Improving the Compliance of Transplantation Medicine Patients with an Integrated Mobile System

Tim A. Majchrzak
Department of Information Systems
University of Münster
Email: tima@ercis.de

Antorweep Chakravorty
amerdis GmbH
Münster, Germany
Email: chakravorty.antorweep@gmail.com

Abstract

Progress in healthcare has brought cure for severe ailments and greatly improved the quality of life of many patients. At the same time, the complexity of treatments has increased: patients do not understand treatment plans or fail to timely take the intended dosage of drugs. The consequence is a lack of compliance. Patients not adhering to medication plans likely face severe health problems. To support the relationship with physicians and patients’ compliance, we developed the mobile platform appmedi. It helps patients to manage, which drugs they have to take, and to document their compliance to medication plans. appmedi has been designed for transplantation medicine because strict adherence is vital to save patients’ lives. We present the medical background and explain the design of our prototype. We also sketch a usage scenario and discuss appmedi. It could be extended to other treatment contexts and might be integrated into holistic patient-care solutions.

1. Introduction

Rapid progress in healthcare and medicine has brought cure for severe ailments and improved the quality of life of patients. Many illnesses can be treated in shorter time, with fewer complications, and with less pain. Patients can often leave hospitals quickly. This even applies to sophisticated forms of treatment such as transplantation medicine.

However, progress is accompanied by increased complexity both for medical staff and for patients. Even though much more information on illnesses and treatments is available, it is very hard for laypersons to fully understand medical conditions. Taking optimal care of patients requires them to be compliant with treatment goals. Compliance is critical with regard to medicine that has to be taken, to patients’ behavior, and to their compliance to treatment plans. Regarding the term, some authors prefer adherence [1]. They feel that compliance suggests a one-directional relationship of patients unquestioningly following what a physician tells them. We believe in a cooperative relationship—a “therapeutic alliance” [1]. For uniformity we use compliance and do not contribute to the ongoing discussion [2], [3].

Unfortunately, a low level of compliance is common, which has a variety of consequences [1]. Patients believing that they are well informed after superficially reading health-related articles might not comply anymore [4]. For many patients it obviously is difficult to comply with prescription plans [1]; they do not refuse to be compliant but are overstrained or overburden and uneasy with their treatment. In this case, a lack of compliance is not caused by reluctance but can be explained with inadvertence and overload.

The rate of chronically ill patients not adhering to their therapy is reported as 50% in developed countries [5]. Chronically ill patients generally require acute care. A lack of compliance does not only lower their quality of life and might have unwanted health effects but also is an economical problem. Money spent on patients’ treatment is wasted; even worse, addressing problems that relate to inadequate treatment requires additional resources. It can be concluded that particularly failed compliance with medication “is costly in terms of health, time and money” [6].

Transplantation medicine relies on a patients’ compliance, specifically with regard to taking the prescribed medicine. Due to progress in treatment a relatively normal life is possible for many patients. Nevertheless, they become patients for life and require to continuously take a number of drugs. If prescriptions are not followed with much care, negative effects ranging from minor issues with patients’ condition to very serious problems such as tissue rejection are expected. The success of the treatment is endangered and patients face health problems. Surprisingly, non-compliance (i.e. any deviation from the treatment plan) is reported for transplant recipients even though it could be expected that they are highly motivated to comply [7].

To investigate new ways of increasing compliance, we collaborated in developing a mobile system for patient support. appmedi was realized in a joint project between amerdis GmbH, a Germany-based service provider for the healthcare industry, and the Department of Information Systems of the University of Münster. It is vital to understand that patients need support rather than to be censured for their lack of compliance [5]. appmedi helps them to take the right
number of drugs at the right time. Physicians can check the compliance documentation to better support patients.

We chose a transplantation medicine scenario since compliance is extremely important for it. Typically, transplantation patients require multiple different drugs and their prescription plans tend to be very complex. Compliance decreases with the number of daily dosages to take [8]. Since a positive user experience leads to increased compliance [9], we developed a prototypic mobile application that is meant to be a handy tool for patients that is not felt to be invasive. Today’s widespread use of smartphones makes them an attractive platform. It has been found that technology support even helps people with little affinity to technology (e.g. elderly people) in following medication plans [10].

