Exploring the Use of Tablet Applications for Emergency Resuscitation Practice

by

Harsh Damania

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved January 2016 by the
Graduate Supervisory Committee:

Brian C Nelson, Co-Chair
Vimla L Patel, Co-Chair
Janaka Balasooriya
Ayan Sen

ARIZONA STATE UNIVERSITY

May 2016

ABSTRACT

As technology has advanced in recent years, tablet devices have started to make their way into all walks of life. Yet, many medical documentation processes still see the use of paper. Though the paper based documentation method has been shown to be effective for some purposes, the introduction of tablet devices has the potential to make the documentation processes a lot smoother. In this thesis, tablet based documentation systems are reviewed, and based on this, a new custom application is developed that medical staff can use with ease. This new application, developed for an iPad is one where users can fully customize their own forms for different uses in the intensive care unit for resuscitation scenarios. The thesis discusses the architecture behind this application along with designing different elements of the system. Through this thesis project, the application was evaluated to see if such a complex documentation process can be easily used and created on a tablet device. The medical staff surveyed, responded positively to the use of the application and agreed that the electronic documentation usage and creation is a powerful tool that could help improve resuscitation practice by making it more efficient.

ACKNOWLEDGMENTS

Special thanks to Amanda Krumm - RN at Mayo Clinic - for working with me on creating this app with valuable ideas and feedback, and on going the extra mile to get survey responses from the rest of the medical personnel at Mayo Clinic. Also a special thanks to all the committee members of this thesis for constant feedback on the project and to improve the overall quality of the work. Lastly, I'd like to thank my parents for their continuous support to help achieve my dreams.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

During medical emergencies, every second that a clinician (doctor or the nurse) can spend on focusing on the patient's problem, rather than documenting the actual event, is precious. Medical staff in this event should spend as little time as possible in shuffling papers and completing medical forms, and rather look for ways to leverage modern technology to facilitate these tasks efficiently, effectively and safely.

Since the introduction of the iPad in 2010, the tablet market has sky-rocketed [1]. The iPad has made the tablet more user friendly, and has pushed other manufacturers and developers to compete actively in this segment of the market. What makes these tablets a huge seller is their size and portability - with the average tablet screen size around 8 to 10 inches and weighing in at a couple pounds it makes the perfect gadget to travel with, completely erasing out the need to carry around a laptop.

Most tablet manufacturers have opened up their SDKs (Software Developers Kit) to other third party developers after seeing the craze of building apps for smartphones in similar environments. Because of this, developers have built millions of apps for people to use everyday - from news to social networking to games - almost every kind of app exists on sale today. The app market is huge, with about 1.3 million apps available on each of the Apple AppStore and Google's PlayStore, and several hundred thousand apps available on smaller similar app stores [2]. There have been an

estimated 138 billion app downloads in 2014, and by 2017, that number is estimated to be around 268 billion [3].

Even with these wide array of applications available, there is a small market that has not been touched on too well: medical assistant apps. While health apps have been around and doing well for the general public, there exist very few apps for medical personnel to help with their daily tasks. A 2012 study showed there were about 40,000 health related apps on the AppStore with around 44 million downloads climbing to 144 million by 2016 [4]. That number goes goes down dramatically when it comes to apps for health records and documentation [5]. There needs to be a way of digitizing many of the clinical workflows that would save medical staff a lot of hassle that is encountered when dealing with paper, not to mention reducing the number of errors. We live in a very digital world, and there is a big push to introduce more technology in to medical environments [6] and reduce dependencies on physical copies of paper forms, and with the ever improving power and usability of the tablets, now is a good time to make the switch from paper to a digital form.

For example, for a resuscitation scenario, currently medical staff spend a great deal of time completing the Code Blue paper sheet to log the events that are currently taking place during the code. An event in this case is defined as any action being performed on a patient during an emergency. These events include anything from recording the patient's pulse to the dosage of drugs being given to monitoring the patient's current condition. All these logs need to make sure that they have the time of logging along with the code timer log to track as to when the different events took place and when they should check for other symptoms or provide medication.

A Code Blue resuscitation scenario in adults is generally called immediately for any patient who's unresponsive, apneic, and/or pulseless. Under American Heart Association (AHA) guidelines, calling for help and initiating CPR (Cardiopulmonary Resuscitation) should be done simultaneously [7].

This particular process can be completely digitized using tablets. Since tablets are light weight, they can be easily taken along from one room to another. With the usability of tablet devices improving with touch screens, it makes for the perfect device that can be used in an emergency condition. The US Department of Health and Services has outlined benefits for using electronic record documentation [8] including reducing errors and saving time, both of which are related to patient safety. Using an electronic documentation system also helps to keep a more accurate time, using a time stamp that helps with different procedures that need to be conducted during regular intervals [9] [10].

The research conducted for this thesis is an extension of one that was conducted by Bokhari et al. earlier in 2015 [11]. They built a basic prototype of a Code Blue form sheet app and concluded that an electronic form of documentation was preferable to the medical staff over a paper-based version. For this current research, a documentation system (iPad app) was built on the basis of the previous app. This new app not only improves the general design and usability - to make it a more user-centric application along customer facing application - but also includes customizable forms that medical staff can add, edit, create or remove by themselves without any technical knowledge.

2

RELATED WORK

There are not a great number of alternative documentation methods available that are reported in the literature [5], but there have been a few similar applications developed for tablets that help to document the events during an emergency scenario

EventDoc

In research conducted by Grigg and colleagues at the University of Washington, they evaluated the ability of an electronic system to document events during a cardiac arrest scenario [8]. They created a tablet application called EventDoc to help clinicians with their documentation. The authors found that using the application resulted in 30%, decrease in the number of reported errors (See Figure 2.1) while also reducing the amount of redundant information captured. They concluded that on the whole, the app captured 24% more relevant information than paper-based documentation.

The app developed by Grigg et al., though cluttered, shows a good display of events as well as the logs of what is currently happening. There is also a handy audio recorder of for recording what is happening at any moment.

Full Code Pro

Another tablet-based application was developed by Peace et al. to evaluate whether electronic documentation improves the precision during documentation in a fast paced

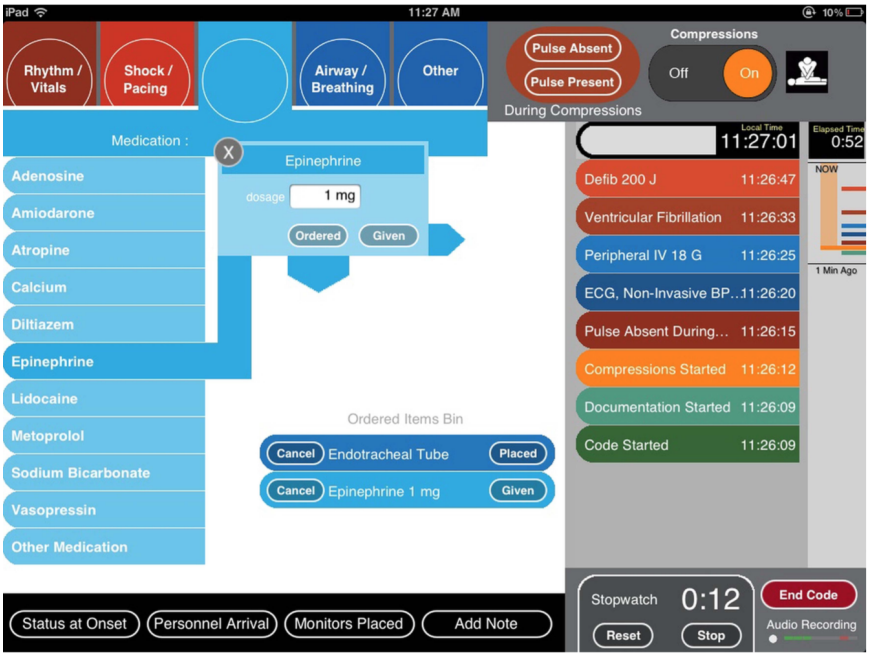
Figure 2.1: Errors as a Function of Paper and Electronic-Based System as seen by Grigg et al. [12]

