Voalte UI/UX

Importance of UI/UX Slide

Summary: As a **consumer and product driven** company, we should push to deliver exceptional user experience, gain **competitive advantage** over our competitors, and **monetize** over value-added features. Focusing in user experience will have long lasting effect for us as a company moving forward.
So what exactly can an exceptional UI/UX do for us? **It allows us for….**

Ease of new client acquisition:
- Reflect the unique and exceptional Voalte experience.
- Allowing our app to reflect consistent branding and marketing goals.

Create truly value-added features:
- Adam mentioned Voalte is **entering** this rapid growth stage with a **maturing** mobile healthcare industry. We will have to find ways to distinguish ourselves, become the **industry leader** NOT the follower. (With **design** being an important aspect of it, and o**f course engineering**, **support** and other **function** of the company that follows)

At the end of the day, we want to

Increase ROI, for the company with:
- Decrease the need for redundant tech support
- **Increase user satisfaction rate**
  - Measure by the decrease in tech support tickets
  - Longer and more "active" users in hospital setting
- Decrease the amount of training time for end users and the amount of onsite ICs, VMAAs…
- Decrease the deployment cost and decrease deployment time.

"**Good design at the front-end** really **suggests** that everything is in **order** at the **back-end**, whether or not that is the case." - The Value of Good Design

There are also **plenty of researches conducted** that have data that states large number of **users assess the credibility** of sites and application **based on the appeal of the overall visual**.

As a reminder, this is what we promised our customers, "**we don't settle for being 'good enough**' ". And if I draw some parallels, here's **Google's number 1 philosophy**.

*With that we will **move on** to see **how usability is evaluated** and how Voalte Platform is doing against these evaluations.*

Using Usability Evaluation

Summary: With the ***importance of UI/UX established***, let's briefly walk over how usability and user experience is evaluated. **Human eyes are extremely sensitive**, and that's what makes design hard to grasp.

A lot of people think "UI is just a matter of opinions and colors".

That's not exactly true, there are **specific ways user experienc**e can be **evaluated objectively**.

One of the mos**t widely adopted metrics** in the industry is the **Nielsen's usability heuristics**.

Here you can find **specific and objective qualification** on if we are delivering good user experience. You can even find these heuristic guidelines being mapped into the **two of the most popular client ecosystems**. IOS and Android. These guidelines were **set up to maintain consistency between different apps** within the same ecosystem.

Having this **consistency is extremely important in helping users to adopt and familiarize expected user behavior**. Can you imagine having users to relearn a new set of rules every time he or she download a new app ? That would be disastrous.

So here you can see how **detailed the Nielsen's usability heuristics are**. We do well in most of these evaluations and I've **bolded** some specific points where we could use some attention. **We will go through these in more detail in the next couple slides as we walk through a typical user flow within our Voalte system**.

Any questions on any of these evaluation metrics before we move on? I know it's a lot of information in one slide.

Nielsen's Usability Heuristics

- Visibility of system status
  - Are you currently online? Offline? Can you make calls/text/ receive alarms? Why or why not?
- Match between system and the real world
  - Word choice within app/refrain from using system oriented language and design
    - We're doing well with using terminology like: Unit/Building/Role BUT

- User control and freedom
    - “undo/back” buttons.
- Consistency & standards
- Error prevention
    - Our [Group Message] & [Log Out] Warning is a good example
- Recognition rather than recall
    - “I know what I need to do here”
        - “How do I log out”? “How do I dismiss all alarms?” “How do I…”
- Flexibility and efficiency of use
    - Allow different types of user(advanced/average) users to do more/less
- Aesthetic and minimalist design
    - Every irrelevant/rarely needed info competes with relevant info.
- Help users recognize, diagnose, and recover from errors.
    - Example: Any re-connection issues/error up to a certain amount of time (30 secs?), give AlertView allowing users to send helpDesk a ticket and [Relaunch] the app. Give PROPER & USEFUL messages.
    - DONT TELL USERS THAT SOMETHING IS BROKEN && CAN’T BE FIXED!!
- Help and Documentation

Apple’s iOS Human Interface Guidelines

- Aesthetic Integrity
    - - How well an app’s appearance and behavior integrate with its function
- Consistency
    - A consistent app implements familiar standards and paradigms by using system-provided interface elements, well-known icons, standard text styles, and uniform terminology
- Direct Manipulation
    - The direct manipulation of onscreen content engages people and facilitates understanding
- Feedback
    - Feedback acknowledges actions and shows results to keep people informed.
- Metaphors
    - People learn more quickly when an app’s virtual objects and actions are metaphors for familiar experiences—whether rooted in the real or digital world.
- User Control
    - Throughout iOS, people—not apps—are in control. An app can suggest a course of action or warn about dangerous consequences, but it’s usually a mistake for the app to take over the decision-making.

