

IPHONE MOBILE DASHBOARDS APPLICATION DEVELOPMENT PLAN

Corporate Applications Development,
January, 2011

TABLE OF CONTENTS

1	PROJECT INTRODUCTION	2
1.1	PURPOSE OF THE PROJECT	2
1.2	PROJECT GOALS	2
1.3	TERMS AND ACRONYMS	2
1.3.1	Tags used in this application's plists	3
2	ACCESS AND SECURITY	4
3	APPLICATION DESIGN	5
3.1	LOGICAL DESIGN	5
3.2	PHYSICAL DESIGN	6
3.3	USER PROCESS FLOW	7
3.4	CLASS MAP PROCESS FLOW.....	9
3.5	USER SCENARIOS.....	10
4	DATA DESIGN	10
4.1	DATABASE SCHEMA	10
4.2	DATA INTERFACE	10
4.3	MOBILE DEVICE MEMORY USAGE	10

1 Project Introduction

1.1 Purpose of the Project

Enable Organization Dashboard users, such as Service (Group Company) managers and Division (Practice) managers to view up-to-date financial performance and sales data from outside the network firewall.

Enable Functional Dashboard users, such as Project Managers to monitor projects and Project Financial Managers to monitor contracts assigned to them using only their telephones.

Because this application, like its desktop computer counterpart, allows only read access to data, it suits mobile device technology. . User interaction consists of navigation, selection, display rotation and scrolling between views, all perfectly suited to leverage touch screen mobile devices.

Demonstrating mobile dashboards to clients provides competitive advantage to Marketing personnel.

This project results in a library of custom iPhone programming that can be applied to many PBS&J proprietary applications.

1.2 Project Goals

Information Technology expectations:

- Leverages speed of iPhone 3gs and 4 and iPhone ability to deliver graphs and color-coded progress indicators (KPI)
- Proof of concept to demonstrate advantages of mobile workstations when distant from the physical network.
- Provides competitive advantage showing customers PBS&J state-of-the-art tools
- First of a large family of iPhone applications; Extensible to iPad and Droid

End-user expectations:

Remotely access data inside the firewall. Secure financial data from external internet access. .

Dashboard performs as it does within SharePoint and the e*net.

1.3 Terms and Acronyms

Term	Explanation
AD	Active Directory-Stores and authenticates access attributes (username, password, organization membership, group membership) for all PBSJ network users.
Ldap	The dictionary format used by AD.
DWH	Data Warehouse
Dashboard	Selection of graphs and tables summarizing state of an organization at a specified point in time.
Apple iPhone	Smart, mobile telephone provides email connectivity and runs web-enabled applications. Trademark held by Apple Computers, Inc.
.xib, .nib	Collections of Interface builder objects such as text fields, data tables, sliders, and pop-up menus.
h, m, pch	Header, compiled header and methods files describing classes

1.3.1 Tags used in this application's plists

Property lists organize data into named values and lists of values

Foundation class	CoreFoundation type	XML Tag	Storage format
NSString	CFString	<string>	UTF-8 encoded string
NSNumber	CFNumber	<real>, <integer>	Decimal string
NSNumber	CFBoolean	<true />, or <false />	No data (tag only)
NSDate	CFDate	<date>	ISO 8601 formatted string
NSData	CFData	<data>	Base64 encoded data
NSArray	CFArray	<array>	Can contain any number of child elements
NSDictionary	CFDictionary	<dict>	Alternating <key> tags and plist element tags

2 Access and Security

Apple offers JanisPlus API, which opens a secure VPN connection to our network behind the firewall. Our application launches web services.

Logon

1. When the user logs into Mobile Dashboards on the iPhone handset, the iPhone sends the login user id and password via LoginWebServices.
2. The LoginWebServices request initiates DWH User.asmx .cs. User.asmx.cs uses LdapHelper.cs to search for the User in AD

Membership

User.asmx.cs also checks AD for user's Group membership.