Table 3 Difference in errors between electronic and paper records

	Omission errors (count)	Noise errors (count)	Timing (% of recorded events)	Specification (% of recorded events)	Commission (% of recorded events)
Average errors using AHA record (constant)	22.2***	8.0**	12.1	8.7**	6.2
CI	(14.3 to 30.0)	(3.2 to 12.8)	(−0.5 to 24.8)	(2.4 to 14.9)	(−4.0 to 16.4)
Difference in errors using EventDoc	−6.3**	−2.9*	−3.2	−3.2*	−4.4
CI	(−10.1 to −2.5)	(−5.3 to −0.6)	(−9.3 to 3.0)	(−6.3 to −0.2)	(−9.4 to 0.5)

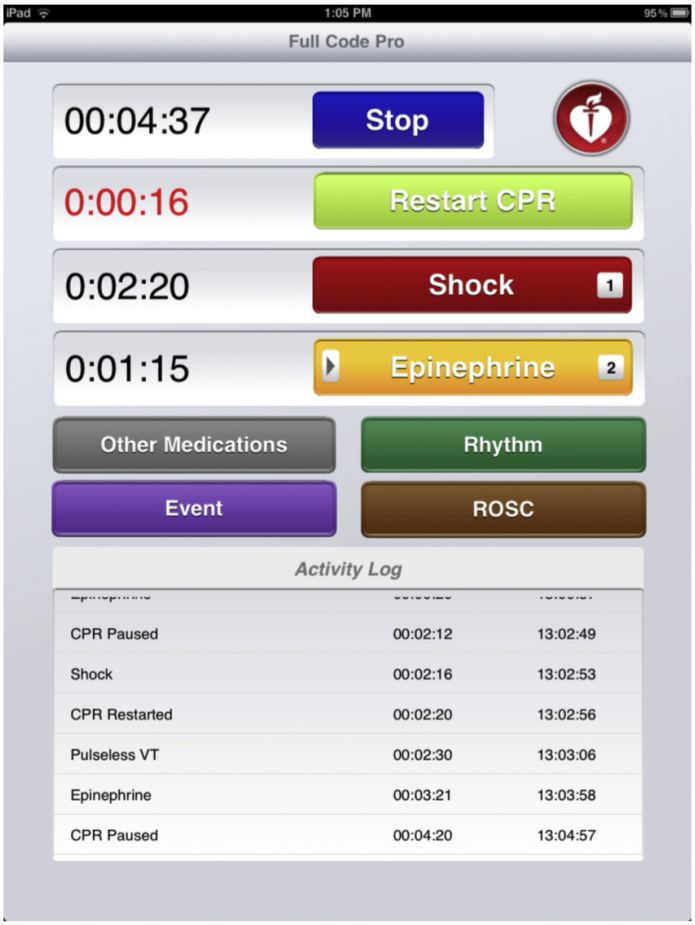
The constant is the control—the estimated number of errors committed by the American Heart Association (AHA) cardiac arrest record. The electronic record row shows the change in errors when using the EventDoc system compared with the control. Omission and noise errors are expressed as counts. Timing, specification and noise errors are expressed as percentages of information recorded.
Repeated measures ANOVA model (32 observations of 16 research subjects). 95% CIs in brackets. *p<0.05, **p<0.01, ***p<0.001. Subject-specific coefficients not shown.

Figure 2.2: *EventDoc* App in Action. [12]



resuscitation scenario. They found that conventional paper-based documentation practices are inaccurate, often misreporting intervention delivery times or missing their delivery entirely mainly due to human error in high pressure situations [13]. However, they also demonstrated that a tablet-based documentation method may represent a means to substantially improve resuscitation documentation quality, which could have implications for resuscitation quality improvement and research [10].

Figure 2.3: *Full Code Pro* App in Action. [10]



The *Full Code Pro* app is a fairly basic app that records the basic events during resuscitation. More events and drugs are left out of this as compared to the *EventDoc* app. This does however make use of multiple timers for different events which is useful.

3

CLINICAL ENVIRONMENT

As mentioned before, during a Code Blue scenario, the medical staff spends a great deal of time completing and managing the paper forms and event timers. For example, below is a detailed breakdown for a Code Blue resuscitation scenario at Mayo Clinic Hospital Arizona -

- Code team roles are dedicated prior to the start of a shift. The code team includes a Primary RN (Registered Nurse), Secondary RN and a Recorder who is responsible for documentation.
- Events are documented on the very cluttered code sheet paper form. (Figure 3.1)
- Vital Signs are documented at least every minute.
- Medications given per ACLS (Advanced Cardiac Life Support - A training paradigm for Continuing Medical Education [14]). Example every 3-5 minutes, etc.
- Energy delivered also per ACLS - every 2 minutes.
- The documentation is then reviewed by the attending MD (Doctor of Medicine)
- Review documentation to make sure whether or not it meets American Heart Association (AHA) Resuscitation Recognition Criteria.

3.1 AHA Resuscitation Recognition Criteria

Below are the list of criteria that need to be followed to be recognized by the AHA [15]

- Cardiopulmonary Arrest (CPA): Time to first chest compressions $\leq$ 1 min in adult or pediatric patients and newborn/neonates $\geq$ 10 min old: Percent of events in adult or pediatric patients where time to first chest compressions $\leq$ 1 minute of event recognition.
- CPA: Device confirmation of correct endotracheal tube placement: Percent of adult or pediatric events with an endotracheal tube placement which was confirmed to be correct.
- CPA: Time to first shock $\leq$ 2 min for VF/pulseless VT first documented rhythm: Percent of events in adult or pediatric patients with VF/pulseless VT first documented rhythm in whom time to first shock $\leq$ 2 minutes of event recognition.
- CPA: Percent pulseless cardiac events monitored or witnessed: Percent of events in adult or Pediatric patients who were monitored or witnessed at the time of arrest.

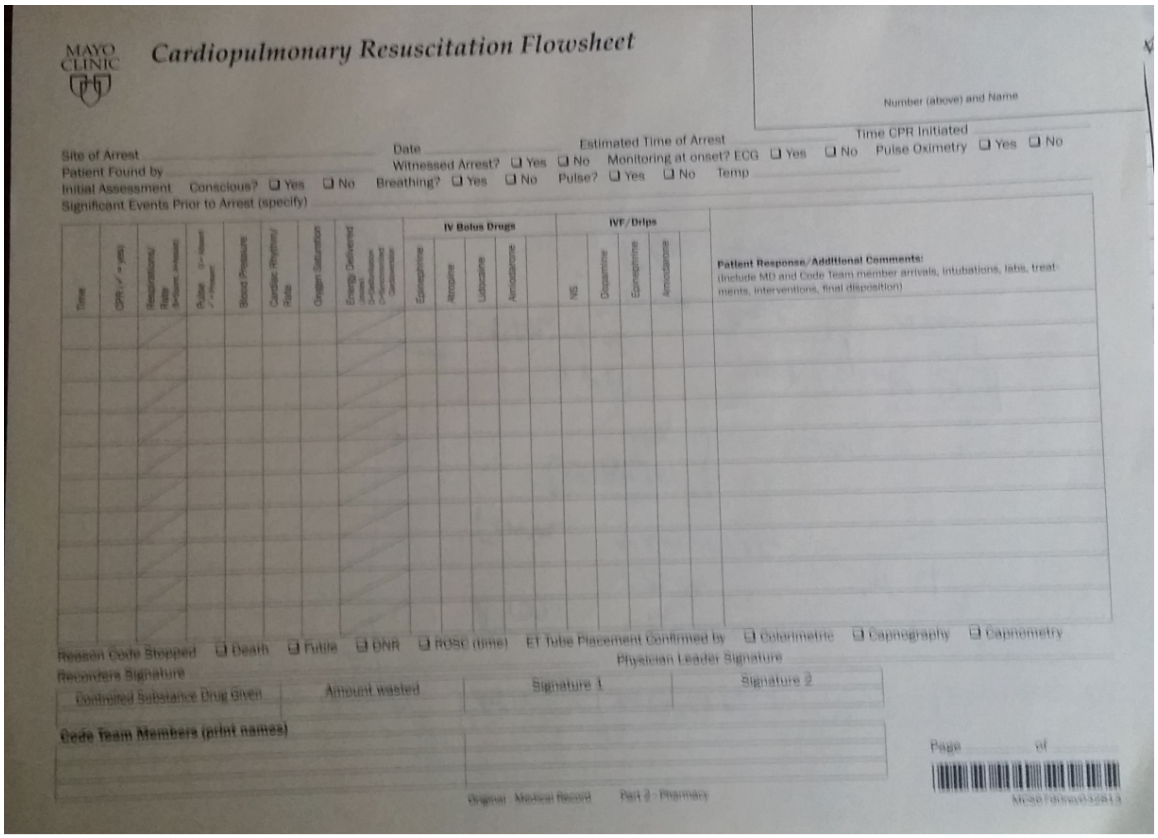
To keep up with the guidelines and also maintain good documentation, there are certain challenges that the medical team face -

- Person documenting the code is unfamiliar with the record or data requested.
- Job of documentation during code is not considered a key, desirable role.