This paper is structured as follows. Section 2 introduces the healthcare background of our work. In Section 3 we give an overview of related approaches. The underlying methodology is explained in Section 4. Section 5 describes the development of our prototype. We then present an usage scenario and discuss our prototype’s capabilities in Section 6. In Section 7 we draw a conclusion.

2. Healthcare Background

Transplantation medicine is one of the most expensive and extensive form of treatment. Transplantation describes the act of moving a body organ from one individual to another or from a donor site to the patient’s own body. The main purpose of it is to replace damaged or absent organs. Organs that can be transplanted are heart, kidneys, liver, lungs, pancreas, intestine, and thymus including tissues such as bones membrane, tendons cornea, skin, heart valves, and veins. Various sources report different costs for transplantations and costs greatly vary by country. High five-digit EUR / USD sums are common for kidney transplants with other organ transplantations often running even higher [11], [12].

To better demonstrate the extent and the importance of transplantation medicine, statistics of the number of transplants in Eurowtransplant countries (Austria, Belgium, Croatia, Germany, Luxembourg, Netherlands, and Slovenia) are given in Table 1. An increase in transplantations can be observed for the last years. The development for various transplant types can be analyzed in order to understand the post-transplantation effects for such patients [13]. This helps to comprehend the necessity to establish a constant channel of communication with patients, have an feasible monitoring system, and a need for preventive care to be in place.

Table 1. Transplantation statistics (sources: [13], [16])

<table>
<thead>
<tr>
<th>Year</th>
<th>2005</th>
<th>2006</th>
<th>2007</th>
<th>2008</th>
<th>2009</th>
<th>2010</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deceased donors transplants</td>
<td>5750</td>
<td>5992</td>
<td>6384</td>
<td>6124</td>
<td>6322</td>
<td>6667</td>
</tr>
<tr>
<td>Living donors transplants</td>
<td>988</td>
<td>1017</td>
<td>1133</td>
<td>1172</td>
<td>1250</td>
<td>1400</td>
</tr>
<tr>
<td>Total transplants</td>
<td>6738</td>
<td>6969</td>
<td>7517</td>
<td>7296</td>
<td>7573</td>
<td>8087</td>
</tr>
</tbody>
</table>

Transplantation patients are immune-compromised. They have special healthcare needs even after returning home, e.g. regarding their living conditions. It is important for their physicians to prepare a proper treatment and communication plan. Besides, patients should receive appropriate monitoring from a short- and long-term [17] perspective. Monitoring does not mean surveillance but observation of the health status. A variety of diseases and symptoms such as infections, bleeding, interstitial pneumonia, decreased appetite, diarrhea, jaundice, and enlarged liver can plague a patient directly after transplantation of an organ [18]. However, there are other long-term conditions depending on the type of organ transplanted that might come into effect. Respiratory issues, musculoskeletal problems, kidney related problems, vascular issues, and psychological problems are examples of such—in some cases life threatening—disorders.

It is essential that patients pursue a preventive care mechanism and regularly screen for early symptoms. Many post-transplantation issues can be prevented or managed by modern medicine. Adequate treatment also ensures an almost normal life. However, patients are usually required to take a high number of drugs. For each drug, the right dosage has to be carefully determined and, if possible, the dosage-frequency optimized [19]. Additionally, many drugs are bound to be taken at a certain time or in correspondence to certain conditions, e.g. before or after a meal. This requires strict compliance by the patient [20] and can be wearing; overstrained patient might abort treatment while understanding the negative consequences. Therefore, ways to ensure compliance while disrupting the desired way of life of patients as little as possible is a major challenge of transplantation medicine (also cf. [21]).