Google’s Material Design & Usability

- Principle: clear,robust, and specific
    - Design with clear layouts with distinct calls to action
    - Design to accommodate a variety of users
        - Short attention span, give users confidence in knowing where they're in your app, understand what's important, include accessibility for disabled people.
- Color & Contrast
    - Use contrasts ratio to ensure the readability of your app.
- Sound
- Motion
- Layout
- Writing
- Hierarchy & Focus
- Implementation

Current U/X

Consistency of the overall Voalte System

FYI: Voalte pink in MenuView = #FF0075
Summary:
Analyzing Voalte from a **higher level overview**, our current system is feature rich and densely packaged. However, anyone who has had experience on-site would be able to **recall incidents** where our end users **didn't realized certain features existed**. Leading them into **frustration** and **negative** experience with the Voalte system. If you remembered the couple bullets I highlighted a slides back, to improve the current UX , we essentially need to find
1**)Consistency & standards,**

bring
2)Aesthetic & minimalist design

3)help users recognize,diagnose, and recover from errors and
4).allow users to recognize an action rather than having them recalled an expectrd action.

We have a great **feature**s, it's a shame if we don't see more users actively using them.

Without going into specifics just yet, and just by looking at this slide right here, you will realize our UI is fairly **inconsistent** as it **deviates** and **follows iOS standard practices** from screen to screen. Our system also contains **5+ colors** and different sizes of fronts to **convey** very **specific emphasises** on different screens. Referring back to the usability heuristic evaluations, this is **not only distracting and inconsistent**, it subconsciously results in **extra**

cognitive load for our users to distinguish from one action to another. Imagine nurses curious about why the color just all the sudden changed, or **why certain things aren't in a certain place.** That's the **discrepancy we want to solve.**
[off-script] talk about inconsistencies on the screen, and distracting colors.

Standardize to iOS UI Practices

There are some **straightforward** changes in the UI that can easily clean the entire interface up:

1) Standardize to latest IOS practices,
2) Limit the amount of colors(valid and consistent) used in the system
3) Have a consistent size of fonts/styles/table cell sizes/icon sizes
4) Standardize the location/font styles/icons in the UINavBar

With that, let's walk through a typical user flow so we can analyze our system in more detail.

1.Onboarding (Signing-in process)

Summary: **Traditionally** for consumer facing products(apps), the **onboarding process refers to initial app launch** where it provides an end-user training before a user is finally landed inside the system.
Understanding our unique niche of consumers, this might not be completely applicable. However, upon looking at these initial screens, we still have **some room** for "**design fun**". We could **guide** our end-user through this **signing** in process while providing a **brief overview** of how our app work.
(this in the end would **leverage for additional trainings** needed from our ICs/Support team) !!!
Aside from the **educational purpose of this onboarding process, these screens are also golden opportunities to express VOALTE as a brand**.

1. Note: ***Inconsistency*** in tapping **Continue** and **Done**.

Main Directory View

Summary: This is **first screen** a user sees after the initial onboarding process. **At first glance**, this is a screen with large table row sizes and easy to read.

If we pull up the **default** [Contacts] UI from the **IOS system**, you'd immediately notice the scroll to **filter feature** that cease to exist in our system. There are also other little things and **minor criticisms**, but **let's move forward** to the **core functions** of this app so we can see what **inconsistencies exist from screen to screen. .**

1. Default icons doesn't "pop"/stand out
2. Awkward grey divider.

Alarm/Text/Call Non-Empty Views

Summary: **So as a nurse**, or hospital staff. I'm finally here in the **core features of Voalt**e. I'm supposedly excited because Voalte is suppose to make **my life better with mobile healthcare solution** right? However, just from navigating quickly between the core feature screens, you can't help but **feel** how the different screens **flickers in front of you**.

