Elation Mobile Apps – iPhone App v1

Project Members: Jeremy Warner, Mike Malkowski, Rachel Pekarek  
Project Managers: David Springhetti and Brian Butterfield  
Date: 10/15/2012  
Innovative Systems L.L.C.

Revision History

|  |  |  |
| --- | --- | --- |
| Project Members | 6/4/2012 | Initial Version |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Table of Contents

[Table of Figures 4](#_Toc338417283)

[Introduction 5](#_Toc338417284)

[Purpose 5](#_Toc338417285)

[General Architecture 5](#_Toc338417286)

[Technologies Used 5](#_Toc338417287)

[Development Environment 6](#_Toc338417288)

[Xcode 6](#_Toc338417289)

[Eclipse 6](#_Toc338417290)

[Versions/Tortoise SVN 6](#_Toc338417291)

[GitHub 6](#_Toc338417292)

[Dataflow Diagram 7](#_Toc338417293)

[User Interface Flow (StoryBoard) 7](#_Toc338417294)

[Class Method Call Diagram 7](#_Toc338417295)

[View Details 8](#_Toc338417296)

[Sign In View 8](#_Toc338417297)

[Set Up 8](#_Toc338417298)

[Process 8](#_Toc338417299)

[Main Menu 9](#_Toc338417300)

[Set Up 9](#_Toc338417301)

[User Event Handling 9](#_Toc338417302)

[My Account 10](#_Toc338417303)

[Set Up 10](#_Toc338417304)

[User Event Handling 10](#_Toc338417305)

[Invoice 11](#_Toc338417306)

[Set Up 11](#_Toc338417307)

[User Event Handling 11](#_Toc338417308)

[Payments 12](#_Toc338417309)

[Set Up 12](#_Toc338417310)

[User Event Handling 12](#_Toc338417311)

[Log Framework 12](#_Toc338417312)

[Set Up 12](#_Toc338417313)

[Developer Use 13](#_Toc338417314)

[Unit Testing 13](#_Toc338417315)

[Appendix 15](#_Toc338417316)

[API 15](#_Toc338417317)

# Table of Figures

[Figure 5 8](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417318)

[Figure 6 8](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417319)

[Figure 7 9](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417320)

[Figure 8 9](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417321)

[Figure 11 10](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417322)

[Figure 10 11](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417323)

[Figure 9 11](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417324)

[Figure 11 12](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417325)

[Figure 12 13](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417326)

[Figure 13 13](file:///C:\Users\rachelp\Documents\GitHub\ECPMA\Documentation\ECPMA%20Design%20v1.docx#_Toc338417327)

# Introduction

## **Purpose**

This application is built on both Android and iOS systems, and the needed support systems. It will allow users to access eBill services via the mobile application.

## General Architecture

Architecture stuff.

## Technologies Used

The Software Development Kit (SDK) that is used for this project is Xcode 4, which uses the objective-C language combined with the Cocoa framework provided by Apple. eBill currently targets iOS 5 and iOS 5.1 with no support for earlier versions. Interface Builder, along with Storyboard, is used to create the UI for the project. CoreData, an Apple framework, is used for database storage/retrieval.

For documentation, the project uses Doxygen along with Graphviz for diagrams. To do unit testing, the project uses Apple’s provided SenTestingKit, which allows for unit tests to be run inside of Xcode.

To report errors, data of interest and the state of the app, eBill uses an in-house framework, known as the Log. The Log allows the developers to log different states and errors as they write code. This comes in handy when debugging customer issues.

Eclipse, API, ….

# Development Environment

## **Xcode**

Xcode is the development environment for Apple products, namely Mac OS and iOS. Objective-C 2.0 is the language that is used in this environment, along with Automatic Reference Counting (ARC). ARC is the equivalent of garbage collection, where object reference counting such as retains and releases are done automatically by the compiler. This allows for quicker development and less debugging to find zombies (over released objects).

Interface Builder allows the developers to create User Interfaces without writing any code. This assists in keeping with the MVC architecture in that the view is completely separate from the rest of the project and is reusable. Storyboard, in addition to creating views, also allows a developer to show the flow of the UI. The current Storyboard configuration can be seen in Figure 3.

## Eclipse

Eclipse was used for the Android Development.

## Versions/Tortoise SVN

SVN was used in the development of the eBill app in order to allow a small team of programmers to work on the project simultaneously. One repository was used to store modified versions of the project where other team members could merge and update working copies throughout development.