3. Related Work

Related work can be identified for the field of medical compliance, for application of information technology (IT) in healthcare, and for improving compliance with the support of technology. Literature on compliance is vast; most articles
have been written by medical researchers. This research justifies our work but an elaborate discussion is not target-adequate. Similarly, the field of adopting technology to solve problems in healthcare is much too broad to discuss it in general. We identified approaches that can be related to our work by kind of application or by implementation steps taken. They can be checked for at least four criteria:

- the ability for patients to self-manage their treatment (in consultation with physicians),
- a focus on the needs of chronically ill patients,
- possible interaction between physician and patient, and
- notifications in case of non-compliance.

QUDAH et al. introduce compliance solutions that are not computer-based such as pill dispensers or dedicated electronic devices [22]. They show that such systems are inferior to sophisticated solutions for mobile devices. At the same time, these systems are established—applicable databases list a higher number of patents for such devices.

The MOMEDA system consists of two modules; one for the patient and one for the physician [23]. Electronic patient records can be accessed wirelessly. Similarly, TeleCardio Mobile provides information access and the possibility for remote consultation requests [24]. Both are examples for telemedicine systems; their main focus is the exchange of information. Some systems also have communication abilities. They might indirectly improve compliance but medication planning is not a focused feature.

Telemedicine systems can be extended with additional functionality. Approaches such as the Patient Safety Services (PSS) address patient care [25]. PSS provides notifications that can also address medication. However, such systems are specialized on patient monitoring [26] rather than on the patient-centered approach we chose. Therefore, these systems are either suitable (and important) for prevention or to look after critically ill patients in hospital environments [27]. They are not suited for chronically ill patients who on the one hand require intensive support but on the other hand want to keep an independent life style. Thus, support has to be non-pervasive yet engaging.

A laptop-based system is proposed by Finkelstein et al. [28]. Medication can remotely be suggested by a physician. However, even with using a laptop, the system has a stationary character. COCOSILA et al. [29] discuss using short message service (SMS) to improve patients compliance. This does not require any software to be installed on a patient’s device but is very limited in functionality. A commercial SMS notification service is patient reminders [30].

Two systems are directly comparable to our work. UbiMedS is an iPhone application for medication compliance that focuses on accessibility [31]. UbiMedS provides notifications and the possibility to track compliance. QUDAH et al. describe a mobile health monitoring system for cardiac patients [22]. Their system incorporates medication plans including a notification function. In contrast to our system, both solutions’ focus is less on self-management and there seems to be only limited support for physicians. appmedi provides support for physicians via a Web portal. However, the concepts proposed by UbiMedS to support disabled patients are convincing. QUDAH et al. address different reasons for non-compliance and discuss how their system can be used to counter them. Therefore, both solutions and our approach can be seen as novel systems that were designed with support of patients’ compliance in mind. Despite similar core functionality, each has different strengths.

Compliance can also be addressed in innovative ways. DE OLIVEIRA et al. propose using a mobile devices-based game that motivates compliance to medication plans [32]. It differs from our approach but still is an example we can learn from.

There is a high number of mobile health systems. An overview including a classification of systems is presented by VOSKARIDES et al. [33]. Further papers are not directly related. Hence, we do not present a survey-like listing.1

A related thread of research is the work on systems for ambient assisted living (AAL). Such systems are integrated into a user’s home and environment. They are particularly proposed to aid elderly people [34]. Despite their characteristics not all of them are called AAL solutions and a variety of names is used to describe them (examples are [35] and [36]). They can be used to observe health-status and provide further information; compliance support can be integrated [37]. Our system could be perfectly integrated into an AAL solution. It could also be used to extend mobile support systems for elderly people such as [38].

4. Methodology

Our work’s focus was not to understand why people are not compliant; we rather wanted to create an information system that supports them in adhering to their prescription plans and thereby increase their compliance. Therefore, we chose a design science [39] research method which reflects our strive to solve practical problems [40].

Activities in design science follow an iteration of build and evaluate [41]. This is particularly helpful if the desired target state of an information system is not known a priori. First of all, requirements for a novel mobile device to support compliance to prescription plans were compiled. After implementing an initial prototype based on the requirements, it could be evaluated and refined [42] in subsequent steps. This enabled a process of “learning via making” [43]. In other words: with each iteration we got a more profound understanding of the problem leading to improved prototypes that helped to further track down design issues.