You **can't pin-point what's exactly "wrong", but it feels awkward to navigate around.**

So let's take some time and analyze what I mean by "**flickering"**.
So notice:

1. Differences in Table row sizes due to differences in **font size** and differences in **icon presenting styles.**
2. **Intruding banners**.
3. Difference in the **presentation of icons.**
    a. Inverted colors vs no border design.

So these represent what I was saying about the inconsistencies that exist in our system.

Potentially missed Features

Like I mentioned previously, **one of the major criticism** for Voalte Platform is that we have a feature rich system, but some of them seem to be "**hiding**" from the end users. Here you will see a couple examples.

User Profile View

In the user profile screen, I**'m expected to know** what my role, team, and **all the information** I inputted **during my shift selection** process. However, I landed on this page. It has my basic

information, now what? We have so much empty space, yet users are still expected to tap on the “my shift” tab. **Referring back to the usability heuristics, this requires users to “recall” an action rather than intuitively “recognize” an expected action.**

And because we want Voalte to look identical on both ios and Android, this **specific gesture(scroll from below)** is almost specific to IOS, which require Android users to **learn** this **expected** gesture.

Detailed Directory View

Similar to the user profile view, detailed directory view has a similar issue. **The viewing by Name/ Role/ Room feature** is basically hidden from the user. I think it **took me a mont**h myself to “**uncover**” this feature. And it wasn't from “playing around with the app, it was from the **code**….
I know I didn't went through any end user trainings but as **we go live with bigger** hospitals and with more phones and end users, it's almost **unavoidable** for some **users to never receive any training** from our ICs. (Ex: Cedars)

It would be **interesting** to see the **actual percentage** of users actively using this features.

Messaging View

The messaging interface is fairly straightforward and follows default IOS UI designs. With interesting features like: **1)Sending Priority messages** and sending
2)quick messages, it would be interesting to ask end-users some opinions about the presentation of these features, and

how many of our users opt in just to type the message, instead of routing through “quick messages”. Or do they know what priority messages or the double exclamation points actually does…

With **some more research and data analysis**, I'm sure ICs and tech support bffs will have a **list of features** to add that are often missed by users when they first encounter our system.

With that being said, **let’s move forward** and see **how we can highlight features** and **provide some in-app educational tutorials** through using **empty state.**

Alarm/Text/Call Empty-State Views Slide

Summary: **Empty state** is probably the most **overlooked** aspect of **ux design**, and it could be the **most important.**

“A useful empty state t**ells you *what* it’s for, *why* you’re seeing it, and *how* you can fill it up**…. (what, why and how) is the formula for re-engaging your user, but also keep in mind that it’s the bare minimum requirement for an empty state.”

- *Benjamin Brandall*

Consider redesigning our **empty states on different screens that tells the user what they should be expecting**. It is an extremely helpful for users to learn about our system with these temporary empty screens. On the slide here, you can see some interesting empty-state designs. Consider how helpful this would be **if it’s applied to a user’s “empty” favorite/contact/message screens?**

One true story I heard from someone that was on site is that the nurses didn't know you could text a doctor, and she didn't want to bother him with a phone call since it wasn't urgent**. This is a fairly extreme example, but things like “Contacts” and “Favorites and “Directory” tend to have some non traditional definitions. Contacts being external contacts vs. Internal directory.**

This will help cope with the low active use rate, and reduce redundant support desk tickets.

A well designed app, should be able to explain itself to users.

Users should eventually learn how to navigate our system with confident and not have to guess what “Contacts” mean in our system.

Another place a we might want to implement this type of system **feedback/documentation is in error messages.** Referring back to the usability evaluation metrics, these **witty feedbacks** are an excellent **examples** of “**helping users recognize, diagnose, and recover from errors.**

*So this **wraps up the current UX** analysis and let's move forward and take some time to look at some of our **competitors' designs.***

Any questions or comments thus far?

Competitor's Design

Summary: Here, **I would like to show you briefly** what our competitor's user interface look like. You will see **Vocera** utilizes its **key brand colo**r. Almost immediately you would **recognize** its **Vocera** without even needing a logo. On the other hand, you will see **Cerner's** fairly **simplistic UI** that matches **highly with the IOS default standard**, making this much of a comfortable UI for anyone new to Cerner's app. These two sample UIs serve as a **reminder** that we should make our **Rubine RED** more prominent in our Voalte system while **sustaining IOS standards** to make sure end users are comfortable to use.

