MEDADHERENCE : COGNITIVE BEHAVIORAL THERAPY VIA MOBILE WEB FOR IMPROVED MEDICATION ADHERENCE

by

VARUNA IYENGAR

(Under the Direction of Eileen .T. Kraemer)

ABSTRACT

After an organ transplant, adherence to a prescribed medication regimen, is vital after-care to avoid organ rejection and ultimately, death. For adolescents, barriers to medication adherence are often tied to social, emotional, and behavioral issues rather than simple forgetfulness. Studies have identified Cognitive Behavioral Therapy (CBT) as an effective approach for improving medication adherence in adolescents. However, following up with patients after every dose and helping them overcome their barriers to medication adherence is not a feasible solution. *MedAdherence* is a Smartphone-based CBT system that incorporates incentives and is designed to improve medication adherence in adolescent kidney transplant patients. The system sends reminders, records adherence data, and provides support for CBT at the moment of non-adherence. In this work, we also evaluate the feasibility of the LogicBlox platform, a Datalog-based, Cloud-delivered platform-as-a-service, for mobile web applications such as MedAdherence.

INDEX WORDS: mobile web, logic databases, medication adherence, Smartphone intervention, ubiquitous computing, computerized cognitive behavioral therapy, CCBT.

MEDADHERENCE : COGNITIVE BEHAVIORAL THERAPY VIA MOBILE WEB FOR IMPROVED MEDICATION ADHERENCE

by

VARUNA IYENGAR

Bachelor of Engineering, Datta Meghe College of Engineering, India, 2009

A Thesis Submitted to the Graduate Faculty of The University of Georgia in Partial Fulfillment

of the Requirements for the Degree

MASTER OF SCIENCE

ATHENS, GEORGIA

2012

MEDADHERENCE : COGNITVE BEHAVIORAL THERAPY VIA MOBILE WEB FOR IMPROVED MEDICATION ADHERENCE

by

VARUNA IYENGAR

Major Professor: Eileen. T. Kraemer

Committee: Maria Hybinette
John. A. Miller

Electronic Version Approved:

Maureen Grasso
Dean of the Graduate School
The University of Georgia
December 2012

DEDICATION

To my family.

ACKNOWLEDGEMENTS

I would like to take this opportunity to convey my heartfelt gratitude to my advisor, Dr. Eileen Kraemer for her valuable guidance and continuous support throughout my Masters program. I also thank Dr. Kurt Stirewalt and others at LogicBlox for their guidance and assistance. I would like to thank my committee members, Dr. Maria Hybinette and Dr. John Miller for their time. Last but not the least I would like to thank my family and friends for being supportive.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Page

CHAPTER 1

INTRODUCTION

Transplantation is an optimal treatment for many children with serious kidney conditions. A kidney transplant brings many changes in the lifestyle of the recipient. As with any organ transplant procedure, after care is of utmost importance to prevent organ rejection, infection or ultimately, death [1-4]. Adherence to prescribed medication is one of the many after care requirements. Statistics show that adolescents have higher kidney allograft loss at 3 and 5 years post-transplant than any other age group except those ages 65 and older, due largely to rates of non-adherence as high as almost 75% [5-7]. Low adherence to medication is also largely responsible for high medical expenses [8]. Advocating medication adherence in adolescents is particularly difficult because it is challenging for adolescents to comprehend the importance of taking regular medications in order to maintain good health. Forgetfulness and difficulty in swallowing pills are considered common barriers to medication adherence. However in adolescents, barriers to medication adherence also range from social to emotional and behavioral issues [9]. Research has identified Cognitive behavioral therapy (CBT), a technique that helps people better understand their thoughts and feelings that lead to certain undesirable behaviors, to be effective in the treatment of several illnesses and disorders in adolescents[10].

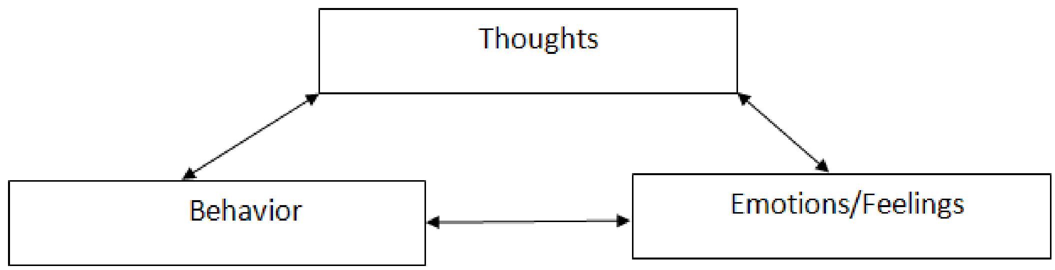

Figure 1.1: Cognitive Behavioral Therapy Triangle

Fig. 1.1 depicts the CBT model popularly known as the *Cognitive Behavioral Therapy Triangle*. It expresses how our thoughts, emotions and behaviors are interdependent. Therefore, a shift in any one of these aspects , affects the others. Thus, channeling one's thoughts and emotions in a direction that helps positively influence ones behavior is the goal of CBT. Studies [11] [12]and [13] have highlighted CBT as a successful approach to encourage medication adherence. A study by our collaborators has also identified barrier scales to medication adherence [9] in adolescent kidney transplant patients and also provided probable solutions to address specific barrier subscales. Ideally, the transplant team would interact with the patients on a daily basis and help them overcome the barriers that make patients non-adherent to their medications. However, it is not feasible for the team to check on each individual patient for every dose and check for adherence. Also, it may be too intrusive to the schedules of the adolescent recipients to talk to the transplant team daily and get help with overcoming their barriers to taking all of their medication. Therefore practical tools are required to empower patients to self - manage their health and for physicians and transplant coordinators to keep track of the patient's adherence behavior. The fundamental challenge is to lower the barriers and increase the motivation for patients to adopt a self-management approach for long term care**.** Smartphones, with their increasing levels of ownership, popularity with the intended audience, and sophisticated features, are an excellent fit for our purposes. Apart from the fact that a cell phone is a personal item and considered a valuable device by the people who own them. Smartphones provide mobile web access and hence facilitate mobile presence. Furthermore, cell phones are lightweight devices and can typically be carried around in one's pocket.

1.1 Contributions:

Our contribution to this work is twofold. In this work, we leverage the power of the mobile web and Smartphone technology to deliver a Smartphone-based Cognitive Behavioral Therapy system to improve medication adherence in adolescent kidney transplant patients. The system captures data about barriers and solutions at the time of non-adherence and maintains this data for future analysis to detect patterns of non-adherence. It also has a reward structure in place, in which the patients are awarded points based on their adherence and reporting. These points are then translated into money transferred to a debit card.

In the context of the development of such a vehicle for the Smartphone-based delivery of Cognitive Behavioral Therapy to pediatric transplant patients, with the goal of improving medication adherence and thus improving medical outcomes, we explore and evaluate the use of LogicBlox, a Datalog-based, declarative system for the creation and deployment of Cloud-based applications, for a mobile web application such as MedAdherence. This application represents the first use of the LogicBlox platform for mobile web.

The rest of the thesis is organized as follows. Chapter 2 discusses the related work and talks about why mobile technology is an optimal approach. Chapter 3 describes the implementation details, system architecture and the working of the overall system. Chapter 4 discusses our evaluations and results. We conclude and discuss future work in chapter 5.

CHAPTER 2

RELATED WORK

Traditionally, access to the web has been through desktop and laptop computers. The advent of Smartphone devices has changed this. Not only do nearly half of the Americans own a Smartphone today [14], but according to research by Pew Internet in 2011, 25% of Smartphone users in the United States , mostly access the web through their phones rather than their computers and a whopping 87% check their emails and access the Internet through their Smartphones [15]. As the number of owners of Smartphone and tablet devices continues to increase, the shift towards mobile web will accelerate. Consequently, many web applications and web sites have been or must be adapted to suit mobile devices.

2.1 Mobile Application Development:

Mobile applications come in two flavors, mobile web apps and native apps. Due to differences in approach and in the underlying technology, each has it own advantages and disadvantages. A whitepaper published by Lionbridge [16], a provider of web translation, development and testing solutions, elaborated on the factors influencing the choice between Mobile Web apps and Native apps.

2.1.1 Native applications

Native apps are mobile applications developed for a particular mobile platform such as iPhone, Android, or Windows. Unlike mobile web apps, these applications are downloaded once from an app store or similar repository and reside in the phone memory. As a result, once downloaded, they generally do not require an Internet connection, unless data transfer is one of the functions provided by the app. Other advantages of native apps are:

- Native apps provide a richer user experience as they can access the camera, accelerometer, GPS and other phone hardware.
- Native apps can be placed on the home screen for easy access. Users only need to tap the app icon and the application launches instantly. In contrast, launching a mobile web app requires typing in the URL or accessing a bookmark and may take anywhere from a few seconds to a few minutes to load, depending on the Internet speed and content.

Native apps do have certain disadvantages as well:

- Since native apps are platform dependent, their development requires expertise in that platform. For an application to work on a variety of platforms, separate versions of the application need to be developed for each platform. This incurs additional cost and a longer development time.
- Maintenance of the application also becomes a tedious task, as it requires expertise in building and testing on a number of platforms, and keeping up-to-date with changes in the those multiple platforms..
- As all the data associated with or collected by a native app resides on the phone, it poses a security threat and results in information loss if the phone is stolen or damaged.

Xcode [17] is the official toolset for iOS development. The Xcode kit includes an iOS simulator, the iOS Software development Kit(SDK) and debugging tools for developing applications for the iPhone, iPad and iPod. Objective C [18] is the primary language used for native mobile application development on the iOS platform.

Native Android applications are typically developed in Java using the Android SDK [19]. The Android SDK also includes a rich set of development tools such as a debugger, an Android emulator, libraries and documentation. Similarly, developing for the BlackBerry® [20] or Windows® [21] platform requires expertise in their respective development languages and SDKs or APIs.

Writing separate versions of native applications for different platforms is time consuming and is not cost effective. Furthermore, developers have to update the application to suit any system upgrades. Therefore, this gave rise to cross platform mobile development tools, which enable developers to write native applications for popular platforms with some modifications required to run on a variety of platforms.

The RhoMobile [22] suite composed of RhoConnect , RhoStudio and RhoElements components offers an open source, Ruby-based framework that allows for development of native apps for a wide range of Smartphone devices and operating systems. It supports the iPhone, Android, Symbian, BlackBerry® and Windows Mobile platforms. The "write once and use everywhere" approach makes development easier across different platforms. The RhoMobile framework also provides access to the phone's hardware devices like the camera, accelerometer and GPS. MoSync [23] is yet another native mobile application development framework, based on common programming standards. The open source MoSync SDK enables developers to create mobile applications for varied mobile platforms in C/C++ or HTML/JavaScript. PhoneGap [24] is an example of an hybrid mobile application development tool. It is a free and open source framework that helps create mobile application by means of platform specific web APIs and provide access to phone hardware such as the camera, GPS and accelerometer.

2.1.2 Mobile web applications

A mobile web application is an application designed to suit Smartphones and tablets and is accessed via the mobile web browser of these hand-held devices. Mobile web apps are primarily powered by HTML, CSS and JavaScript technologies. The HTML components typically define the static text, images and the structure of the page , the CSS defines the style and presentation [25], and the JavaScript defines interactions and animations [26]. Mobile web apps are typically downloaded from a central web server each time they are run. One of the main advantages of the Mobile web over native applications is that mobile web applications are platform- and device-independent. Other advantages include:

- Owing to cross-platform compatibility, development and maintenance costs are significantly lower than native apps.
- Users need only access the application via a web browser and need not download or install applications on the phone or tablet device.
- The data is stored at a remote web server and not on the user's phone; thus, data security is guaranteed in the event of theft or physical damage to the phone.