- Several different clocks need to be used to enter times in to the records.
- Data on record is incomplete.
- Data on record is often illegible.
- Names of all members not present on record.

The application presented in this thesis is built with these challenges in mind. While there are some cases where technology cannot help - such as being unfamiliar with the records, or it not being a desirable role - the others can be tackled.

Figure 3.1: Code Blue Paper Form Used at Mayo Clinic Arizona

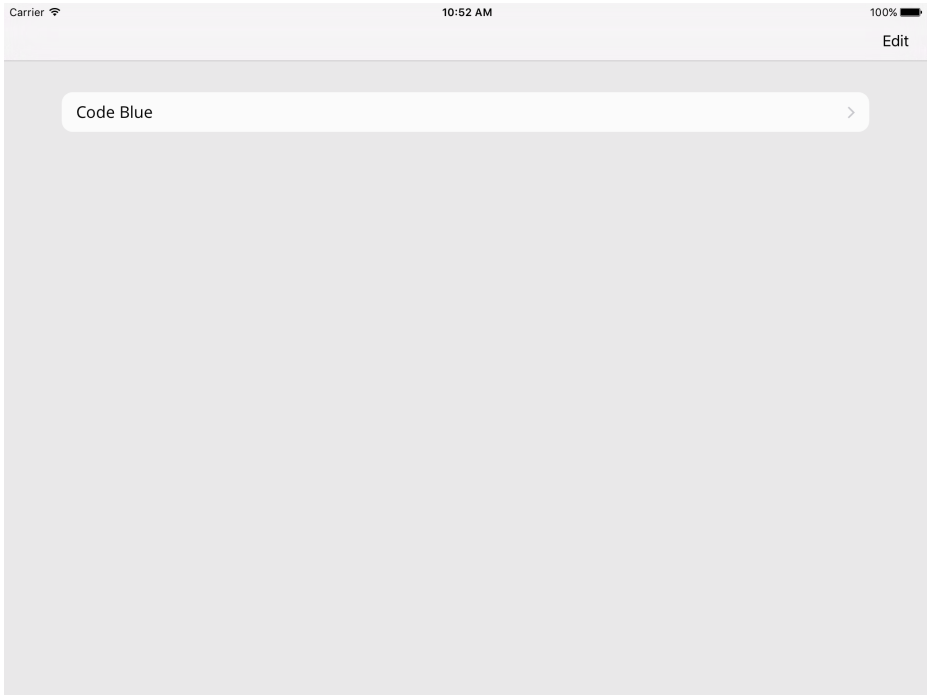


4

DESIGN

The main menu for the app includes the list of forms that have been built out as we see from 4.1. The *Edit* button on the top right takes us to the form editing page where users can add or edit forms. Selecting a form navigates the user to the form.

Figure 4.1: Main Menu



4.1 Forms

The form screen is divided into 3 columns, as seen in Figure 4.2 for Code Blue. The first column is used to display and select the different events for the form. The users have control over the order in which the events appear so more critical events can be

Figure 4.2: Code Blue Form

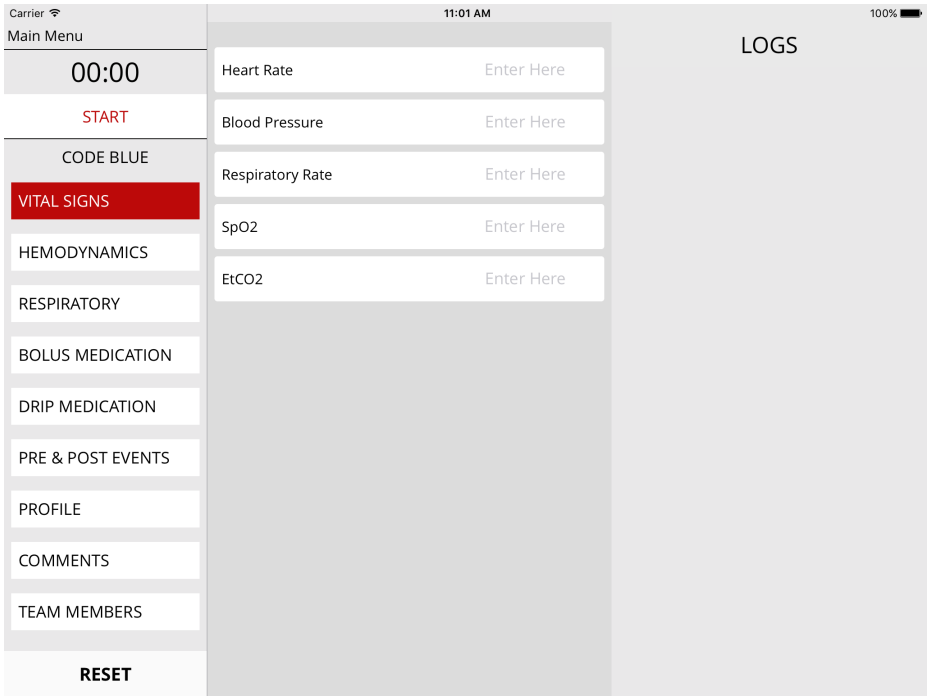
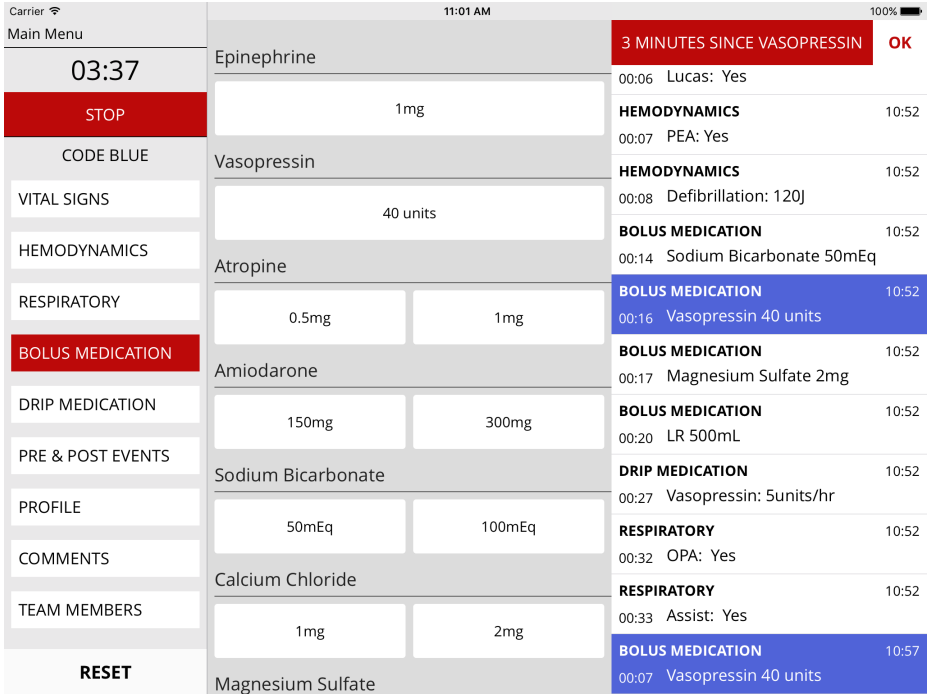


Figure 4.3: Code Blue Form Simulation



placed on top and the user does not need to scroll to get there. The timer is also contained in the first column on the top. The location is static and does not scroll with the events. There is also a *Reset* button at the bottom of the first column to reset all the fields, logs and the timer for the forms.

The middle column is used to display the main UI elements for the form. There are a number of different UI elements used to display different kinds of data which will be discussed later.

