Detailed Design Document

SlackList, a user list app for Windows 8.1 and Windows Phone 8.1

Contents

[Revisions 1](#_Toc428124993)

[Description 2](#_Toc428124994)

[Initial requirements 2](#_Toc428124995)

[Functional Specs 2](#_Toc428124996)

[Initial Concept 2](#_Toc428124997)

[Getting a List 2](#_Toc428124998)

[The Menu 3](#_Toc428124999)

[The Search 3](#_Toc428125000)

[The Member 4](#_Toc428125001)

[The Update 4](#_Toc428125002)

[Viewing a Profile 4](#_Toc428125003)

[Windows Version 5](#_Toc428125004)

[Technical Specs 6](#_Toc428125005)

[Overall Design 6](#_Toc428125006)

[Colors 6](#_Toc428125007)

[Font 7](#_Toc428125008)

[System Requirements 7](#_Toc428125009)

[Slack Integration 7](#_Toc428125010)

[Users.list 7](#_Toc428125011)

[User.getPresence 7](#_Toc428125012)

[Unit Testing 7](#_Toc428125013)

[Known Issues 7](#_Toc428125014)

[Future Enhancements 8](#_Toc428125015)

# Revisions

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Name | Comments | Date |
| 1.0 | Erik Umenhofer | Initial Document Creation | 8.23.15 |
|  |  |  |  |
|  |  |  |  |

# Description

This document will catalog the major design choices used for the application. It will detail functional and technical specifications.

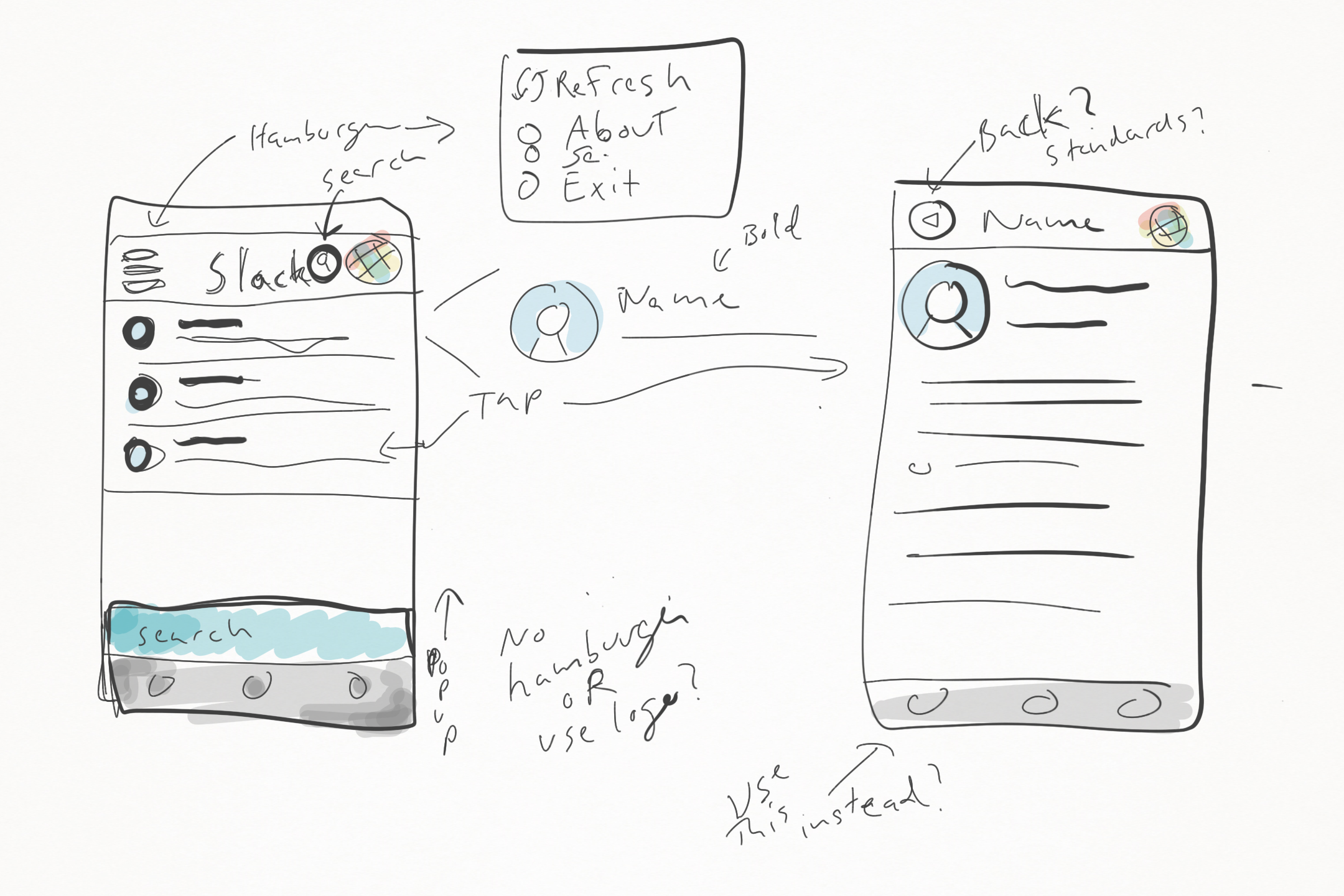
## Initial requirements

1. Create a Windows Phone 8.1 Compatible app
2. Display a list of Users
3. Display a Drill Down Profile Page
4. Must function correctly offline
5. Must function correctly from tombstone

