Alternative Keyboard Input In a Digitalized Trauma Journal

Milad Taba Tabaei Ranjbar
Abstract

Alternative Keyboard Input In a Digitalized Trauma Journal

Milad Taba Tabaei Ranjbar

Trauma personnel have limited time to stabilize their patients depending on their injuries and overall state. This introduces problems related to how the whole trauma team can fast and accurately note and find specific injuries or other types of disorders. Akademiska sjukhuset in Uppsala together with Uppsala University has a vision of introducing a new tablet application that replaces the original paper trauma journal. This thesis describes the development of the alternative keyboard solution which is a part of the digital trauma journal application. The purpose of the application and its alternative keyboard solution is to reduce the latency of information within the trauma team and to portray what information has been gathered from the patient. The work is done based on the proof of concept design. Based on the design, prototypes of their proposed alternative keyboard were created in an iterative process to refine and build the keyboard solution to demonstrate its usefulness. The conclusions are that the alternative keyboard solution does provide a more efficient way of handling the input phase for the T1 nurse. It has all the necessary components available the T1 nurse would have to use and further enhances the ability to be able to display their content across several other devices in real time. If the application and its keyboard solution are proved successful, Akademiska sjukhuset is willing to pursue further towards utilizing the application into their daily methods and practice.
Acknowledgements

I would like to express my appreciation to my supervisor Mats Lind, who provided and made me involved in this thesis project. His support and guidance have been valuable throughout the whole project.

I am grateful to my reviewer, Mikael Laaksoharju, for his support and advice both during the development of the application and the report itself.

I would also like to thank Diane Golay and Petra Söderlund for giving me access to their design, facts and other material needed to implement my part in this thesis project successfully.
Contents

1 Introduction 3
  1.1 Purpose ......................................................... 4
    1.1.1 Questions Investigated .................................. 4
    1.1.2 Delimitations .............................................. 4
  1.2 Structure of Thesis .......................................... 5

2 Background and Theory 6
  2.1 The Trauma Journal ........................................... 6
  2.2 Proposed Application Design ................................. 10
  2.3 Proposed Keyboard Design .................................... 13
  2.4 Prior Work ....................................................... 14
  2.5 Alternative Layouts ........................................... 14
  2.6 Use Of Gestures ............................................... 15
  2.7 User Experience ............................................... 15

3 Constructing the On-Screen Keyboard 16
  3.1 Methods .......................................................... 16
    3.1.1 Investigation of Given Build ............................ 16
    3.1.2 Interviews .................................................. 16
    3.1.3 Iterative Process ......................................... 17
  3.2 Frameworks ....................................................... 17
    3.2.1 NodeJS ...................................................... 17
    3.2.2 SocketIO .................................................... 17
    3.2.3 Cordova/PhoneGap ......................................... 17
    3.2.4 AngularJS .................................................. 18
    3.2.5 Express ..................................................... 18
  3.3 Hybrid Mobile Applications ................................. 18
  3.4 Results .......................................................... 20
    3.4.1 Application Flow ......................................... 22

4 Fine-tuning the Keyboard 27
  4.1 Hardware Support .............................................. 27
  4.2 Prototype and Application Design ........................... 27
  4.3 Methods ........................................................ 28
    4.3.1 Usability Evaluation ..................................... 28
  4.4 Results ........................................................ 29

5 Analysis ............................................................. 30

6 Discussion .......................................................... 31

7 Conclusion .......................................................... 33

8 Future Work .......................................................... 33
  8.1 A Final Word .................................................. 34

9 References .......................................................... 35
1 Introduction

Within the medical domain, trauma refers to injury or damage caused by external physical force, following from, for instance, vehicle accidents, falls and knife stabbings. It is the leading cause of death and disability in the age group 1-46 years old and the third biggest cause across all age groups [1]. In the United States, trauma injury accounts for 30 percent of all life years lost [2].

When a trauma victim arrives at a hospital, the first priority of the medical personnel is to stabilize the patient, after which they focus on identifying the need for further treatment. The medical personnel form a trauma team, which can typically include an attending surgeon, surgical residents, an anesthesiologist, an orthopedic surgeon, nurses, a respiratory therapist, a pharmacist, and an x-ray technician [4]. Each member of the trauma team has their role with defined responsibilities.

Information about a trauma is noted in a so-called trauma journal, which is carried and utilized by the so-called T1 nurse, who is responsible for recording information in the trauma journal [3]. It is crucial that correct data has been recorded in the journal, for the surgeon, neurologist or any other specialist to be able to give the patient proper medical treatment as fast as possible.

Within this multi-professional team, there are several problems with today’s pen-and-paper based practice, such as the trauma journal getting cluttered with unrelated information, wrong type of information in wrong sections, and difficulties to share information between team members. These problems could potentially be mitigated by a digital trauma journal, which would simplify the situation for the T1 nurse.

The motivation for this thesis is to contribute to a digital version of the trauma journal. The use of a digital trauma journal does not only have to be fast, responsive and accurate but the information in the handheld trauma journal also must be possible to present on large, wall-mounted displays in the trauma resuscitation room, in order to share information with the whole trauma team.

Prior work has revealed that this is not a trivial design problem. The Trauma Pen [5], for instance, is a digital pen that captures and sends information to wall-mounted displays. It allows collecting live data as the T1 nurse fills in a familiar paper-based trauma journal. This type of input has proven good for unobtrusively collecting live data as the T1 nurse does her daily work. However, there are also design challenges due to the combination of regular pen-and-paper and digital information capture. For instance, since the Trauma Pen captures handwriting, the legibility of information depends on the nurse’s penmanship.

Other attempts, display the overall difficulties that nurses and other specialists have when it comes to introducing new methods and standards in their everyday work as described in [6]. A lot of advantages can be gained by using technology, but for medical personnel to be able to use this type of application in their daily work, it is necessary to create a simple, good design that requires little to no prior experience.