GitHub

GitHub was used to store public documents

# Dataflow Diagram

Figure x

# User Interface Flow (StoryBoard)

Figure x

# Class Method Call Diagram

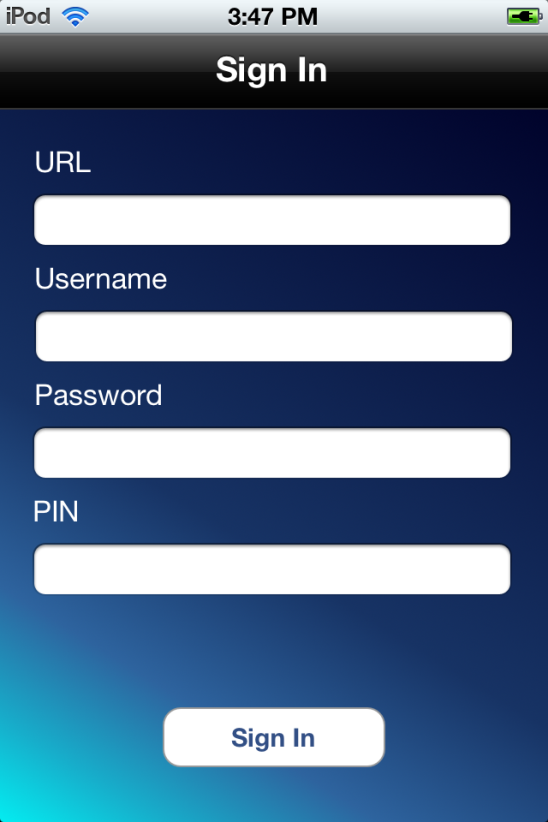
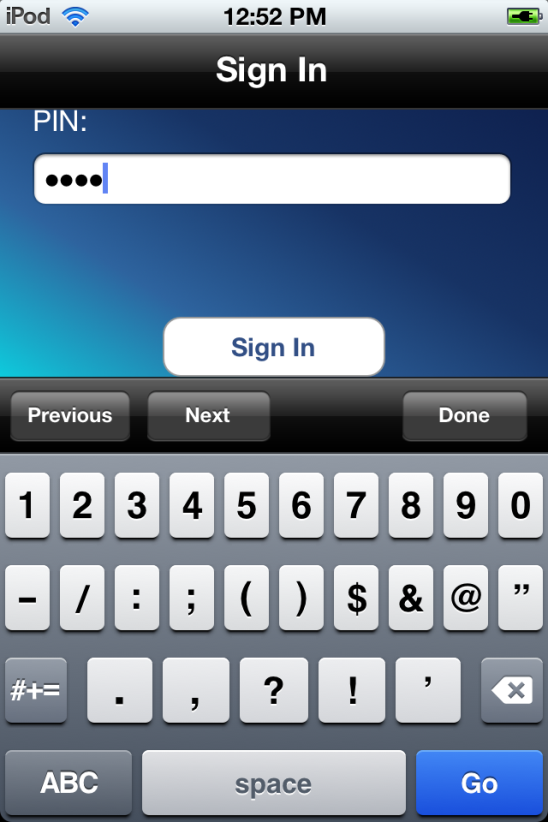
Figure x

# View Details

## [Sign In View](\\\\ISRCDC\\Shared\\Projects\\Apple iOS\\ManageMyVM\\Documentation\\VoiceMail.docset\\html\\interface_modal_login_view_controller.html)

### Set Up

* Displays user information prompts dynamically



Figure

Figure

### Process

* Sign In Button pressed
  + Verify no fields are empty
  + Connect to the domain
  + Login the user
  + Send APN token to service
  + Once logged in, go to Inbox View
* Text field active
  + Toolbar displayed on top of keyboard
  + Previous button-tap event
    - Makes the previous text field active if not currently of the first field
  + Next button-tap event
    - Makes the next text field active if not currently on the last field
  + Done button-tap event
    - Dismisses keyboard
  + Go button-tap event
    - Verify no fields are empty
    - Connect to the domain
    - Login the user
    - Send APN token to service
    - Once logged in, go to Inbox View

## [Main](file:///\\ISRCDC\Shared\Projects\Apple%20iOS\ManageMyVM\Documentation\VoiceMail.docset\html\interface_voice_mail_view_controller.html) Menu

