Software Design Specification

for the Digital Medical Assistant

Version 1.0 **Date: (05/01/2010)**

Prepared by Suneil Berajawala

Table of Contents

1 Introduction

1.1 System Overview

The Digital Medical Assistant serves as an aid to a healthcare provider needing to keep patient information organized and available on a BlackBerry smartphone. The Digital Medical Assistant will offer an intuitive interface that effectively aggregates patient information in a task-oriented way. In version 2.0 of the Digital Medical Assistant, the information will be presented in the context of a personal calendar and scheduler, which is likely to be an essential tool for all healthcare providers in the performance of their duties during each day. Patient information will be included with patient appointments that would be overlayed on the provider's personal calendar managed within Google Calendar. To begin with, version 1.0 will offer access to patient electronic health records on a BlackBerry smartphone that are managed within Google Health. The healthcare provider will also have the option to write and store electronic notes and other media depending upon the capabilities of the device in version 1.0. All of this data will also be available offline so that a mobile device without network connectivity can still gain access to this information. For version 2.0, if the calendar entry includes location information such as where the appointment is, then the application will interface with the mapping application on the mobile device if one is available and offer directions.

The benefits of the Digital Medical Assistant will be far-reaching. The value of electronic health records is well documented as a part of eHealth initiatives and it is surely going to become even more important as it becomes necessary to further automate and streamline workflow in healthcare along with the benefits that come from making communication between pharmacies, clinics, etc. easier. With the popularity of smartphones such as the iPhone, BlackBerry, etc, and the increasing capability of these devices, an opportunity exists to extend these kinds of capabilities beyond client PCs to smartphones. A healthcare provider no longer needs to locate a workstation in order to access patient information among other things. He or she can instead access this data from anywhere at anytime with a mobile device interface.

1.2 Definitions, Acronyms, and Abbreviations

Acronyms and Abbreviations:

a. DMA: Digital Medical Assistant: the software application that is being discussed in this design specification.
b. SRS: Software Requirements Specification: previous document
c. SDS: Software Design Specification: this document
d. eHealth: a term used to refer to healthcare practice that is supported by electronic processes and communication.
e. mHealth: a term used to refer to an extension of eHealth concepts whereby electronic healthcare processes and communication are supported by mobile devices.

f. EHR: Electronic Health Record: a collection of electronic health information about
g. individual patients that's in digital form and can be shared amongst different health-care providers. In the context of this document, EHR and Google Health record are used interchangeably.
h. HIPAA: Health Insurance Portability and Accountability Act: an act enacted by Congress that dictates national standards for electronic health care privacy among other things.
i. JEE: Java Enterprise Edition: a widely used platform for server programming in the Java programming language.
j. JME: Java Mobile Edition: a Java platform for mobile devices.
k. RSS: Really Simple Syndication: family of web feed formats used to publish frequently updated works such as news headlines, blog entries, etc. It is most closely related and an alternative to the ATOM Syndication Format. Represents a format commonly used in REST-based Web Services.
l. REST: Representational State Transfer: style of software architecture synonymous with that of the WWW. A central principle is that of the resource, which are sources of specific information, each of which is referenced with a global identifier. Components of the network communicate via a standardized interface, such as HTTP, and exchange representations of these resources.

Definitions:

a. iPhone: refers to a line of smartphones made by Apple that are both Internet and multimedia-enabled. The smartphone incorporates a multi-touch screen with a virtual on-screen keyboard, camera phone, media player, and Internet client.
b. BlackBerry: refers to a line of wireless devices made by Research in Motion. Capabilities vary amongst the line of devices, and they are known for being the most popular smartphone among business users. In this context, the class of BlackBerry device being referred to is the smartphone.
c. Java: a system for developing cross-platform application software.
d. ATOM: consists of a pair of related standards. The ATOM Syndication Format is an XML language used for web feeds, while the ATOM Publishing Protocol is a simple HTTP-based protocol for creating and updating web resources. These standards are often commonly used in REST-based Web Services.