Updated UI/UX

After an extensive scrutiny of our system, now I'm going to show you what things could look like with this "**prototype**" developed in inVision. **This demo is by no means perfect or contains every improvement I aforementioned**. However we will walk through this minimally modified prototype so you can **grasp a sense** of what the app **could look like** and **more importantly, feels like**. Most of these changes **will not require** changes to the app **architecturally** and it will probably take an **experience developer a day or two** to modify every changes made here. The majority of the changes was done to **incorporate** our **branding** Ruebin P!nk and make sure **aesthetically** matches what we represent as a company.

So you would notice this **modified onboarding process**, where it incorporates the old user flow but with the addition of user tutorials. **This process** is extremely **valuable** in that it explains to the user **why we are expecting** you to input all these information, and it allows them to understand **"how" Voalte system works in the background**.

I also proposed a **new design** for the **user profile page** so our design matches with the real world scenarios. You will see here the updated UI that contains the **hospital badges**.

You will also notice I've been **playing around** with **switching** some **colors** in our **call history** and in the overall system in general. **I cut down the number of colors** used in the system and I **updated some warning messages** to align with IOS standard practices.

Noticeable changes

1) Adding the onboarding process for educational purposes.

2) The [Done] button is the signing-in process
3) Added a couple **Empty-State screens**
    a) A place to reflect Voalte's company culture.
4) Added [Home] +[Favorite] to Main Directory
    a) An intuitive user flow
5) Update the [Shift Overview] section to be consistent with [My Profile]
    a) Redesigned the "My Profile" view with a hospital badge to reflect an "interface" our users that are already familiar with.

Looking Forward

Summary: One of the **goal** for me and this presentation is to encourage managers to **incorporate this concept of iterative process design in the deployment process**.

Generally we have this structure, **id problem, hypo**... to ensure the **final product we ship has top of class UX**.

From **what I've learned** the past couple months here, we are already doing a lot of what this structured out to do, but we could use some **emphasis** on **DESIGN** aspect of things instead of **pure FEATURE** driven releases.

This design process could even be independent of the regular deployment process. We could create prototypes and do some **initial pilot** testing with **none of the back-end hookup**s(using tools like inVision, Origami and etc).

Essentially this should be part of the **product-business review process** and final decisions are then converted into the regular triage ticketing system.

Identify the Problem

So.
We **can generate** a list of problem from different sources, from analyzing in-app UX data to asking our customer facing teams. **Generating quality** points to address is the **hardest** part of this process. Here is a set of **questions** based on a **framework** that Jon Lax, a Design Director at Facebook, instilled among designers

"What are you trying to solve?

1. How do you know it's a real problem?
    a. Have "Human Goals" - value added features/fixes.
2. What data, anecdotes, etc. lead you to believe this?
3. Why solve this problem over others?
4. You have limited resources and time — why apply them to this problem above all else?

So in **our case,** this **translate** to something like:

Nurses and hospital staff don't use the Voalte system because they do not trust the Voalte system and that the Voalte system is **unintuitive** to use.

Now for those onsite (PMs, ICs, VMAAs) we often receive **hands on complains** directly from nurses and staffs.

With that being said, we need a **more objective way** to analyze our **end-user experience**. We need to start **collecting** some **data/survey** to reflect how nurses and hospital staffs are **actually** using our app.

What's the **avg. rating** a nurse would give after a new app update? What is the **average active rat**e (in % of all deployed phones), **how long** do they spend in the app per shift? How many **calls/alarms/messages** gets **ignored**? Consider **sending** our **surveys** to those nurses that **do not like to use** or try to **avoid** using the Voalte System, ask them **what keeps them** from using the Voalte phones that are assigned to them.

Note this is different from the v**oluntary feedback** form we have within our app, we need to be more **active** (aggressive) on **collecting** these data. We don't just want bad feedbacks, we want an accurate representation on how our apps are being used on a daily basis.

And yes.. There's also **strategies** in how to **prompt for feedbacks** with the highest integrity. [Sample Survey Questions attachment]
.

Moving forward, let's come up with a testable hypothesis to improve UX.

