A MOBILE PLATFORM FOR ADMINISTERING QUESTIONNAIRES AND SYNCHRONIZING THEIR ANSWERS

Maria Germana Ginardi and Giordano Lanzola
Department of Electrical, Computer and Biomedical Engineering
University of Pavia, Via Ferrata 1, 27100 Pavia, Italy

ABSTRACT

This paper describes a platform for administering questionnaires on smart-phones and tablets. The project arises from the need of acquiring data for monitoring the outcomes of different homecare interventions. First a model has been defined for representing questionnaires, able to support adaptivity in the dialog with the user and enforce some simple consistency rules for checking his input. Then an application has been implemented able to run instances of those questionnaires. It downloads questionnaires over the air in terms of XML files from a server and stores them locally into a local repository. At that point questionnaires become available for the user who can select which one to fill in according to his needs and the specific treatment protocol. A questionnaire may be filled in all at once or be completed in subsequent steps over time since the input provided is persisted on a local database. Finally, when a questionnaire is closed all the answers are transparently synchronized to a server for further evaluation or statistical purposes.

KEYWORDS

Mobile Applications, Multimedia Applications, Computer-Mediated Communication, eHealth Policy and Practice, Evaluation and Assessment.

1. INTRODUCTION

Techniques for administering questionnaires are varied and evolved over time in parallel with technological progress. Early techniques involving face to face interviews or paper-based questionnaires sent by surface mail turned into telephone interviews and more recently into massive e-mails, phone calls automatically collecting answers, and web interviews. That evolution is motivated either by the possibility of reaching an increasingly larger population sample as well as by performing an improved analysis on the outcomes with a dramatic decrease in the time and costs required for setting up and processing the study [Dillman, D.A. et al., 2009].

Coming to the medical domain, the use of surveys for epidemiological studies is popular among pharmaceutical companies and organizations dealing with public health care [Valente, T.W. 2007]. Results emerging through those studies often have a great impact on starting public health care campaigns. The specific point of view of the patient is also gaining focus by the modern medical practice since it provides a deeper insight on the perception of the quality of care and also accounts for its personalization [Baker, D. et al., 2005]. To this aim questionnaires are being used in health care and social welfare contexts for assessing resource use, access to treatments, quality of services and patient satisfaction in order to improve the overall efficiency and effectiveness of the services.

A different perspective exploits surveys to assess the patients' quality of life. This provides an indirect estimation of the outcomes of medical actions in terms of treatments efficacy or the patient's own health perception and constitutes the foundation for economical (cost/utility) studies considering the costs sustained by the national services [Carr, A.J. et al, 2001]. Measuring the quality of life is also important for assessing the evolution of chronic diseases which are increasing due to population aging and changes in people lifestyle. For example, the EuroQol-5D is a popular questionnaire developed by the EuroQol Group [Rabin, R. and de Charro, F., 2001] administered to patients in many countries and in a variety of clinical areas to assess a generic measure of health status in terms of a simple descriptive profile and a single index value.

This paper describes a mobile platform available on smart-phones and tablets running the Android operating system for rendering questionnaires and collecting answers from users. The questionnaires are downloaded over the air and stored into a local library. Once in the library they become available for entering answers at user will. This means that they can be filled all at once or incrementally at different times. Answers provided by the user are checked against a model which also provides the basis for personalizing the navigation across questions: our platform allows the dynamic adaptation of questions according to previously provided answers. When the user is finished with an instance of a questionnaire, the instance is closed and all the answers are synchronized with a central repository so that they can be accessed for a subsequent analysis. The platform also supports the complete substitution of a questionnaire by downloading a new one, which is a key issue for dealing with new emerging scenarios.