Design science is particularly suited for designing healthcare-related systems. In most cases, no technical

---

1. A broad comparison of existing systems in the field of patient support could be a topic of future work.
breakthroughs are needed. Rather, existing technology has to be adopted, reconfigured, and recombined in creative ways to target yet unsolved problems. This helps to focus on the research’s objective—improving the quality of life of patients—while using modern technology as an enabler.

Requirements Engineering (RE) followed well-established practices [44]. In particular, healthcare industry consultants that work closely with their clients were interviewed. Designing and implementing the prototype followed common procedures from Software Engineering (SE) [45]. All phases had an incremental character. Progress was checked with experts in the field of healthcare system development. Moreover, we adopted best practices of Human-Computer Interaction (HCI) [46] in order to provide an adequate interface that can be used ergonomically. Besides, adequate visualization can engage patients. Since our paper’s focusses not on technical details, an elaborate discussion of the implementation process is out of scope.

5. Prototypic Implementation

The following sections describe the development of our prototype. Steps are presented in sequential order.

5.1. Requirements Analysis

Requirements engineering for appmedi comprised a series of interviews with consultants of amerdis GmbH, who are from the domains of pharmaceuticals and IT. Based on their ideas and our analysis the problem statement was defined as an *Android-based smartphone system (appmedi) that uses certain functionalities of amerdis’ Web-based patient information system (PIS) along with sharing data with it and having complete operational independence*. Two different areas were identified for investigation to gather and analyze the requirements for the proposed application. One set of requirements was identified for appmedi and another set for extensions to the existing PIS in order to make it capable of supporting appmedi.

appmedi should give patients access to their medical plan at any time and provide a scheduling platform to keep track of their medical progress, both in terms of recovery and development of health. Moreover, it should offer a communication channel to patients’ physicians; a good physician-patient relationship has been recognized to increase compliance [15]. Real-time reminders have been found to increase compliance [47]. Therefore, appmedi requires an advanced notification functionality. Functional requirements have been divided into six categories. Each category reflects a module, which enables a better understanding of the different aspects of the application. Each module can be described as a use case. Exemplary use cases in Unified Modeling Language (UML) notion [48] are given in Figures 1 and 2.

These requirements are associated with the modules:

- **User Profile.** The system should be accessed with a predefined user identification number and a password. Once the patient has logged in for the first time, credentials should be stored by appmedi. Patients should be able to update their credentials at any point of time.
- **Overall Medicine Plan.** Patients have to be able to view their overall medicine plan that gives them an overview of their weekly prescriptions. Along with it, they should be able to post a query for their physicians and view replies from them. The use case for this module is shown in Figure 1.

![Figure 1. Overall medicine plan use case](image)

- **Daily Medicine Plan.** Patients should be able to have an in-depth review of their medication plan for a particular day. The view should show the medicines they have to consume that day and the corresponding dosages. They also have to be able to update the status of their consumption history. They should furthermore be able to navigate to a previous day (history function) to verify whether they have complied to their plan. Navigation to future days should help them to gain knowledge of what medicine they have to consume.
- **Synchronization.** The system should be able to synchronize the medical plan (i.e. retrieve updates made by physicians) and other related information (e.g. send medicine intake and consumption history) with the PIS. The patient has to be able to trigger synchronization at any time. Comments to physicians and replies from them should also be exchanged during this process.
- **Notification System.** appmedi has to notify patients of any local events associated with the system such as synchronization reminders, prescriptions that have been sent to their physician, and profile updates. Even more important, it has to alert patients with an alarm whenever it is time for their medication. The use case for this module is shown in Figures 2.
- **E-Prescription.** Patients should be able to request a prescription from their physicians by using the system. This should be possible along with being able to view and select the medicines associated with their medicine plan to be included in the request.

appmedi utilizes the patient information system from amerdis GmbH to gain required information. However, in
order to support appmedi, the PIS should implement a flexible interface through which it is able to provide its build-in functionalities to appmedi while staying open for other applications that require data from it. Accordingly, it should enable appmedi to use its authentication. It should let appmedi gain access to the medical plan, dosage information, and physicians’ communication for the appropriate patient. The consumption history and any queries posted by the patient should also be reflected in the PIS. While these requirements lead to some changes to the backend PIS, we will not focus on them in the remainder of this paper since our research’s focus is compliance support with a mobile application. Since we use a standardized interface, the backend PIS can be easily replaced.