This thesis is written within the context of a bigger project aiming to produce a proof-of-concept prototype for the trauma resuscitation process at Akademiska Sjukhuset in Uppsala. My contribution to this project is to determine how to design a software keyboard so that it is efficient and reliable for use by the trauma personnel in the stressful trauma resuscitation situation.
1.1 Purpose

The purpose of the thesis is to design and construct an alternative, on-screen keyboard solution that will enable fast and accurate data entry for use in the digital trauma journal proposed by Golay and Söderlund [7]. Once developed, the on-screen keyboard will be evaluated in relation to the expectations of the T1 nurse.

1.1.1 Questions Investigated

- When constructing a fast and accurate on-screen keyboard, can basic software frameworks be found that in a good way will support its construction? And using the found frameworks, how could the proposed general design be constructed?

- How should the on-screen keyboard be designed in detail from a visual and interactive point of view for it to be fast and accurate?

1.1.2 Delimitations

Initially, the aim for thesis project was to create the whole application as a proof of concept, while also demonstrating some functionality. However, both I, the designers and other stakeholders involved from Uppsala University noticed that this is out of scope for this particular thesis. The complexity of the project could possibly result in a lot of time being spent into elaborating design choices rather than implementing the tools needed. This thesis was therefore redirected towards creating the alternative keyboard tool. The alternative keyboard is a tool proposed by the authors Golay and Söderlund [7] who designed the application to make it easy, fast and accurate to input useful data in a digital version of the trauma journal.

During development, no implementation has yet been done towards any database. Figure 10 in this thesis, the application architecture, displays how a future database handler and database would be connected.

No security measures or handling have been implemented regarding the integrity of user data. Only brief discussion about the integrity of the data on safety-related issues is brought up and discussed in this thesis.
1.2 Structure of Thesis

- Section two provides a background and all the relevant theory needed to understand the different design and concepts done in previous, and current work.

- Section three describes the following: the different methods and investigation done to find the proper frameworks needed to create the keyboard solution; the methods used to create the keyboard solution with the chosen frameworks; the resulting solution.

- Section four finishes the thesis project and addresses the second research question to how the keyboard can be designed in greater detail to accomplish the different tasks it will be assigned in the future in an adequate way.

- Section five is an analysis of the results discussing both relevant theory and all the relevant findings during this thesis project.

- Section six discusses the project and its implications.

- Section seven contains the conclusion.

- Section eight discusses possible future work.
2 Background and Theory

Understanding how the pen-and-paper version of the trauma journal is being used and handled by the T1 nurse determines the design for the digital representation. It is crucial to understand how the different pages, their individual sections, color codings and other relevant information are being used, as well as understanding the proposed design created for the digital representation of the trauma journal and the general theory about prior work done within the field.

2.1 The Trauma Journal

The purpose of the trauma journal is to quickly identify injuries or any other kind of disorder [8] shown in Figure 1. It consists of four parts. The first section of the journal is the prehospital page. This is where the T1 nurse inputs the information she gets from the ambulance personnel transporting the patient towards the hospital. The data in the prehospital page is used to determine if the patient is in any kind of trauma. Depending on the injuries and state of the patient, the T1 nurse is determining if any trauma alarm should be signaled to treat that patient as soon as she arrives at the hospital.

![Figure 1: Prehospital Page [8]](image-url)
The second page represents the initial evaluation and interventions depending on the type of injury or disorders the patient has occurred. On this page, many vital components are being signed and addressed to accurately get correct information about the kind of damage, breathing, etc. Many interventions are also listed on this page to carefully follow the story of how the treatment is proceeding as shown in Figure 2.

Figure 2: Primary Evaluation & Interventions [8]
The third page, seen in Figure 3, is where the T1 nurse fills in all type of monitoring data from the patient. Pupil size, medicine, pulse, etc. Everything is noted and placed within their categories depending on the treatment in the trauma clinic.

<table>
<thead>
<tr>
<th>Patient</th>
<th>Medication</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>AP</td>
<td></td>
</tr>
<tr>
<td>B</td>
<td>SAT</td>
<td></td>
</tr>
<tr>
<td>C</td>
<td>Cl</td>
<td></td>
</tr>
<tr>
<td>D</td>
<td>RLS</td>
<td></td>
</tr>
<tr>
<td>E</td>
<td>IV5</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3: Monitoring [8]
The fourth and last page of the trauma journal consists of information that describes what kind of accident, trauma or disorder the patient had before treatment. Damage report, information regarding close relatives and other miscellaneous information are also written on this page shown in Figure 4.

Figure 4: Summary [8]
2.2 Proposed Application Design

This thesis is a collaboration with HCI (Human-Computer Interaction) students Golay and Söderlund who have created the design for the application [7]. The design suggestions from their work that are relevant for this thesis will be included and discussed.

The first page that the T1 nurse interacts with within the application is the start page as shown in Figure 5. It has been developed and constructed in this particular way to get all the vital information the original trauma journal initially had and to make it possible to display in portrait mode.

Based on the current trauma journal, the difficulty lies within creating a digital application which potentially can replace the current paper journal. However, this is not an easy task. It requires clever design choices to accurately, responsively and quickly input data as a patient is undergoing their treatment in the trauma clinic. The design concerns require a deeper understanding of the proposed design created by Golay and Söderlund [7] to determine what type of functionality and layout is appropriate for the proposed keyboard solution.

Figure 5: Starting page for incoming patient.[7]
The ambulance report includes the information about the current patient status and other vital information shown in Figure 6. This page is filled in by the T1 nurse who gets the information from the ambulance personnel who are the first responders that interact with the patient determining the different damages and status. This information is immediately transferred to the trauma personnel to gather as much information as possible.