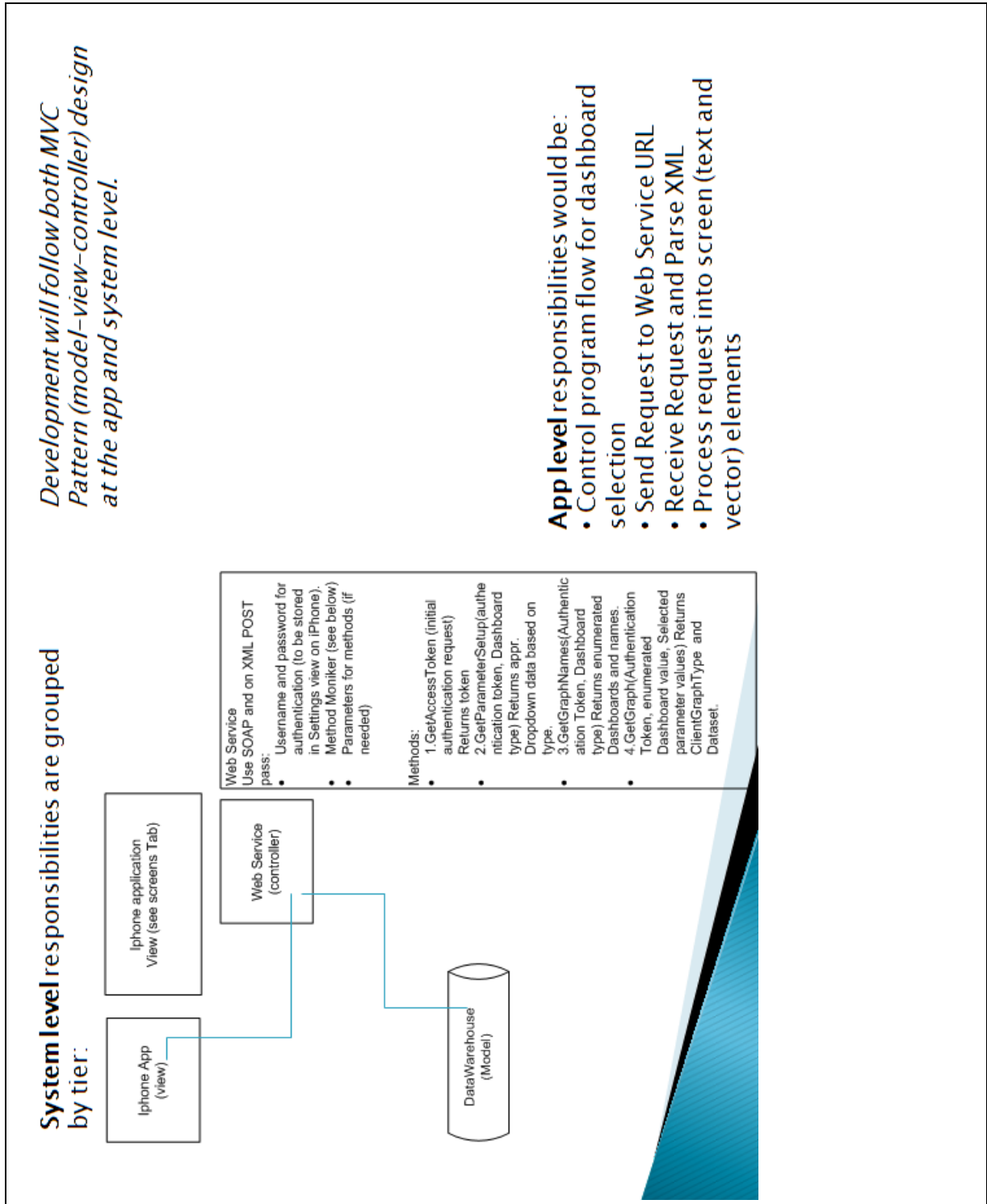
Access Level

Equipped with the user's job title and group, the remaining Web Services obtain and return data for all dashboard s that user is authorized to see. The mobile device stores the data it receives in an area called Core Data.