### Set Up

* Static cells display menu options

Figure

Figure



Edit Mode

### User Event Handling

* My Account Cell-tap Event
* Display My Account page
* Invoice Cell-tap Event

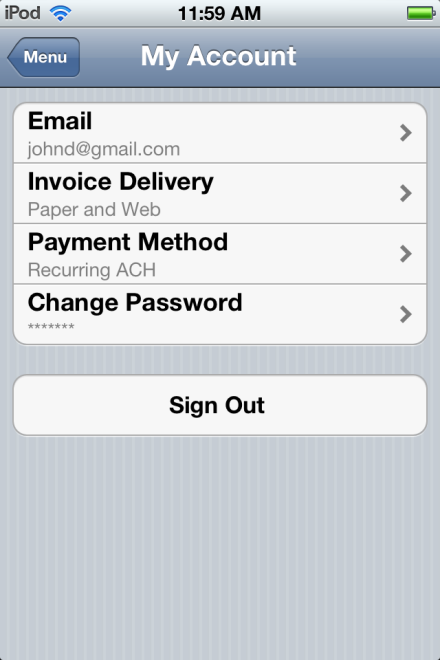
- Display Invoice page

* Payments Cell-tap Event
* Display Payments page
* Usage Summary Cell-tap Event
* Display Usage Summary page
* Support Cell-tap Event
* Display Support page

## [My Account](\\\\ISRCDC\\Shared\\Projects\\Apple iOS\\ManageMyVM\\Documentation\\VoiceMail.docset\\html\\interface_settings_view_controller.html)

### Set Up

* Display URL, Username, PIN as read-only fields



Figure

### User Event Handling

* Email Cell-tap Event
  + Displays E-mail page
  + E-mail page allows for adding, updating, and managing addresses
* Invoice Delivery Cell-tap Event
  + Allows user to change Invoice Delivery settings
* Payment Method Cell-tap Event

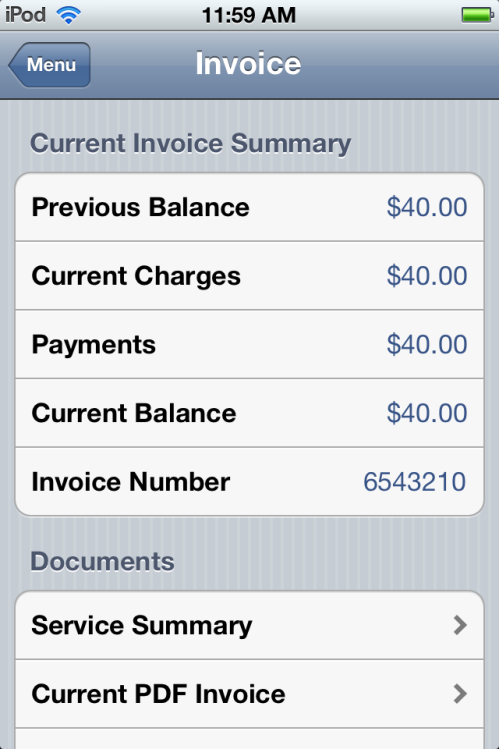
- Allows user to change payment method settings

* Change Password Cell-tap Event
  + Displays Password page
  + Password page allows user to change their password
* Sign Out Button-tap Event
  + Signs user out of eBill application

## [Invoice](file:///\\ISRCDC\Shared\Projects\Apple%20iOS\ManageMyVM\Documentation\VoiceMail.docset\html\interface_detail_view_controller.html)

### Set Up

* Display Current Invoice and other Invoice viewing options



Figure

Figure

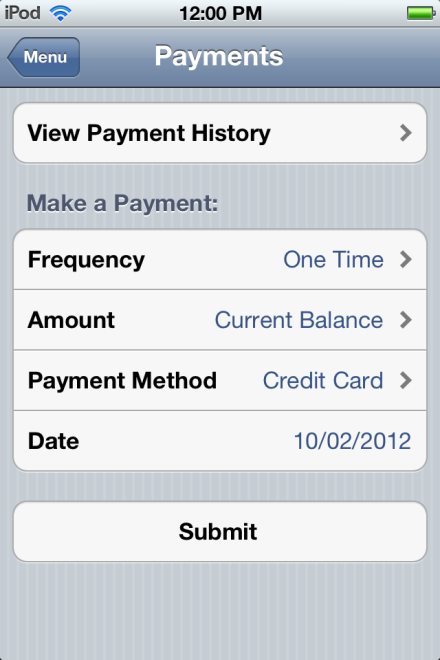
### User Event Handling

* Service Summary Cell-tap Event
  + Display current Service Summary
* Current PDF Invoice Cell-tap Event
  + Display current Invoice as a PDF file
* Past PDF Invoices Cell-tap Event
  + Display list of Past Invoices to select

## [Payments](\\\\ISRCDC\\Shared\\Projects\\Apple iOS\\ManageMyVM\\Documentation\\VoiceMail.docset\\html\\interface_settings_view_controller.html)