Testing Hypothesis

Continuing with our previous example, we're at why solve this problem over others.

"This **problem is critical** to solve because the **trust** between the nurses/**hospital staff** is the **foundation** of what makes **Voalte a successful solution**, and it impacts what Voalte represent as a brand as it **stand us ou**t from the **increasingly** competitive industry"

Sample hypothesis would be something like:

1) *We think nurses will trust our system more **if we improve the visual experience** for the "**quick reply message" and "Viewing by role/names" interfaces**. If we **improve** our **UI** and **remove the barriers** to the "**hidden**" **features** presented in our app, we **will gain overall in-app activity** and **proven** to hospitals that Voalte is a **positive investment** decision.*

After this process, we then test our hypothesis with either **design prototypes**, or if we have the **resource to do A/B testing** on different controlled release sites and collect and **analyze data** accordingly.

Test, Collect, Observe behaviors

So now… we're finally at testing and collecting data to observe user behaviors…

There a lot of third-party analytic services that can help us in this process, or we can even **start collecting/analyzing our own data** from our **servers** to start. **Establishing benchmarks** for what identify as a successful "change" is another **challenging and lecture itself**. But imagine, we could use **data to derive potential paid features**, potential of **paid white labeling** (through modifying our **VoalteCommonClient** UI layer) our apps for hospitals with strong marketing purposes and etc.

Prioritize Human Center Design. Instead of **simply** measuring **number** of text/calls/alarms received, **measure the meaningful exchanges(interactions)** within the app. Similar to what was mentioned in the webinar from UCSF, examples are what are "**actionable alarms**".

What's the ***net positive contribution*** to patient experience?. [See attached for sample KPIs]

Refine

From either hard data we collected through regular end-user usage or through different design specific iterations, **we will then finally refine our users' needs** and turn them into triage tickets for developers.

These can be mark as **low priority or aesthetic fixe**s but remember it's important first **step for them to be on the board.**

Moving forward, we can talk about some potential new features once resources permit.

Potential New Features

Voice messages

1. Voice messages is a feature widely popular with app like Wechat/Line and currently is implemented by Apple's default message app. I know there has been some talks and ideas floating around product…
    a. This feature is **already implemented** by some of our competitors. One being **HippaBridge**.
    b. "Walkie-Talkie"

React to Text/Voice Messages

1. Create a simple interface allowing users to respond quickly using emotions.
    a. Example: Facebook's & Slack's multiple emotion response.
    b. Positives: Facilitates quick responses other than using "quick messages"
    c. Possible implementation:
        i. Long press on the conversation bubble to initiate a response.
        ii. Change the background color of the conversation bubble base on user response.

// Random Thoughts...

Smart Wearable Product

Targeting a specific audience with one step at a time. We have Voalte One which is largely used by Nurses and Voalte Me for Physicians. If we're a company truly care about patient experience, why not develop a product centered around the patient experience specifically for patients?

Developing a product centered around the patient experience specifically for patients

" We've seen an explosion of apps that do help "replace" the pager already for iPhone and Android. Most of these apps have not been transformative, but if they were able to utilize the Apple Watch they could completely disrupt the ecosystem." - iMedicalApp

Potential Features:

- Maybe demo WeChat Android App
- Redefine **Users** with **Rooms**
  - Associate each watch with a room instead with a specific user
- Utilize the "**Voice Message**" from above
- Potential to utilize some **Siri**'s app extensions for Voalte's purposes
- Sent quick preconfigured **Nurse Call Alarms**

Existing Competition:

- https://www.vocera.com/watch

Selling Value Added UX

Super long stretch….

- After libVoalte gets "stabilized" and well tested….
  - Consider remodeling the current "Client UI Cake in Code"
    - Consider enabling "white-labeling" for hospitals for a fee.

Attachments:

[Sample Survey Questions attachment].

Keep it short and prompt users to fill it out in-app to increase response rate, data size and objectivity.

1. What's the number one factor that holds you back from using the Voalte phone?
    a. Size of the phone
    b. Trust in receiving proper alarms
    c. Are you receiving redundant alarms?
    d. Unintuitive to use?
2. Did you know about the [X,Y,Z] new features in this release?
3. Rate [X] feature.
4. Thoughts for improvement for [X] ? (Free response)

["Todo" List]

Overall: Standardize to iOS/Android UI Practices

Thing like
"Read to log out" warning box is the same style as the "Group Messaging" box. Why associate "Ready to log out" with "Group Message"?