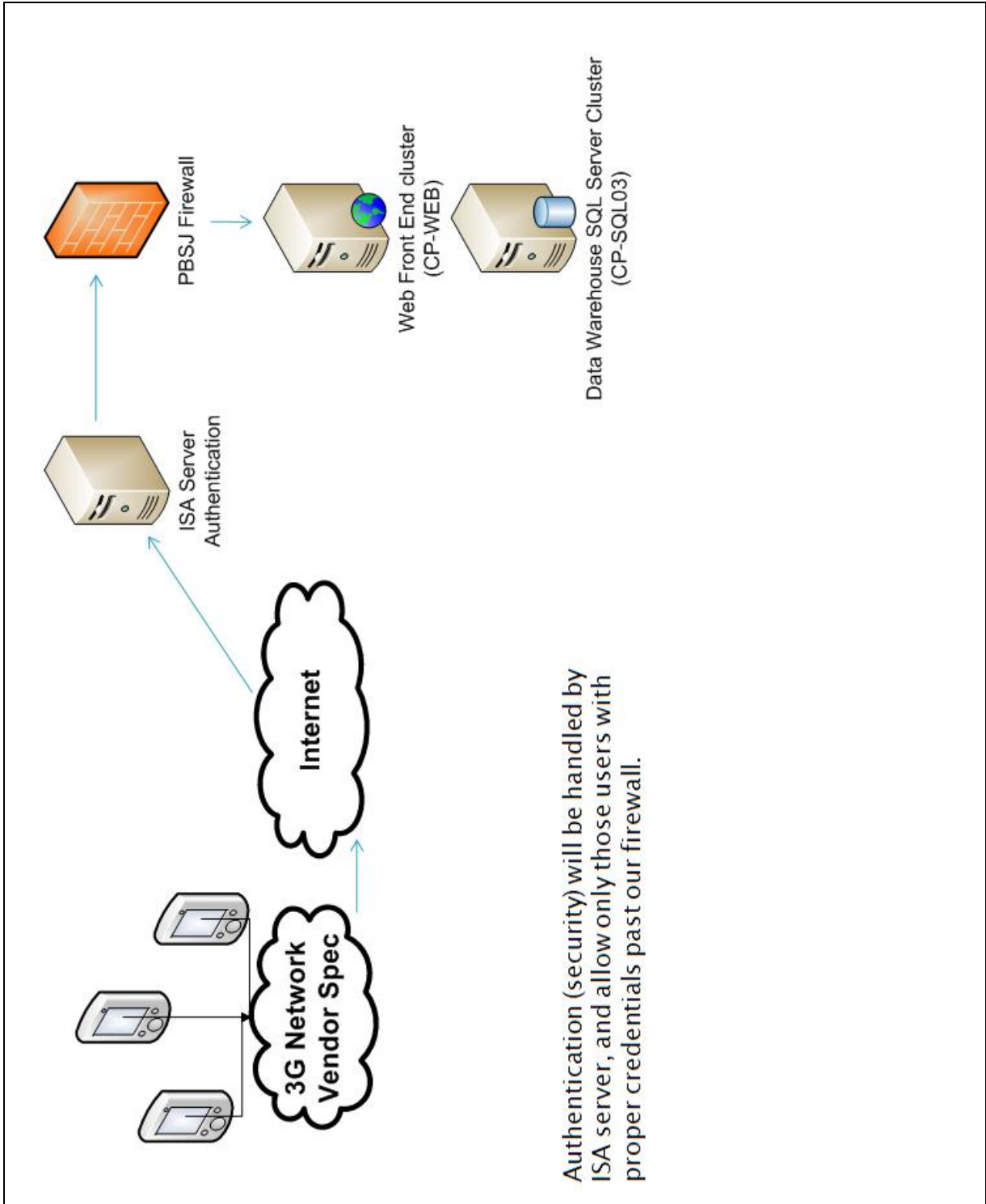
The user can break the vpn connection. Subsequent View Controllers will display Core Data.