The last column is the log column which populates as different events are being logged. These log views can also be set different colors to make sure that certain events stand out. This log column also contains the alert view for ones that are set by the user for different items. These can be seen in Figure 4.3.

4.1.1 UI Elements

There are different UI elements used for different data types that the users need. Using these different elements, numerous forms can be created.

4.1.1.1 Flags and Values

Flags and *Values* represent buttons. *Flags* have a highlighted state to correspond to its *Yes* or *No* values. These are a part of the *Collection* section view that will be discussed later. Figure 4.4 shows an example of a *Flag* field. *Assist* has been marked as *Yes* and is thus highlighted. *Value* types would not have a highlighted and would reset back to the normal state after selected.

Figure 4.4: Flags and Values

4.1.1.2 Numeric and Text Fields

Numeric and *Text* types represent single line inputs. They render a text field with the option for showing a detail label which can be used to represent units as seen in 4.5.

Figure 4.5: Numeric and Text Field

When using numbers with textfields on CocoaTouch, the numeric keyboard for the iPad is not quite ideal for a fast paced environment like an emergency room. For that reason, a custom numpad for the use of *Numeric* types was created. Using the numpad makes it convenient to simply use the buttons provided there rather than find one's way through the keyboard for different symbols.

4.1.1.3 Comment

The *Comment* item type is used when the user needs to enter multi-line text. For the use of comments, there is a "Stock Comments" option where the user can enter some commonly used phrases as seen in Figure 4.7

Figure 4.6: Numeric Field with Numpad showing

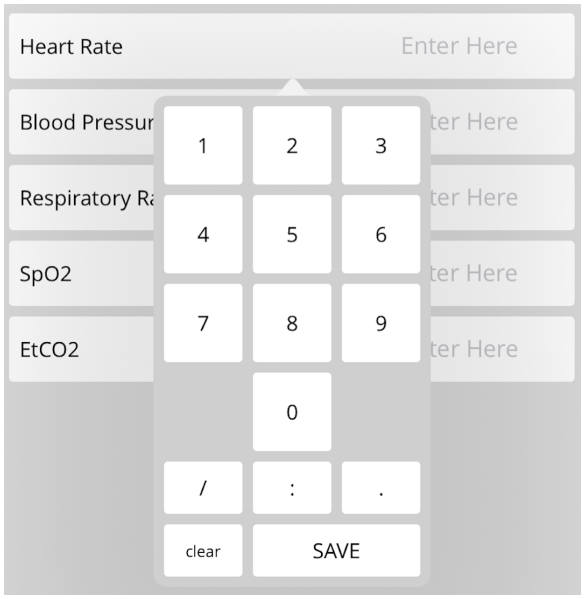


Figure 4.7: Comments

4.1.1.4 Signature

The *Signature* item type is used when there is a signature required as seen in Figure 4.8. Users have to sign and enter their name to be considered a valid.

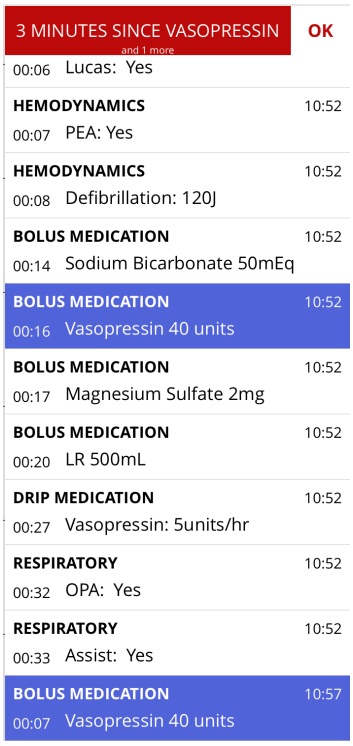
Figure 4.8: Signature

4.1.1.5 Logs and Alerts

Log messages are displayed with only the information required so as not to seem cluttered. The log views also display the event from which it was fired (For example - *Hemodynamics*, *Respiratory*, etc seen in Figure 4.9) for easy finding. Logs can also be customized to display with different colors to make different events easily discoverable.

Alerts are shown on top of logs if any. They have a completely different eye catching red color so as to direct attention towards them. If there happens to be more than 1 log, then they are stacked and they display counts at the bottom as shown in Figure 4.9.

Figure 4.9: Logs with Alerts

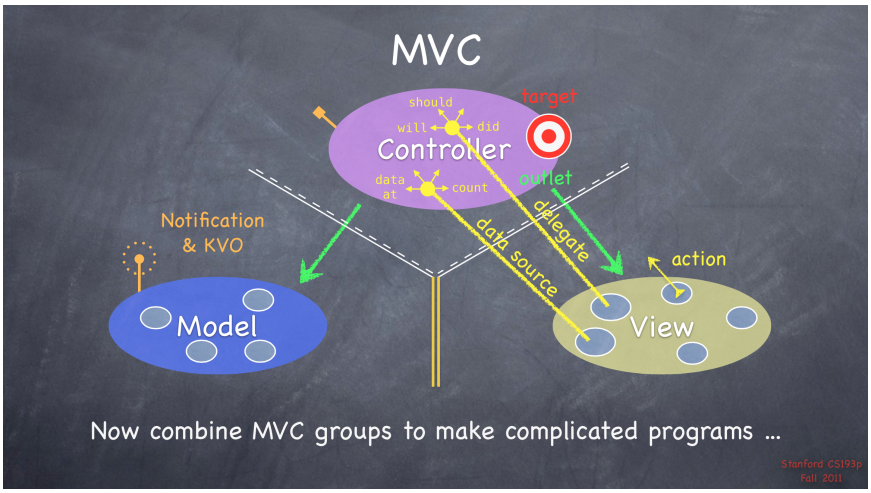


5

TECHNOLOGY

This is a native iOS app written completely in Swift 2.0. Since the app depends on customizability, arranging all the elements for the forms also had to be customizable. To do that CocoaTouch's `UICollectionView` class was heavily used. It makes writing custom layouts easy, while efficiently handling UI memory load. The application was built using the MVC (Model-View-Controller) pattern, one that is most commonly used when developing iOS applications or applications in general that rely on UI feedback.

Figure 5.1: Overview of the MVC Paradigm [16]



The MVC pattern is mainly used when developing for application that is dependent on how a user interacts with the interface. This design pattern is one where every class has a specific purpose and does not tread on others. The view manages the graphical and/or textual output to the portion of the bitmapped display that is allocated to its

application. The controller interprets the mouse and keyboard inputs from the user, commanding the model and/or the view to change as appropriate. Finally, the model manages the behavior and data of the application domain, responds to requests for information about its state (usually from the view), and responds to instructions to change state (usually from the controller) [17].

5.1 Model-View-Controller (MVC)

As mentioned earlier, the application was built using the MVC paradigm. While Figure 5.1 does a good job of giving an overview of MVC, building out a larger application - like this - generally requires multiple MVCs working together.

Figure 5.2: Overview of Multiple MVCs Working Together [16]

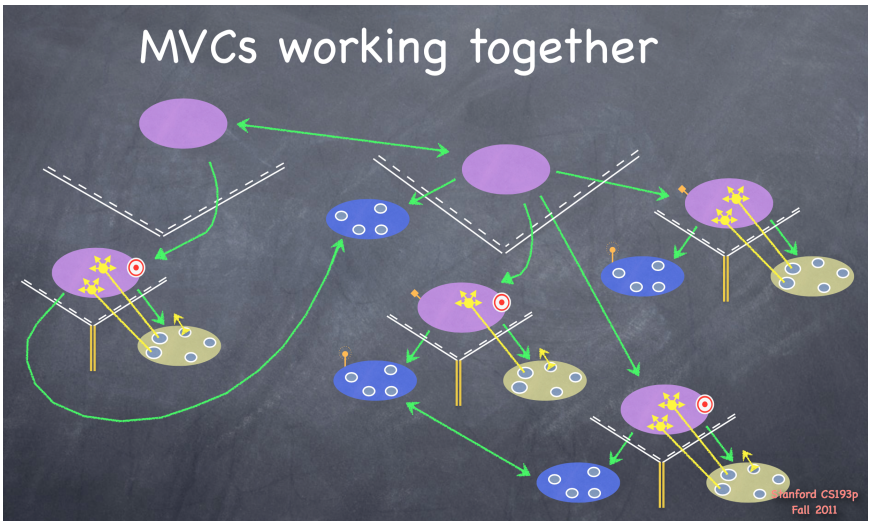


Figure 5.2 shows that a view controller is responsible for solely one view. View controllers talk to other view controllers to send data back and forth. Models can talk to multiple view controllers and also other models, but - as mentioned before - never to a view.

