Technical Specification Document

Notifyi

IPhone Application

05 September 2012
Rapid value

Document Information and Approvals

Version History

Version #	Date	Revised By	Reason for change
1.0	06/09/12	Veena Chandran	Initial draft

Document Approvals

Approver Name	Project Role	Signature/Electronic Approval	Date

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This project deals with development of an iPhone application for Notifyi users to communicate effectively in the network through an email system which works on iPhone.

This document is intended to address the following users;

1. Notifyi/ Redshift team, the end user who is supposed to approve this document
2. RapidValue Development team includes the Project Manager, Engagement Manager, developers and testers

This document discusses the technical functionality of the system. This document will be used by Designers, Developers and Testers. User screens, technical flow; code snippets (wherever required) and data design will be explained in detail using UML. Document owner would be “

SECTION 1. INTRODUCTION

This document describes the high-level design for developing an iPhone application for **Notifyi users** clinical management system, which would help its clients [doctors] to manage their patients and related task on the move.

1.1 Purpose

The purpose of the system under development is to develop an iPhone application that would enable the physicians in the notifyi network to send/receive messages , touch base, view a physician's profile, view his/her schedule etc.

The proposed system is intended to be deployed on iPhone 5.x and iPod 5.x, while users will be able to install and run the application in ipad but the UI will remain same as the iPhone.

1.2 Scope

The following functionalities cover the scope of the application:

- The application is only targeted for "Apple iOS iPhone only"
- Only the existing users of the notifyi web application can use the iPhone app.
- The web services required for the application shall be developed by Notifyi/Redshift.
- The management of users is not part of the app and will be an external web application.
- The iPhone application will have the following modules

SECTION 2. FUNCTIONAL DETAILS

Login:

- User can login by providing user name and password
- If the user is an authorized, then the server send a response with success status.
- If the user is not an authorized, then the server send a response with failure status.
- A user with success login can enter into the main screen of the application.
- If user forgets password, option is provided for emailing the password to user after answering a secret question.
- 'Forgot Password?' and 'Contact Us' link will redirect to the corresponding webpages.

Main Menu:

- As this is a tab bar based application, user will be able to select any of the below menu items:
    - Inbox
    - Touch Base
    - Directory
    - More (My profile, coverage calendar, mobile settings)
- On clicking any of the above items users will navigate to corresponding pages.

1. Inbox:

- The Inbox tab contains all of the messages the user has sent and received.
- The messages are displayed in the chronological order where the latest messages are displayed first.
- It allows the users to view and manage any incoming and outgoing messages.
- All items in any folder will be removed after 30 days.
- If there is any unread message in the inbox, it will be notified in the inbox icon on the bottom bar.
- Users can read a single message on clicking on that message.
- A user can navigate to any of the folders like **Sent items**, **Drafts** and **Trash** by clicking on the menu which is located on the top bar.

1.1 Sent item folder:

- When the user clicks on sent items, all send messages are displayed similar to the inbox.
- There is also an option to delete that particular sent item.

1.2 Draft folder:

- When the user clicks Draft, all of their drafts or uncompleted messages should be displayed in a paginated list similar to Inbox Folder.
- Viewing Messages from the Draft Folder, the user is directed to the partially filled out “Send a Message” form.

1.2 Trash Folder

- When the user clicks on Trash, all deleted messages are displayed similar to the inbox.

- Details will be displayed when a user selects a message from the trash folder list.
- Details page contains a “Restore” button to restore the particular message.

Inbox Details:

- On clicking on a message from the inbox, the user can read that message.
- From this screen the user can reply to the message, forward the message or delete the message.
- On clicking on the “To” header the user can see all the recipients of that message.
- On swiping from one message screen, the next message will be loaded .

compose mail:

- User will be able to compose mail by clicking on the button which is also located at the top bar.
- It contains a standardized form that provides needed information to effectively communicate information about a patient.
- On clicking on the patient information, the form will expand to receive the patient information like patient name and date of birth.

search option:

- There is a search option which will help the users to get an item quickly.

Delete option:

- And users can easily delete mails from the list.

2. Touch Base :

- Touch Base allows physicians to have ongoing group conversations about patients.
- This section is intended to allow physician to easily collaborate with a number of physicians at once about a shared patient.
- When the user clicks on Touch Base, the latest chat is displayed in the main view.
- The user can click the “Start a Discussion” button on the top to create a new chat.
- On clicking the “Drafts” button the view will be changed to the drafted messages and from there the user can come back to the discussions.