Some disadvantages of mobile web applications are:

- As the performance and user experience of mobile web apps depend on the quality of the Internet connection, performance may vary considerably depending on whether the user is on a Wi-Fi network or on their provider's 3G or 4G network.

- Applications that collect substantial amounts of information have the potential to incur additional expense to the owner of the phone.
- Unlike native apps, mobile web apps do not have access to the phone hardware such as the camera and accelerometer and hence applications that require camera control or the control of any other phone hardware cannot currently be built as a mobile web application. However, this is changing due to tools like PhoneGap.
- Native apps provide a richer user experience with aesthetic UI controls that mobile web apps are, as of now, lacking.

JavaScript frameworks such as Sencha Touch, and jQuery Mobile are popularly used to create mobile web applications with a native-app like feel. For the UI development of MedAdherence, UIBlox , a declarative UI framework developed and supported by LogicBlox was used. Each of these frameworks are discussed below:

Sencha Touch Framework:

Sencha Touch [27] is a mobile UI JavaScript framework, specifically built for the mobile web. It is completely built on web standards like HTML, CSS and JavaScript. It generates a DOM from the objects created in the JavaScript layout. Over the last two years, Sencha Touch has grown to support several mobile platforms and Webkit - enabled browsers such as Safari, Bada Mobile Browser and Google Chrome for Android. Sencha Touch provides rich UI tools for a native app like experience. The latest version of Sencha ships with several different icons, tab layouts, map elements and the popular Carousel Layout, which enables swipe-navigation between screens. However, since Sencha Touch apps do not support non Webkit - based browsers, installing Webkit - based browsers to access the Sencha Touch based web application becomes mandatory.

jQuery Mobile Framework:

jQuery Mobile [28] is a touch optimized web framework built on top of the popular jQuery [29] framework , a multi-browser JavaScript library used to create AJAX applications, handle events and create animations. Unlike Sencha Touch, it supports a wide range of mobile browsers and mobile platforms. jQuery Mobile transforms HTML code into touch friendly elements, thus simplifying the development process.

UIBlox Framework:

Datalog^{LB}, the version of Datalog developed by LogicBlox, supports its own UI framework for web application development. Therefore, the UI for MedAdherence was developed on the UIBlox framework. UIBlox is a declarative UI framework developed by LogicBlox [30]. Formatting of UIBlox elements is done through CSS. In order to be adaptable to the mobile platform, mobile CSS techniques such as viewport adjustments and mobile web browser compatible forms were used. UIBlox is based on the W3C Xforms [31] standard and hence follows the Model-View-Controller approach for web application development. UIBlox compiles native YAML forms [32], a hierarchical data serialization language into HTML and dynamically renders them to the browser using JavaScript at runtime. UIBlox includes a pre-defined entity type called form whose instances in the workspace or database are in one-one correspondence with the form instances on the web browser. UIBlox offers a wide range of UI tools such as Text fields, Label fields, Drop Down menus, Multi-tiered Select menus, Checkboxes, Grid layouts and Date fields. In UIBlox, user initiated events, such as button - click events are handled by pulse predicates. When the user activates a submit button for instance, it triggers a pulse predicate and asserts the form data entered by the user into the workspace. Since UIBlox is primarily a web application framework, it is compatible with almost every browser. The only disadvantage of the version of UIBlox used in the development of MedAdherence is that since it is a server-side framework, form interaction and data flow speed is slow and any changes to the forms require the entire application to be rebuilt.

Other mobile web application development tools such as WidgetBox [33] , foneFrame [34] and Tiggzi [35] that utilize HTML, JavaScript and CSS technologies are also available. These tools are compatible with most modern web browsers. WidgetBox and Tiggzi require a monthly fee, whereas foneFrame is a free tool.

HTML5, still in development as of the writing of this thesis, is a candidate for use in the development of mobile web applications [36]. This latest version of HTML will provide developers with tools for offline web storage, a geolocation API, canvas drawing, and CSS3. As a result, developers may be able to achieve a look and feel that is closer to that of native apps, and run offline without an Internet connection. Both iOS and Android support HTML5, as does RIM, however different browsers interpret HTML5 differently and this lack of consistency complicates its use when targeting a market containing a wide variety of Smartphones.

Although client-side UI frameworks are typically used in web applications, a few server side frameworks also exist. Apart from UIBlox, which is based on XForms, other frameworks such as Apache MyFaces [37] and JBoss RichFaces [38] are AJAX-enabled frameworks based on JavaServer Faces. Server side validation is considered to be more stable than its client side counterpart, as components are validated at compile time. Server side frameworks lead to high bandwidth usage, as every data from interaction and render request is sent over the network for server response. Client side frameworks use less bandwidth and hence are faster and less expensive on the network. This is one of the primary reasons why client side UI frameworks like JavaScript are commonly used over server side frameworks.

Mobile web design considerations:

Although mobile web applications are in fairly common use today, they still have a long way to go before they provide a user experience that is comparable to traditional desktop web applications. Several factors contribute to this condition. First of all, due to the reduced screen size and low resolution of mobile phones, the web app interface should not be cluttered with UI elements. As a result, all the information that would generally be displayed on a single web page needs to be split into different screens with simple and minimal navigation UI controls. Although mobile web apps adapt to the device dimensions, they may require some fine-tuning in order to work well on each device. Furthermore, mobile web browsers have less memory than desktop browsers. As a result, caching capability is much lower in mobile web browsers than desktop browsers.

2.2 Medication Adherence:

A number of studies have been dedicated to understanding the causes of medication non-adherence in general. With the elderly or adult population, aging, inadequate health literacy and declining cognitive and physical functions have been identified as major causes of non-adherence [39, 40]. Meanwhile in the case of adolescents, studies show that no single cause of non-adherence exists. Apart from common reasons such as forgetfulness and difficulty in swallowing pills, several social and behavioral factors contribute towards non-adherent behavior in adolescents. In a study conducted to assess medication adherence in pediatric asthma patients [41] it was found that although older children have a better understanding of their heath condition and assume more responsibility of disease management than younger children, their adherence was lower than that of younger children. A high percentage of adolescents are found to skip doses for fear that their friends or peers may discover their health status [42]. In adolescents, peer acceptance and approval is an important factor. Anything that is perceived to be "different" from their peer group leads to social stigma. This is especially true if the treatment regimens bring about certain undesirable cosmetic changes. Due to the wide range of barriers to medication adherence that are unique to each adolescent patient, a one-size-fits-all approach is inadequate to tackle this issue. A more personal, tailored approach is required to address individual barriers to medication adherence. In a study performed to indentify barriers in adolescent transplant patients [9], a promising set of *Parent Medication Barrier Scales (PMBS)* and *Adolescent Medication Barrier Scales (AMBS)* were developed. These scales help in estimating and understanding probable categories of barriers to medication adherence for adolescent kidney transplant patients. This approach precisely identifies different barrier classes and supports the identification of appropriate solution strategies that can be recommended to improve medication adherence.

2.3 Mobile applications in Health Care:

Internet based disease self- management tools and applications have been proved to be effective against chronic diseases [43]. Other Internet based applications like Fearfighter [44] , which attempts to aid panic and phobia treatment and MoodCalmer [45] for depression and low mood treatment have also found considerable success. Mobile phones have also increasingly been used as a support tool for delivering health interventions. A considerable body of literature shows that mobile health interventions have positive outcomes. The Short Messaging Service (SMS) offered by mobile phones has been proven as an effective tool in intervention studies for adolescent diabetes patients and as a support tool for smoking cessation services in young adults [46],[47]. In general, the use of computerized Cognitive Behavioral Therapy (CBT) in adolescent depression and anxiety disorder cases has shown encouraging results [48],[49], [50]. The text messaging capability of mobile phones has been known to be used in behavior modification interventions and has found considerable success [51]. [52] proposed a mobile phone based system to promote health self-management for patients suffering with chronic illnesses. This hypothetical system proposed to apply social-behavioral theories to engineer the system to positively influence patients' compliance behaviors, by means of mobile–triggered questionnaires and contextual reminders. It also discussed the research challenges in realizing such a system. Thus we can say that mobile phones have proven to a solid support tool for delivery of health self management practices. The rapidly growing Smartphone market brings more avenues for mobile health care technology. A survey conducted by Statista [53] in 2012 shows that the number of Smartphone users in the United States alone has doubled in the last two years and is estimated to reach a whopping 192.40 million users by 2016. Currently, a number of software applications are available for the Smartphone market that focus on medication adherence for a diverse audience that includes the elderly, adolescents and middle aged population.

MyMedSchedule® [54] is a free application that one can access from a desktop and it also has an app version that can be installed on an Android Smartphone or an iPhone. In this application, patients have the flexibility to set up their medication schedules themselves or have their health caregiver set it up for them. The patient or provider can then print their schedules and keep it for their reference. The application is designed to send a text message to the patient's phone when it is time for the patient to take medication. While this system does remind the patients to take their medications on time, it does not record adherence or non-adherence information by following up with the patients to see if the medications were in fact taken, nor provide any intervention if they do not take the medication.

The OntimeRx® [55] application is another system that is available on the Android, BlackBerry and iPhone platforms. It also has a version that is compatible with the Windows operating system. Additionally, users can choose from email, phone or text reminders. Keeping our focus only on the mobile application, we can draw certain similarities between OntimeRx® and MyMedSchedule®, in that both enable users to set up their own med schedules and edit them as per changes in their prescriptions. While MyMedSchedule® sends a text or a email reminder to the patients at the scheduled time, OntimeRx® sounds an alarm to indicate that a dosage is due. OntimeRx® takes further steps to record if in fact the meds were taken, by soliciting a response from the patient. Additionally, if the patient does not respond within 15 minutes of the dosage due time, it sounds another alarm and continues to follow this pattern until an hour after the scheduled dosage time. Furthermore, it calculates and maintains compliance/adherence information in a log that enables doctors and the patient themselves to track medication adherence rate. A number of such applications are available for the mobile phone market that attempt to aid medication adherence through reminders and logging and/or journaling capability. While such applications seem to have received substantial popularity, they do not tackle the primary causes of non-adherence. From other studies on medication adherence [9, 11] we learn that reasons for non-adherence are not only forgetfulness, but may range from lack of health awareness and behavioral issues to finding it hard to keep up with complex medication regimens. Consequently, any system that proposes to tackle non-adherence should make use of this fundamental knowledge. Secondly, it should be fairly easy to use or learn, so that the users do not find the system to be overwhelming to use. Lastly, the system needs to have an impact on the target group and should solve the underlying problem of non-adherence to a reasonably good extent.

CHAPTER 3

APPROACH

MedAdherence was developed in Datalog^{LB}, a strongly-typed extension of Datalog from LogicBlox. Datalog is a declarative logic programming language that is a subset of Prolog [56] and is also a query language for deductive databases. In the last few years, Datalog has attracted significant research interest in both the logic programming and database research communities because it preserves SQL's property of guaranteed termination and it allows recursive queries. Datalog has two types of constructs: rules and queries. The general syntax of a rule is given by

head ← body

Following are examples [57] of a rule in Datalog:

```
ancestor(x,y) ← parent (x,y).
ancestor(X,Y) ← parent(X,Z),ancestor(Z,Y).
```

In the above examples, the first line can be read as X is the ancestor of Y if it is known that X is the parent of Y. The second rule can be read as X is the ancestor of Y, if X is the parent of Z and Z is an ancestor of Y. Relations appearing in the head are called derived relations and those appearing only in the body are called base relations. Datalog has been found to be useful in semantic web applications such as Ontological modeling and reasoning [58].

