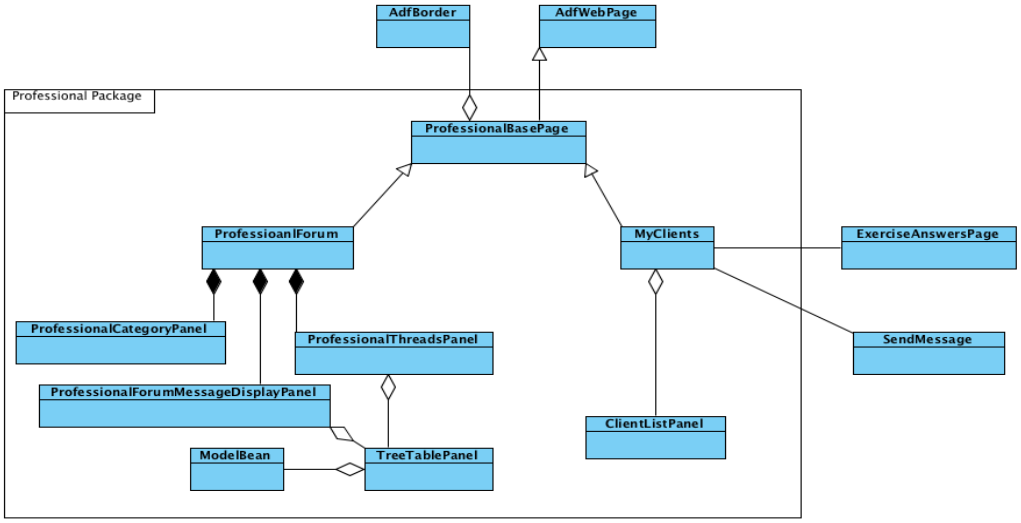
Package adf.page.user:



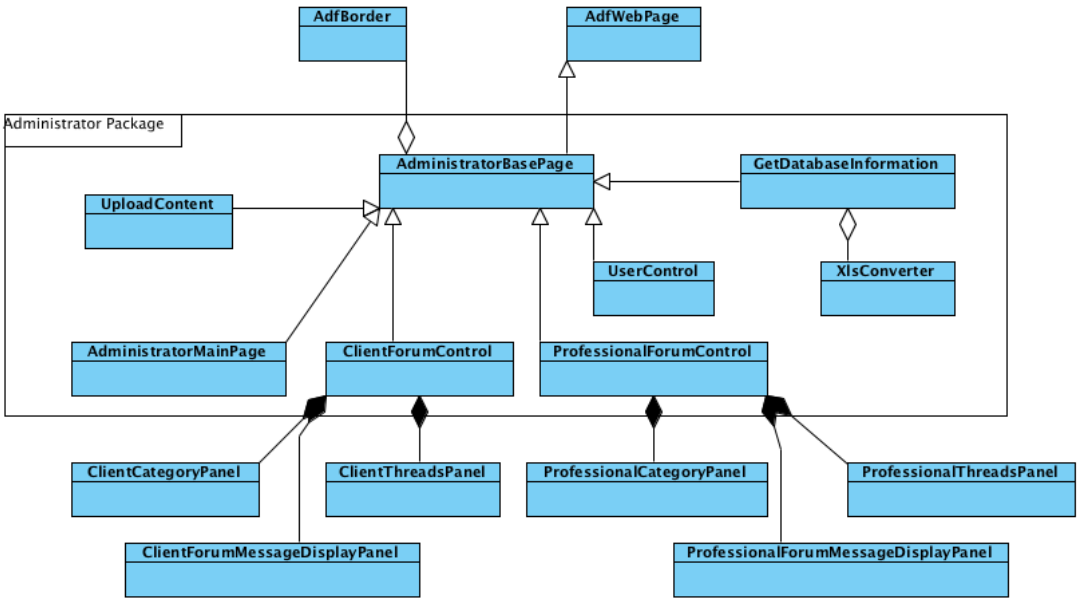
Package adf.page.client:



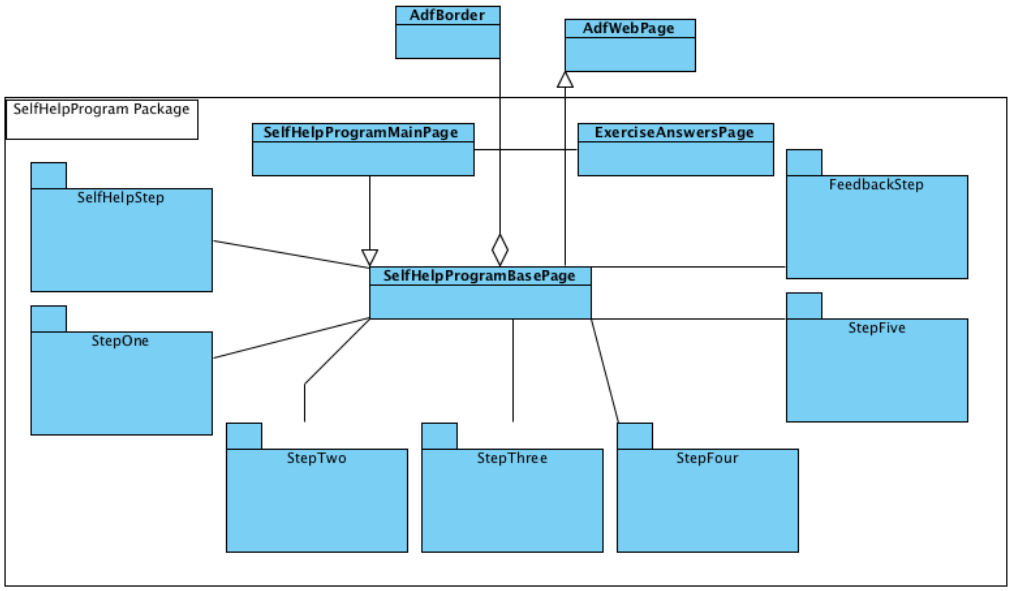
Package adf.page.professional:



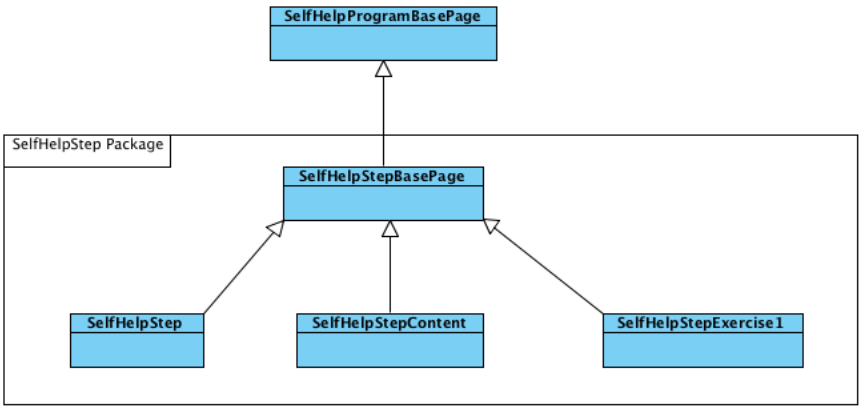
Package adf.page.administrator:



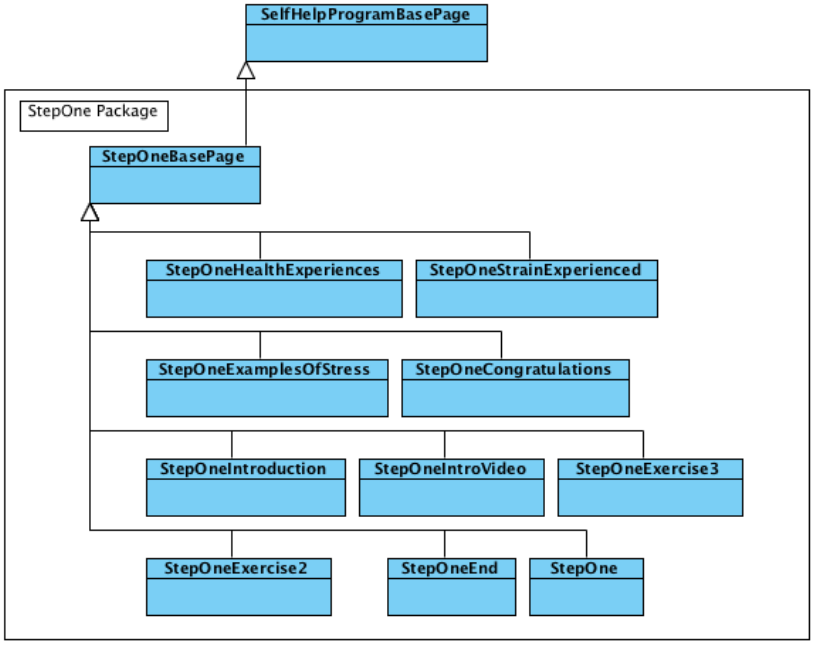
Package adf.page.SelfHelpProgram:



Package adf.page.SelfHelpProgram.SelfHelpStep:



Package adf.SelfHelpProgram.StepOne:



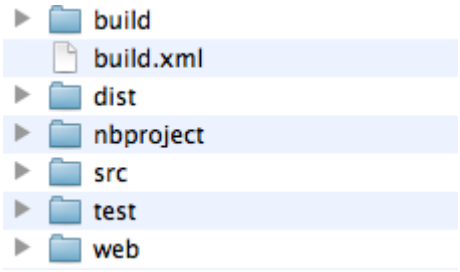
APPENDIX D

Notes on the report's accompanying CD structure and on how to run the prototype.

Inside the CD:

Inside the CD three folders and a PDF file can be found. The Dissertation-Miel Zozaya Garcia.pdf file holds this document. One of the folders is called 'Related Documents'. Inside is folder can be found the documents that are referenced from this project report as 'see in CD'. 'Library' folder contains the libraries that have to be added to run the project.

The 'ADFWebsite' folder contains the developed software's NetBeans 6.8 project. Inside it the following folders can be found:



Instructions on how to run the software:

1. Firs thing that is needed to run the program is to have NetBeans 6.8 IDE, Apache Tomcat 6.0.28 and PostgreSQL database installed.
2. Open the NetBeans' project. (ADFWebsite is a NetBeans project)
3. Make sure that Apache Tomcat server is configured in NetBeans.
4. Change the database details in the applicationContex.xml file that is located on the default folder of the source files. The database location, username and password have to been set to meat the way that you configured the PostgreSQL.
5. If the libraries are not correctly inserted on the project, erase the bar referenced libraries and add the ones provided in the CD's 'Libraries' folder.
6. Clear and compile the program.
7. Run the application.