1.3 References

3rd Party APIs

a. Google Calendar (ATOM)

Google Calendar Data API: http://code.google.com/apis/calendar/

Developer's Guide: http://code.google.com/apis/calendar/data/2.0/developers_guide.html

Reference Guide: http://code.google.com/apis/calendar/data/2.0/reference.html

Client Libraries and Sample Code: http://code.google.com/apis/calendar/code.html

b. Google Health (ATOM)

Google Health Data API: http://code.google.com/apis/health/getting_started.html

Developer's Guide: http://code.google.com/apis/health/docs/2.0/developers_guide.html

Reference Guide: http://code.google.com/apis/health/docs/2.0/reference.html

Client Libraries and Sample Code: http://code.google.com/apis/health/code.html

c. MedWorm (RSS)

Homepage: http://www.medworm.com/

3rd Party Frameworks

a. Spring: http://www.springsource.org/about

b. Spring Security: http://static.springsource.org/spring-security/site/index.html

c. ROME: https://rome.dev.java.net/

2 Design Considerations

2.1 Assumptions

The initial version of the BlackBerry OS being targeted for the DMA client is v5.0 because of the built in support for the relational database engine SQLite. The operational environment required for the server-side is a SpringSource server such as the dm Server or the tc Server.

2.2 Constraints

No specific constraints exist other than the resource limitations imposed upon applications that are being written for BlackBerry smartphones. Also, certain other data constraints should also be in effect since excessive data transmission can drain device battery life, result in substantial data network charges due to excess bandwidth being used, etc.

2.3 System Environment

The software that the Digital Medical Assistant server will run on is the SpringSource dm Java Application Server (http://www.springsource.org/dmserver). The Digital Medical Assistant client will support BlackBerry OS v5.0 initially with support extending to other BlackBerry OS versions at a later time.

Integral to the operation of the Digital Medical Assistant are certain services that manage data and business logic. These services that the DMA interacts with are Google Calendar (v2.0), Google Health, and MedWorm.

2.4 Design Methodology

The design methodology being employed here is the UML object-oriented technique insofar as necessary since large parts of the DMA functionality are implemented by 3rd party services that have been designed and implemented already. Links to that design documentation are offered in this specification as reference material.

3 Architectural (High-level) Design

3.1 Overview

The DMA client will run on the BlackBerry smartphone. It consists of classes that implement the BlackBerry DMA UI along with using certain classes in the BlackBerry API Collection that allow for network transport access, xml parsing, SQLite DB access and manipulation, etc. No 3rd party libraries are utilized and so the only dependancy for the DMA client is the BlackBerry v5.0 API Collection.

The DMA server application is a Spring MVC JEE-based web application that runs within the SpringSource dm Server. It consists of Spring controllers that respond to requests for data from the BlackBerry DMA client and Spring views that present the model as the response to the client requests. These controllers/views access the 3rd party services that manage the requested data (Google Calendar, Google Health, and MedWorm), build an appropriate and optimized model for transmission to a resource-constrained mobile device, and build the view of that model for transmission to the mobile device.

3.2 Rationale

Given the stated requirements of the DMA in the SRS, it's quite clear that those requirements can be mapped onto certain 3rd party services that offer the desired feature/function. Instead of reinventing the wheel, it would be far more prudent to reuse existing services. As a result, Google Calendar, Google Health, and MedWorm were chosen. All 3 services offer RESTful APIs that support standard formats such as ATOM and RSS. ATOM and RSS are XML-based formats, and with XML possibly being quite verbose, this characteristic may be at odds with the stated goal of sending as little data as possible to the resource-constrained mobile device. As a result, the architecture calls for a server-based application that can consume the feed formats from these services and build a model that only contains the data that is required by the DMA client. This generated model will then be significantly more efficient for the mobile device to process. Also, with data being "cached" locally in a device-side SQLite database, even less data needs to be transmitted to the client achieving the stated goals.