Discussion Details:

- While selecting a particular discussion from the list, we can see the details about that discussion.
- From the discussion page,user can add comments.
- The user can also stay away from that particular discussion by clicking "Remove me" button.
- And also user can easily add participants to that discussion.
- On clicking on the comments button, a text filed will come for entering comments about that discussion.

Start new discussion:

- It contains a standardized form that provides needed information to effectively communicate information about a patient.
- When the user clicks "Send", the discussion will get sent.

3. Directory :

- Directory contains list of physicians with name and speciality.
- Users can easily select a particular physician from the list and can navigate to its details page.
- There is an option to search for a particular physician from that list.
- And also an option to delete a particular physician.

Directory Details :

While clicking on a particular physician from the directory user will navigate to the details page which includes the following data.

- Name
- Practice
- Specialty
- City
- State

- Status
- Phone

There will be an options to send text message and touch bases the contact from this screen.

4. More:

There are mainly three sections comes under More menu.

- My Profile
- Coverage Calendar
- Mobile settings

4.1 My Profile:

- From the my profile menu, the user can see his/ her profile.
- However, for changing the profile the user needs to login to the web version.
- From the profile screen, the user can go back to the 'more' menu on clicking the back button.

4.2 Coverage Calendar :

- The coverage calendar will be a read only page which displays the schedule of the particular user.
- The calendar will display the schedule of that user for 2-3 weeks which can be scrolled and seen.
- On clicking a calendar entry the detailed page will appear.
- The addition or update of a calendar entry can be done only at the web side.

4.3 Mobile settings

The users mobile preferences can be set from the mobile settings page. This page will have two sections

- Mobile alert settings

  The message alerts like the vibration alert and the ring on message are set from the mobile alert settings section.

- Push Notifications

  From the push notification section we can turn the push notifications on or off.

SECTION 3. DATABASE DESIGN

INBOX:

Field Name	Data Type	Default Value
messageId	Integer 32	
MessageType	Integer 32	SentItem – 0 InboxItem – 1 DraftItem – 2 TrashItem – 3
Date	Date	
subject	String	
TextMessageBody	String	
patientFirstName	String	
patientLastName	String	
patientDOB	Date	

TOUCH BASE:

Field Name	Data Type	Default Value
discussionId	Integer 32	
discussionType	Integer 32	Draft – 0 Normal – 1
TextDiscussion	String	
patientName	String	
comments	String	

MESSAGE RECIPIENTS:

Field Name	Data Type	Default Value
UserName	String	
UserId	Integer 32	
MessageId	Integer 32	
messageType	Integer 32	SentItem – 0 InboxItem – 1

DISCUSSION PARTICIPANTS:

Field Name	Data Type	Default Value
participantName	String	
discussionId	Integer 32	

COMMENTS:

Field Name	Data Type	Default Value
discussionId	Integer 32	
commentsId	Integer 32	
comments	String	

DIRECTORY:

Field Name	Data Type	Default Value
DoctorName	String	
DoctorId	Integer 32	
practice	String	
Speciality	String	
city	String	
state	String	
status	String	
phone	String	

MY PROFILE:

Field Name	Data Type	Default Value
userName	String	
speciality	String	
hospitals	String	
practice	String	
contactInfo	String	

COVERAGE CALENDAR:

Field Name	Data Type	Default Value
date	Date	
startTime	String	
endTime	String	
title	String	
details	String	

SECTION 4. API INFORMATIONS

API INFORMATIONS (NOTIFYI-IPHONE APPLICATION)

Service response is common with all API response. Response code consists of 600 – success, 604 – failure and 605 - internal error.