![Ambulance Report](image)

Figure 6: Ambulance Report [7]
The apprehend action page displays all the possible actions that can be taken depending on what section of the trauma journal the T1 nurse is trying to input data. Figure 7 shows the different options and their effects are shown in the right portion of the application which then affects the results gathered and displayed in the left part of the application. Having a progressive left and right side transferring and displaying data as the user proceeds gives valuable visual feedback and flow throughout the whole application, while also saving some screen space utilizing this type of behaviour.

![Figure 7: Apprehend Actions](image-url)
2 Background and Theory

2.3 Proposed Keyboard Design

In order for the application to become fast, responsive and accurate regarding its usage, the alternative keyboard solution has a distinct and straightforward design with the intended users in mind. To encourage and propose a customized keyboard instead of utilizing the already familiar and well-developed soft keyboards of tablets and mobile devices is motivated by the fact that the proposed design by Golay and Söderlund [7] is tailored to the user's context. This means that the design of the alternative keyboard is adapted to best suit the T1 nurses. It is based on the QWERTY layout, which is the easiest layout to model a keyboard. Its familiar design and layout allows users who already are familiar with the QWERTY layout to further increase their performance [9] by adding context-specific functionality and features.

The keyboard design proposed by Golay and Söderlund [7], shown in Figure 8 is not final but merely a visual representation that during development is open for change to improve the core functionality and availability.

![Proposed Keyboard Design](image)

Figure 8: Proposed Keyboard Design [7]
2 Background and Theory

2.4 Prior Work

Prototypes such as the TraumaPen which have a paper-digital interface created by Sarcevic, Weibel, Hollan, Burd, Randall S [5], showed potential in supporting the trauma team in the dynamic and safety-critical setting of the trauma bay. This, however revealed limitations and challenges using said type of technology. Some constraints and challenges that occurred during the development of the TraumaPen were related to users being able to real-time present the input data. There was too much information that required more large-screen displays and handwriting recognition in the dynamic work setting.

Keyboard and mouse using existing electronical flow sheets is another concept that has been tested in three different hospitals in various configurations with one or more terminals set up in the trauma bay [7]. In all three hospitals the configuration of the terminals were similar to each other in regards to both process and use. In order to fully be able to utilize the system properly, the nurses had to be trained prior to their implementation which all varied from a 3 hour session to monthly 3 hour sessions before the nurses were able to fully use the system with the primarily task of documenting the trauma resuscitation.

Data entry through ASR or speech recognition systems is another idea that has not yet been attempted for the trauma resuscitation scenario but has been considered and discussed in published literature [7]. However, with the trauma bay being a very noisy environment having several of the trauma team communicating at the same time makes it hard to fully utilize any kind of ASR systems. It would be difficult for the system to differentiate background noise from actual commands while also contributing with additional cognitive effort in an already stressful and highly cognitive workload setting.

Based on these findings, it is clear to say that redefining how the trauma team work and behave using new technology is not an easy task. However, gathered all the insight and knowledge about these challenges determine and affect how future design and prototype such as ours, can potentially solve this.

2.5 Alternative Layouts

Viewing other keyboard layouts besides the proposed QWERTY design made by Golay and Söderlund [7], the Dvorak layout is also an interesting alternative. The Dvorak layout shown in Figure 9 is distinguished by the different placement of its keys in comparison with QWERTY. According to Norman, Fisher [9], having this type of key layout allows the Dvorak layout to show a 5% performance gain over QWERTY [9]. The result, however, is based on how well the different users already knew the various keyboard layouts. With the noticeable performance gain, the Dvorak layout can offer, it still is not appropriate to use it in this thesis project, due to not many people are familiar with its layout.
2 Background and Theory

2.6 Use Of Gestures

The possibility of utilizing gestures have been considered in order to do different type of commands within the keyboard solution. However, this is all depending on how the gestures can be used efficiently to produce better user experience. iOS applications by default allow particular type of gestures within some applications such as left and right swiping maneuvers on different objects, being able to drag the main page away to view underlying pages and more. Framework7 by iDangero.us [17] allows the application to easily get the look and feel from how a native iOS app behaves. Features such as gestures and different type of swipe maneuvers are all included within Framework7.

2.7 User Experience

In interaction design, the goal is to develop useful interactive applications, meaning that they should be easy to learn, efficient to use and provide a pleasant experience throughout the application. It is also important to consider what type of mental models or activities people usually use when working with a particular kind of alternative solution or during their daily routines based on their environment, the technology available and how we as humans behave and interact with individual objects.

Although the on-screen keyboard solution is developed for a particular type of users, they are a broad group, meaning that it is not always possible in advance to know what kind of users they are, how they interact with the solution and respond towards it.

Löwgren introduces his model based on usability, which is relevant when developing the alternative keyboard solution, as REAL: Relevance, Efficiency, Attitude and Learnability [10]:

- **Relevance** is described by how well the system serves the users’ needs.
- **Efficiency** states how efficient and easy it is for a user to complete certain required tasks.
- **Attitude** describes the user’s subjective feeling towards the system.
- **Learnability** explains how a new user learns the system and maintains knowledge of how to use it.
3 Constructing the On-Screen Keyboard

Prior to start the software development process for the digital keyboard solution, certain methods have to be set and acknowledged to determine what the requirements, specifics and other important details are needed, before starting the actual development. Methods such as conducting interviews, evaluations, and investigation of the current build are some examples of different methods. Knowing which one to use depends on the situation. Most of the benefits can, however be seen by using the combined insight from different methods to fulfill the requirements depending on the context of use [11].

Software development is an iterative process where several methods and their results are used in order to meet the demands of the consumer. User observation, design proposals, and refined solutions, together with evaluations define how the proposed system should work in its intended environment.