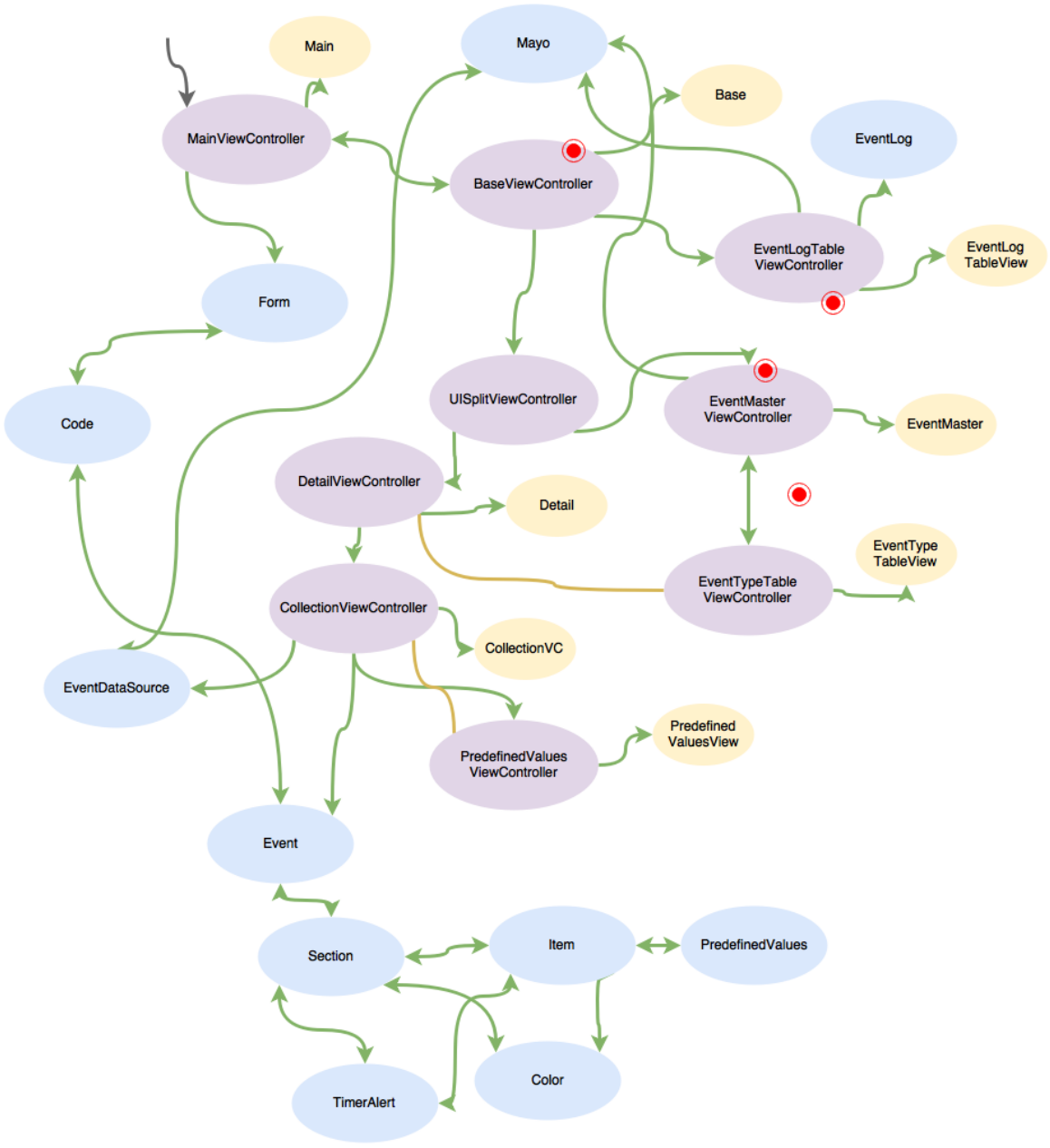
In Figure 5.3 we can see how this works in this app. The `Mayo` model class is a singleton that stores all the `Form`s, logs and other relevant data for the app. It is the sole class responsible for maintaining the timer. The rest of the models are discussed in the Models section below. The red targets on a few view controllers are *Notification Observers* for different events that are fired by other objects, which could be Models or view controllers. By using notifications, an object does not have to have an instance of another where it is trying to send the message, rather those instantiated objects would already be listening for that message. Another pattern seen is that of the Delegation pattern, represented with yellow lines in Figure 5.2 and 5.3. The Delegation pattern is used when a class needs to pass off a feature to another class, delegating the work to another class. This is mainly used when communicating data back to the object that contains a objects reference. An example used in the app, `PredefinedValuesViewController` has a delegate that it uses to pass data back when a selection is made to a *value*, in this case to the `CollectionViewController`.

5.2 The Models

The application is presented as a `Form` class. Each of these has a number of `Code`s that the medical staff selects for the appropriate setting. Each `Code` has a broken down into different `Event`s that make organizing the data intuitive and easy to find. An `Event` has `Section`s that helps clubbing together similar data. Each `Section` has a number of `Item`s. An `Item` is what is represented on the screen as an individual UI element.

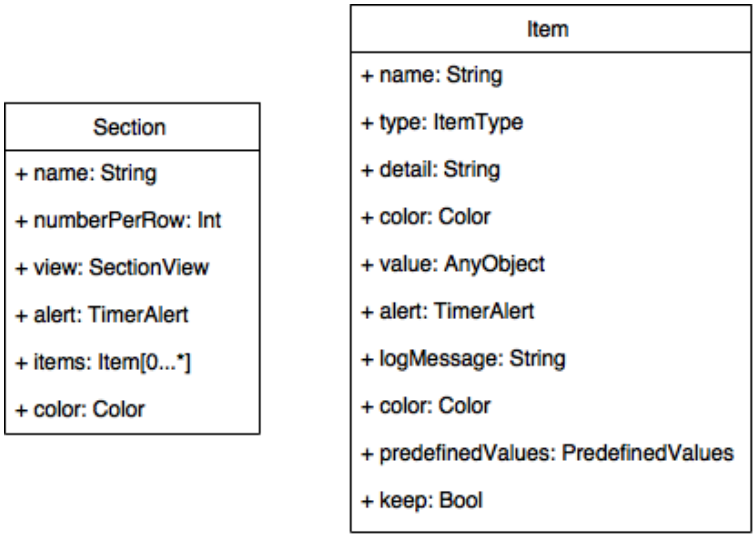
Each of the `Code` and `Event` classes have only a name attribute and their list of dependencies. `Section` and `Item` classes have some more attributes that help the

Figure 5.3: Multiple MVCs Working Together in the App



model determine what to do with them. They are outlined in Figure 5.4. Both `Section` and `Item` classes have a `Color` and `TimerAlert` attribute. `Color` helps set a preset color for different log types so that they can be easily noticed. `TimerAlert` sets up a customizable alert to set a reminder for the user.

Figure 5.4: Class Diagram of Section and Item Classes



5.2.1 Section Class

The important attribute for the `Section` class is `view` which basically tells the model that all items in a block would be of a certain type, thus telling the renderer how much screen real estate to allocate for those items. The `view` is an enum type `SectionView` with the elements -

Collection Used for button trigger types. One tap events.

List Used for text fields that would take in input.

Signature Used for signature fields only.

Comment Used for comment type fields only.

`numberPerRow` value is used for `Collection` elements to determine how many of them the user would want to be displayed in a row. (Defaults to 1 for the rest). This helps emphasize some view elements over the other. Setting a `TimerAlert` or `Color` for a section would set those values for all the items within it.

5.2.2 Item Class

An `Item` has quite a few more attributes that help with displaying the particular UI element. `name` and `detail` fields are used for displaying any relevant data for input or such. `logMessage` is used to determine the log message fill in along with the actual value. `type` is an enum used to determine the type of UI element to be displayed. They are -

Flag Used as a on/off button.