The work has been motivated by the need of supporting two different scenarios related to ongoing research projects in health care at our institution. The first one is meant to improve after-discharge home assistance for *fragile* patients, specifically preterm newborns and elderly renal patients. The need to keep medical expenses under control are pushing towards home care with respect to hospital confinement for this kind of patients. A questionnaire has been designed to collect the needs of those patients and their home caregivers, as a preliminary step for the development of a tele-homecare system. The second scenario addresses the MobiGuide project funded by the European Union and dealing with support for patients guideline enforcement anywhere and anytime including at their domiciles. Patients are provided with tools helping them in adhering to the guideline related with the treatment, and are asked to periodically report information about their satisfaction with the support provided by MobiGuide as well as their clinical status and progress with the treatment. In both cases questionnaires are used as a means for reporting information to the clinic center.

2. DESIGNING THE PLATFORM

The first task to be accomplished when setting up a survey, soon after the main goal of the study is defined, deals with identifying the way in which the questionnaire is to be administered and how data should be collected, as well as with the actual design of the items building up the questionnaire. This involves preparing a clear textual definition for each question, rendering them with a proper wording and partitioning all of them into a suitable set so that they can be proposed into the most appropriate order to the user [Morrison, R.L. et al., 2010]. Structuring the questions is particularly important as the target user perceives their sequence as a virtual conversation through which the survey designer elicits his answers. In order to properly motivate the user and capture his attention, that conversation should avoid being repetitive, trivial or ambiguous. To this aim in a written questionnaire directions are helpful for making explicit the meaning of each question and the specific intent associated with their answers, although a careful tradeoff should be considered. While the absence of directions could be misleading for the user and possibly result in eliciting the wrong answers from him, an overly long set of instructions could be perceived as obtrusive and demotivate the user thus yielding the opposite result [Smyth, J.D. et al., 2006].

2.1 The Questionnaire Model

Designing a platform for rendering questionnaires implies as a prerequisite the capability of representing the questionnaire itself. Thus we developed a model with a high level of flexibility in order to better capture and shape the dialogue with the user, which is described in the following. Within the model developed, a *question* is a group of information that must be shown together to the user. Its main components are the *question text* and one or more *answers*. The former conveys the primary meaning of the question and provides a framework for the input to be supplied by the user. Answers represent instead some ancillary information acting as placeholders and shaping the context for the actual input. Thus a single answer may include multiple input items which are all considered related and should be consistently filled in.

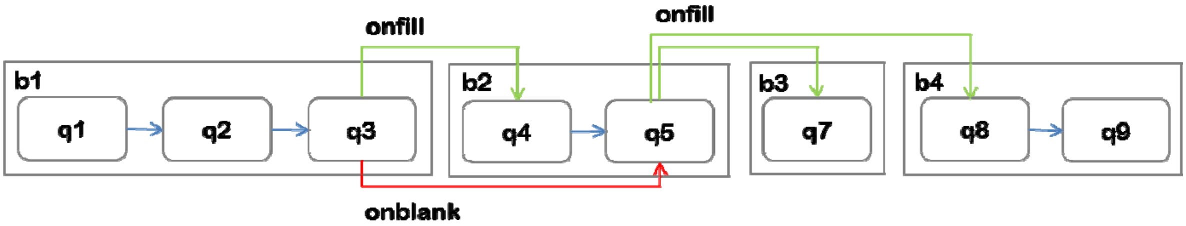


Figure1. Navigating through the questions of a questionnaire.

Following a short review concerning the question types addressed in the literature we decided to model an answer with fixed parts called *labels*, which are not user editable, serving the purpose of introducing *inputs*, where the user actually enters information. Data may be entered into an input either by typing or by selecting one of the available choices associated with it. In designing the semantics of input elements we capitalized on the Html experience allowing a broad set of types such as *text boxes*, *checkboxes*, *radio buttons* and *choice selections*. We also allow structured input types, such as the *yes-no* choice which is rendered by the application through primitive types such as buttons or multiple choices. Finally, for those answers requiring it, *answer instructions* may also be available to better specify the meaning of an answer or clarify the question context altogether in order to support the user choice.