3.1 Methods

The methods discussed in this paper are common paths to undertake during software development. This is done by gathering all the different requirements needed for the keyboard solution, from both the lead designers and stakeholders from Uppsala University involved in this project. The methods findings and their results are discussed and evaluated.

3.1.1 Investigation of Given Build

Upon the start of this thesis project, the given build already had a current core architecture utilizing NodeJS, Express, AngularJS and Socket.IO. This lead towards investigating and understanding how and what the different types of frameworks did and how they affected the application in its current state. Based on that, different assumptions, and code snippets were created to get more understanding and trying to realize the various parts needed for the keyboard solution to be successful.

3.1.2 Interviews

Several interviews were conducted during the development of the keyboard solution, with the lead designers of the project, Golay and Söderlund [7], but also with two stakeholders from Uppsala University, Mikael Laaksoharju and Mats Lind. The interviews consisted of talking through each part created during development, getting their feedback to make further adjustments towards the different relevant tasks the functions were assigned to solve.

A usability evaluation interview was also conducted with the lead designers Golay and Söderlund [7]. The evaluation was performed near the end of the thesis project when almost all of the essential functions and their behaviour was implemented.
3.1.3 Iterative Process

Once the given build was investigated, small code snippets were created and functionally tested individually. When the different code snippets were evaluated and tested according to their task, if successful, they would be implemented in the core software. When all the pieces are then evaluated and tested, the whole application and all the small components were tested in correlation to each other to find any weaknesses that could be improved, given the time left on the thesis project. The concept of using divide and conquer in this type of project proved to be successful as the code snippets independently were easy to build and evaluate, but as a whole, difficult to realize how they might work together.

3.2 Frameworks

The alternative keyboard solution is built utilizing several frameworks in order to best support the features required, based on the findings made by Golay and Söderlund [7]. All of the following frameworks contribute within the application in various ways and simplify many of the different programming aspects in regard to client-server communication, data binding, deployment, etc.

3.2.1 NodeJS

NodeJS is a JavaScript runtime built on Chrome’s V8 JavaScript engine [13, 14]. NodeJS uses an event-driven, non-blocking I/O model that makes it lightweight and efficient. It is built to create scalable web applications with optimized throughput [14] and enables the developers to use JavaScript on the server side of the application. A variety of tasks such as listening on TCP ports or handling HTTP get or request methods that react to certain events is easily done with NodeJS which is powerful and very easy to use in applications such as ours.

3.2.2 SocketIO

Socket IO is a JavaScript library that enables real-time bi-directional communication between web clients and servers [17]. It consists of two parts, the client side library that runs in the browser, and a server side that runs for NodeJS. The Client-Server communication is handled through SocketIO which in turn allows broadcasting functionality across all connected devices on the web server. This to display the keyboard entry input.

3.2.3 Cordova/PhoneGap

Apache Cordova is an open-source mobile development framework. It enables the developers to use their knowledge of HTML, CSS & JS to develop mobile or tablet applications with little effort. Applications execute within wrappers targeted to each platform and rely on standards-compliant API bindings to access each device’s capabilities such as sensors, data, network status, etc. Deployment-wise, Cordova/PhoneGap also provides additional support for developers by being able to deploy your application towards their cloud-based build service. This allows developers to upload their ongoing or finished application to Adobe® PhoneGap™ Build cloud service to distribute the application [19] efficiently, whether it is for evaluation, collaboration or distribution of the application.
3.2.4 AngularJS

AngularJS is a framework for dynamic web applications. It lets the developer utilize web technologies such as HTML, as a template language and extends its syntax to further express the application components clearly [16]. AngularJS’s data binding and dependency injection eliminate much of what the code developers otherwise have to write. Utilizing the models created by AngularJS together with data binding allows for an automatic way of updating the view when the model changes, as well as updating the model itself whenever the view changes. This eliminates much of the DOM manipulation that otherwise has to be handled by the developer. All of these functionalities happen within the browser, which makes AngularJS ideal with any server technology.

3.2.5 Express

Express is a minimal and flexible NodeJS web application framework that provides a robust set of features for web and mobile applications. Its main purpose is to provide the MVC (Model-View-Controller) architecture on the server side [15].

3.3 Hybrid Mobile Applications

Hybrid mobile applications are developed using the web technologies HyperText MarkUp Language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS). These web technologies together with the help of Cordova/PhoneGap framework [12], allows developers to deploy their application to multiple platforms. The PhoneGap framework allows the application to run locally on most operating systems including Apple iOS, Android, Windows and more, with little or no modifications necessary.

With applications that are more computationally intensive, other programming languages such as Objective-C or Swift, would be more appropriate to use towards the Apple iOS operating system. However, hybrid applications using Cordova/PhoneGap and NodeJS behave and interact almost at the same level of performance and responsiveness as a native application. There is almost no distinction between these that can be seen by the keen eye. One of the primary challenges, to produce such a complicated application in limited time with multiple stakeholders involved, is what framework to choose. This leads towards some questions that had to be answered in order to pick a good framework appropriately. Such as, what framework is suitable for rapid development? Easy deployment? Good for evaluation? These are just a small collection of questions, but for this project Node JS together with the frameworks AngularJS, Express and Socket IO became the obvious choice.
3 Constructing the On-Screen Keyboard