A number of non-functional requirements could be identified. They are implicit constrains on how the application should work, which have to be explicated for purposes of design and implementation. Non-functional requirements concern appmedi, the backend PIS, and their interoperability.

- **Usability.** The system should be designed for smartphones based on the Android 1.6 operation system with upwards compatibility without varying look-and-feel of the graphical user interface (GUI). The application should be intuitive and efficiently guide even technologically inexperienced patients through its core modules. Once patients log into the system, they should stay logged in until they manually log out—it is assumed that a smartphone is not shared among different persons. It has been found by BICKMORE et al. that the “degree of perceived politeness of interruptions is positively correlated with predicted long-term adherence” [47]. Thus, usability is essential for our approach and specifically notifications have to be designed with much care.

- **Reliability.** appmedi should have permanent access to the PIS and be able to synchronize any time. The client, the communication channel, and the backend have to be robust. Data retrieved has to stay available locally.

- **Performance.** appmedi should use minimal system resources. It should be runnable on smartphones with slow processors and little main memory. Future scalability has to be reflected by the interface.

- **Response Time.** The exchange of data between appmedi and the PIS should be limited. Only information that is required should be transmitted (incremental updates) to minimize response, download, and upload times.

- **Security.** An appmedi installation should be allowed to be configured for only one patient by specifying a valid identification number and password. Prior to any data exchange between appmedi and the PIS, the user has to be authenticated. The system should ensure that patients cannot view other patients’ medical data. Since no data that could identify the patient is needed, such data should not be exchanged with the client. Data that might be needed for medical reasons but could allow identification (e.g. body weight) should be encrypted.

- **Infrastructure Requirements.** Android-based smartphones with version 1.6 to 2.2 are to be supported. Compatibility with future version should be ensured as far as possible. For synchronization, Internet access is required. However, appmedi should not demand a minimum data rate or fixed quality of service.

- **Internationalization.** It should be possible to configure other language packages and date/time-formats without the need of programming.

- **Automation.** Medicine intake status updates should be locked once they were synchronized. Patients should not be able to deactivate the default medicine alarm setting. Any local notifications that are older than seven days should be automatically deleted.

Although the requirements are presented sequentially, they were gathered incrementally and modified throughout the development process. Building components brought insights about further needs or revealed misconceptions in requirements from earlier iterations. We discussed them with healthcare practitioners and incorporated their feedback. Nevertheless, we chose an sequential, waterfall model [49] like way of describing requirements and subsequentially design and implementation in this paper for comprehensibly.

### 5.2. Design

For synchronization appmedi needs to initiate communication with the PIS. The communication protocol that best suits our needs is the XML-based Simple Object Access Protocol (SOAP). It typically is used for Web services and provides easy exchange of structured data. It is particularly suited to enable communication by a variety of systems. Other clients can access the PIS using a well-specified Web service; at the same time, the PIS could be replaced by one offering a Web service with the same specification.

Our system has been designed as a modified three-tier application. Using the Web services, it would also be possible to provide it as a 4-tier Web application following the Model View Controller (MVC) design pattern or to provide a rich client application. A simplified architecture sketch is given in Figure 3. MVC is suited for Web applications, which currently do not have good offline capabilities—even
Figure 3. Architecture of appmedi and its backend PIS

though this will change with HTML5 [50]. A (rich) PC client might be a future addition.

The client tier is the smartphone-based appmedi. It contains the core business logic of our application as well as the associated views to provide the user interface. An embedded SQLite database [51] is storing all data related to patients’ health. The SQLite adapter and the KSOAP adapter establish channels for data transfer to SQLite and the PIS respectively. While frequent synchronization is desired, the client is capable of working without a permanent network connection. All components are designed to be light-weight.