# Functional Specs

## Initial Concept

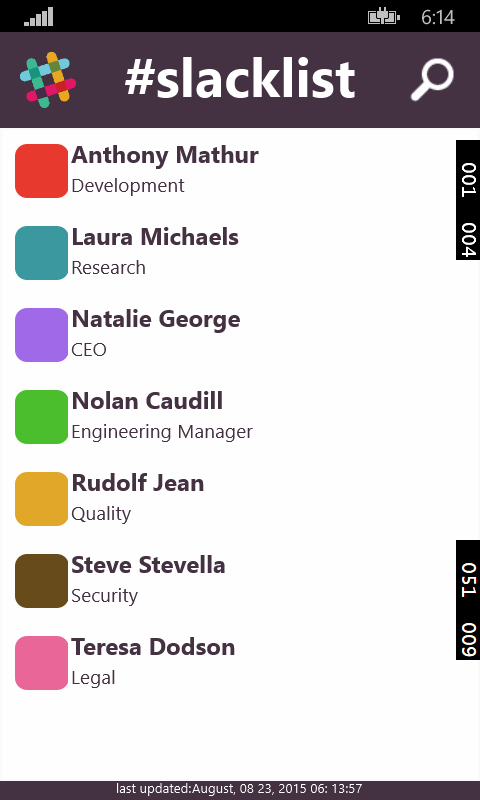
These drawings detail the original concept.



Initial Concept

## Getting a List

The main purpose of the application is to retrieve a list of users. The main screen of the application will load a list of users from the Slack api.



The Main Screen

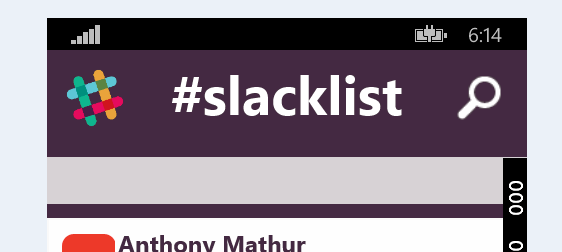
The diagram above show the main page. There are four main areas to call out.

### The Menu

The Top Left logo will refresh the data. In the future this might open a menu.

### The Search

The Top Right icon will open a search feature. It will refine the list by letters. It is a simple search. Tapping the mag glass will toggle the search.



Search feature

### The Member

In the center is a list of members. If there is a network issue, the list will be populated with data from last use. You can still click on a user with the network down, but the presence information will not function.

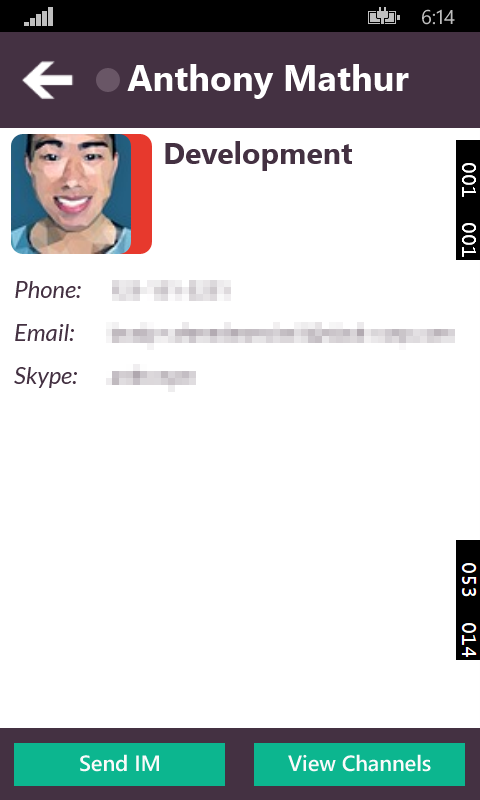
Click the member to view a profile.

### The Update

This date will update each time the application successfully retrieves data.

## Viewing a Profile

The profile page seen below is the main details page for the user.