In Figure 10 the application architecture displays how the different frameworks chosen for this thesis project work together. The important aspect of Figure 10 is that the architecture applies to all devices running the application. On the client side of the keyboard solution, Angular together with Jade, CSS, and Javascript allows us to model and display the graphical content created for the keyboard solution. All the different pages for the application are loaded from the server via HTTP request. To display the text content written from the keyboard solution in real-time on various other devices, the application connected via web-browser is handled by the Socket IO client handler sending messages through a web socket pipeline [17]. When the message is received on the server side, Socket IO broadcasts the incoming message towards all the other devices connected to the server via a browser. In the future, when the application and the keyboard solution is ready for deployment, a future database handler is implemented in this iteration of the architecture with its database to demonstrate how this would be applied in the current system. Also, the additional image of PhoneGap utilizing its frameworks describes what type of frameworks within the PhoneGap framework is deployed in future work.
The core project structure of a Node JS application utilizing Express [15], AngularJS [16] and SocketIO [17] follows the traditional MVC (Model-View-Controller) design pattern. It allows for great flexibility and satisfies having high cohesion and low coupling. Meaning that the dependencies and modularity between the different code files are flexible for future improvements or changes. NodeJS together with the other frameworks used in this project are all easily configured in their respective files. All NodeJS configurations in this project are made in the file app.js. Here we initialize most of our selected libraries, determine our view engine, etc. SocketIO handling and all the routing using the Express framework for our different views are handled in the routes folder. The Public folder is where we handle all the client-side services and make sure their logic and styling are functional and work as intended. Working in this type of structure for any software application gives a good overview of all the components and allows for greater flexibility regarding new modules for future implementations.

Node JS [14] together with AngularJS [16] and Socket IO [17] allow the developer to easily utilize model-based data binding with additional server handling through the web socket protocol. This allows forms and other valuable data to be sent to the model created by the developer. This particular practice becomes handy especially in this kind of application where a lot of relevant information is being written within several specific fields, inside the application. Therefore, these fields need to pass their data in a good fashion before being able to display their content.

3.4 Results

The given build at its core architecture has all the typical components needed, to produce a functional web application. It utilizes smart front and back-end tools to produce and display relevant information, with the least amount of stress trying to set up various types of dependencies to work. With the feedback gathered from Golay and Söderlund [7], several visualization issues were resolved. The keyboard solution was also tested against eight different cases the T1 nurse would typically use for writing information. The keyboard solution as tested by Golay and Söderlund [7] passed all of these tasks. This indicated that the core functionality and visual representation of the keyboard solution were successful in the sense of accomplishing the given tasks.

The Application is created with Node JS together with the frameworks AngularJS and Socket IO for handling the models needed and their data bindings. To test the application and the keyboard solution on a desktop, Node JS is launched with its local server. Once the local server and application are up and running, the keyboard solution can easily be tested and evaluated. To verify the application on the iPad Pro [20], accessing the application through the own device browser and the local area network IP address will connect towards the application. This allows the program to run directly on the iPad Pro [20] with similar behaviour and looks as it would be if deployed through frameworks such as Cordova/PhoneGap [12]. By accessing the application through the localhost on the same network, allows the development process regarding actual deployment of the application to be handled in a later matter.

The keyboard solution developed for the trauma application has since its initial development changed both in its design and functionalities to better compliment what was proposed by Golay and Söderlund [7]. Visually, the design of the alternative keyboard solution is almost the same, some visual elements such as the text area, five most common selectors, numpad and the keyboard have been slightly modified regarding their visual design to best suit the iPad Pro device with its specified resolution. Moreover, functionalities such as backspace have been more redefined to better suit how an actual
user, the T1 nurse, would interact with the keyboard solution entering any text and later on be able to edit depending on the situation. Adding the highlighting functionality upon the backspace function allows the T1 nurse to use the touch interface to choose any text she wants to edit and to continue input text. These kinds of tweaks and design decisions were realized by creating several prototypes of the different functionalities before integrating them into the project entirely.

When constructing an alternative keyboard solution based on the design proposed by Golay and Söderlund [7], it is not always clear how it will work or even be better than the device specific soft keyboard. That means that there are several risks taken when considering a development over an already implemented solution within a particular appliance. The implemented soft keyboard solution within an iPad, for example, is by all means an excellent keyboard. However, if used in the trauma resuscitation scenario, it will not be enough to enable the T1 nurse fast data entry. Based on its current layout and the different options the soft keyboard has available. Therefore, doing the keyboard solution for the trauma application and making sure it does apply to the particular setting and scenarios, will hopefully enable the T1 nurse to access faster data entry within the trauma application. If the alternative keyboard solution is better than the already implemented, soft keyboard solution, it has to be determined by conducting usability evaluations and tests in its designed setting.
## 3.4.1 Application Flow

When the application is started, and all the configuration and settings have been set for the T1 Nurse, the user has some options going through the whole page which is a digital render based on the paper sheet trauma journal as shown in Figure 11. From this page, the T1 nurse has a digital representation of all the relevant fields and sections as previously known from the paper version of the trauma journal.

To now access the alternative keyboard solution entered in the digital version, the T1 nurse clicks the keyboard icon which all are based next to the “OK” button in each respective field.

![Digital Trauma Journal](image)

Figure 11: Digital Trauma Journal
Once the user has clicked the "OK" button, the keyboard solution for the trauma journal slides in from the bottom of the screen. Once in position, the keyboard is a modal popup view in front of the digital trauma journals main view which has been blurred out slightly.

![Keyboard Solution](image)

**Figure 12: Keyboard Solution**

Within the keyboard view, the user now has the ability to input any message that corresponds to its field. Displayed in Figure 12, the user can either use the numpad, the QWERTY keyboard or the five most common states gathered from the observation and the data collected by Golay and Söderlund [7]. The five most common states are so far programmed in such fashion that it allows the developer to have a more dynamic approach if the words tend to change with time as the trauma resuscitation will evolve. The text for the five most common state buttons does, therefore change depending on which field the user is upon, allowing the application to dynamically adapt if new common states would occur in the future. Noticeably the QWERTY keyboard characters...
are in lower case, while the five most common states are in upper case. The motivation for this decision is to further emphasize the visible differences between the words that the users input via the keyboard or if the users choose to input by using the common states. Another important aspect to why the common states have more benefit of being in upper case, is that the common states are keywords that describe the critical states a patient can be in when doing the assessment. Meaning that the words MEDVETSlös or AVSVIMMAD, for example, are valuable keywords that describe the patient’s current state. It allows the T1 nurse to identify these key states in the different text fields quickly, especially in cases where there might be a lot of text written. Whether or not the lower or upper case characters, both for the QWERTY keyboard or the common states, bring any value needs to be evaluated in an actual work setting to best tailor this accordingly to their needs.