The server tier includes a Web server, which hosts the PIS and provides its Web services. From appmedi’s perspective it is a black-box providing the required Web services.

The data tier is an database that supports the PIS with patient data, physician-related information, and lab data. It stores the original and current patient medical plan, dosage information, and intake plan of a patient. It also keeps record of intake status of medicines that patients consume in order to provide compliance-related support. Thereby, it allows gathering data for all desired analyses and should be the target of backups (rather than appmedi).

Since appmedi is specifically made for Android-based smartphones, its Java-based development framework was used. A modular design was chosen. The advantage of this approach is the loose coupling of objects with the possibility to modify implemented objects without affecting others. This is an important factor since an incremental approach was used for building appmedi. The modules that were derived from the module sketch made during requirements analysis are explained in the following.

The patient module captures essential information related to patients such as their unique identification number and their encrypted key. It also flags whether the patient accessed the system before to prepare it for first time usage through a synchronization process with the PIS. The doctor module captures contact information of patients’ physicians. Additionally, the synchronization module communicates with the PIS and synchronizes the current medicine plan, consumption history, and patient-physician messages.

The medicine module stores data and provides information about the medicines assigned to patients. It is accessed by various other modules. Information about the possible consumption pattern for a medicine is held by the dosage intervals module. It describes the intervals within which a medicine should be consumed. Various consumption patterns are available. The dosage plan is represented by a corresponding module that describes information about the prescribed starting date of a medicine, the times of the day it has to be consumed, and the dosage to be taken. Compliance related information of the patient is stored by the intake plan module. It captures information on whether patients consume their medicine by logging their consumption history.

There are several modules for plans. The abstract medicine plan module provides the complete medicine plan of a patient and acts as the core module to perform any actions related to updating plans. It is extended by the overall plan module, which provides information related to the overall medicine plan of a patient. The daily plan module provides a detailed view of patients’ daily medicine plan for a particular day. In combination with the intake plan module it enables patients to track their medicine consumption history.

Two modules are triggered by events or the current time. The notification module captures system events. It keeps log of synchronizations made and prescription requests sent. The alarm module is a background service that notifies patients when it is time for them to take their medications.

As an additional unit, the prescription module provides patients with list of medicines assigned to their plan and the responsible physician. It enables patients to send an email and request a prescription for the drugs they require.

While the source code of appmedi is currently closed, technologies used are open. Due to the modular design and the interface specification, replacing single parts or attaching another backend PIS would be possible. Besides specifying the architecture, three notable design decisions were made. Firstly, an embedded SQLite database has been chosen to store data on the mobile device. The SQLite library provides a helper class for typical tasks, which simplifies its usage [51]. Only minor adjustments to Java classes are needed; the approach is light-weight.

Secondly, communication had to be set up. The Android application programming interface (API) offers broad support but no SOAP functionality [52]. A prevalent SOAP library for embedded devices is KSOAP. We used KSOAP2 for communicating with the PIS. KSOAP2 allows SOAP objects to be created with various parameters and values. It also provides container objects as a vessel to carry SOAP objects as requests. Using its transport objects, a communication channel can be established with the PIS and calls of remote services performed.

Thirdly, we designed a synchronization algorithm that keeps data in appmedi consistent with the data in the PIS. The database content of appmedi is a subset of that in

2. Apparently, Google tries to reduce the computing effort on the client side, which might not be ideal from a programmer’s point of view [53].
Figure 4. Prototyping of the GUI

the PIS. During the first synchronization, data is merely copied. For consecutive synchronizations a two way data exchange is required. Physician occasionally update medical data while appmedi collects compliance data. Interchange must not lead to an inconsistent state and should require as little data exchange as possible. Synchronization consists of finding the corresponding record pair and updating the content. Having closed schema structures, unique identifiers can be used for mapping purposes. For every record in each relation in the PIS there is a created and modified timestamp along with a performed action type identifier that reflect operations. appmedi also keeps timestamps enabling it to determine which data it has to synchronize.