1. Login :-

input : username, password, operationType

output :

```
service response
{
        response code:
                600 - success
        message:
        operationType:
}

Login Details
{
        token (userId);
}
```

2. Inbox :

input : userId, lastModifiedDate, operationType, requestCount

output : service response + senderName, recipients, subject, TextMessageBody, patientFirstName, patientLastName, patientDOB, Date, messageId , messageType(i.e. SentItem - 0, InboxItem – 1, DraftItem - 2 and TrashItem - 3), balanceCount, lastModifiedDate

3. Delete a message :

input : messageId, messageType, operationType

output: service response

4. Restore a message :

input : messageType, messageId, operationType

output: service response

5. Compose :

input : userId, lastModifiedDate, operationType, sendersName, patientFirstName, patientLastName, patientDOB, subject, TextMessageBody

output : service response + messageId

6. Touch Base :

input : userId, lastModifiedDate, operationType, requestCount

output : service response + UserName, ParticipantsName, patientName, lastUpdatedDate, TextDiscussions, discussionId, discussionType(Draft - 0 and Normal - 1), comments, commentsId, balanceCount

7. Remove me:

input : userId, operationType

output : service response + TextDiscussionMessageId

8. Add Participants:

input : userId, operationType, discussionId, participantsName, participantsId

output : service response + participantsId

9. Start a discussion:

input : userId, operationType, sendersName, participantsName,patientFirstName, patientLastName, patientDOB, TextDiscussionMessage

output : service response + discussionId, TextDiscussionMessage

10. Directory :

input : userId, lastModifiedDate, operationType, requestCount, sortOrder, searching(1 or 0) (will be null if the user need not using searching feature)

Output : service response + physicianName, physicianId, practice, speciality, city, state, status, phone, balanceCount

11. Delete physicians from Directory:

input : userId, operationType, physicianId

Output : service response + physicianId

12. My Profile :

input : userId, operationType

output : service response + userName, speciality, hospitals, practice, contactInfo,

15. Coverage Calendar :

input : userId, operationType

output : service response + date, startTime, endTime, title, details

SECTION 5. KEY DESIGN ASSUMPTIONS

- Application would be developed on iPhone SDK 5.0.
- User interface is not designed for iPad and thus user will only be able to install and run the application on iPad but with application screen size same as that of iPhone

SECTION 6. APPLICATION FRAMEWORK

iPhone SDK provides a collection of frameworks for accomplishing our development activities like UIKit, messageUI, CoreData etc. The application would be developed with the aid of these frameworks. The architectural pattern used in designing these frameworks is the Model View Controller [MVC] architecture. The advantage of using this architecture is that the presentation logic and business logic are separated from each other. This makes the software system flexible to changes and also helps in dividing responsibilities between development team. Here Model contains the business logic of the system, View renders the data in Model into a visual format for the users and Controller manages the presentation logic of the View and provides feedback to Model regarding user inputs.