3.3 Conceptual (or Logical) View

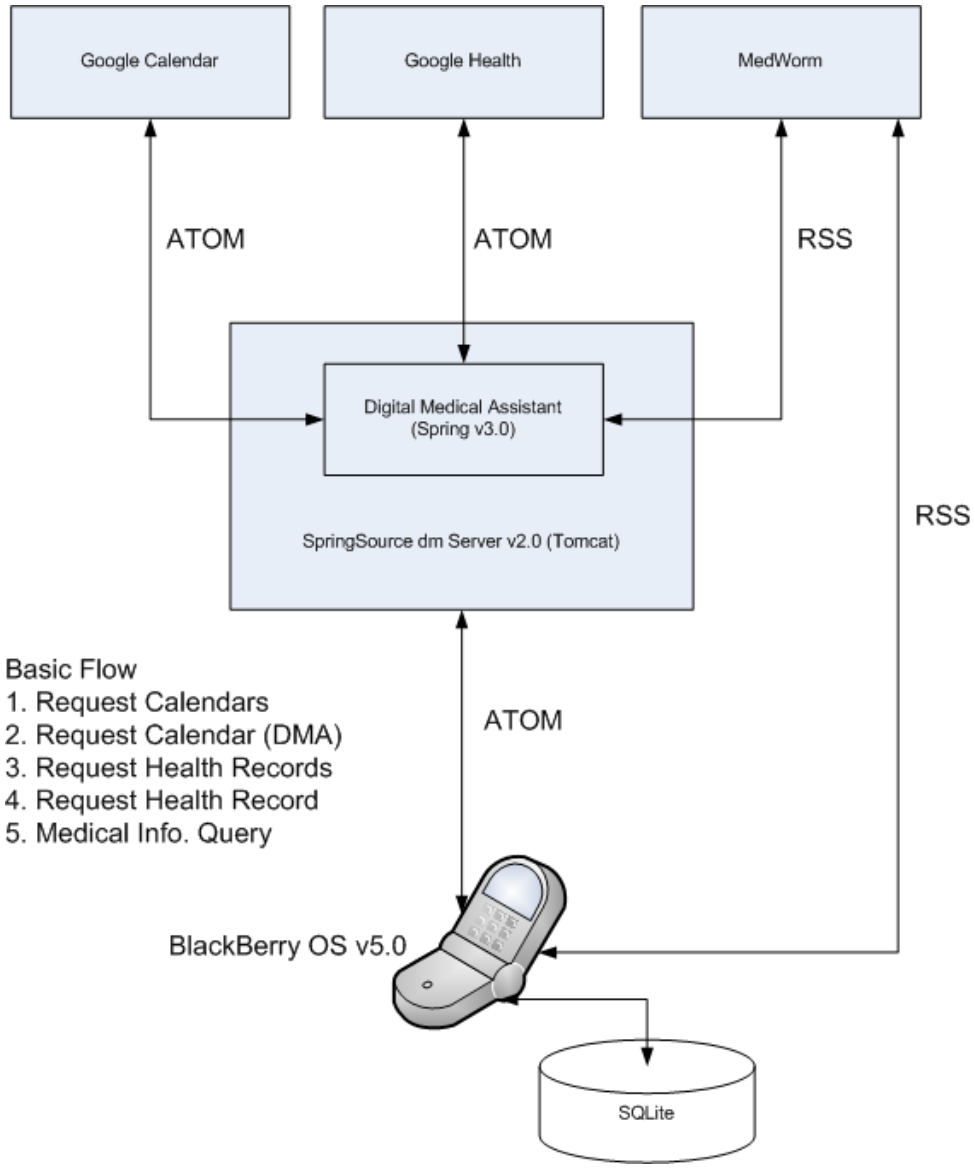


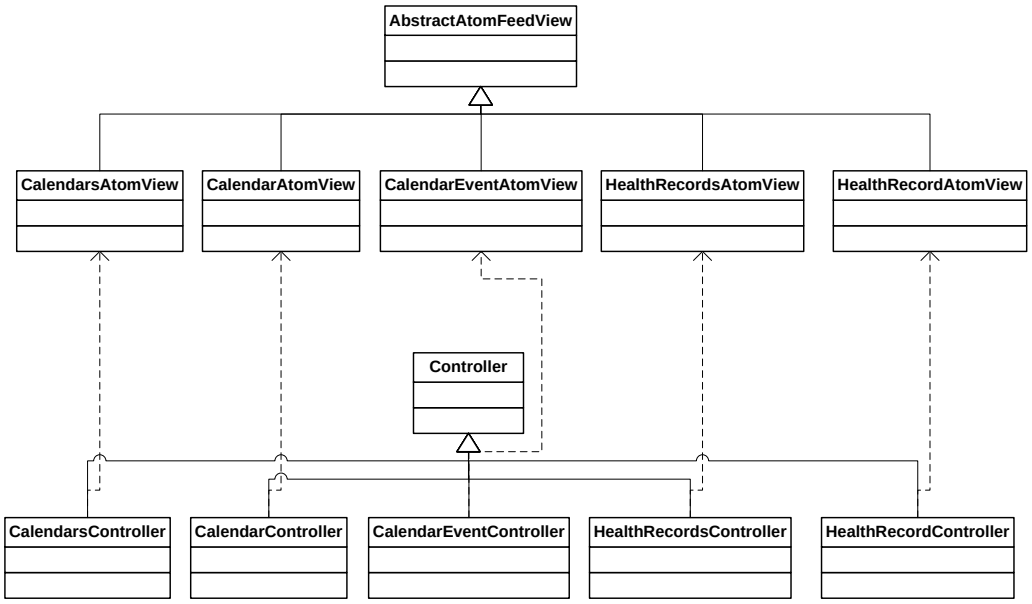
Figure 1.1: DMA Block Diagram

The diagram depicts the principal parts of the DMA along with the relationship between those principal parts. All requests originate from the DMA client running on the BlackBerry smartphone. These requests access calendar information (v2.0), patient record information, and medical term reference material. Data is also cached in an SQLite relational database for offline availability. Not shown in the diagram is the storage of recorded audio files (dictated notes), which will be stored initially within the BlackBerry filesystem SD Card. The DMA server application provides optimized and efficient to process ATOM feeds that contain the

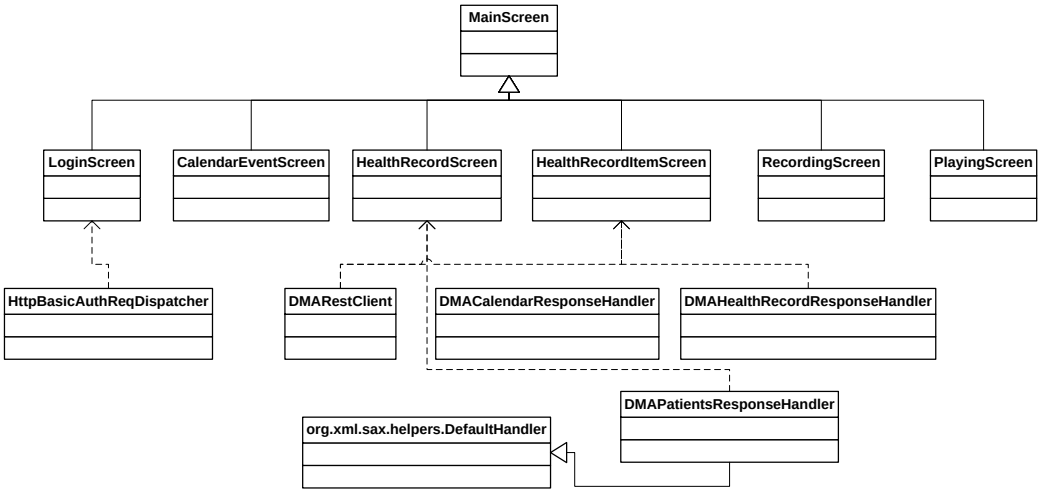
requested information. The DMA server application needs to consult the 3rd party services that manage the data in response to requests from the DMA client. For calendar information, the DMA server application will query the Google Calendar service. For health record information, the DMA server application will query the Google Health service. For medical reference material, it will consult the MedWorm service.

4 Low Level Design

4.1 Server (Spring Web App)



4.2 Client (BlackBerry Rich Client)



5 User Interface Design

5.1 Application Control

The primary user interface consists of a tabbed interface as can be seen in the following screenshots. A custom manager is used on every screen such that the header and footer stay visible even while scrolling. Other Fields used are standard for the most part. The desire was to not customize as keep the familiar BlackBerry UI look-and-feel that the end-user would be accustomed to with using the BlackBerry native applications.