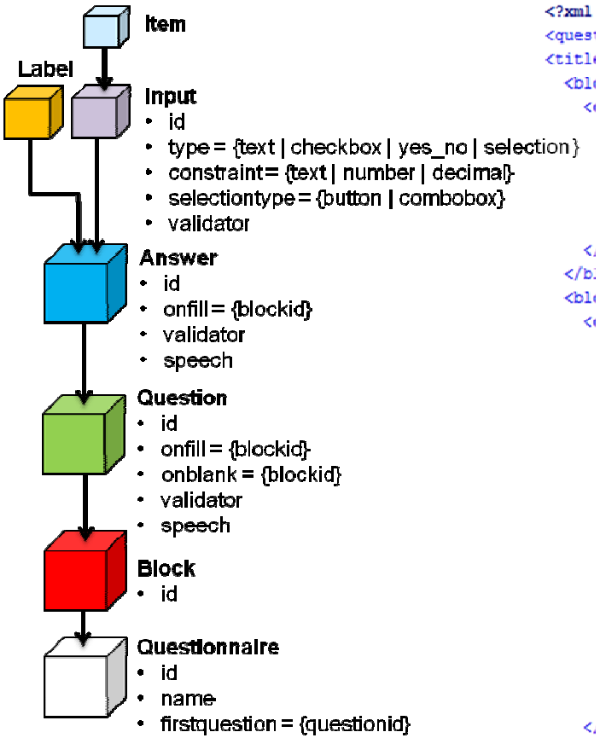
Another important requirement of the model concerns the definition of constraints for the questionnaire which help in reducing errors and checking the correctness of the input provided by the user. We adhered to a very simple constraint model for fields where input can be restricted to the available choices, and possibly be marked as mandatory. Additional constraints may also be defined at the answer or question level, for specifying the number of mandatory input or answer combinations.

In order to provide a dynamic adaptation of the questionnaire which better captures the needs of a monitoring scenario in a clinical context, the model also includes *navigation rules* for shaping the correct sequence of questions according to the previous answers provided by the user. For example, a compound question might be "did you require assistance from medic/paramedic staff during the treatment" ? If the user answers "no", no further inquiry concerning the staff is performed. However if the answer is "yes" a section is entered asking specific information detailing the assistance required. This should be repeated as many times as there are invocations of assistance issued by the user.

Navigation rules can be set up on a whole question or on specific answers, as shown in Figure 1. In fact questions are arranged into *blocks* inside which navigation proceeds sequentially unless a specific rule is established. For example in block *b1*, question *q1* is always followed by question *q2* no matter how the user replies to the answers included in question *q1*. At the question level two different rules may be defined addressing the fact that the user provided some input (i.e. *onfill*) or not (i.e. *onblank*). This is represented by the two lines departing from question *q3*, meaning that if the user provides some input to question *q3* then question *q4* is shown, if he doesn't question *q5* is accessed instead. Answers may only specify an *onfill* rule pointing to the next question to be accessed if the user provides some input to that answer. For example, the two lines departing from question *q5* and pointing to questions *q7* and *q8* represent the *onfill* rules relating to two different answers available in *q5*. If there are no navigation rules for a terminal question of a block the questionnaire is considered finished, as it happens in questions *q7* and *q9*.

The navigation rules devised make it possible to skip some questions or move to explicit ones depending on the user answers, therefore better shaping the virtual interviewing conversation with him. By properly structuring a question in terms of answers and inputs, placing constraints at the question, answer or input level, and defining suitable navigation rules a great degree of flexibility is achieved, able to capture the requirements of almost any complex questionnaire scenario. For example this accounts for excluding blocks altogether depending on previous user input or repeating some questions several times in order to collect all the user's information required for a study.