Many Datalog extensions and dialects exist, and Datalog^{LB} is one such variant tailored for enterprise software systems.

Datalog^{LB} was chosen amongst other available platforms for the following reasons:

- **Declarative specification and configuration** -- The declarative nature of the Datalog^{LB} approach supports the specification of the core functionality with a base collection of rules and constraints, and the configuration of the individual application through additional, application-specific rules and constraints.
- **Constraint optimization and maintenance** -- Scheduling of initial reminders and dynamic scheduling and rescheduling of follow-up interactions is an essential element of the core functionality of this family of applications. Datalog^{LB} provides good support for the specification of these scheduling requirements as constraints. Further development of MedAdherence would impose additional constraints on the system such as changes to a patient's prescription, changes to a patient's contact information or methods of medication ingestion, which would have to be immediately reflected by the application. Constraint optimization and maintenance, a strongpoint of the Datalog^{LB} platform, naturally supports the automated generation and incremental maintenance of the schedules of alerts and other interactions required by this application.
- **Assurance** -- Health and safety-critical systems will likely require the approval of regulatory agencies when in full production. That the code is logic-based makes it amenable to verification, which would support the ability to obtain approvals that other platforms may lack.

3.1 Datalog^{LB} Architecture:

The system architecture is comprised of several logic files and corresponding user interface screens. The user interface has been developed using the UIBlox framework from LogicBlox®. This server-side framework provides tools for developing interactive user interfaces. UIBlox implements the Dojo framework for cross platform JavaScript-based application development.

3.1.1 Workspaces

Datalog^{LB} programs manage *workspaces* also called databases, which contain both the program and data. Workspaces hold collections of facts, each which is concerned with a predicate. In logic terms, predicates are properties that may be held by individual things or relationships that may apply to multiple things. In a workspace, a collection of facts associated with a predicate is called that predicate's population. In terms of relational database systems, a stored predicate corresponds to a table and the facts of a predicate population correspond to table rows. In addition to built-in primitive data types, Datalog^{LB} allows the programmer to define their own entity types. The following code snippet shows an example of a predicate declaration:

```
sdcs:barrier(b) -> .
sdcs:barrier(b), sdcs:barrier:name(b:n) -> string(n).
```

Here, the first line defines `sdcs:barrier(b)` as an entity. The second line defines `sdcs:barrier:name` to be another predicate where the domain of the barrier name is a barrier and the range is of type string. The period symbol "." Indicates the end of the predicate declaration.

3.2.2 Logic Files

The core logic of the system is implemented in the form of logic files . These logic files are written to manipulate the predicates in the workspace. A logic program is a set of clauses such that each makes a claim about the facts in a predicate. A clause comprises a head and/or a body, each consisting of atoms. An atom is a predicate name with a list of arguments, each of which correspond to one of the predicate's roles. Additionally, a delta modifier may precede the atom in the head, to control changes to the predicate content in the workspace. A delta rule has a non empty body and a head containing only delta modified atoms. A program's delta logic comprises its fact assertions/ retractions and its delta rules. Clauses are typically of three types: fact assertions/retractions, constraints and derivation rules. A fact assertion is used to add facts to a predicate's population; a fact retraction is used to delete a fact from the predicate's population. In Datalog, ordering of clauses does not matter.The following code snippets show an example of fact assertion and fact retraction:

```
…
+sdcs:medication(med1),
     +sdcs:medication:brandName[med1] = "Ciprofloxacin",
     +sdcs:medication:genericName[med1] = "Ciprofloxacin‟,
     +sdcs:medication:imageString[med1]="‟http://cs.uga.edu/~iyen
gar/Smartimages/no_image.png”.
…
```

Here, we are asserting certain facts about a medication into the workspace viz., the medication's brand name, its generic name and an image associated with that medication. The following code snippet shows a fact retraction example:

…

```
-medSet(s,ms)<-
+viewAccountResponse(f),
medSet@prev(s,ms),
forms:FormInstance:session[f]=s.
```

...

Here we are removing a fact about a set of medications, `ms` from a previous session `s`. `medSet(s,ms)`, which is a combination of a predicate name and a list of arguments, is called an atom.

Constraints are used to declare predicates or limit the facts that can populate predicates. The following code snippet shows an example of a predicate declaration constraint.

...

```
sdcs:person:firstName[a]=b -> sdcs:person(a), string(b).
```

...

In the above code snippet, we declare a predicate, `sdcs:person:firstName[a]=b` such that, `a` is a type of `sdcs:person` and his first name, `b` is a type of `string`.

This next snippet depicts a constraint on the entity `sdcs:person` such that any fact populating the predicate `sdcs:person` should have a first name. The underscore "_" character is a "not null" character in Datalog[LB]. In the context of this example it means that for a person fact to be asserted in the workspace, that person should have a first name and it should not be null.

Derivation Rules are used to programmatically infer new facts from known facts in the workspace. The following code snippet shows an example of a derivation rules.

...

```
onTimeWindow(pt,ms,dt) <-
     medSet(s,ms),
     medSetTime(s,dt),
     userReply[s]=r,
     sdcs:replyDateTime[r]=dtNow,
     sdcs:replyPatient[r]=pt,
      datetime:offset[dt, dtNow, "minutes"] = offset,
      offset>-59,
     offset < 60.
```

…

In the above code snippet, we derive facts for the predicate `onTimeWindow`. `onTimeWwindow` is a predicate which decides if a patient's response is within an 2- hour time window of a scheduled dosage, i.e., an hour before the scheduled dosage and until an hour after the scheduled dosage. The facts for `onTimeWindow` are derived from rules consisting of atoms such as `medSet(s,ms)`, `medSetTime(s,dt)`, `userReply[s]` , `sdcs:replyDateTime[r]` and `sdcs:replyPatient[r]`. In Datalog, the "," represents a conjunction (AND) and a ";" represents a disjunction(OR). The `medSet` and `medSetTime` predicates hold facts about medications due at a certain time of the day. The `sdcs:replyPatient and sdcs:replyDateTime` predicates record the user id of a patient and the date and time of her visit. If the difference between the scheduled time of the dosage and the time of the patient's response is within the on time window, the `onTimeWindow` predicate is populated with the corresponding fact about the patient (`pt`), medication set due (`ms`) and the time of the due dosage (`dt`).

3.1.3 Workspace Management

A Datalog^{LB} program manipulates two kinds of predicates: extensional database (EDB) predicates and intentional database (IDB) predicates. EDB predicates are normally used to hold facts explicitly entered into the workspace. In contrast, IDB predicates are computed by the logic program with its IDB rules. Datalog^{LB} also supports conditional modification to predicates in the workspace. These are known as Delta modifiers.

3.2 MedAdherence System Description:

The MedAdherence system has a database of patient information. It stores sensitive data such as patient contact details, prescribed medication dosage, and schedule times of dosage for each patient. To ensure security of sensitive and private data, we propose to secure user logins with user-specific passwords. The system is programmed to send out text message reminders through the Datalog^{LB} code, approximately 15 minutes before a dose is due. The message reminds the patient that a dose is due and includes a clickable link to their personal website. Fig. 3.1 shows the home page or the Welcome screen.

Figure 3.1: Welcome Screen

When the user first enters the site, a "reply" predicate (`sdcs:reply`) is created and associated with this session. As the user moves through the site, additional information, such as the patient's id, date and time of visit, barriers and solutions indicated etc., is associated with this reply forming a time-stamped log of the user's interaction with the site and their adherence behavior. Such real-time logging is an improvement over the state-of-the-art in the collection of medication adherence data, which typically occurs in paper-and-pencil format at clinic visits, at which patients are asked to retrospectively report their adherence over the past few months. When the user enters the Welcome page, the current time and user id is compared to the database of scheduled dosages for that user. Because the risk of rejection is so high, health care providers typically specify tight time windows in which patients should take their prescribed medications. The MedAdherence system calculates the times at which patients should take their medications based on the scheduled time of the patient's dosage and the allowable time window specified by the healthcare provider. If the user enters the Welcome screen at a time when she is too early or too late for a given dose she is directed to a page, i.e. the Early/Late screen in Fig.3.2, which tells the patient that she is either too late or too early for a dose.

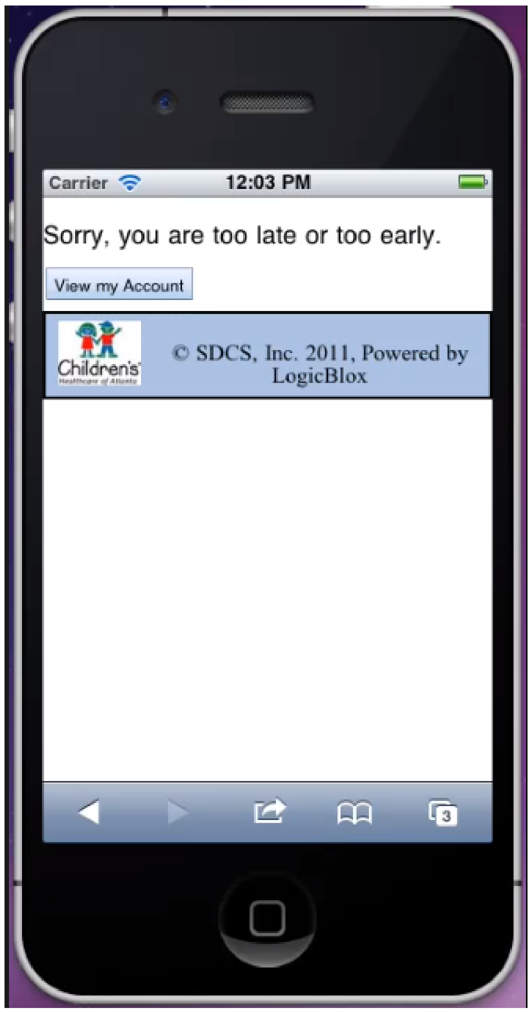

Figure 3.2: Early/Late Screen

If the patient enters the Welcome screen within the allowable time window around a scheduled dosage, the Medication screen, seen in Fig.3.3, displays the doses to be taken at that time.