5.2 Screenshots

Figure 5.2.1: Login Screen

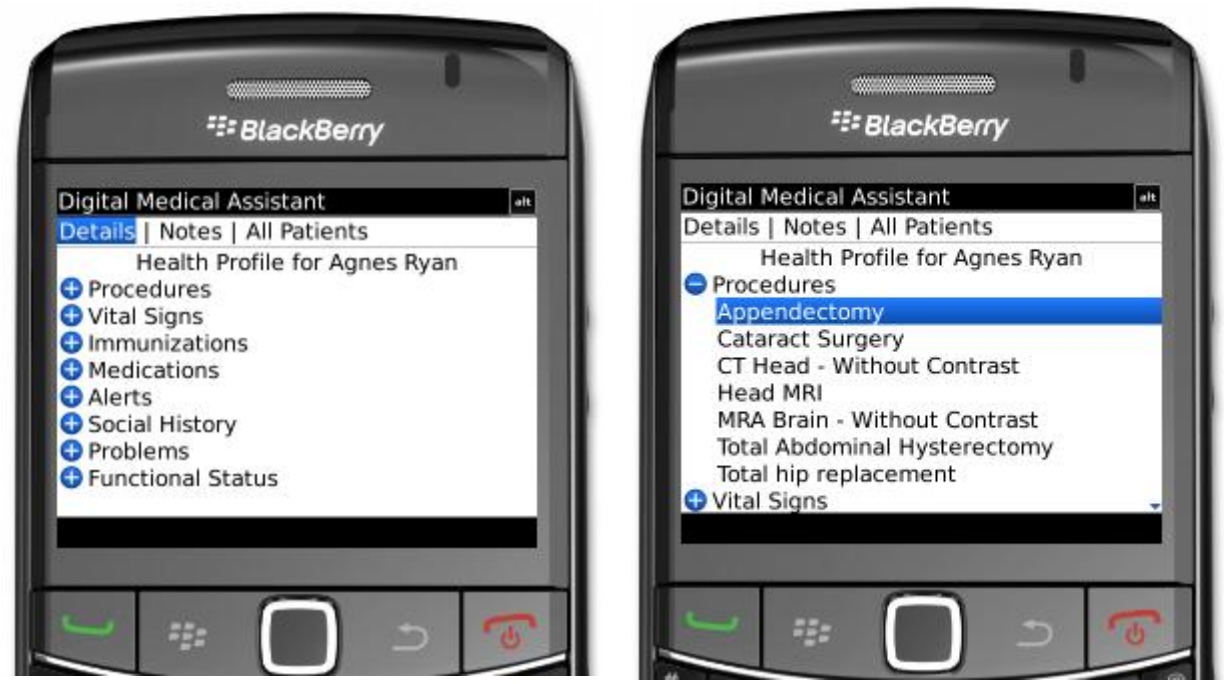


Figure 5.2.2: Patient Health Profile (Google Health) Screens

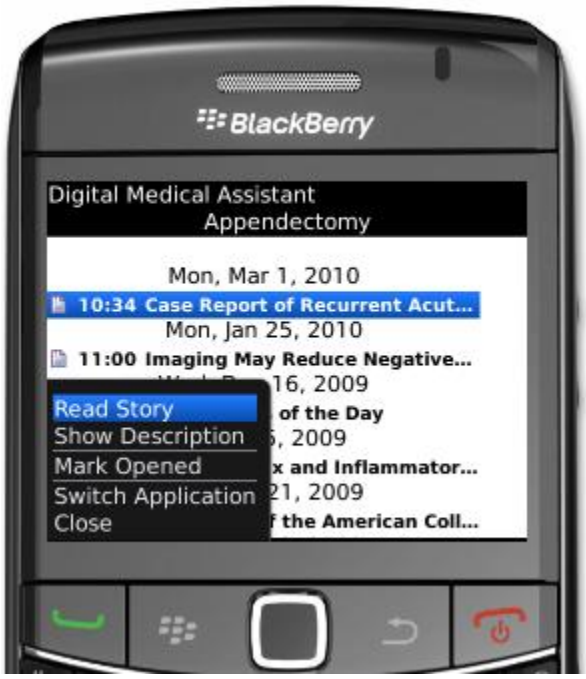


Figure 5.2.3: Medical Term Lookup (Web Feed from MedWorm) Screen