Given the description of the model proposed, XML seemed to be the most appropriate representation language for encoding a questionnaire. Figure 2 illustrates on the left all the elements referenced in the XML schema while an excerpt of a questionnaire is reported on the right. The root element is *Questionnaire* including *Block* elements which then encompass *Question* elements. Each question is composed of several 7 *Answers* including *Labels* as fixed components or *Input* elements representing placeholders for the user input.




```xml
<?xml version="1.0" encoding="UTF-8"?>
<questionnaire id="1" name="questionnaire" firstquestion="q1">
<title>Home context and need for assistance questionnaire</title>
  <block id="b1">
    <question id="q1" onfill="b2" onblank="b3" validator="-1"
      speech="Did you have problems with drug consumption?">
      <title>Did you have problems with drug consumption?</title>
      <answer id="a1" validator="-1">
        <input id="i1" type="checkbox" validator="-1"/>
      </answer>
    </question>
  </block>
  <block id="b2">
    <question id="q2" onfill="b5" validator="-2"
      speech="Which kind of problem?">
      <title>Which kind of problem?</title>
      <instruction>Specify at least one</instruction>
      <answer id="a2" onfill="b3" validator="-1"
        speech="Specify at least one">
        <input id="i2" type="checkbox" validator="-1"/>
        <label>Wrong dosage</label>
      </answer>
      <answer id="a3" onfill="b4" validator="-1">
        <input id="i3" type="checkbox" validator="-1"/>
        <label>Wrong preparation</label>
      </answer>
      <answer id="a4" validator="-1">
        <label>Other: </label>
        <input id="i5" type="text" validator="-1"/>
      </answer>
    </question>
```

Figure 2. The XML elements representing the questionnaire on the left and an excerpt on the right. The excerpt refers to the questionnaire for the parents of preterm newborns.

The XML section also shows some navigation rules set up for question and answers in terms of *onfill* and *onblank* attributes. Constraints are also defined through the *validator* attribute holding a code corresponding to the specific action. For example the validator code *-2* in question *q2* means that it is mandatory to fill in at least one answer, but more are allowed as well. The validator code for any answer included in question *q2* is instead *-1*, which means that their filling is optional, as it also happens for any included input item. In fact, in that case each answer only includes a single input item so the same validator is used at both input and answer level. Finally, the *instruction* element and *speech* attributes provide suggestions which may be rendered as side notes or as spoken language by the renderer.

2.2 The Functional Architecture of the Platform

The model proposed allows the definition of a questionnaire along with all the associated constraints and navigation rules. However this only helps in representing the set of questions and still leaves open the problem of collecting feedback from patients during clinical studies involving experimental protocols. Thus we needed an application able to support running instances of those questionnaires and we selected smart-phones and tablets as the target devices for the implementation. This is in line with current studies concerning clinical diaries showing that users are more comfortable with portable devices since they are available and ready to use at any time and in every place whenever they are needed [Franc, S. et al., 2011]. We also selected the Android operating system because it is open-source and available on multiple hardware.

Figure 3 shows the overall functional architecture of the platform illustrating the full workflow supported by it. That workflow starts with the *design* of a new questionnaire, which basically entails writing the XML file representing it, and *deploying* it as a module on a server repository as shown in the upper part of the figure. The mobile application notices the availability of a new questionnaire module, then *downloads* and saves it into its local module repository. A questionnaire may either supersede an old one, therefore substituting it in the application library, or be stored as a new one representing an additional choice for the user.

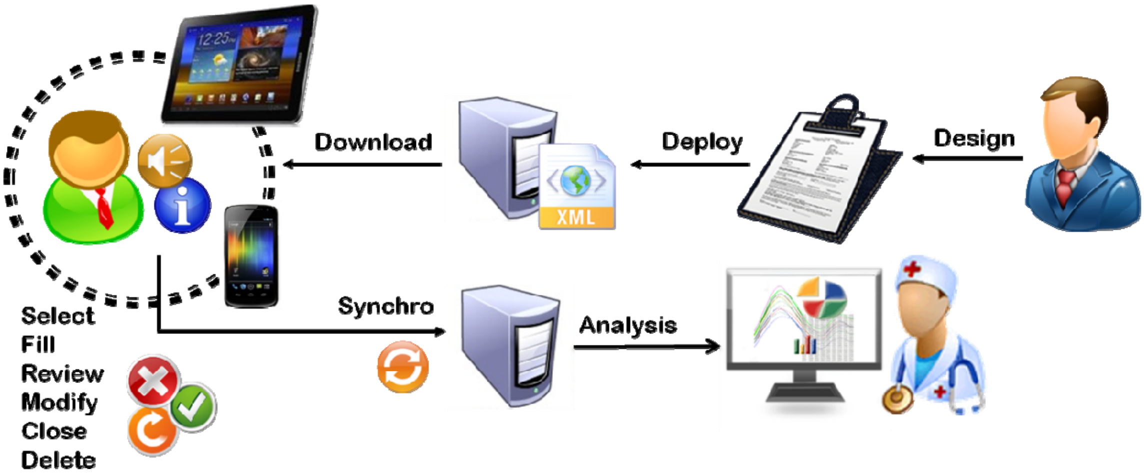


Figure 3. The functional architecture of the platform.