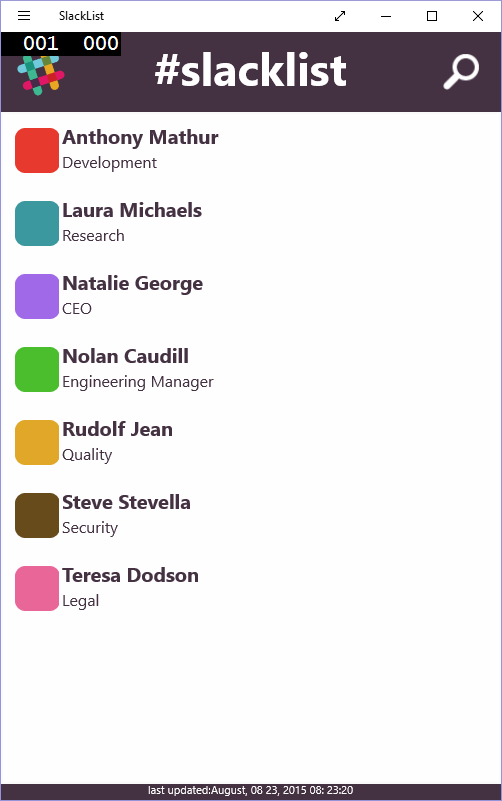


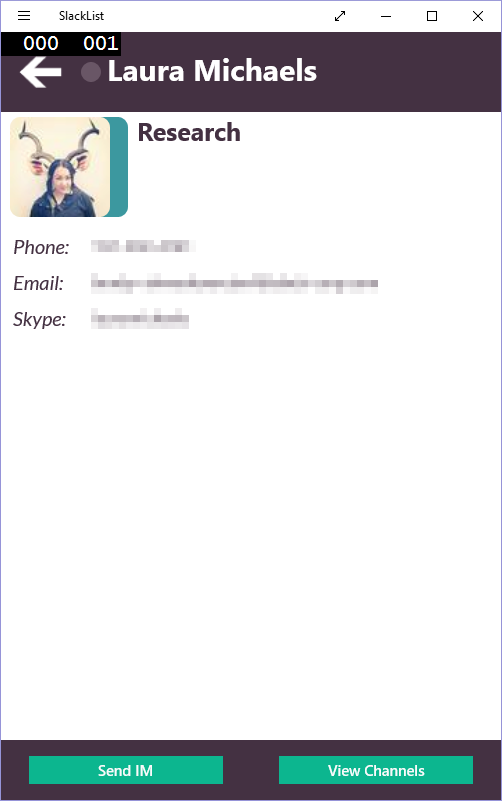
The main features of this page include

* Name at the top, name of profile selected.
* Presence indicator next to name. This small indicator will let you know if they are active.
* Picture with color band. This color is saved in the member’s profile.
* User data. Phone,
* Buttons at the bottom for future use. These could eventually load other portions of the application.
* Back Button (so you can go back in both platforms from one control.

## Windows Version

Here are some screen shots of the desktop version. Note the similar layout, but font size issues. This would be addressed with a multi UI approach instead of a unified approach.





# Technical Specs

## Overall Design

Key Points

* Universal Windows App 8.1
* Shared Project used extensively

The major architectural choices on the platform revolved around creating an application that was easily maintained and worked similarly for both the Windows and WP platforms. The UIs have been moved to the Shared Project section of the solution. This restricts many of the standard apis in the .Net framework as they have to span both phone and desktop operating systems.

### Colors

* 555459 - black
* 443142 - plurple dark
* fafafa - white
* ffffff – off white
* 0cb68f - blue1
* e80d68 - red
* eba438 - yellow
* 5ac7d7 - blue2
* 7f9947 - green (light)
* 695666 - unlit purple

### Font

* Lota Bold, Black and Reg

## System Requirements

* Window Phone 8.1
* Windows 8.1
* Internet Connectivity
* Windows Phone with 512 megs ram 480x800 screen min.
* Windows Desktop with 4 gigs ram, 800x640 display min.

## Slack Integration

Main site is located here: <https://api.slack.com/>

The two main methods used in the api are the users.list and user.getPresence calls. These give us access to all the user information we need to display as well as the presence feature.

### Users.list

https://api.slack.com/methods/users.list

### User.getPresence

https://api.slack.com/methods/users.getPresence

## Unit Testing

Unit testing does not work on shared code. Shared code is a representation of different platforms in a single file, but there isn’t a good way to share the assembly with another project because the platform tools split up the assembly at compile time.

The solution is to use more platform specific code and avoid shared.

## Tools

* Visual Studio Community 2015
* Photoshop 2015 CC for images

# Known Issues

Check all TODO documented in the code.

Back navigation is not working as the application was built with 100% Shared UIs. The solution is breaking up each UI per platform. This results is less shared code, but more control. The trade-off is worth the effort as the WP experience suffers when back navigation is broken. The App.xaml.cs is located in the shared project and does not have access to the same framework as the phone, and does not understand back navigation. This is why the code should be split.

The profile page doesn’t store user images, so if the network is down and the profile page is loaded, it will not display the profile image.

Profile page image needs a default in case of slow internet or no connection.

# Future Enhancements

* Hub integration, use Windows 10 features.
* Refactor HTTP Code. This needs to be cleaned up and moved around so that it can take use of reuse and clean exits.
* Add authed user login, add a authenticated user framework.
* Update to JSON.net Serializer. Shared code can’t access the same assembly from both platforms, the code must be split out or find a version of JSON.net that works as shared.
* Add Toast Notify when presence changes. This would just be to add some flair.
* Add animations. Add story boards for all the icon and list movements.
* Add some sort of QR system or notify system, so that you can take the conversation on the go. If at you desktop you can move it to your phone for easy access without having to dig into your app and search.
* Add sorting to list
* Look to third party controls such as Telerik virtualized lists etc.