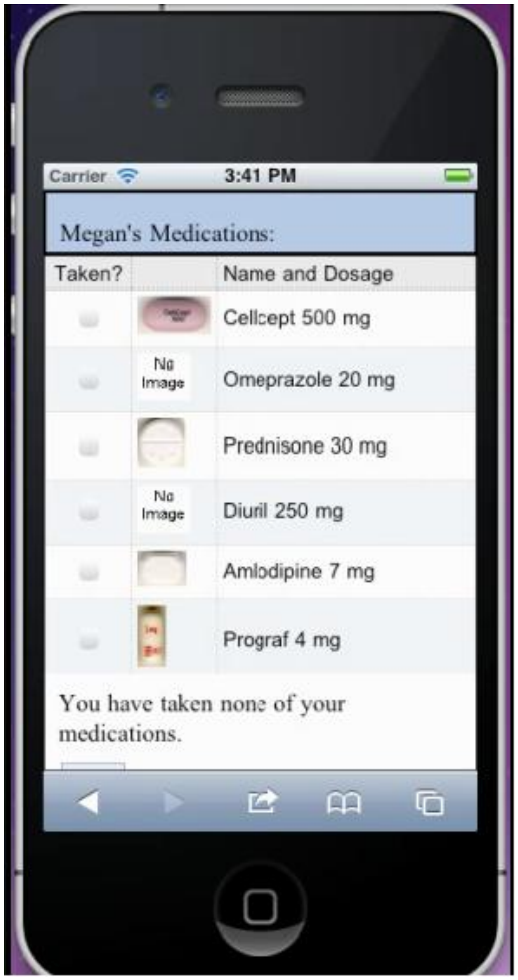

Figure 3.3: Medication Screen

Next, the patient checks off the medications that she has taken. Fig. 3.4 shows this interaction. The string at the bottom of the page reflects whether the patient took all, some or none of her medications. The system asserts facts into the workspace indicating the medications taken by a patient and the time at which this was reported.

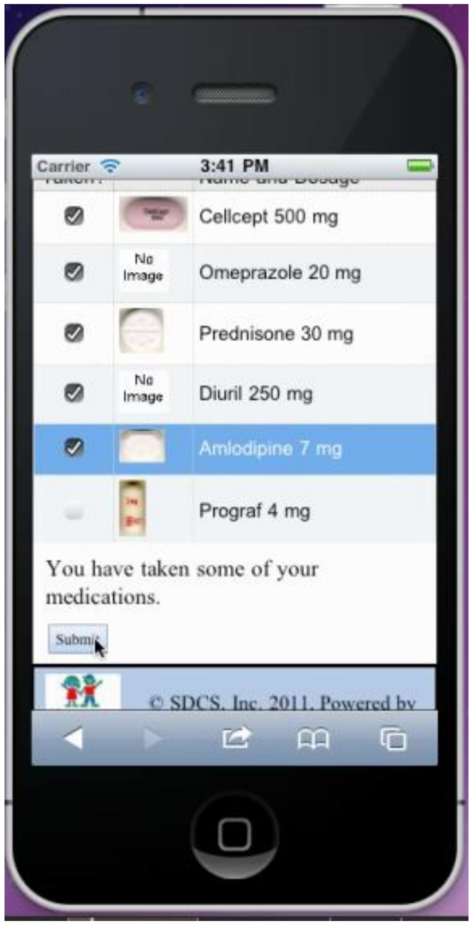

Figure 3.4: Patient checks off medications

Next, depending on whether the patient took some, none or all of the medications, she is taken to one of several pages that instructs her about further actions required. Figs. 3.5, 3.6 and 3.7 depict the screens for these respective scenarios.

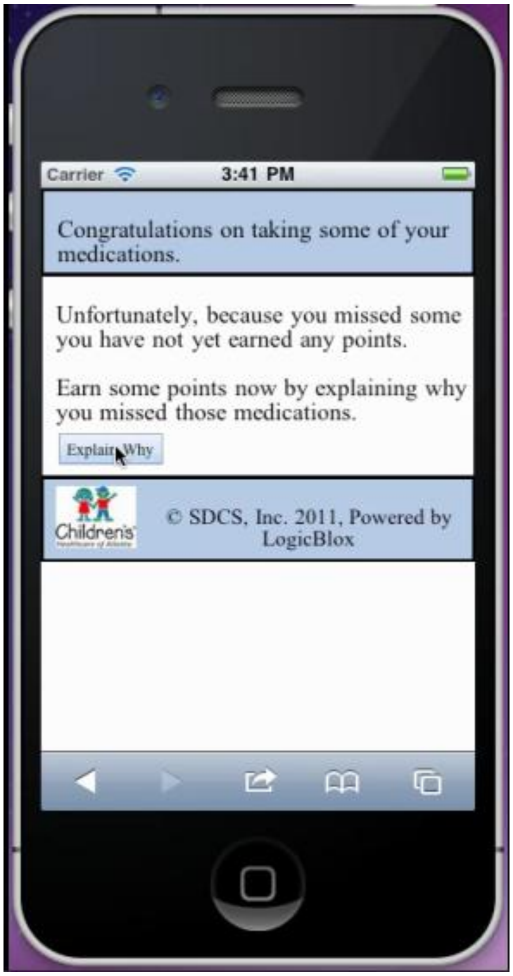

Figure 3.5: Screen after patient indicates taking some of her medications

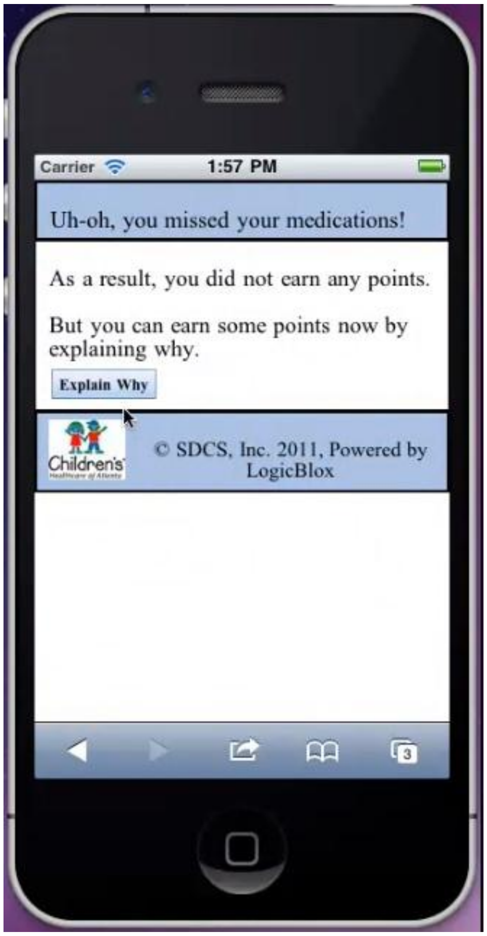

Figure 3.6: Screen after patient indicates taking none of her medications

Figure 3.7: Screen after patient indicates taking all of her medications

If the patient took some or none of her medications, she is directed to a “barrier” page, which retrieves from the database the “barriers” to adherence – the user-specific reasons that this patient may have missed all of her medications or the user- and medication-specific reasons this patient may have missed each particular medication. If the patient misses all of her medications, she is prompted to respond to a single set of barriers to taking medications. These barriers are based on the patient’s previous responses to a survey. The idea behind this is that since all medications were missed, the barriers weren’t specific to any particular medication but were instead based on general barriers to taking any medication. Fig.3.8 shows a barrier menu in this case.

Figure 3.8: Barrier screen in case all medications are missed

Then, based on the barrier selected, the patient is presented with a set of potential solutions that she can adopt. Again, these solutions are specific to the patient, medication, and selected barrier and are based on prior surveys for this patient and healthcare provider or researcher-selected solutions that should work for this patient. Figure 3.9 shows the corresponding solutions menu pertaining to the selected barrier. This presentation of potential solutions takes this system beyond a mere reminder system, and into the realm of in-the-moment, personalized, cognitive behavioral therapy. If none of the potential solutions presented to the patient are satisfactory for the current issue, the patient can have a caregiver contact her to help work out a solution for the current issue that prevents the patient from taking her medication.

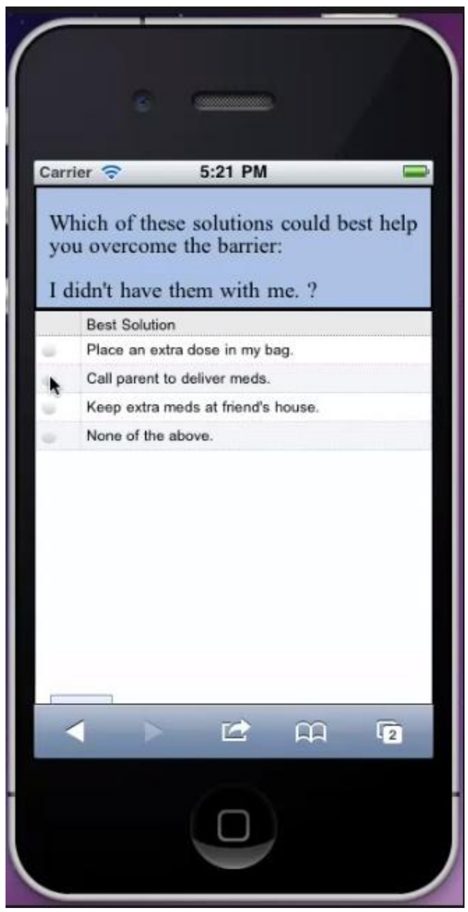

Figure 3.9: Solution screen in case all medications are missed

Based on whether the patient took all, none or some medications on time and provided they responded to the problem-solving barrier and solution menus, they are awarded points. If they took all of their medications on time, they are awarded 20 points. If they took some of their medications but responded to the problem solving menus, they receive 15 points. Lastly, if they indicated taking none of their medications, but responded to the problem solving menus, they receive 10 points. We propose to translate the total points earned in each week into a monetarily equivalent amount placed on a debit card. This incentive scheme is a way to further motivate the patients to practice adherence. Fig. 3.10 shows a sample account summary page. This page displays the points earned by taking the current dose and responding to the problem solving menus, if any and also the total points earned in the current week.

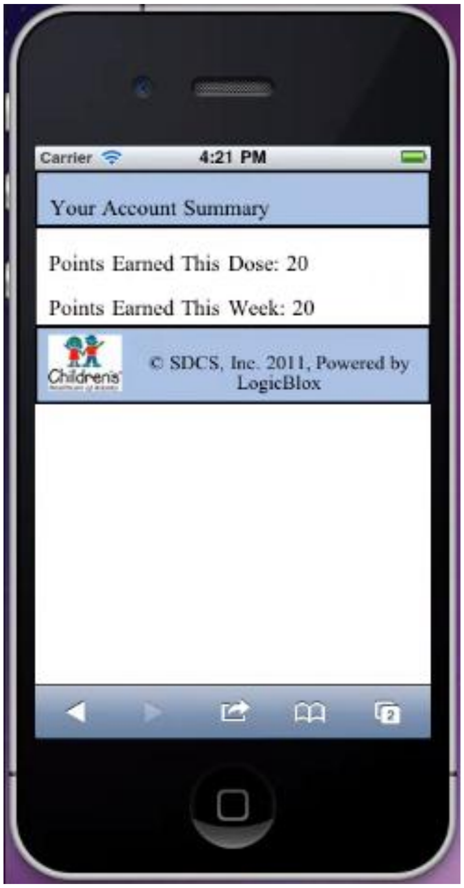

Figure 3.10: Account Summary Screen

A cron job runs each night, during which the system exports data from the workspace to comma separated value (CSV) files. These files are then made available on a daily or weekly basis to the patient's psychologist, physicians or researchers for further analysis and to keep track of their patient's progress.