When the user accesses the mobile application he may *select* a questionnaire among those available in the local repository. Then he can start *filling in* answers as required by its treatment protocol. We provide two different help modes supporting the user in filling in a questionnaire. The XML model allows the definition of an associated instruction text for each question which is rendered by the application as a side note dynamically shown or hidden on user demand. Furthermore, the same model allows an optional explanation text for both questions and answers which the application renders through the embedded text-to-speech library provided by Android. Similarly when providing input the user may exploit the embedded speech recognition engine which is constrained by the type associated with the input field. Taken as a whole the possibility of using the speech both for hearing directions and entering input simplifies the task of filling in a questionnaire and improves both the usability of the application and the user experience.

The answers provided by the user are incrementally persisted in a local database so that they are not lost if the user stops a session before finishing the whole questionnaire. The next time he accesses it he faces the option of continuing with the suspended one. In that case he may *review* the previously provided answers, *modify* them and finally *close* the questionnaire or *delete* the suspended work altogether. Each questionnaire may be answered multiple times by the user according to the treatment protocol, in which case the answers will be correctly grouped together by instance and remain consistent. When a questionnaire is finished, the answers provided by the user will be *synchronized* to a database in order to perform the required *analyses* by the clinicians. The synchronization task requires authentication in order to save the results. Depending on the study each patient can be allotted a different credential for providing his answers or use a shared space enforcing anonymity.

2.3 The Computational Architecture of the Android Application

The application for rendering questionnaires has been developed on top of the Android operating system, thus it is able to run on all the android-powered devices which have very different capabilities ranging from small smart-phones to wide tablets. The feature impacting the application behavior at most is the screen size, and depending on the complexity of the information conveyed in the questionnaire it is therefore advisable to select a suitable device.

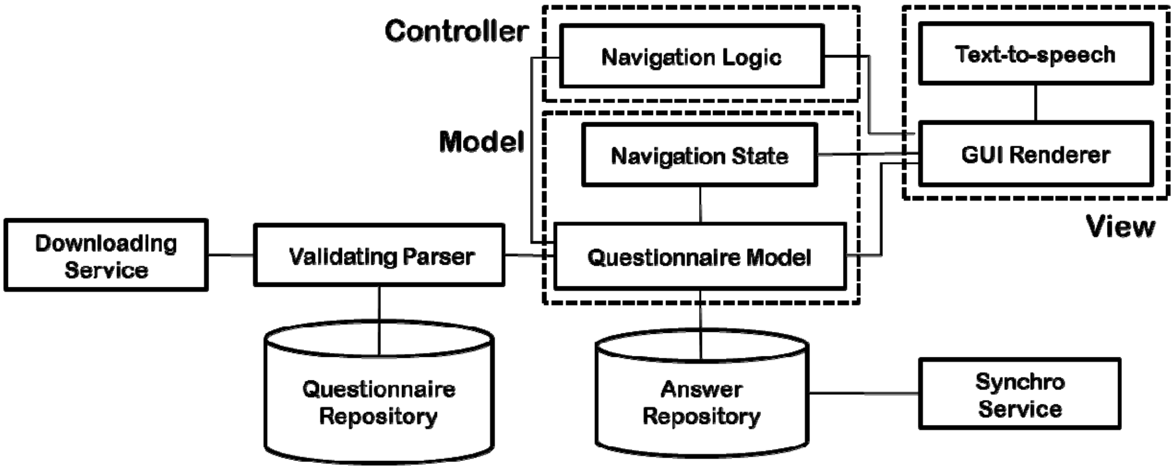


Figure 4. The computational architecture of the Android application.