SECTION 7. FUNCTIONAL ARCHITECTURE

5.1 System Architecture Overview

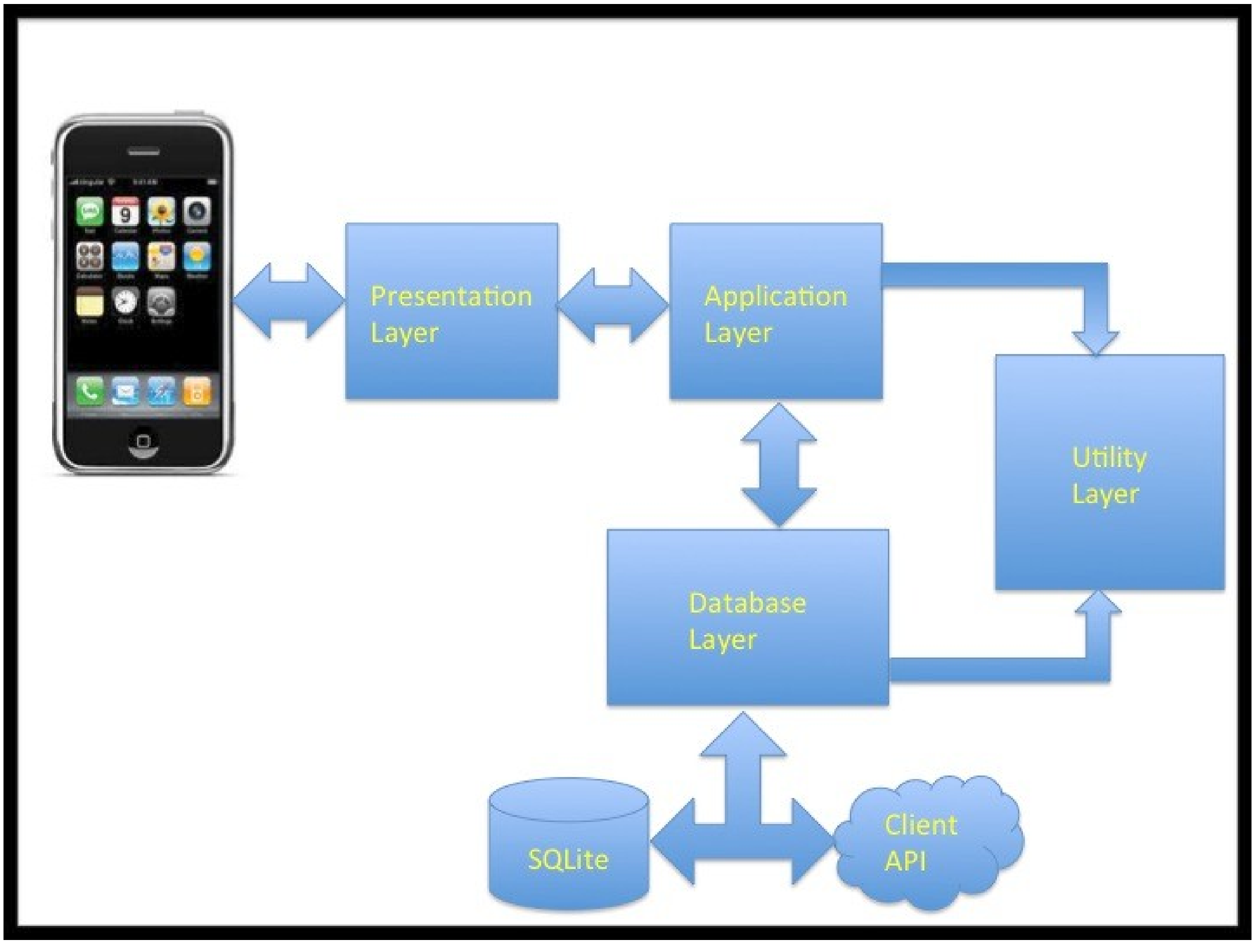

Fig. System architecture

Detailed description of each component in the architecture view:

a) **Presentation Layer:** This layer implements the functionality of *controller* in the MVC architecture. It will manage the presentation logic for the view and track user input to the system. Based on the user input, this layer will trigger appropriate application layer class for handling the messages.

b) **Application Layer:** Business logic of the application is contained in this layer. It sets the rule for manipulating the application data. This layer would form a part of the *Model* unit in the MVC architecture. The layer manages the flow of data through

Database layer. Utility layer classes support this layer.

c) **Database Layer:** This layer forms the interface of our application to the data storage resource like SQLite and client side API's [web services]. In this layer data is not manipulated but simply retrieved or saved back to their resource based on request.

d) **Utility Layer:** Utility layer implements most of the common functionalities across the layers. This helps in easy maintenance and availability of the utilities to other layers. Few examples of functionalities that can be grouped into utility layer are XML parsing, database connectivity, string formatting etc.

e) **SQLite:** It is an internal data storage resource available in iPhone. It is light weighted and can be used for the purpose of storing mostly static data or for caching frequently used data.

f) **Client API:** Since the client data is large and dynamic; we would be utilizing the API's provided by client for retrieving and storing data into client database.

SECTION 8. OTHER REQUIREMENTS

Nonfunctional requirements focus on the qualities that must be applied to design and implement the system.

Type of Requirement	Description
Reliability	The application should give an up-time of 99%.
Operational Recovery	Manual/Automatic backup facility of data should be provided. On the event of a System/Application crashes, recovery should be quick with minimum process. If possible, automate the recovery wherever possible.
Performance and Response time	Every data should load without fail and incase of error, appropriate messages should be displayed like if unable to send a prescription to pharmacy or unable to change a schedule etc.
Scalability	The application architecture should allow any number of users to be added to the user groups, for instance currently only physicians can login to the application but later clinic staffs could also be able to login to the application with minimal changes.
Security and privacy	Users should not be able to view content/data restricted by clinic or patient.
Interoperability	The application should be able to integrate and exchange data or services with other systems.

SECTION 9. DEVELOPMENT ENVIRONMENTS

Technology	**Version**
IPhone SDK	5.0+
Xcode	4.3+
Web services	SOAP

APPENDIX A: ACRONYMS AND ABBREVIATIONS

Acronym	Definition
XML	*Extended Mark-up Language*
DB	*Database*
Rx	Prescription
UI	User Interface
SDK	Software Development Kit