Fig. 3.11 shows the complete system flowchart. The following steps explain the steps in the flowchart:

1. When the user visits her homepage, simply known as the Welcome page, there are three possibilities:

a) If the patient visits her home page within a two hour time window of her scheduled dosage, i.e. an hour before her scheduled dosage or within an hour after her scheduled dosage, he is said to be in the “on time window”. Intake of medication in this time window is highly recommended and thus maximum points are awarded with the intake of medications in this window. Assuming a dose is due at 9 am on a given day, 8am to 9:59am is called the on time window. A patient visiting the Welcome page in this window is directed to a page which displays her medications. This page is called the Medication page.

b) If the patient is found not to have checked into the system after the initial text message, a second text is sent an hour after the scheduled dosage time. Thus, from an hour after the scheduled dosage time until four hours after the scheduled dosage is demarcated as the “late time window” Assuming a dose is due at 9 am, the time period between 10 am to 1pm is known as the late time window. Naturally, reward points associated with this window are low. In this case, a patient visiting the Welcome page is first directed to a page that checks for the patient’s reason for a late visit. This page is known as the “No Response/Late Response ” page. After the patient has finished reporting the reason for a late response, he is directed to the Medication Page.

c) In case a patient visits her home page at any other time, apart from the on time or the late time windows, it is assumed and understood that there is no dose due at that point and so that patient is directed to a page that informs her, that she is either too late for the current dose or too early for the next dose. This page is known as the “Early page”. The Early page contains a link to the patient’s account summary, which displays the total points earned in that week.

2. Once the patient is on the Medication page, there are three possibilities:
   a) Patient takes all medications: If the patient indicates taking all of her medications then it means that he was completely adherent to her regimen. He therefore does not need to respond to any problem solving menus and is awarded maximum points. In the on time window, 20 points are awarded if complete adherence is observed and in the late window, 12 points are awarded for complete adherence.
   b) Patient takes some of her prescribed medications: Taking some medications but not all shows moderate adherence to the prescribed medication regimen. Therefore, the patient is required to follow problem solving menus for each missed medication, in order to overcome the barriers to non-adherence and subsequently earn points for her response. In the on time window, 15 points are awarded for taking some medications and responding to the problem solving menus for the missed medications . In the late window, 8 points are awarded for following the problem solving menus.
   c) Patient takes none of her medications: If a patient reports taking none of her medications, it shows complete non-adherent behavior. In this case, the patient is required to respond to a general problem solving menu of barriers and probable solutions. The reason for this being that, since all medications were missed , there is likely to be a common barrier for not taking any of the medications. In the on time window, 10 points are awarded for responding to the problem solving menus whereas in the late window, 4 points are awarded for completing the problem solving menus.

Fig. 3.12 depicts the points structure in the on time scenario and Fig. 3.13 shows the late window scenario.

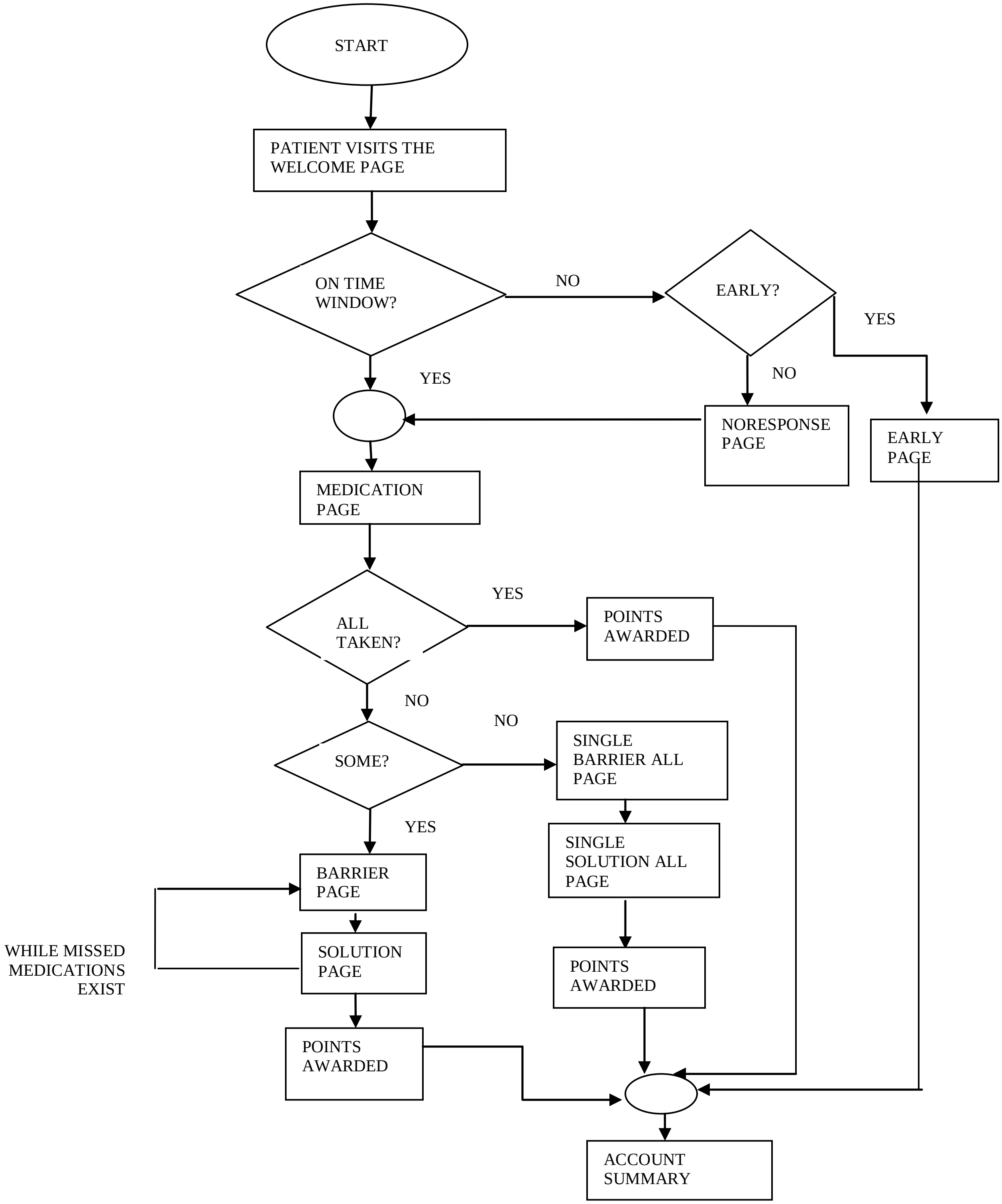

Figure 3.11 System Flowchart

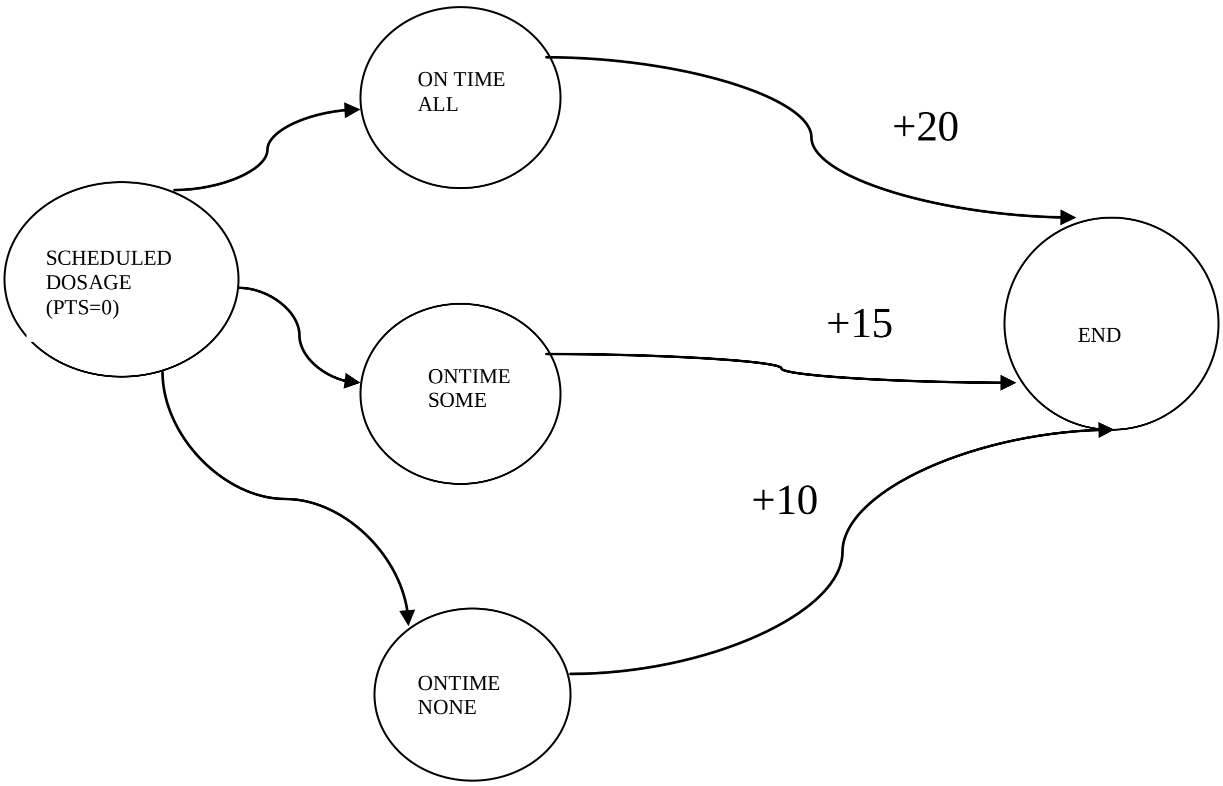

Figure 3.12: Points: All/Some/None taken in the on time window

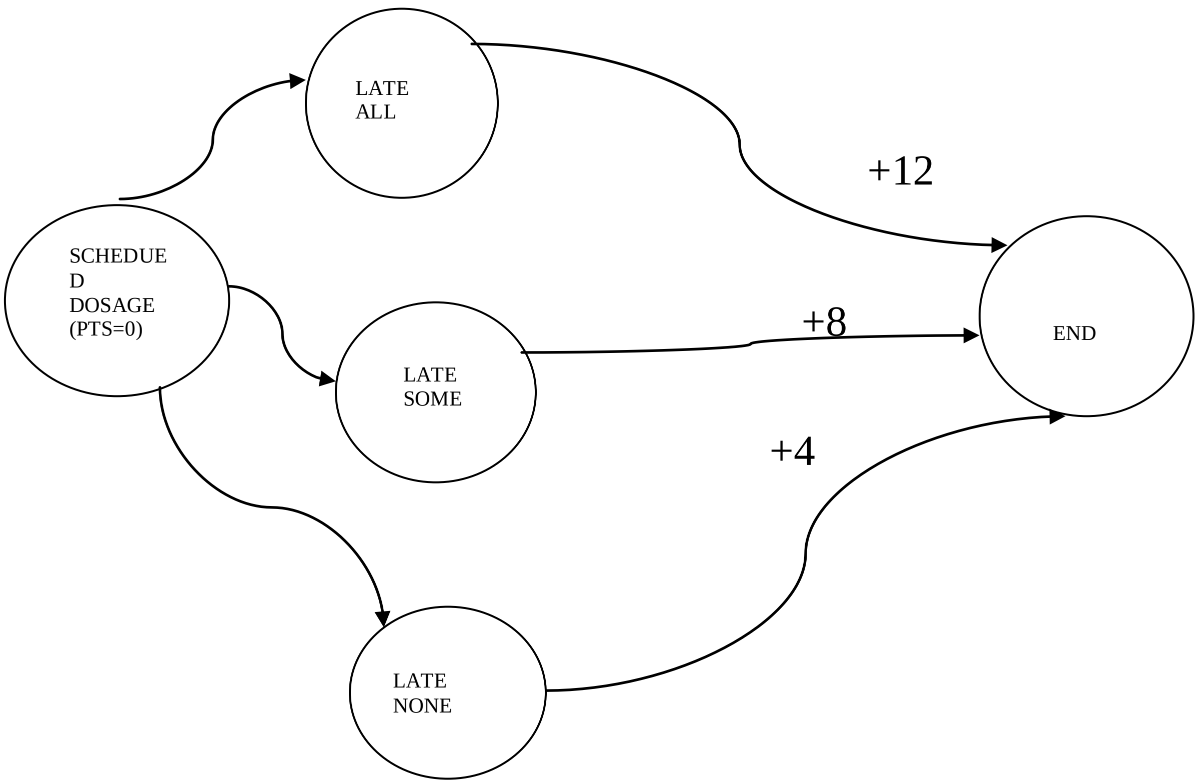

Figure 3.13: Points: All/Some/None taken in the late window

In addition to the patient log-in prompted by the arrival of a text message, patients may pre-emptively log-in using a prior text message or bookmark. Also, patients who have already logged in and reported their adherence behavior have the capability to log in again, within the allowed time window, to indicate if they have taken any additional medications since the last login session. They can log in multiple times but the points are awarded only for the initial response and one additional response. If the patient indicates taking none of her medications in the initial response, they she can either indicate taking some or all her medications for that scheduled dosage. If she takes some, she gets an additional 5 points, bringing her total to 15 points. If she takes all, she gets 10 additional points, bringing her total to 20 points. Similarly, if she takes some in the first session, she can take some more or all in the second session. Consequently, she gets an additional 2 points for taking some more, bringing her total to 17 points or an additional 5 points, bringing her total to 20 points respectively. Figs. 3.14 and 3.15 depict these two cases.