Warning vs. Functionality

1. "Clean up" the user interface to allow only the Voalte Pink & Green accent for emphasis.
2. Decide on a consistent **font style and size** from one TableView to another.
    a. Main Directory/Call/Alarm/Chat/Menu view all have different font sizes & color
    b. I noticed different **sizes** of icon assets for the **same placeholder (Profile icon)**
    c. Different UI for changing password and PIN
3. Inconsistent locations of **Titles** in the different UiNavigation Bar
    a. Examples:
        i. **Contacts** and **Messages** title bar appears below the regular Title bar
        ii. "Interesting" additional title bar design for the string "VIEWING NAMES" when navigated to a specific unit. (Also not intuitive that it sorts by Role/Unit)
4. Position of the "Ready to Log Out" box SHOULD be CENTERED
    a. Style the "Ready to log out" box to Apple's standardized UIKit
5. Give a **Title** & **Body** to the warning messages.
6. Consider redesigning the dialer with a Voalte PINK accent

a. Another branding opportunity
b. **What's the "official Voalte Color" in the app?**
    i. (#FF0075) -- IOS Menu Screen BG (in the production code…)
        1. Android looks like they have a different **hex**
            a. Might be due to screen resolution...but worth checking
    ii. (#E20177) -- from branding guideline (our logo color) --Rubine Red

Sign-in:

Criticism:

1. ***Inconsistency*** in tapping **Continue** and **Done**.
    a. Don't have to necessarily remove the **Done** in upper right hand corner, but consider adding a big **Done** button in the same position as the **Continue** button in the previous screens.
2. This **grey** empty-state background
    a. This is common for other part of the app, consider designing specifically **Voalte Branded empty-states.**

Consider:

1. Review inVision prototype

Main Directory View:

Criticism:

1. Default icons doesn't "pop"/stand out enough.
2. Grey divider?
3. List view can get very long and tedious.
4. Consider adding scrolling base on A->Z order on right hand corner.
5. Consider changing the [Green Bubble] to [User Profile icon] for consistency's sake.
    a. It's more common to see [User Profile] related actions to be associated with the [User Profile] icon. (That icon should change colors to indicate the current status of user) Train users to associate specific icons for specific actions. Referring back to the heuristic evaluation, this means less cognitive load on the user's' end. We shouldn't rely on user's cognitive memory to reflect what this green bubble should/shouldn't be doing.

Core Feature Screens:

Consider:

1. Consider having "Empty State" role avatars
    a. Have generic avatars based on user roles (Dr./Nurses/Tech..?)
        i. Given a lot of our sites doesn't have profile pictures uploaded
2. Consider changing out the default "color pallet"(phone dialer) [Contacts] icon
    a. We're trained by Apple/Android and other system default icons.(usually has a generic avatar on it)
3. Consider changing the default [Nurse Call] icon.

    a. This has a transparent icon background
    b. The exclamation point creates a sense of distress for non-emergency Nurse Calls.
    c. Consider changing the [Saturation] of the icon & Nurse Call Banner base on the [Priority] of incoming Nurse Call.
4. Consider applying the feature " Changing Font Size" to the global system
    a. If a user will have trouble reading texts, they will have trouble reading directory/incoming Alarms/texts presented in the tableViews of the app
    b. Allows for user control and consistency of presentation across the entire system
5. Considering having "Always Top" feature for Text Messages
    a. Probably a good feature to have for staffs that are traveling and need to communicate with a 'home unit' constantly

Empty States:

Empty state examples: click me , TedTalk-404s

Minor Criticism:

1. Already neatly designed, although the TEXT empty state icon is different in size from the other 2.
2. The TEXT interface has an extra "MESSAGES" banner. Consider moving that up to the Title field and change "New Messages" to "New" to make room for the title field.