### Set Up

* Display payment options



Figure

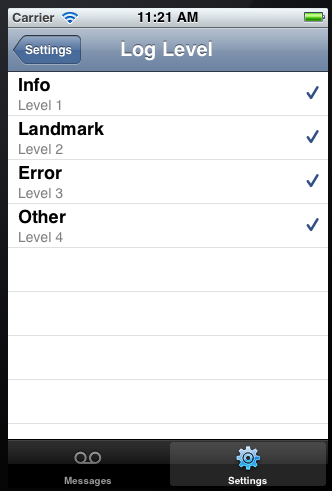
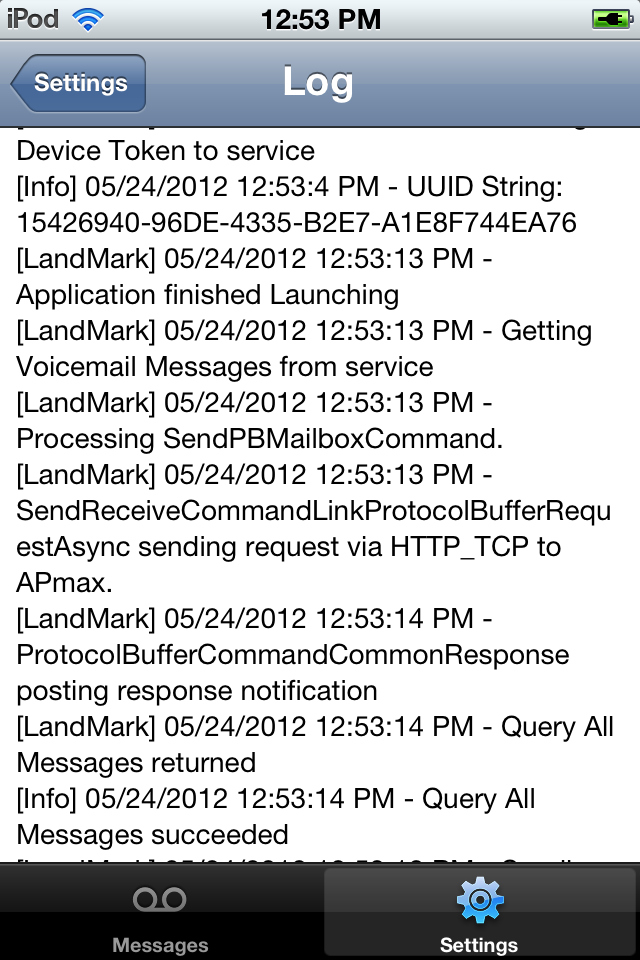
### User Event Handling

* View Payment History Button-tap Event
  + Displays Payment History
* Frequency, Amount, Payment Cell-tap Events
  + Allows user to change these fields
* Submit Button-tap Event
  + Prompts user to confirm payment
  + Submits payment if ‘OK’ pressed, cancels if ‘Cancel’ pressed

## 

## [Log Framework](file:///\\ISRCDC\Shared\Projects\Apple%20iOS\ManageMyVM\Documentation\VoiceMail.docset\html\interface_log.html)

### Set Up

* Display concatenated logs in a text-view
* Read-only for user

Figure

Figure

### Developer Use

* Log method
* Optional force save-to database parameter
* Used to track what the app is doing
* Hidden from the user
* Log Level vs. Log Type
* Log level set at beginning of program
* Log type used throughout program each time log is created

# Unit Testing

SenTestingKit, Android testing

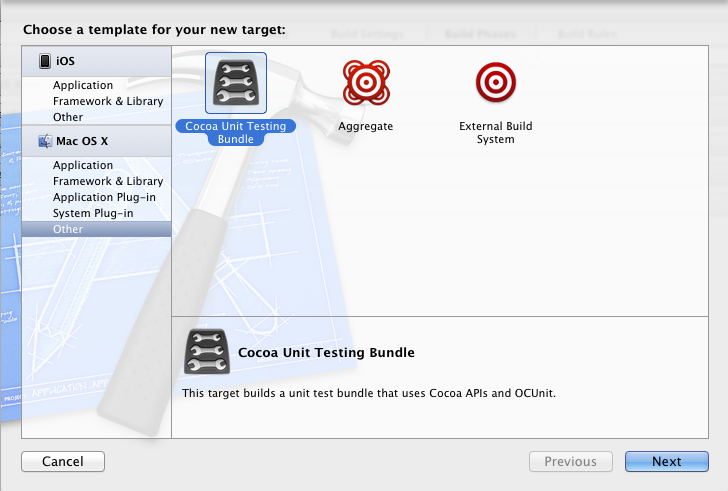


Figure x

To do setup before the tests are ran, -(void)setup must be overridden, as well as –(void)tearDown, if additional code is required to clean up after all the tests are done. The tests are run in no particular order. This makes sure that the developer keeps the nature of unit testing in mind as the unit tests are written. Each unit test should test a different feature of code, and should not depend on previous tests.

# Appendix

## API

API