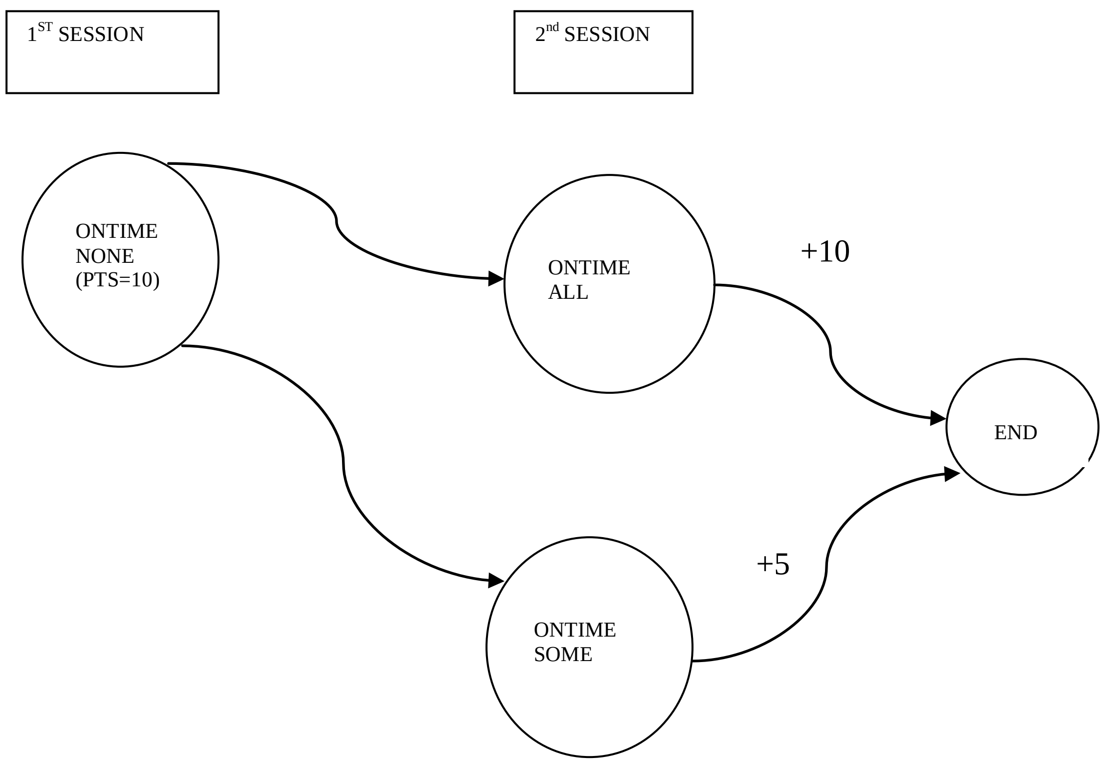

Figure 3.14 : Points : None to All taken and None to Some taken transitions in the on time window

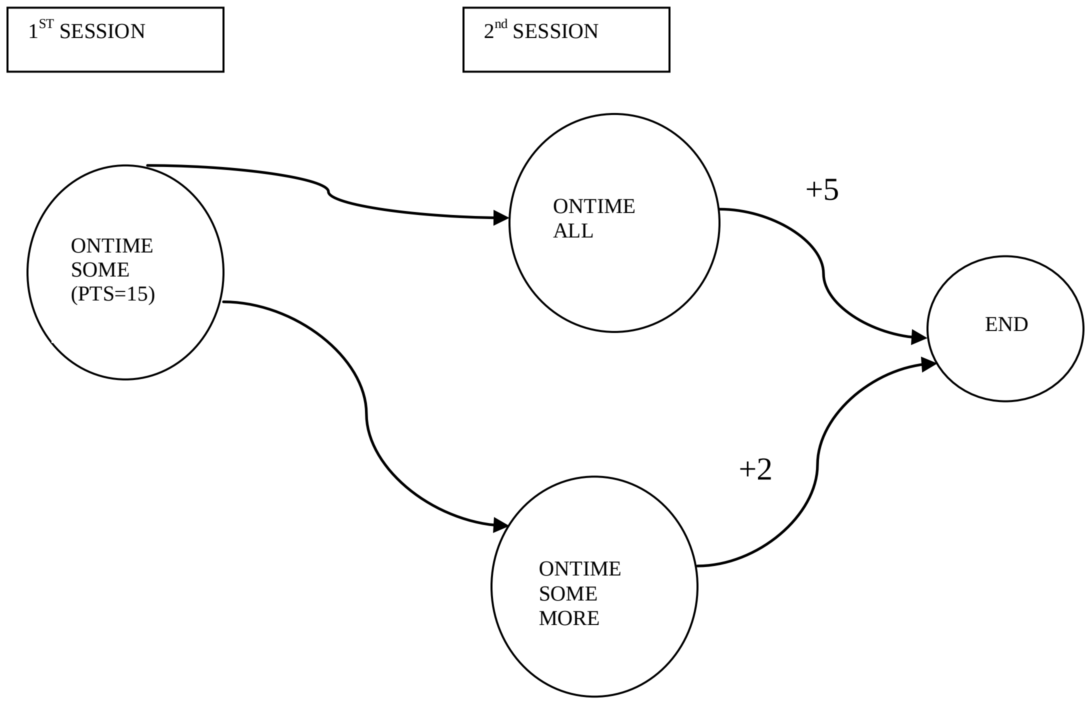

Figure 3.15: Points: Some to All taken and Some to More taken transitions in the on time window.

The points structure becomes a little more complicated when a patient logs in once in the on time window and then logs back in to report in the late window. The points structure in this case can be broken down into the following scenarios:

1. If a patient indicated taking some medications in the on time window and then she logs back in the late window and indicates taking all of her remaining medications, she is awarded 3 additional points.

2. If a patient indicated taking some medications in the on time window and then logs back in the late window and indicates taking more but not all the remaining medications, she is awarded 1 additional point.
3. If a patient indicated taking none of her medications in the on time window, then logs back in the late window and indicates taking some medications then she is awarded 2 additional points.
4. If a patient indicated taking none of her medications in the on time window, then logs back in the late window and indicates taking all of her medications, she is awarded 4 additional points.

Fig. 3.16 depicts the points structure when the patient logs in a late time window following an on time session. In this case too, points are awarded only for the initial on time response and one additional late window response

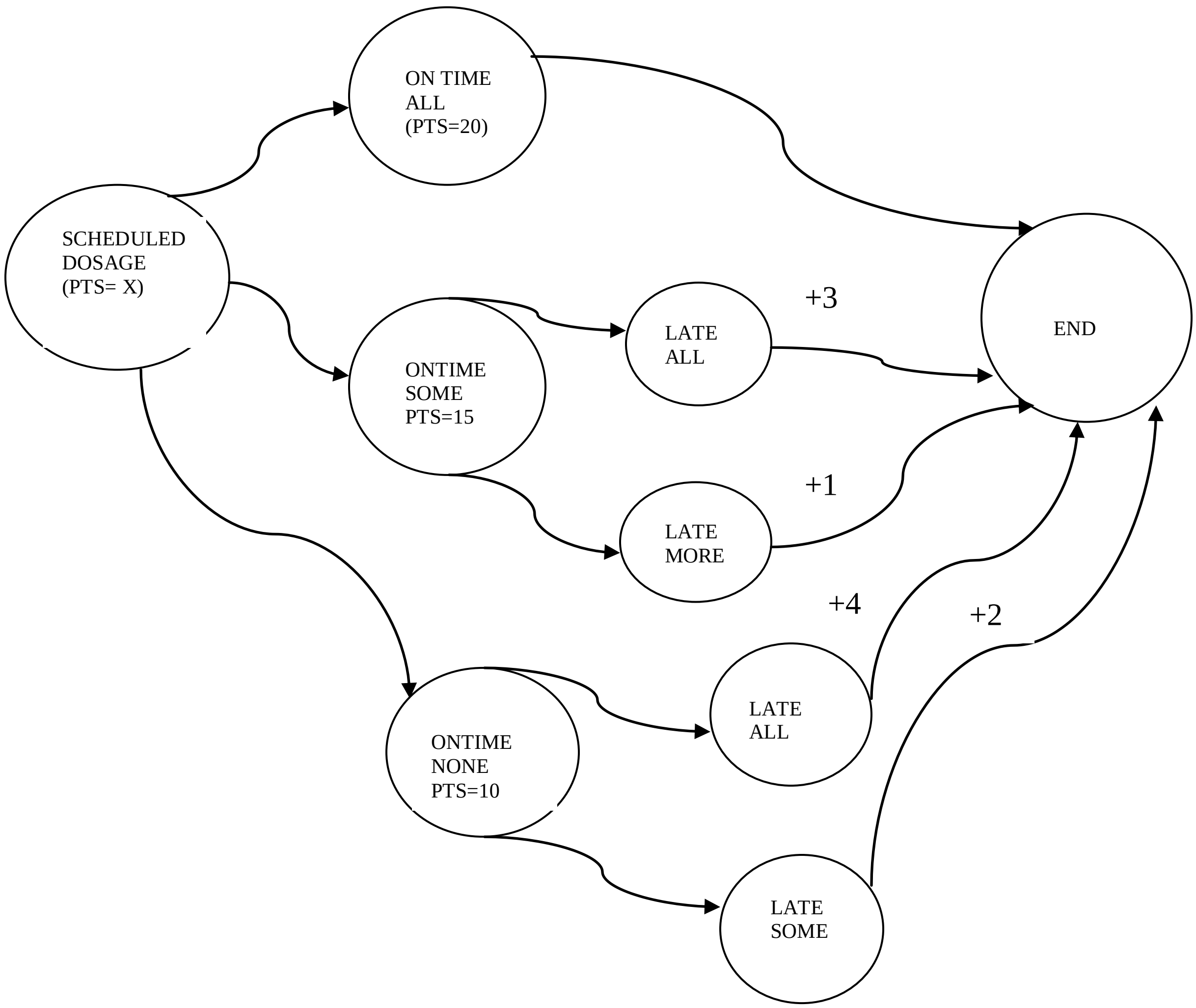

Figure 3.16: Points: On time window and subsequent late window transitions

If the patient does not respond to the first text message, a second text message is sent after an hour of the dosage time. When the patient logs in, they are asked why they did not respond to the initial text and the reason for not responding is recorded. This is called the No response screen. Then, her medications are displayed and the subsequent interaction depends on whether all, some or none of the medications are taken. As opposed to the on-time interaction, multiple logins are not permitted in the late window. This case was depicted in fig. 3.13. After the user has completed an interaction in the late window, if he logs back in, he is taken to a page that notifies her that he has already responded to that dose. Fig. 3.17 and 3.18 show the No response page and the Taken page, respectively.