Value Used as a selectable button logs something every time it is tapped.

Text Used for a standard text field.

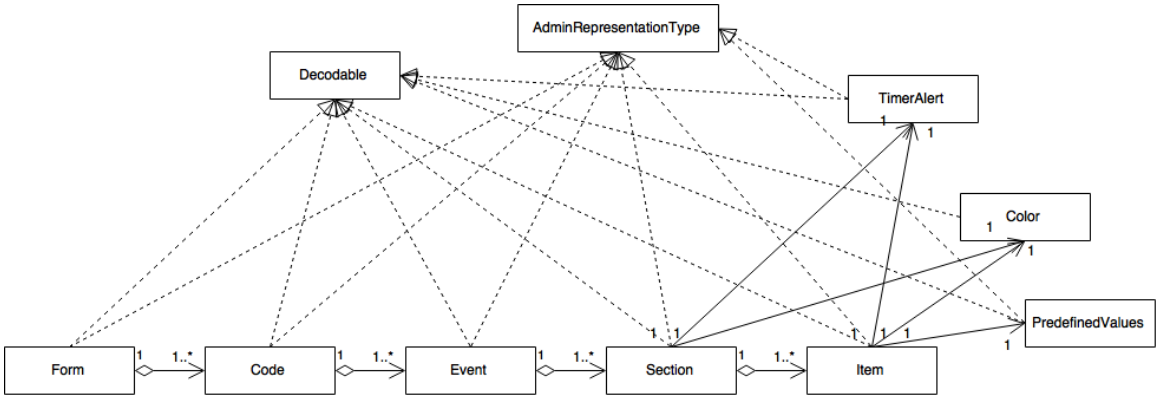
Numeric Used to input text using a custom numpad.

Comment Used to enter comments.

Signature Used to for drawing signatures.

`value` is classified as an `AnyObject` (Swift for representing any kind of data type from the library) since it can take `String` or `Bool` values. Setting a `TimerAlert` or `Color` for an item overrides any set for its parent `Section`. `predefeinedValues` is used for item types that would allow the user to set some properties predefined for quicker use. For example, having top Comments set beforehand can make it very quick document during an actual emergency. `keep` is a handy boolean property to tell the model whether that field needs to be cleared after logging.

Figure 5.5: UML Diagram of Model Classes



5.3 Making it Flexible

All the representable data for a form is stored locally on the device and is stored in a standard `JSON` (JavaScript Object Notation) format. This key-value format is ideal for representing models as it can simply be taken and parsed to any other platform. This ensures us that we can use this data in the future on any other device or even the web, not strictly on iOS.

Since swift objects cannot directly parse and map such `JSON` objects and its attributes to itself, they have to be manually done. From Figure 5.5 we can see that all the model objects conform to `Decodable`. `Decodable` is a Swift protocol (similar to a Java interface) that defines an initialization function for all the classes that conform to it so that they can map it's attributes to the key-value pairs of the `JSON` object being read in.

Apart from reading in the data, we also should be able to write to it - thus making it customizable and future proof. On the landing page for the app on the top

right, there is an *Edit* button that takes the user to a settings menu where they can making any change they want to the forms. Each of these screen represent the data models discussed earlier along with their attributes. (Figure 5.6 and 5.7) Users can select and edit text fields to change property data as well as create, reorder or delete codes, events, sections, items, colors, alerts and predefined values. To view their data, these model classes also conform to the `AdminRepresentationType` protocol that has functions to help them determine what users can edit for it. `Color` does not conform to it because it is a simple pop-up that helps set the *RGB* values instead of a new view.

A good advantage of using different protocols here is that only the classes that are required to perform certain functionalities would conform to it thus making the application more modular. A good example of that is of the `Color` class. Since it does not push on the navigation stack for editing its data in settings, it does not need to conform to the `AdminRepresentationType` protocol.

Figure 5.6: Settings Menu Code

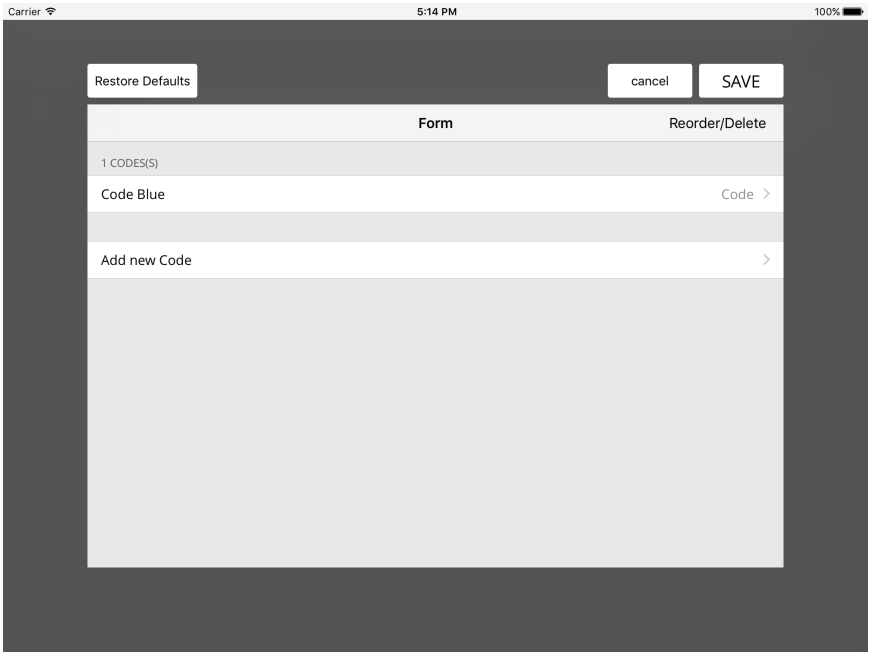
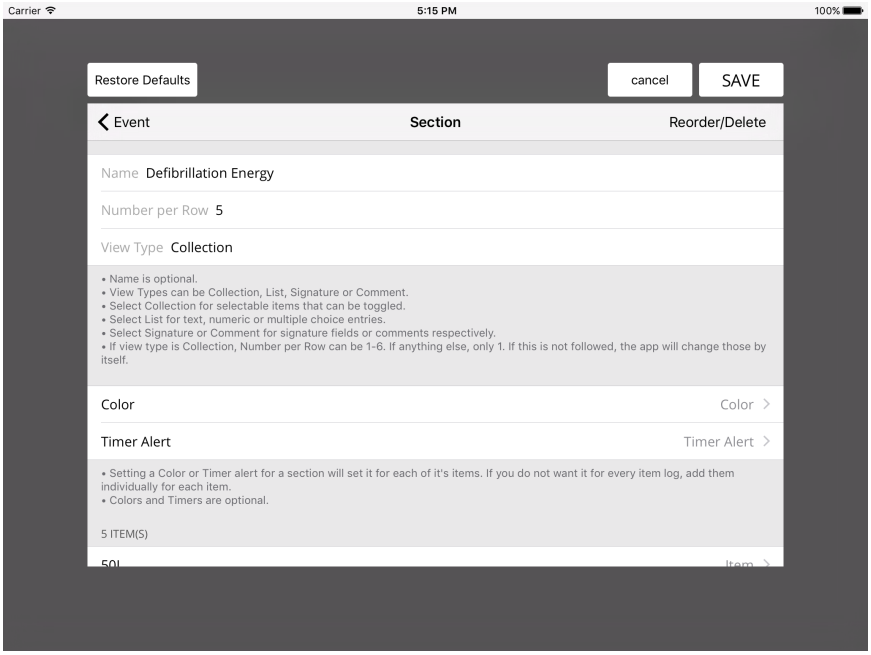


Figure 5.7: Settings Menu Section



The layout for the Settings screens is shown in the above screenshots. New items can be added, removed or re-ordered easily using a single tap. Other attributes for the data models are displayed here along with the technical details that are required for a valid model. Users also have the option here to restore to defaults, which in this case happens to be only a Code Blue form.