Figure 4 shows the computational architecture of the Android application illustrating all the software components building it. The files including the XML representation of questionnaires are checked on the server repository through an internet connection and possibly downloaded by the *Downloading Service*. Then they are processed by the *Validating Parser* and eventually stored in the *Questionnaire Repository* where all them are available for the user selection. The parser checks each file against the structural and syntactical correctness of the included XML and creates the *Questionnaire Model* which is a memory structure used by the application for navigating across the questions. There is mostly a one-to-one correspondence between the elements available in the XML model and the object instances available in that memory structure. Among the checks enforced by the validating parser are those concerning the availability of all the required attributes for each item and the consistency of the navigation rules expressed in terms of the onfill and onblank attributes. The *Questionnaire Model* is represented through a series of JavaBeans extending a common class, which contains properties and behaviors shared by all the different elements referenced by a question. The model is also responsible for managing the user input which is saved into the *Answer Repository* and recovered from it whenever previously answered questions are revised. That repository is implemented using the *Content Provider* feature exposed by Android.

The application is shaped against the architectural pattern named *model-view-controller*. In our case the model provides the in-memory representation of the questionnaire where both the definition of the questions and the user input are stored. The view is responsible for assembling the layout of the questions and activating the text to speech concerning the help, while the controller enforces the *Navigation Logic* and the consistency rules defined in the model for the user input. Our implementation sees each model element matched with a view part responsible for producing its graphical representation. There is a single Android *Activity* for displaying the questionnaire which is dynamically updated by the view parts reshaping its layout whenever a new question is accessed. In fact the view part is responsible for creating all the widgets representing the title, labels, checkboxes, radio buttons and textual input fields according to the specific organization of the current question.

The *GUI Renderer* displays each question either in read-write mode or in read-only mode according to the navigation rules. The latest question is always editable and the user can input a value or select one from the available choices depending on the case at hand. The editable question also has the capability of displaying help messages. Those messages may also be uttered in spoken language by the *Text-to-speech* service based on user preferences.

If the user navigates backward and accesses a previously answered question, that one will be recovered from the Answer Repository with inputs precompiled and disabled. On any previous question he is given the chance to reopen it in order to modify the answer. However, since the navigation is affected by previous answers, reopening a question implies the truncation of all the input provided from the reopened one onwards. Once the questionnaire is finished, the renderer always gives the user a chance to review the full questionnaire in read-only mode, in order to delete it or accept and synchronize the answers through the *Synchro Service*.

3. DISCUSSION

Our primary interest in questionnaires was for using them in performing evaluation studies on home patients. Thus in designing and implementing our platform we addressed the typical needs of those patients, as we tried to combine the advantages of both paper and web based administration methods. More specifically, the availability of an application running on tablets for rendering questionnaires eliminates the requirement of possessing any skill on the user side for navigating the internet. The application can be started with an action as simple as a click on an icon proposing to choose one among the available questionnaires. Furthermore, the implementation on a mobile hardware improves the overall experience since the user is not required to be sitting in front of a networked PC. The task of completing the survey may be accomplished with fewer restrictions and is more comfortable for the user who can bring the device with him while he is busy with his everyday activities and answer the questionnaire in the most timely fashion. In fact, once the questionnaire is downloaded and stored in the local library, it may be accessed anytime even when no network coverage is available. The user is free to navigate among the questions he has already seen, possibly revising the answers, and when the survey is completed, the input becomes eligible for synchronization.

Another major concern affecting the quality of a survey relates to the questionnaire layout preparation. More specifically, the analysis of web questionnaires has determined that question ordering and visual disposition capture the user attention and affect the response rate [Couper, M.P., 2000]. It has been found that preceding questions may affect how users consider and interpret latter questions especially when some kind of relationship exists among them [Tourangeau, R. et al., 2004]. We capitalized on those results devising an adaptive model for representing the questionnaire, so that whole question blocks may be skipped whenever their answers can be inferred by those provided earlier or they turn out to be inapplicable altogether. Visual disposition arranges separate questions on different screens so that each one may be associated with help to support the user in providing the answers. A special effort has been devoted to implement an enhanced help supporting the user in properly understanding the question. This is available in the form of annotations to the questions but may also be rendered in spoken language using the text-to-speech engine available in Android.