Once the user has input any form of data within the text area, closing the keyboard solution is made by tapping outside of the modal view. This also saves any input text within the text area and sends the string towards the server using AngularJS data-binding and model controls through Socket IO. Once the model has handled the string and sent to Socket IO, the intention is to store the data within the hospital’s database which handles the form and all the data concerning the trauma journal. However, this part is not yet implemented caused by this project still being a proof of concept.
Figure 13: Keyboard Solution, New Common States

Figure 13 displays how the five most common states have now been changed to other common states that would be used in for the “AF” keyboard view. Depending on the keyboard view open for each field, the new common states is easily changed to match accordingly against the fields, and their common states as the application is used.
3 Constructing the On-Screen Keyboard

Figure 14: Editing Text

Seen in Figure 14, editing the text in the keyboard solution by deleting certain words or characters, is done by highlighting. The highlighting sequence is started by the user interacting with the device touch interface. This is done by selecting any given text or position in the text area and start a dragging movement over the target. Upon release of the touch interface, the highlighting event is stopped, and the target is visually coloured to emphasize the selected target to be modified. When any part of the text is highlighted, the user can now press the backspace button to remove that object. If nothing is highlighted and the user presses backspace, it will edit any text character by character. Once removed, the user can then type a new word in its place and continue wherever they might be at that moment. Utilizing this kind of feature with a touch interface, gives the keyboard solution an intuitive feel when the users need to edit any part of the input text.
4 Fine-tuning the Keyboard

4.1 Hardware Support

To visualize the application based on its created design, the Ipad Pro together with its
companied pen is the device of choice for this project. It enables the user to utilize
the extra screen size compared to the original Ipad, to enter data more efficiently. The
iPad Pro has two models with different screen sizes. The smaller one which is a 9.7-inch
screen with the pixel resolution of 2048 x 1536. And a bigger 12.9-inch screen with the
resolution of 2732 x 2048 [20]. For this thesis, the goal is to develop the application and
the alternative keyboard solution towards the bigger 12.9-inch model. Thus, this allows
us to utilize more space to get the best possible solutions for this kind of application.
The iPad Pro 12,9 has a 64bit A9X chip processor which enables the iPad Pro to have
great performance with multi-tasking or higher graphical applications. The iPad Pro is
designed and considered to be used for both professional and ordinary everyday use.
The digital application of the trauma journal will be developed primarily for iOS de-
vices. This is based on limiting particular platform specific problems. However, since
the project is using Node JS and Angular framework as its core framework in its de-
velopment, there isn’t much of an issue to make it work for other platforms such as
Android while also resolving any platform specific problems.
The digital trauma application and its keyboard solution are developed with scalability
in mind. Meaning that the application does not only run and perform well on the
desired device, the iPad Pro, but also on all other types of iOS devices such as older
generation iPad, iPhone, laptop and desktop devices. In order to support all of these
different devices with different resolutions, it requires development to be done according
to responsive design. Responsive design allows the visual representation to support these
various types of devices regardless of what resolution they are capable of delivering [21].

4.2 Prototype and Application Design

The trauma journal application is developed and created with the purpose to digitize
information and make it easier for the T1 nurses to handle information more fluently
with little effort. Therefore, the design of the application is vital to match the user’s
requirements and specifically them, as users of the application in mind.

Based on the complexity of the pen-and-paper journal, the digital version and the key-
board solution is not an easy task to design. Therefore, the design of the alternative
keyboard solution and overall design of the trauma journal application is handled by
two master students Golay and Söderlund [7]. These students have during many months
of investigation and research created a design that should meet the requirements based
on their interviews with multiple T1 nurses and how the overall look and feel should be
for the application.

During development of the application, design consideration is taken into account when
the development itself is being done on a regular computer desktop or laptop. Because
later on, the application will be run on the iPad or iPad Pro which both have signifi-
cant resolution differences, in comparison with desktop or laptop’s [21]. Therefore, it is
important to remember and develop a responsive design which utilizes viewport tech-
niques, or CSS media queries to make the application look and run smoothly on the
many types of resolutions different devices may have.
4 Fine-tuning the Keyboard

However, as with prototyping, the design and actual development of the application is an iterative process. Different solutions and design choices that are made at the beginning of the project are well prepared to be changed if it means better user experience or in some cases make more sense from development or design standpoint. For each design, the application and its implemented solutions are evaluated and tested separately and later on together with the designers Golay and Söderlund [7] and other stakeholders involved.

4.3 Methods

4.3.1 Usability Evaluation

To maintain the goal and usability, as defined by REAL [10], user testing of the application can be used to indicate and find issues that can occur and provide valuable input for a redesign. In this project a formative usability evaluation based on Löwgren [10] was done with the two designers Golay and Söderlund acting as proxy users. The attempts and tasks they tested were all based on their findings gathered from their contextual inquiries from the T1 nurse and other relevant actors.