Figure 3.17 : No response Screen

Figure 3.18: Taken Screen

Predicates that keep track of the “all/some/none” interactions in every session are computed as IDB predicates, evaluated by the underlying IDB rules. Points are updated according to the interactions described above and are stored as delta facts. The system stores other EDB predicates for information gathering.

These EDB predicates capture patient responses to medications taken, barriers and the subsequent solutions adopted. Other EDB predicates in the system track a patient's every visit to the welcome page.

Thus we have seen how $Datalog^{LB}$ is A viable candidate for mobile web application offering a rich declarative framework, especially in the context of applications related to health care where code verifiability are vital.

CHAPTER 4

EVALUATION

MedAdherence will be used in the clinical trial of Smartphone-based delivery of Cognitive Behavioral Therapy for pediatric transplant patients. To this end, pilot testing was performed with the researchers and graduate students of the Pediatric Psychology Lab at the University of Georgia posing as students. A local web server was installed and fake patient data was loaded into the system.

4.1 MedAdherence Prototype:

The sample group consisted of about 7-8 people owning a variety of Smartphones and using a variety of service providers, such as Motorola Droid on Verizon, iPhone on AT&T network and so on. The test subjects received text message reminders approximately 15 minutes before their scheduled dosages. Minor latency was observed by the subjects in receiving the texts. However this was noted as a minor problem, as the texts would arrive within 2-3 minutes of their scheduled times. Certain performance issues were also observed. Users on a wireless network reported having a better user experience than those on their respective 3G or 4G networks. This may be attributed to the signal strength in the location at that time. From our observations, we found that iPhone with AT&T connection had the best user experience with pages loading within a matter of seconds and Android on Verizon had the worst load time of over 50 seconds per page. These observations and this thesis should in no way be treated as an evaluation of the service provider networks or the phone devices nor does there exist a bias of any kind with either the service provider and/or the phones. Based on user feedback and the above observations, certain major changes to the working of the application were made. These changes include adjusting the core logic to accommodate multiple user logins, adjustment to points structure relative to multiple user logins and other syntactic and textual changes to imitate first person thought process. In addition we found that upgrading from the "development" web server (which does not permit client-side caching of JavaScript) to the "production" web server (which does permit client-side caching of JavaScript) reduced the worst-case page load times from 50+ seconds down to just a few seconds. This feedback to LogicBlox was instrumental in their decision to discontinue the use of UIBlox framework, since they saw how the combination lightweight mobile browsers and mobile network strengths impacted application performance.

Some users reported difficulty in checking the checkboxes on the Medication, Barrier and Solution pages, both with the size of the checkboxes on some mobile browsers and with latency in the response to clicking on these checkboxes. They requested making the column click-able or making the checkboxes bigger. Although neither was a viable option, we made use of the "loading modal" element which tells the user that their response has been recorded and that information is being sent to the server. This is in accordance with the recommended techniques outlined in the W3C guidelines for minimizing perceived latency [59]. The loading modal widget also informed the user that the application was not halted and that prevented the users from checking and un-checking controls to make sure their response was recorded. Initially, pill images reportedly took a long time to load and would fail to load on rare occasions. This could be attributed to the fact that the pill images were high resolution . High resolution pill images may have been too intensive for mobile browsers to process, resulting in higher load times. Also, since the caching capability of mobile browsers is lower than desktop browsers, caching of high resolution pill images was assumed to be a problem as well. We thus reduced the resolution of the images and the load times of the medication page became acceptable . The users wanted to have different kinds of user controls like drop down menus, lists, radiobuttons etc. But due to the limited UI controls, imposed by the choice of the grid layout, we were unable to include these controls. We wrote additional logic to emulate radiobuttons, although we were unable to include lists or dropdown menus. Apart from these, feedback on syntactic and textual changes was received and corresponding modifications were made to imitate first person thought process.

The high load times and the limited UI controls were also perceived also posed development challenges which we overcome using the approaches discussed above. Furthermore, limited documentation on UIBlox was another challenge.

At first, the application functionality was fairly simple: A patient was sent a text message 15 minutes prior to a scheduled dosage. The patient was allowed only a single login opportunity within a tight time window of 75 minutes, i.e. 15 minutes before the due dosage to an hour after the scheduled dosage. If the user happened to make a subsequent visit to the welcome page outside this window, she would be redirected to the taken/DueTaken page . On the other hand, if the patient's first visit to welcome page was after an hour of the scheduled dosage, she would be redirected to the Early/Late page, indicating that she is too late or too early for the next dose. In the summer of 2012, changes to the application were proposed by the Pediatric Psychology Lab members, multiple logins and subsequent modification to the points structure was requested. According to this proposition, two permissible time windows were defined – a two hour On time window starting from a hour before the scheduled dosage to an hour after the scheduled dosage. The Late window is defined as the window extending from an hour after the scheduled dosage to four hours after the scheduled dosage. Multiple logins were permitted in these two windows where the patients could indicate medications taken and respond to the problem solving menus if any. Although multiple logins are allowed, points are awarded on responses recorded in at most two logins.

On the whole, users reported application personalization as a huge plus point. Users also reported the several screens to be informative and helpful in understanding the step by step flow of the application.

Table 4.1 summarize the usability evaluation results.

Issue	Action Taken/Resolution
Difficulty in checking checkboxes due to its small size.	Increasing the size was not a feasible option
Not knowing if the application has halted or is loading.	Addition of the "loading" modal widget to the forms, to keep the user informed that the application has not halted. Addressed the checkbox issue to a certain extent as well.
Limited UI controls	Additional logic written to emulate missing UI controls, viz. radiobuttons.
Syntactic changes to text to imitate first person thought process.	Suggested changes were incorporated.

Table 4.1 : Usability Evaluation

The following table summarizes the performance evaluation results. High page load times was considered both a performance issue and a development challenge.

Issue	Action Taken/Resolution
Slow loading pill images or no images loaded on rare occasions	Reduced the image resolution.
High page load times.	Upgraded from the "development" server mode to "production" server mode that enables client-side JavaScript caching. Lead to acceptable load times. LogicBlox to discontinue the use of UIBlox framework.

Table 4.2 : Performance Evaluation

Apart from the development challenges discussed above, challenges with the Datalog^{LB} compiler were observed where it provided little support for debugging. However, upgrading to a newer version of the compiler helped overcome this issue. Table 4.3 summarizes the software development evaluations

Issue	Action Taken/Resolution
High Page load times	Upgraded from the “development” mode to “production” mode to allow client side JavaScript caching.
Restricted UI development tools	Additional logic written to emulate missing UI controls, viz. radiobuttons.
Limited flexibility with the choice of grid layouts	Using the LB ClientGrid layout was the only option as only that provided support for image display.
Difficulty in debugging	Earlier versions of the compiler provided little support for debugging. Upgraded to a new version with better debugging support.
Documentation not up to date with constant the software upgrades	

Table 4.3: Software Development Evaluation

Table 4.4 summarizes the feature modification and addition aspect of the application.

Issue/Request	Action Taken
Allow multiple logins across multiple time windows for a given dosage	Additional logic written to accommodate multiple on time and late window logins
Changes to the points structure to account for multiple logins	Points structure logic expanded to account for multiple logins

Table 4.4: Feature Modification and Addition

4.2 LogicBlox and UIBlox as a mobile web platform :

In spite of the availability of other mobile web frameworks, the LogicBlox platform for mobile web application development was chosen due to the reasons elaborated in Chapter 2. The UIBlox framework is currently the only UI framework supported by the LogicBlox platform and this was the initial reason behind its usage. Initially, the learning curve was steep, as it was a paradigm shift from having worked on routine procedural language frameworks to working with a completely declarative framework. However once the initial hiccups of grasping the fundamentals of the framework were overcome, it was easier to understand and implement a logic-based declarative mobile web application. UIBlox, as mentioned earlier, is based on the MVC framework and is thus inherits its structured and modularized architecture. The form predicates in UIBlox are tied to logic predicates in the underlying logic files. Therefore, workspace management is primarily concerned with effective predicate management. Formatting and styling of the UI seemed to be simple on paper, given that the YAML specification in the form files dynamically generate HTML and styling of form elements is achieved through CSS. However, the end interface was a little clunky and it was difficult to come up with styles and controls that suited multiple platforms and browsers. The W3C Mobile CSS [60] specifications provided guidelines in this matter. Also, in order to be adaptable to any device dimensions, the viewport meta tag, <meta name="viewport" content="width=device-width, height=device-height,initial-scale=1"/> had to be applied as a patch to the UIBlox source at deployment time. Ideally, UIBlox should implement a “mobile” feature that would generate HTML with this tag in place. Similarly, to prevent stray form navigation, the navigation bar on the form template was removed. Again, this should be a feature of UIBlox rather than a custom patch at deployment. On the whole, since the LogicBlox platform was used in its entirety, connecting the UI to the database and launching the database and web server on the cloud was automated. One of the main disadvantages of UIBlox version used for this application is that it is a server – side framework. This results in slower loading forms as all form actions are evaluated at the server side. Also, any changes made to the forms, require the entire application to be rebuilt. Currently, LogicBlox is working on providing a web-services based access to the underlying Datalog^{LB} workspace, which will promote the use of other existing mobile UI development and access the Datalog^{LB} workspace through a web services API.

CHAPTER 5

CONCLUSION

In this work, we contribute to two areas: support for the investigation of the impact of Cognitive Behavioral Therapy delivered via Smartphone for pediatric transplant patients, and an evaluation of the performance, usability, and ease of creation of mobile web apps using the Datalog language on the LogicBlox platform and the UIBlox framework for creation of user interfaces that are tied to Datalog workspaces.

5.1 MedAdherence:

Medication adherence is a must post-transplant. Non-adherence to medication results in organ loss and ultimately, death. Several medication reminder applications are available for the phone and desktop market. However, contrary to popular assumption, barriers to medication adherence for pediatric transplant patients are tied to social and behavioral issues rather than mere forgetfulness or difficulty in swallowing pills. Cognitive Behavioral Therapy (CBT) has been found to be useful in improving medication adherence in adolescents [10]. In this work, we introduced MedAdherence, a mobile web application developed in Datalog^{LB} , that delivers personalized CBT to adolescent kidney transplant patients. The system sends text reminders to patients approximately 15 minutes before a dosage is due. The text contains a click-able link which takes the patient to his or her personal webpage. The system enables the patients to indicate medications taken, elicits reasons for non-adherence to medication and provides strategies to overcome the indicated barriers. Furthermore, the patients are awarded points after every complete and thorough interaction. These points are then translated into cash placed on a debit card which can be made available to the patients for personal spending. Although in this work, we have restricted the target group to pediatric kidney transplant patients, the approach followed in this work is applicable to a variety of other acute and chronic illnesses that can potentially benefit from CBT, like ADHD disorders, weight management, diabetes management etc. Also, as the ownership of Smartphones increases, the mobile web is destined to become even more ubiquitous.