As a final notable activity, we performed GUI prototyping. An example is given in Figure 4. Instead of implementing the GUI based on the requirements, we first prototyped it using graphical mockups. We discussed them with medical experts and applied changes iteratively. Eventually, the actual GUI was implemented, which does not necessarily resemble the early prototypes (cf. Figure 5, page 8).

5.3. Implementation

The technology required for appmedi is well understood. Thus, we do not describe the implementation from a technical point of view but highlight some notable observations. We do not introduce Google Android; developer resources are widely available (e.g. [54], [55]).

Keeping in mind the targeted audience for the application, the interface has been kept simple with easy access to all functionalities through an integrated menu. The integrated menu acts as an one-stop navigational panel accessible to patients at any point of time by the click of a button from which they can traverse to any of the features such as daily medicine plan. It keeps the application intuitive and allows the patient to perform any action without the need to search for it. Along with giving a textual description for each feature, it also has a pictorial representation for each of them providing a better understanding for a glanced view.

Further related functionalities to a feature are provided to a patient through a single view. A good example to explain this is the daily medicine plan, where patients can view the medicines they have to consume on a particular day along with information regarding their dosage and a description of their medication. They can view and update their intake status by just touching the checkbox for the medicine they had to consume at the particular time of day.

With the synchronization feature patients trigger data exchange directly from the navigational menu. After confirmation, appmedi synchronizes with the PIS. It moreover posts a local notification comment about the synchronization result which is stored along notifications from local events. appmedi’s background service even runs when appmedi is not started. It posts a global notification along with an alarm when it is time for a patient’s medication. Patients can directly view the global notification of Android to see further information and to navigate to their daily medicine plans view to have a detailed view. They get there by touching the notification, which starts appmedi.

6. Evaluation and Assessment

To offer a better understanding of appmedi, we give a usage scenario including several features in the next section. We then discuss our findings and explain some limitations.

6.1. Usage Scenario

The following usage scenario shows how appmedi helps patients to comply with their medicine consumption pattern. The weekly medicine plan provides patients with an overview of all medicines they have to take in the current week. As shown in Figure 5 (left), each medicine on the medicine plan of the patient is color coded for every day of the week with green representing “is to be consumed” and red representing “not to be consumed”. The color code combination helps to bring into focus the medicines that are important to have in stock for the week.

In the daily medicine plan (Figure 5, center), patients check the medicines they have to consume on a particular day, the prescribed quantity, the status, and the time for consumption. It divides a day into morning, noon, afternoon, and evening. Patients furthermore log the status of consumption for a particular medicine for the appropriate time. This removes ambiguity even if a drug has to be taken more than once a day. It also keeps a record of the consumption history.

The request prescription view (not shown) provides patients with a list of drugs prescribed to them. Patients can
select medicines they require, write a comment, and send a
message to their physician. This simplifies timely restocking
of medical supplies and is another indicator for compliance.

Access to all features of appmedi is provided at a single
point of the navigation menu (Figure 5, right). It enables fast
and easy navigation; any feature is just one “touch” away,
with nested features requiring merely two or three actions.

6.2. Discussion and Future Work

Our prototype is a first step towards a mobile system to
support transplantation patients. After a successful evalua-
tion with patients, it could be extended for use in other fields
of medicine that are highly depended on medication—like
the treatment of most chronic illnesses. There is much room
for extensions; a number of patient-related issues have to
be checked and patients’ interaction with the system under-
stood. Additionally, appmedi is designated to be integrated
into holistic healthcare systems.

A remaining challenge of patient support application is
the motivation of patients (also cf. [56]) to use them. Such
systems are useless if they are either not accepted or not fre-
quently used by patients. Merely providing a health benefit is
not sufficient for many patients; they also forget to take their
drugs despite knowing the consequences. Therefore, systems
have to be non-intrusive, easy to use, and motivating. Ideally,
patients should perceive working with them as relaxing or
even fun. How to achieve this is a aim of future work.