Figure 5.2.4: Dictated Patient Notes and Media Player Playback Screens

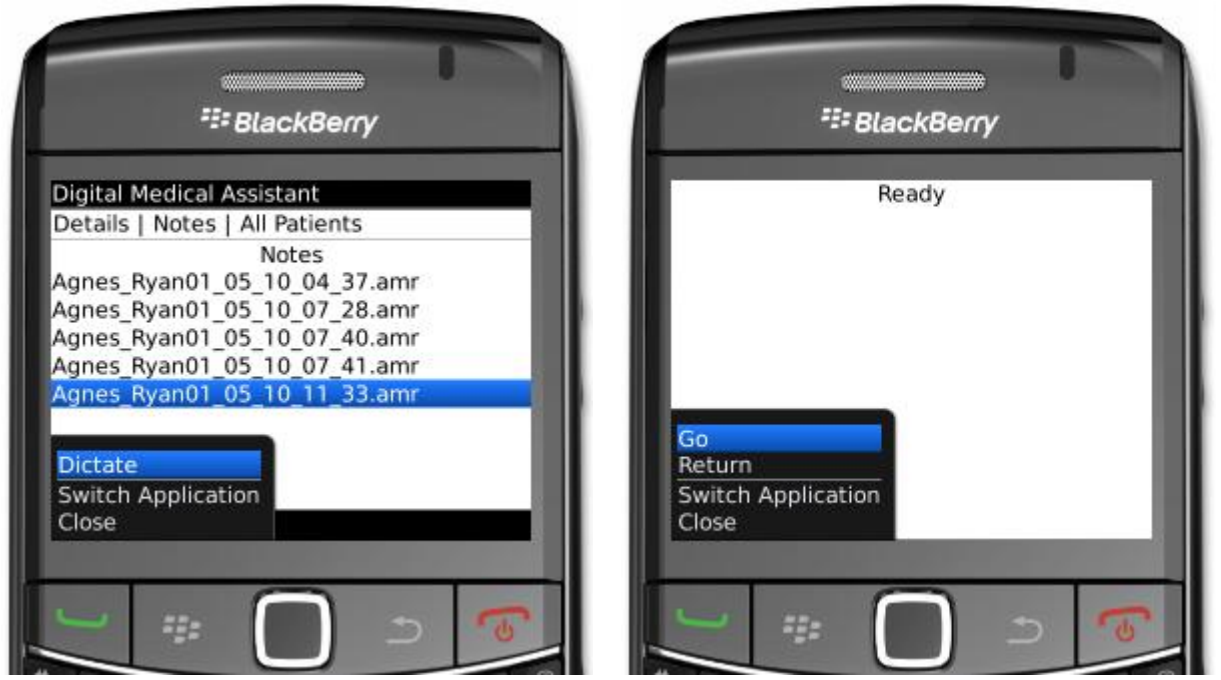


Figure 5.2.5: User Interface for recording Dictations

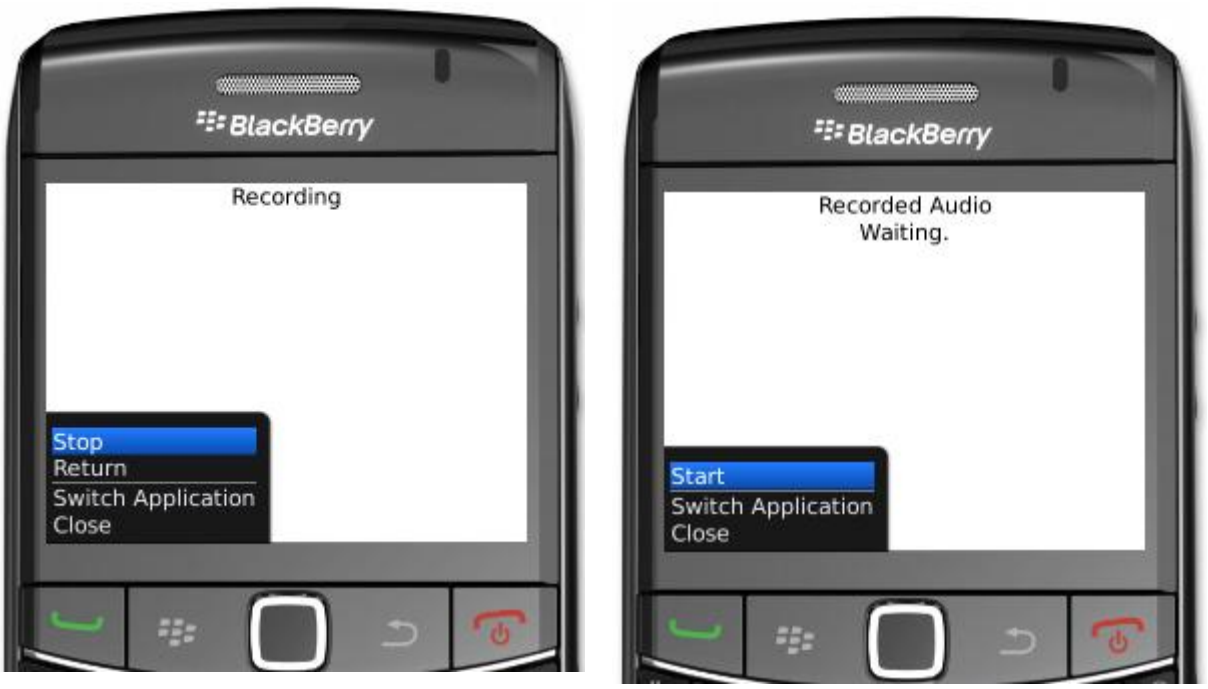