Consider:

1. Allowing users to initiate a call/text by tapping on the empty state icon.
    a. Refer to sample empty states, create a persona out of empty states :)

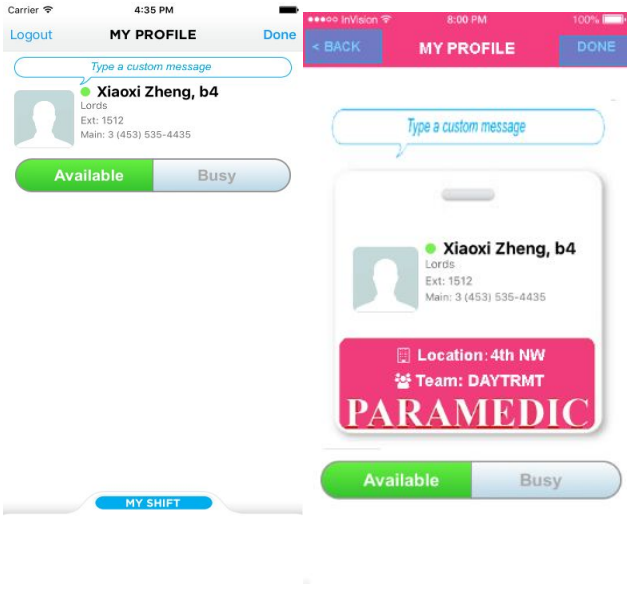
User Profile View

Criticism:

- Too much empty space without the SHIFT INFO slided up
- Awkward gesture to look up "My Shift" info.

Consider:

- Redesign to fit hospital setting [Badges] ?
- Customized status bubble
    - Increase the **size** of bubble for readability
    - Give **Roles** and **Location an icon**
    - Play around with having **SHIFT INFO** display by default instead of having users to slide up
- Make logging out more **prominent**
    - The most performed task



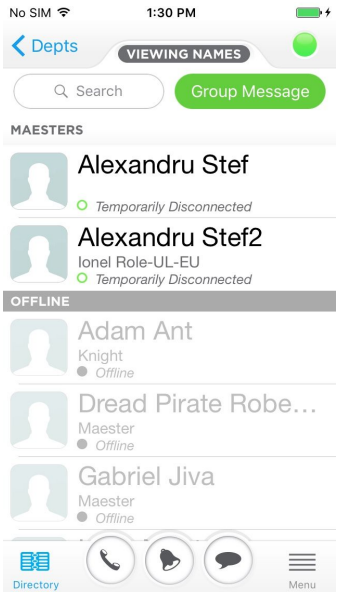
Detailed Directory View

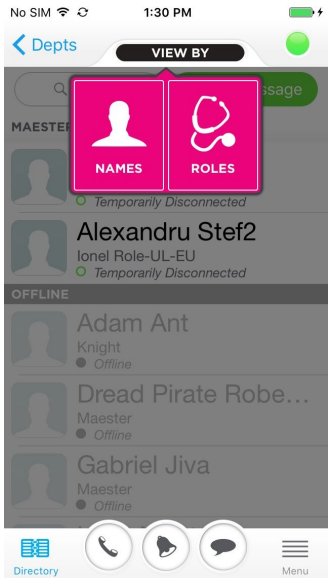
Criticism:

- Awkward positioning of [Search] and [Group Message]. Seem to be fighting over space.
- **Viewing Name** doesn't reflect it's "tappable" and thus hiding the feature that the directory can be filtered by Names/Roles.

To do:

- Considering removing the existing "Tab" design to present "Viewing Names/Roles".
    - Put the Name of Unit in the default iOS Navigation Title bar
    - Use the current "secondary nav bar" to incorporate the Toggle between viewing by [Names||Roles] with the **icons for search** and **group message**.
        - Sample "Redesign":
            - *replace Yes || No
                - With
                    - Names || Role





[Sample KPIs]

- Attempt to create “Human Goals” that improves patient experience. Minimize the data Daniel Shirley is manually collecting right now, eventually we need to evaluate ourselves against previous app versions instead of just evaluating against pagers and old systems.

- Third party analytic tools to reduce pressure on server?

1. Avg response rate from call/text/alarms
2. Measure the amount of **critical** nurse calls and the response time to these **critical** calls. (in-app)
    a. We should aim to increase this KPI overtime. (I know this is a different metric from Voalte Insight)
3. Measure the **average amount of back and forths in a single conversation** thread (Let that be 1-1 chats or group chats)
4. Set a benchmark on **how long a conversation** should be before it becomes “**meaningful**”.
5. Average call time/Avg. time spent in app during avg. shift duration