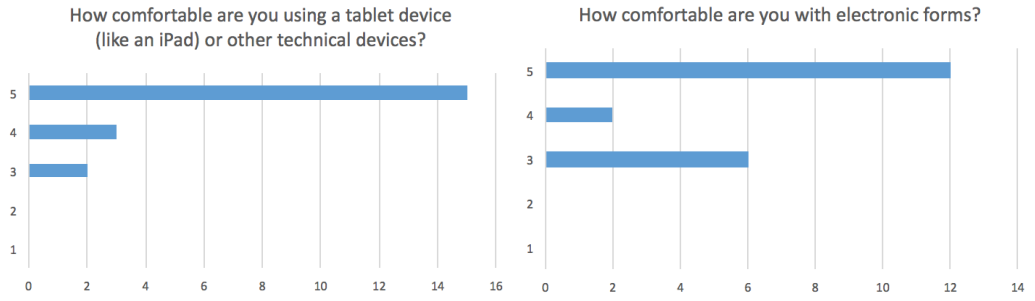
6

DISCUSSION

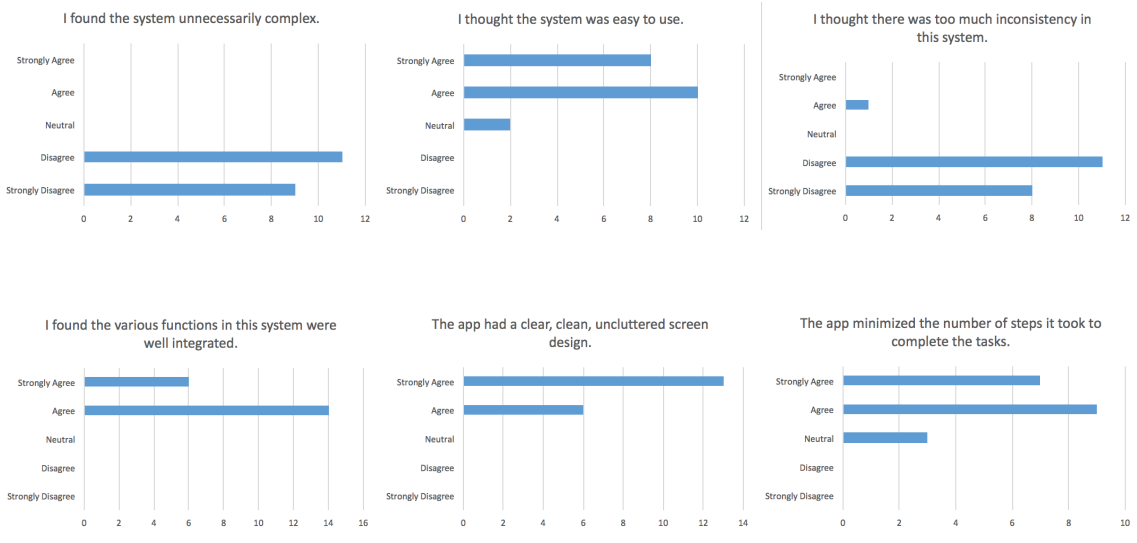
The application was built using an incremental approach while getting regular feedback from the medical staff on new additions and changes (iterative design process). The design and different elements of the application have evolved through time to what they are currently. The application initially had all the Vital Signs statically placed on top so that it's easy to log those events quickly. However, after some use in the clinical environment and consideration, the application was redesigned to have the vital signs placed with the other events because they were taking up a lot of valuable screen real estate. Another big change was that of the use of a numpad. While using the application in the early phases for logging numbers, it was observed that the iPad keyboard was not very efficient for fast paced use in the emergency room. So, a custom numpad was built that is easy and quick to use, and only shows those keys that are required for use.

The application was used by some of the medical staff at Mayo Clinic, and 20 of the clinicians were evaluated (See Appendix A) on the usability and the applicability of the system. The evaluated questions were based on those provided by the Healthcare Information and Management Systems Society (HIMSS) when dealing with medical applications for tablets or smartphones [18]. This being an electronic documentation system on a tablet device like an iPad, we tried to gauge their current knowledge and ease of working on such devices. On a scale of 1 to 5 Likert scale, with 5 being the

highest (best), the users mean response was 4.65 for how comfortable participants felt using an iPad and 4.70 for the use of electronic forms rather than the paper, with nobody responding below a 3.0. This shows that clinical staff were generally comfortable with the technology and using such a system for documentation purposes.

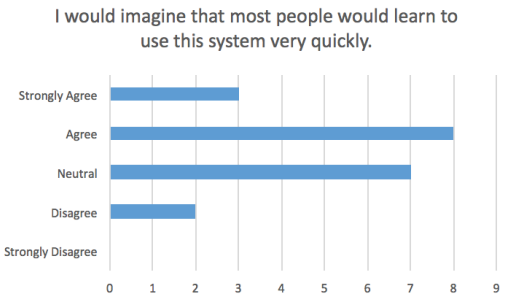


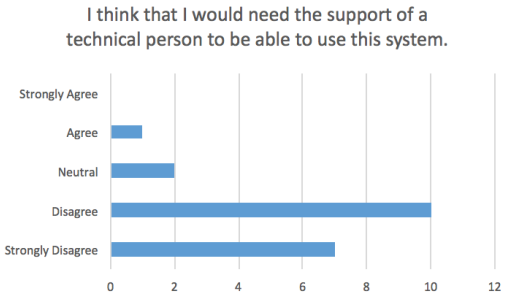
As mentioned before, this system aims to be user friendly and easy to use. The following questions on the survey were related to the usability and design of the app.



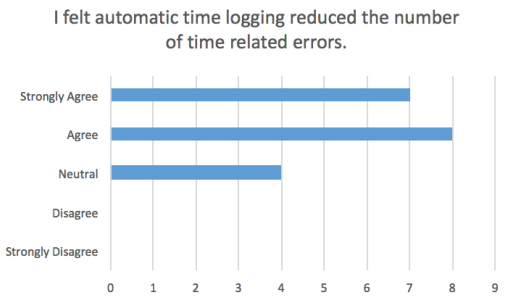
On the whole, users in general felt that the app was designed well in a way that was not complex. Tablet application design should have clear elements presented on the screen with nothing cluttered, and the surveyed staff certainly felt that way. The

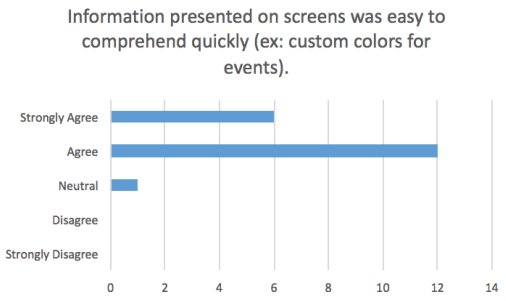
staff also responded positively when asked if the application reduced the number of steps required to complete a task. This certainly attributes to added features like automatic time logging and alerts.

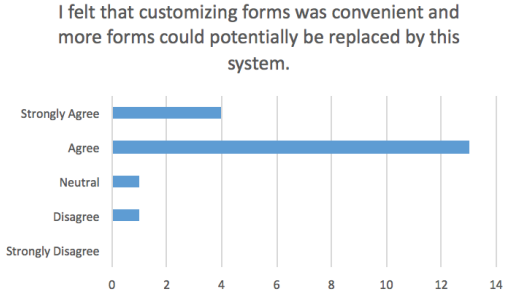




This application needs to show more evidence for relatively easy learning curve so that it would be smoother to onboard new clinical staff, without the need to bring in technical person every time. More than half the surveyors agreed that new users would not take time to understand the application. Many of them disagreed that they would need the help of a technical person to use the application.