Figure 5.2.6: User Interface for recording Dictations

Figure 5.2.7: User Interface for reviewing recorded Dictations

Figure 5.2.8: New Dictation listed

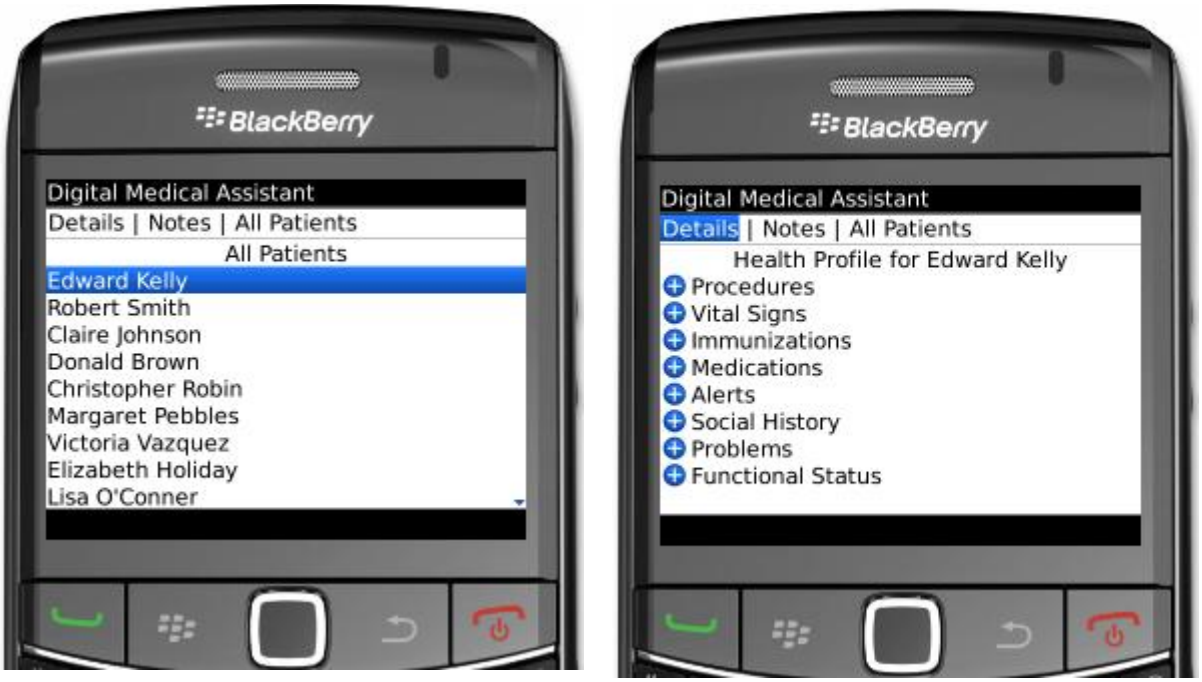


Figure 5.2.9: New Patient record