3 Application Design

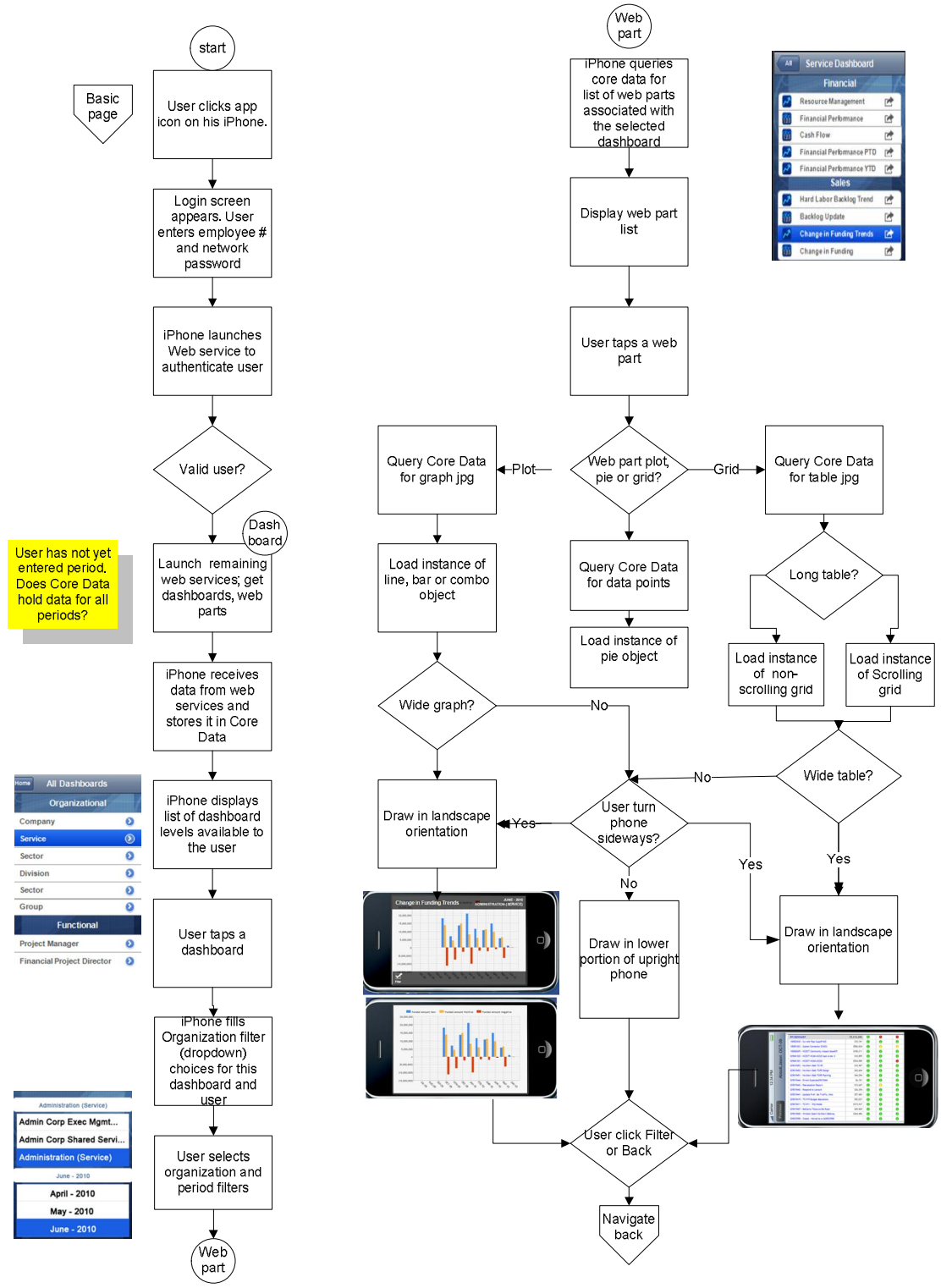
3.1 Logical Design



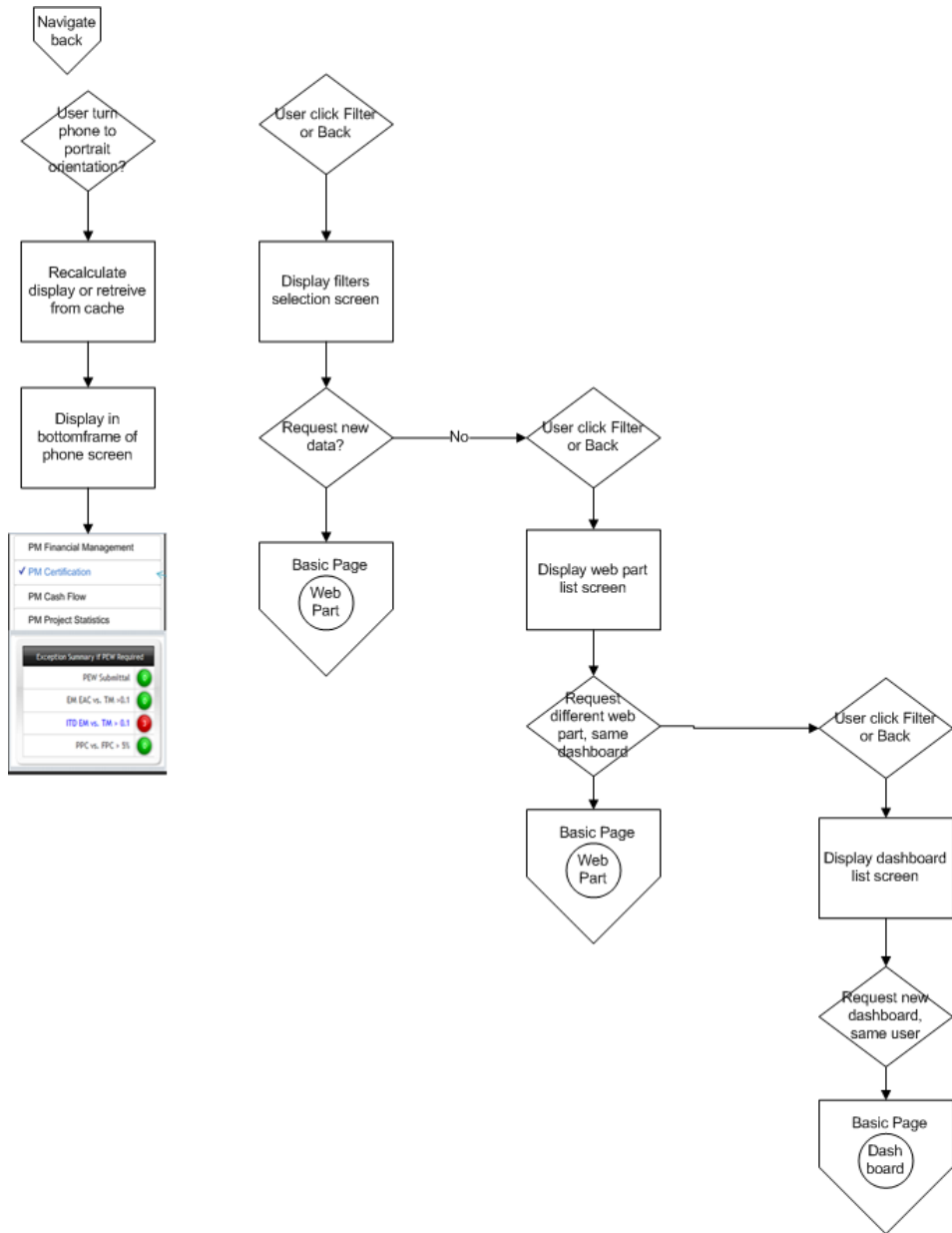
3.2 Physical Design



3.3 User Process Flow



Process Flow Continued



3.5 User Scenarios

Same as SharePoint Dashboard Users.
User must have iPhone.

4 Data Design

4.1 Database Schema

Refer to Data Warehouse dashboards

4.2 Data Interface

Refer to [Storyboard](#).

4.3 Mobile Device Memory Usage

The navigation based Mobile Dashboards application has many views on stack and keeps traversing between them as needed. .

Every View Controller must include the following methods:

`WillAppear` is called before the view is loaded into the memory to perform customization on the view such as: status bar, setting orientation, navigation bar and so forth. Called every time the view is presented.

`viewDidLoad` called after the view has been allocated space in the memory to perform initialization for a view loaded from the nib. Called only once when the view is first loaded on the stack.

`didReceiveMemoryWarning` - The iPhone OS limits an application to a small amount of memory. As the application nears the memory limit the OS, it calls the `didReceiveMemoryWarning` method which in turn tries to reload all the views on navigation stack, and thus calling the `viewDidLoad` for each view again,
Releases cached data if no longer in use.

`viewDidUnload` releases any retained subviews

`dealloc`