An initial idea is to provide patients with a ranking
list. Patients could be issued points for documenting their
drug consume and for synchronizing appmedi. Points are
accumulated and patients can view an (anonymized) ranking
in comparison with other patients with similar treatment
plans. Rankings are adjusted after a while (i.e. “older” points
expire). As reported in Section 3, even game-like ideas
could be incorporated into appmedi—providing a pleasant
experience or even fun might increase compliance.

Additionally, compliance could be bound to financial
incentives. If health insurances would grant a (partly) reim-
bursement of premiums for compliant patients, this would
strongly motivate many of them. This of course has to be
subordinate to medical aims. Our maxim is not a situation in
which patients blindly follow prescriptions but one in which
they actively discuss their treatment with physicians.

Besides motivation, it has to be taken into account that a
system like appmedi cannot certainly know whether a drug
has been taken, or not. This would require blood titer.
Means of automatically tracking an assured compliance
might be developed for future versions of it. This could
include hardware for additional diagnosis.

Automatic assessment of compliance data should be im-
plemented in future versions. This can e.g. by done with an
pattern matching algorithm or a self-learning system. Ana-
lyzing the patterns of compliance could lead to conclusions
about patients’ risks. For example, the probability of organ
rejection could be calculated based on statistics of missed
drug consumption or wrong dosages. Moreover, the pattern
of compliance might reveal a decreasing motivation to take
a drug. If the patient agrees to such analysis, physicians can
proactively intervene and thereby prevent treatment failure.
This would be especially helpful for patients that get weary
of their treatment or have doubts about a medicine’s use-
fulness but refrain from seeing their physician. For patients
that have a high risk of organ rejection, additional data could
be requested that has to be supplied by the patients. For
example, getting daily values of creatinine and body weight
would be indicators for possible upcoming health problems.

The vision is to embed appmedi into an overall medicine
information system that is accepted by patients and physi-
cians. It could automatically synchronize with a PIS and
enable assessment of the patients’ health by physicians in accordance with the treatments plans. At the same time, it could be linked with an ambient assisted living system that supports patients in their daily routine and monitors health related information. Holistic health systems promise to dramatically increase patients’ quality of life.

6.3. Limitations

appmedi is a prototype. Thus, the research behind it and itself are subject to limitations. Further refinements will have to be applied after evaluating its usability with the help of patients. Our work is patient-centered and therefore we consider patients’ opinions to be of high importance—after all, the improvement of compliance can be expected to be bound to the satisfaction with appmedi. No patients were involved for building the first prototype. This is not a flaw; however, to justify our patient-centered idea we need to evaluate appmedi with them. We also need to include the opinion of physicians. Resistance of physicians can be a major barrier for adoption of systems in healthcare [57]. Currently, physicians were involved only indirectly.

Furthermore, our prototype does not yet provide all desired functions. Despite being modular and having interfaces that allow future extensions, the producibility of all intended future features is not yet clear. Finally, we have not yet conducted a field test of appmedi. It would be valuable to qualitatively verify its usability [58]. This would give us strong evidence whether we reached our usability targets, or not. Besides, a survey could include a qualitative assessment to learn about patients’ ideas of improvement.

7. Conclusion

We presented appmedi, our prototypic implementation of an application for mobile, Android-based devices. It aims at increasing patients’ compliance with medicine plans by reminding them to take the right dosage of drugs at the correct time and by supporting them in documenting their compliance. appmedi emphasizes patient-physician communication: our prototype not only synchronizes with a patient information system to request new prescriptions and submit compliance data but supports patients in staying in contact with their physicians. Thereby, patients are motivated and non-compliance due to information overload is prevented.

appmedi targets transplantation patients since compliance is essential for a successful treatment. Despite being limited in functionality, there are many prospects for future developments. Thus, we named several objectives of our future work. A medium term goal is the integration of our system into an overall approach. Ideally, the full “Prescription – Dispensation – Administration – Compliance (PDAC) medication chain” [59] would not only be covered but supported by the system.

Acknowledgments

We thank amedis GmbH—in particular Mr. Matthias Barndt—for feedback on the manuscript. Major improvements to the paper were made based on the comments of the anonymous reviewers. Their feedback is highly appreciated.

References