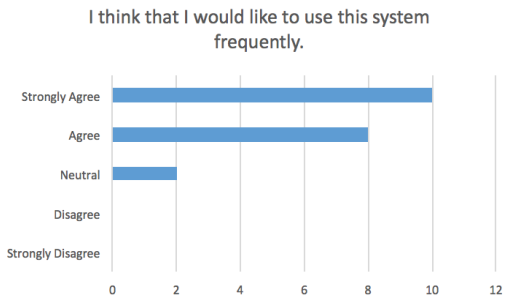


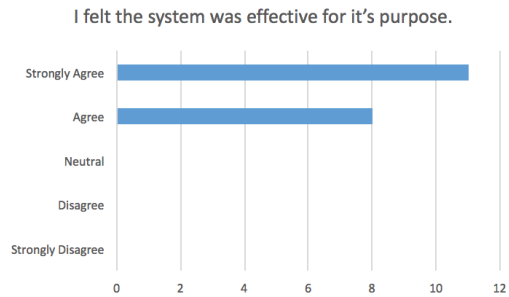


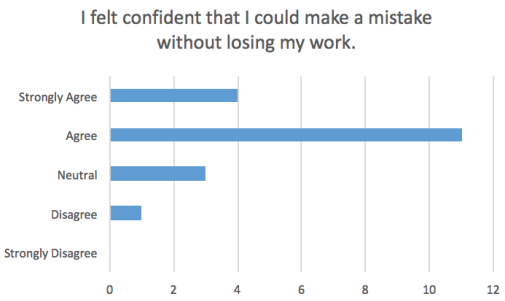


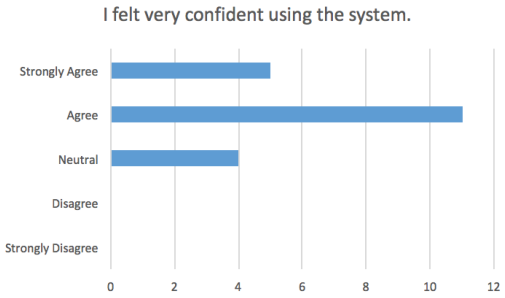
The app also has a few features that really separate it from it's paper based version, and in ways makes the documenting process easier. Automatic time logging helps the staff in not having to check the timer for every event being logged, but rather focus on the situation on hand. About 80% of the clinical personnel evaluated agreed that the automatic time logging helped them reduce the number of time related errors, which can be large for repeated events such as medication and checking for vital signs. Since the medical staff deal with these timed events regularly, there are customizable alerts in the app that notify the user of such events. Almost all the users agreed that this feature was beneficial and made them feel at ease about these repeated events. A feature that was added later in the development cycle was that of colored logs. For every event that was logged, a custom color can be assigned such that it was highlighted and discoverable when scrolling through. Many of the staff felt that this - along with other UI elements - were very beneficial.

The main aspect of the app is its customizability. This gives the system the power to potentially replace more forms that are used in a similar setting. Users can go in and add or edit the forms to suit their needs. Forms can be laid out and presented in such a way that all users feel comfortable with. Many of the staff felt that form customization was easy to use and these electronic forms could replace more paper-based forms.









The application in general was received well. Ninety percent of the users mentioned that they would like to use the app frequently, while everyone agreed that the app does a good job of serving its purpose of a good documentation system. In any use of technology, it is always fair to think whether or not the system would lose your progress or updates. In this app, as users navigate from one event to another, all partial data entries are stored for when the users return to it. Most of the surveyed staff were comfortable with using the app without having to fear that a mistake would lose their data. On the whole, 80% of users were confident in using the system.

7

FUTURE WORK

There are always improvements that would make the system better. It is important to get constant feedback from the medical staff in order to plug in any holes and make the system better on the whole. For this app, one of the key additions is that of multi-device form management. As of right now, when editing and creating data documentation forms, it is only for the device where the changes are being made. The straightforward approach to improve the system would be to upload the JSON document powering the forms on a server where different devices can fetch from. It is also important to note that right now, anyone can go in and make changes to the forms. Going ahead, there should be a system of *Roles* implemented where only a few users have access to the customization engine to add, edit or create documentation forms.

Another feature which would greatly benefit the staff is to transmit all the logs during an emergency to the internal servers which would attach it to the patients current medical data. This would help in a move towards a digitally-oriented environment with minimal dependencies on paper based forms. This needs to be done securely and made sure that only the clinical staff would have access to the data in the future.

As the staff members would get more comfortable with these forms, it would be beneficial to create different forms for different purposes within the app. Because the app

was built to be very flexible, adding additional UI elements in the future would not be a major task. Developers would simply need to add in or build their new element for the new item type and it would be good to go.

Although the medical staff were surveyed on the use of the application, the staff evaluated the application on how they perceive the system would perform in a clinical setting. The application should be evaluated in a real scenario or even a mock one where the staff would get a better idea whether such a system would be efficient and improve the outcomes of resuscitation events.

REFERENCES

[1] Statista The Statistics Portal. Number of tablets sold quaterly. http://www.statista.com/statistics/272070/global-tablet-shipments-by-quarter/.

[2] Statista The Statistics Portal. Number of apps available in leading app stores as of july 2014. http://www.statista.com/statistics/276623/number-of-apps-available-in-leading-app-stores/.

[3] Statista The Statistics Portal. Forecast of Mobile App Downloads. http://www.statista.com/statistics/266488/forecast-of-mobile-app-downloads/.

[4] SG Pelletier. Explosive growth in health care apps raises oversight questions. *AAMC Reporter*, 2012.

[5] Maged N Kamel Boulos, Ann C Brewer, Chante Karimkhani, David B Buller, and Robert P Dellavalle. Mobile medical and health apps: state of the art, concerns, regulatory control and certification. *Online journal of public health informatics*, 5(3):229, 2014.

[6] Fred Selck and Sandra Decker. Health information technology in the emergency department. http://academyhealth.org/files/2013/monday/selck.pdf.

[7] Janet E Jackson and Amy S Grugan. Code blue: Do you know what to do? *Nursing2015*, 45(5):34–39, 2015.

[8] HealthIT.gov. US Department of Health and Human Services. Benefits of electronic health records, 2015.

[9] William Kaye, Mary Elizabeth Mancini, and Tanya Lane Truitt. When minutes countthe fallacy of accurate time documentation during in-hospital resuscitation. *Resuscitation*, 65(3):285–290, 2005.

[10] Jack M Peace, Trevor C Yuen, Meredith H Borak, and Dana P Edelson. Tablet-based cardiac arrest documentation: A pilot study. *Resuscitation*, 85(2):266–269, 2014.

[11] Wasif Bokhari. Improving resuscitation outcomes during intensive patient care: Development and use of an iPad-based resuscitation code blue sheet.

[12] Eliot Grigg, Andrew Palmer, Jeffrey Grigg, Peter Oppenheimer, Tim Wu, Axel Roesler, Bala Nair, and Brian Ross. Randomised trial comparing the recording ability of a novel, electronic emergency documentation system with the aha paper cardiac arrest record. *Emergency Medicine Journal*, pages emermed–2013, 2013.

[13] William Kaye, Mary Elizabeth Mancini, and Tanya Lane Truitt. When minutes countthe fallacy of accurate time documentation during in-hospital resuscitation. *Resuscitation*, 65(3):285–290, 2005.

[14] Jeoffrey K Stross. Maintaining competency in advanced cardiac life support skills. *Jama*, 249(24):3339–3341, 1983.

[15] American Heart Association. Get with the guidelines resuscitation recognition criteria.

[16] Paul Hegarty. More Xcode and Swift, MVC, 2014. Lecture Notes, Stanford University.

[17] Steve Burbeck. Applications programming in smalltalk-80 (tm): How to use model-view-controller (mvc). *Smalltalk-80 v2*, 5, 1992.

[18] Healthcare Information and Management Systems Society. Selecting a mobile app: Evaluating the usability of medical applications. http://s3.amazonaws.com/rdcms-himss/files/production/public/HIMSSguidetoappusabilityv1mHIMSS.pdf.

APPENDIX A

APPLICATION USER STUDY

Working with the application.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
I think that I would like to use this system frequently.					
I found the system unnecessarily complex.					
I thought the system was easy to use.					
I think that I would need the support of a technical person to be able to use this system.					
I found the various functions in this system were well integrated.					
I thought there was too much inconsistency in this system.					
I would imagine that most people would learn to use this system very quickly.					
I felt very confident using the system.					
I felt the system was effective for its purpose.					
The app had a clear, clean, uncluttered screen design.					
The app minimized the number of steps it took to complete the tasks.					
I felt automatic time logging reduced the number of time related errors.					
Information presented on screens was easy to comprehend quickly (ex: custom colors for events).					
Using Alerts for events made me comfortable for timed events.					
I felt confident that I could make a mistake without losing my work.					
I felt that customizing forms was convenient and more forms could potentially be replaced by this system.					

How long have you been working in this position? _____

On a scale of 1-5 (5 being the highest)

	1	2	3	4	5
How comfortable are you using a tablet device (like an iPad) or other technical devices?					
How comfortable are you with typing on a tablet device?					
How comfortable are you with electronic forms?					