During the evaluation, different text annotations related to patient evaluation taken from real trauma flow sheets (and collected within the framework of the master thesis “Computerized data entry and display in trauma resuscitation: a case study”) were typed with the keyboard. The application was open in Google Chrome on a laptop whose touchpad was used to press the needed keys on the virtual keyboard. Among the entered annotations, there were both full annotations and parts of real annotations. The parts of the annotations that were left out for the test were pieces of information that can be documented via a structured input in another part of the trauma flow sheet application to which the keyboard app belongs. It was considered more realistic to type only the information that would be required to be entered as a free text annotation in the app.
4.4 Results

The usability evaluation conducted by using Golay and Söderlund as proxy users gave valuable feedback and insight towards different issues and functionalities that were overlooked and forgotten during development. Simple visualizations such as being able to display which keyboard and corresponding field are about to enter data were easily fixed by implementing a single label for each keyboard opened for the relevant fields. Annotations such as $O^2$ values were found difficult to enter in the previous versions of the keyboard solution, where the characters were in upper case letters. However, based on these findings several changes were made in order for the T1 nurse to understand the keyboard better. One of these changes was the transition of upper case characters to lower case whilst the most common states changed vice versa, this to further enhance how the T1 nurse is able to note within the application. Other features such as notifying the users that the text field is available for input, by focusing on it visually, was beneficial for the user to be implemented.

The possibility to display the current input data from one device to several others, such as bigger displays in a trauma resuscitation room, was suggested and implemented during development. It’s beneficial to share the data with all the other relevant personnel involved during that session. Utilizing SocketIO to provide the client server interaction allows the application to broadcast any incoming data from one device to others and displaying the data on other monitors [17]. This type of broadcasting technique is quite common in an application such as a real time chat.
5 Analysis

The QWERTY layout for the keyboard solution is the most common and efficient layout to use which takes advantage of users’ familiarity [9]. Having the QWERTY layout together with the 5-word-at-the-time alternatives and numpad based on the design from Golay and Söderlund [7] as shown in Figure 8 allows the users in a fast manner to input data and continue with the rest of the fields within the application. This was one of the primary goals. Adding functionality to the keyboard solution such as popup slide animation, overlay view, backspace, tap on the text and simple modal behaviour allows the user in a good fashion to be able to rather quickly get the keyboard upon activation via the main screen.

Different concepts and designs regarding the input speed of some words have been considered. Some concepts like for example the applications Swiftkey [22] or Swype [23], make it possible to swipe phrases into text by using word dictionaries, or by having a suggested list of words which assist and makes the user input words faster by using this type of gestures and autocompletion [24]. For our project regarding the keyboard solution for trauma resuscitation, the swiping gesture can be both beneficial or not. Based on how the T1 nurse would approach and use this way of input, it would whether this kind of gesture would be beneficial or not. To be able to evaluate if the swiping gesture increases the input speed, it requires an actual field test. Other design suggestions based on the layout of Swiftkey are for example having both the alphabetical, numerical and special characters on the same keyboard, but on different pages that are activated by the user pressing certain icons on the keyboard. This is also a design that needs more development and evaluation time in order to properly be convinced towards that type of design implementation, which would be different from the original design made by Golay and Söderlund [7].

When reflecting upon how well Golay and Söderlunds [7] virtual keyboard solution works compared to the hardware specifics in iPad, Computers, etc, is that this kind of application allows more content specific input. The trauma resuscitation event once ongoing is a hectic moment that does not allow the T1 nurse to have all the time necessary for long valid answers or meaningful data. This hectic moment negates the fact that an ordinary keyboard with all the current functions could be beneficial rather than a custom solution.

Evaluation during development gives a lot of valuable feedback towards the right direction regarding how the application should behave and act. Particularly in this case, for the keyboard, many functions, and actions are based working closely with the design authors, reviewer, and others involved in this project.

When having several frameworks to work on a project like this, which uses Node JS, Angular JS, Socket IO, there are some confusion and time spent on finding the best suitable solution each framework gives, depending on the problem. With that said, much time regarding debugging are in some cases caused by the frameworks themselves, trying to figure out how different assets work rather than actual errors in code. However, the process using NodeJS and all the other frameworks build upon using standard HTML, CSS and JavaScript language which has the benefit of rich documentation available.
6 Discussion

What separates the alternative keyboard solution from the already integrated soft keyboard solutions available on any device, is based on the functionality it offers, such as the numpad and the five most common state buttons. The five most common state buttons in the alternative keyboard solution offer a fast way to input these five states based on studies conducted by Golay and Söderlund [7] looking at previously filled trauma journals. Viewing how the T1 nurse notes information in the old trauma journal before Golay and Söderlund [7] proposed their design for the digital application, all the input was done by hand, and several times the T1 nurse repeatedly entered the same type of information. This causes the T1 nurse to lose some time having to repeat herself several times but in different sections in the trauma journal. Having the proposed five most common states menu, therefore, is a tool which allows the T1 nurse to quickly input text by tapping on of these buttons to enter the specified states the patient is within at the particular moment. They no longer have to go through the effort of actually repeatedly typing this in by hand anymore using this kind of solution. The numpad is another tool also based on what kind of information the T1 nurse entered in the previously filled trauma journals. Having a numpad available at any given time in the keyboard solution allows the nurse to quickly enter any value required in the particular fields that have values as a requirement or simply within a text area. To make the application easy and intuitive for the T1 nurse, a lot of the button placements, visual reactions of the buttons, and core functionality have been considered throughout development.

What motivated this thesis to pursue and develop an own unique type of soft keyboard for tablet devices, is based on the premise to create a contextual alternative keyboard that would have more useful functions and ways of interaction to assist the T1 nurse note data within the digital trauma journal.

To enable the nurse to find the right keyboard for the different fields in the digital trauma journal it is simply done by both visual recognition and unique button placement. Meaning that every field in the digital trauma journal has its unique keyboard button, which upon opening also has the unique field’s label. The T1 nurse can quickly switch between the different keyboard selected for any field by just tapping outside of the keyboard pop up on the iPad Pro. Navigation becomes more fluid and intuitive to use having this kind of interaction integrated.