5.2 Viability of the LogicBlox platform for mobile web applications:

In spite of several available mobile application development frameworks, Datalog and UIBlox were chosen for developing MedAdherence. Given the logic fundamentals and declarative nature of Datalog, it supports the core functionality of the application through a base collection of rules and constraints. Furthermore, MedAdherence entails maintenance of medication schedules and generation of automated text reminders at specific dosage times. Constraint optimization and maintenance features of $Datalog^{LB}$ readily support this requirement. Also, given the sensitive nature of the application and the fact that the code is logic based, makes it amenable to code verifiability by the regulatory agencies.

The UIBlox framework is tied to $Datalog^{LB}$ and is currently the only supported LogicBlox UI framework. Hence UIBlox was used to develop the application UI. However, due to the server-side nature of UIBlox, latency in performance was observed, and the creation of UI with a good look-and-feel on multiple platforms was difficult to achieve.. On the other hand, the one-to-one correspondence between form predicates and logic predicates eased the development process.

UIBlox provides a rich set of UI tools for web development, which are more or less adaptable to the mobile platform. However, certain features that were necessary for mobile web were not available as UIBlox features and had to be applied as patches as deployment time

5.3 Future Work

A cloud instance of MedAdherence has been deployed (https://medadherence.logicblox.com), courtesy of LogicBlox®. Cloud computing enables scalability, better data maintenance and backup. This could potentially lead to better performance results and a consequently a better user experience. We foresee the footprint of this application to grow in the next few months, both in terms of the population of the target group and the system functionality based on future user feedback. Currently, all the patient and prescription data is manually asserted into the system. Building the administrative interface for entering patient and prescription information is underway. We are in the process of building UI interfaces for entering patient information and their prescriptions. This physician friendly interface will facilitate easy insertion and updating of patient data. Also, we are currently in talks with Brightwell payments to materialize the points to cash translation and hope to have something worked out soon. LogicBlox is in the process of providing a web-services based access to the underlying Datalog^{LB} workspace. This will promote the use of other existing mobile UI development and permit access to the Datalog^{LB} workspace through a web services API, thus leveraging the combined power of a declarative Datalog^{LB} framework and richer mobile application development framework to create better, developer and user friendly mobile applications.

BIBLIOGRAPHY

1. Meyers, K.E.C., P.D. Thomson, and H. Weiland, *Noncompliance in children and adolescents after renal transplantation.* Transplantation, 1996. **62**(2): p. 186.

2. Shaw, R.J., et al., *A typology of non-adherence in pediatric renal transplant recipients.* Pediatric transplantation, 2003. **7**(6): p. 489-493.

3. Smith, J.M., P. Ho, and R.A. McDonald, *Renal transplant outcomes in adolescents: a report of the North American Pediatric Renal Transplant Cooperative Study.* Pediatric transplantation, 2002. **6**(6): p. 493-499.

4. Simons, L.E., et al., *Multidimensional Adherence Classification System: Initial development with adolescent transplant recipients.* Pediatric Transplantation, 2009. **13**(5): p. 590-598.

5. De Geest, S. and F. Dobbels, *Transplantation: Increasing adherence to immunosuppression: a clinical priority.* Nat Rev Nephrol, 2010. **6**(3): p. 139-40.

6. Dew, M.A., et al., *Meta-analysis of medical regimen adherence outcomes in pediatric solid organ transplantation.* Transplantation, 2009. **88**(5): p. 736-46.

7. Fredericks, E.M., *Nonadherence and the transition to adulthood.* Liver Transpl, 2009. **15 Suppl 2**: p. S63-9.

8. Pinsky, B.W., et al., *Transplant outcomes and economic costs associated with patient noncompliance to immunosuppression.* Am J Transplant, 2009. **9**(11): p. 2597-606.

9. Simons, L.E. and R.L. Blount, *Identifying barriers to medication adherence in adolescent transplant recipients.* Journal of pediatric psychology, 2007. **32**(7): p. 831-844.

10. Spirito, A., et al., *Cognitive-behavioral therapy for adolescent depression and suicidality*. Child and adolescent psychiatric clinics of North America, 2011. **20**(2): p. 191.

11. Velligan, D.I. and P.J. Weiden, *Interventions to improve adherence to antipsychotic medications.* Psychiatric Times, 2006. **23**(9): p. 50-53.

12. Turkington, D., et al., *Cognitive-behavioral therapy for schizophrenia: a review.* Journal of Psychiatric Practice®, 2004. **10**(1): p. 5-16.

13. Safren, S.A., et al., *A randomized controlled trial of cognitive behavioral therapy for adherence and depression (CBT-AD) in HIV-infected individuals.* Health Psychology, 2009. **28**(1): p. 1.

14. Rainie, L., *Smartphone Ownership Update: September 2012* Pew Internet & American Life Project, September 2012(September 11, 2012): p. http://pewinternet.org/Reports/2012/Smartphone-Update-Sept-2012.aspx November 24, 2012.

15. Smith, A., *Smartphone Adoption and Usage* Pew Internet & American Life Project, July 2011: p. http://pewinternet.org/Reports/2011/Smartphones.aspx November 24 2012.

16. Lionbridge. 2012; Available from: https://en-us.lionbridge.com/kc/mobile-web-apps-vs-mobile-native-apps.htm.

17. Xcode. *Xcode*. Available from: https://developer.apple.com/xcode/.

18. Library, M.D. *Objective C*. Available from: https://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/ObjectiveC/Introduction/introObjectiveC.html.

19. Android. *Android* Available from: http://developer.android.com/index.html.

20. Developer, B. *BlackBerry*. Available from: https://developer.blackberry.com/design/.

21. Developer, W. *Windows Mobile Developer Center*. Available from: http://msdn.microsoft.com/en-us/windowsmobile/bb264318.aspx.

22. Suite, R. *RhoMobile*. Available from: http://msdn.microsoft.com/en-us/windowsmobile/bb264318.aspx.

23. MoSync. *MoSync*. Available from: http://www.mosync.com/.

24. PhoneGap. *PhoneGap*. Available from: http://phonegap.com/.

25. W3C, *Web Design and Applications*, in *HTML and CSS*, W3C.

26. W3C, *Web Design and Applications*, in *JavaScript Web APIS*.

27. Sencha. *Sencha Touch*. Available from: http://www.sencha.com/products/touch.

28. Mobile, j. *jQuery Mobile*. Available from: http://jquerymobile.com/.

29. jQuery. *jQuery*. Available from: http://jquery.com/.

30. LogicBlox. *LogicBlox*. Available from: www.logicblox.com.

31. Group, T.F.W. *The Forms Working Group*. Available from: http://www.w3.org/MarkUp/Forms/.

32. Contributors, W. *YAML*. 27 November 2012 09:19 UTC 3 December 2012 06:44 UTC]; Available from: http://en.wikipedia.org/w/index.php?title=YAML&oldid=525101981.

33. WidgetBox. *WidgetBox*. Available from: http://www.widgetbox.com/.

34. foneFrame. *foneFrame*. Available from: http://foneframe.com/.

35. Tiggzi. *Tiggzi*. Available from: http://tiggzi.com/home.

36. Contributors, W. *HTML 5*. 3 December 2012 06:15 UTC 3 December 2012 08:47 UTC]; Available from: http://en.wikipedia.org/w/index.php?title=HTML5&oldid=526132518.

37. Apache. *MyFaces*. Available from: http://myfaces.apache.org/.

38. JBoss. *RichFaces*. Available from: http://www.jboss.org/richfaces.

39. Insel, K., et al., *Executive function, working memory, and medication adherence among older adults.* The Journals of Gerontology Series B: Psychological Sciences and Social Sciences, 2006. **61**(2): p. P102-P107.

40. Murray, M.D., et al., *A conceptual framework to study medication adherence in older adults.* The American journal of geriatric pharmacotherapy, 2004. **2**(1): p. 36-43.

41. McQuaid, E.L., et al., *Medication adherence in pediatric asthma: reasoning, responsibility, and behavior.* Journal of Pediatric Psychology, 2003. **28**(5): p. 323-333.

42. Rao, D., et al., *Stigma and social barriers to medication adherence with urban youth living with HIV.* AIDS Care, 2007. **19**(1): p. 28-33.

43. Lorig, K.R., et al., *Internet-based chronic disease self-management: a randomized trial.* Medical care, 2006. **44**(11): p. 964-971.

44. online, C.L.H.C. *FearFighter*. Available from: http://www.fearfighter.com.

45. online, C.L.H.C. *MoodCalmer*. Available from: www.calmerseries.com.

46. Rodgers, A., et al., *Do u smoke after txt? Results of a randomised trial of smoking cessation using mobile phone text messaging.* Tobacco Control, 2005. **14**(4): p. 255-261.

47. Franklin, V., et al., *" Sweet Talk": text messaging support for intensive insulin therapy for young people with diabetes.* Diabetes technology & therapeutics, 2003. **5**(6): p. 991-996.

48. Abeles, P., et al., *Computerized CBT for adolescent depression (" Stressbusters") and its initial evaluation through an extended case series.* Behavioural and cognitive psychotherapy, 2009. **37**(2): p. 151.

49. O'Kearney, R., et al., *Effects of a cognitive-behavioural internet program on depression, vulnerability to depression and stigma in adolescent males: a school-based controlled trial.* Cogn Behav Ther, 2006. **35**(1): p. 43-54.

50. O'Kearney, R., et al., *A controlled trial of a school-based Internet program for reducing depressive symptoms in adolescent girls.* Depression and Anxiety, 2009. **26**(1): p. 65-72.

51. Fjeldsoe, B.S., A.L. Marshall, and Y.D. Miller, *Behavior change interventions delivered by mobile telephone short-message service.* American journal of preventive medicine, 2009. **36**(2): p. 165.

52. Chen, G., et al. *MPCS: Mobile-phone based patient compliance system for chronic illness care*. in *Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous' 09. 6th Annual International*. 2009. IEEE.

53. Statista. *Smartphone users in the U.S. 2012 -2016*. Available from: http://www.statista.com/statistics/201182/forecast-of-smartphone-users-in-the-us/.

54. MyMedSchedule. *MyMedSchedule*. Available from: http://www.mymedschedule.com/.

55. OntimeRx. *OntimeRx*. Available from: www.ontimerx.com.

56. Contributors, W. *Prolog*. 20 November 2012 14:20 UTC 3 December 2012 09:41 UTC].

57. Contributors, W. *Datalog*. 26 November 2012 08:50 UTC 3 December 2012 09:45 UTC]; Available from: http://en.wikipedia.org/w/index.php?title=Datalog&oldid=524923061.

58. Gottlob, G., et al., *Datalog and its Extensions for Semantic Web Databases.* Reasoning Web. Semantic Technologies for Advanced Query Answering, 2012: p. 54-77.

59. W3C, *Mobile Web Application Best Practices*, Copyright ©2010 W3C® (MIT, ERCIM, Keio), All Rights Reserved.

60. W3C, *CSS Mobile Profile 2.0*, 2008, Copyright © 2008 W3C® (MIT, ERCIM, Keio), All Rights Reserved.