Finally, an important issue for surveys is represented by the need to ensure an adequate quality for collected data. This should be achieved through a careful design phase where the researchers prepare a suitable set of questions whose answers are consistent with the study to be accomplished. However much too often a survey is deployed just to find out, soon after data collection has started, that it cannot be used for establishing valid and reliable explanations concerning the investigation because of some mistakes occurring in the way questions are structured [Saris, W.E. and Gallhofer, I.N., 2007]. As we are mainly concerned with surveys for collecting information during clinical studies we faced a similar problem when dealing with unexpected situations. In those cases there is no way to anticipate the problem providing consistent questions in advance. The only way to cope with that situation was to account for the adaptivity of the questionnaire itself. Once again, having implemented the engine as an application on a mobile device, we are able to dynamically update and download new questionnaires over the air as the clinicians deem to do so in a transparent way for the user.

4. CONCLUSION

The paper reported on a platform for administering questionnaires to patients enrolled in clinical studies, possibly involving experimental protocols. The specific application context required features not available on software in public domain, such as adaptivity in the virtual dialog established with the user and consistency checks on the answers supplied by the user. This called for the design of a questionnaire model and the implementation of an application able to run its instances. The application was developed on the Android operating system so that it is available on smart-phones and tablets, which allows for a better comfort and user experience. Questionnaires are made available on a server repository and automatically downloaded as new ones are deployed. At any time the user is allowed to select a questionnaire to fill in according to the specific treatment protocol. Once questionnaires are completed by the user, the input provided is synchronized to a server for a subsequent analysis step. The platform is currently undergoing evaluation for monitoring patients affected by atrial fibrillation.

REFERENCES

Baker, D. et al., 2005. A Telephone Survey to Measure Communication, Education,Self-Management, and Health Status for Patients With Heart Failure: The Improving Chronic Illness Care Evaluation (ICICE). *Journal of Cardiac Failure*, Vol. 11, No 1, pp. 36-42..

Carr, A.J. et al, 2001. Is quality of life determined by expectations or experience? *British Medical Journal*,.Vol. 322, No. 7296, pp. 1240–1243.

Couper, M.P., 2000. Web surveys – A Review of issues and approaches. *Public Opinion Quarterly*, Vol. 64, pp. 464-494.

Dillman, D.A. et al., 2009. Response rate and measurement differences in mixed-mode surveysusing mail, telephone, interactive voice response (IVR) and the Internet. *Social Science Research*, Vol. 38, pp. 1–18.

Franc, S. et al., 2011. Telemedicine and diabetes: Achievements and prospects. *Diabetes & Metabolism,* Vol 37, No. 6, pp. 463-476.

Morrison, R.L. et al., 2010. Questionnaire Design Guidelines for Establishment Surveys. *Journal of Official Statistics*, Vol. 26, No. 1, pp. 43–85.

Rabin, R. and de Charro, F., 2001. EQ-5D: a measure of health status from the EuroQol Group. *Annals of medicine*, Vol. 33, No. 5, pp. 337-343.

Saris, W.E. and Gallhofer, I.N., 2007. Design, *Evaluation and Analysis of Questionnaires for Survey Research*. Wiley, New Jersey.

Smyth, J.D. et al., 2006. Effects of Using Visual Design Principles to Group. Response Options in Web Surveys. *International Journal of Internet Science*, Vol. 1, No. 1, pp. 6-16.

Tourangeau, R. et al., 2004. Spacing, position, and order – Interpretive heuristics for visual features of survey questions. *Public Opinion Quarterly*, Vol. 68, pp. 368-393.

Valente, T.W. 2007. *Networks and Public Health: A Review of Network Epidemiology: A Handbook for Survey Design and Data Collection*. Oxford University Press, New York.