Since this thesis focuses more on the digital application of the trauma journal and the alternative custom keyboard solution, the application is not entirely finished nor able to be tested quite yet in its designed environment. The alternative keyboard by Golay and Söderlund [7] is implemented and works based on its original design. However, more implementation and design choices on how it should be connected further within the application and further into live production has to be considered. Since the design of the keyboard solution and overall input alternatives is rather simplistic, much complexity lies within however the use of these choices affect and contribute to making it an easier and more accurate type of input choice for the T1 nurse.

At the time Golay and Söderlund [7] created the design for the keyboard solution no prototype had yet been made. When developing the actual solution based on this design, several functionalities and design choices were debatable. Essential functions and their behaviour, such as the backspace functionality. It was never considered how it would behave until a prototype was done and demonstrated its behaviour. What was noticed with backspace, in particular, was the amount of time it took to edit and delete characters when text had been input. This, therefore, becomes inefficient when
the purpose of the keyboard is to provide a faster alternative to the standard keyboard option available. Creating prototypes to get a better sense of how different components behave is beneficial to achieve the goals needed to be successful.

The fact that there are so many people involved during the trauma resuscitation [3] and that the T1 nurse does not only input and write into the current trauma journal but also what the other doctors, specialists, etc. say regarding the different subjects it can easily become an issue. Based on that, almost all the input and values gathered from the doctors and specialist arrive at the T1 nurse orally. Being then able to utilize this digital format of the trauma journal would most likely put the T1 nurse under less stress and effort of trying both to catch up with the actual input being given orally while simultaneously writing it down.

At the current state of the application, based on the design from Golay and Söderlund [7], there is nothing reported about security and how the application will handle the actual data. This is potentially a big problem, at least when connecting the application towards the hospital’s database and records. A sudden stoppage at a Swedish hospital made the hospital lose all their data due to that the database went offline. Furthermore, when the database became corrupted, it was discovered that their backup system had not been working for 34 months [25]. This indicates how fragile and critical the handling of data is and what the consequences are of poor handling or bad design decisions. These kinds of issues where the hospital loses connection to their data is vital for our application where much is connected towards the same type of database and needs countermeasures for a sudden event such as a possible stoppage. Furthermore, the integrity of the data and its accessibility are important topics that need to be addressed when handling any sensitive data.

I believe that digitalizing the trauma journal for all kinds of trauma situations as the patients arrive the hospital is a huge benefit. Not only does it have great potential of providing fast, accurate and responsive information in an instant, but also gains many types of possibilities of displaying and showing the results of the inputted data throughout all the persons involved in a trauma resuscitation. The desired effect could be possible by utilizing hardware and software technologies like how the Google Chromecast [26] displays content towards bigger screens. If this type of equipment is configured with the required software implemented in the application, then this could be a valuable and important feature to have.
7 Conclusion

The alternative keyboard solution allows the users within the digital trauma application to easily, fast and responsive input customized data within their respective fields accessed through the application. This allows the users to use the 5-most-common-states tab, the numpad, and the regular keyboard which is designed and constructed according to the design suggested and visualized by Golay and Söderlund [7]. With all the different functionality, both for frontend and backend component allows the users now to visualize and use the keyboard solution as designed. All the functionality and their results in the application have been evaluated and found sufficient to the extent that was possible within the timeframe of this thesis project.

8 Future Work

Since the application is not yet finished other than the start page for incoming patients and the alternative keyboard solution, some development and work are still needed to release the application into a live production setting. There are several considerations such as the core architecture for the application, functionality still missing from the other pages required in the application that needs further development, and time needed to evaluate the system, both during its development and once finished. For the alternative keyboard solution, additional functionality or behaviour such as swiping gestures for faster inputs would be the next step to increase the performance of the input. The core development is rather straightforward and has rich documentation available for future development [27]. Other variants of solutions can or apply to be implemented depending on how the whole application revises itself during development. Based on these types of issues and considerations, additional development time would address this.

There are other design choices that could be developed or investigated further to get a better visual representation of the graphical interface. One of these design choices would be to investigate further into how well the labelling of each keyboard and their respective fields works. In the keyboard application, the only notification that allow the users to acknowledge which keyboard they have opened is based on the title label which is matched for each field. This could be more elaborated by also highlight the actual button icon in the background of the modal view to get further visual feedback upon which keyboard has been opened. Spending some time on investigating and evaluating these types of design choices could be more beneficial for the end user.

Being able to undo text that has been input via the keyboard or deleted using backspace should most likely have this type of functionality implemented, in order to make sure that we handle all cases if the user by accident would highlight and remove a certain amount of text. The undo functionality has not been mentioned or proposed by Golay and Söderlund [7]. It has neither been implemented in the alternative keyboard solution thus far but should be considered and applied for future improvements of the keyboard solution.

For deployment purposes, as mentioned earlier, Cordova/PhoneGap allows for flexible deployment alternatives that can be built and sent to an iPad or any other smart device. Using Cordova/PhoneGap later on in the process to actual deploy the application fully on the iPad would allow using its services and servers for further integration towards the iPad. With the future integration of Cordova/PhoneGap allows for Framework7, which by default enables gestures usually used in iOS applications [18]. The Framework7
framework should be evaluated and tested together with the stakeholders and different scenarios to determine if the functionality it provides gives more benefit for the T1 nurses that will be using the product.

8.1 A Final Word

Even though the alternative keyboard solution for the digital trauma application is only a small piece of the overall product, it provides all necessary functions sought after, that is to be fast, accurate and responsive for its purpose. Given more development time and additional work towards final revision, the application would surely be beneficial for the medical care in Sweden.
9 References


References


[22] Swiftkey. Swiftkey | About Swiftkey. URL: http://www.webcitation.org/6mmIMrNq1 (visited 12/15/2016)


[27] Krishna Bharadwaj. How Swype works. URL: http://www.webcitation.org/6mmI0McNK (visited 12/15/2016)