Table of Contents

[**1.Introduction** 1](#_Toc531801814)

[**1.1 Purpose** 1](#_Toc531801815)

[**1.2 Overview of Document** 1](#_Toc531801816)

[**2. Overview of Project** 1](#_Toc531801817)

[**2.1 Why was this Project Developed?** 1](#_Toc531801818)

[**2.2 Electronic Task Logger Functionalities** 2](#_Toc531801819)

[**2.3 Deployment** 3](#_Toc531801820)

[**3. Implementation** 3](#_Toc531801821)

[**3.1 Problems and Solutions** 3](#_Toc531801822)

[**3.2 Deviations** 4](#_Toc531801823)

[**3.3 Known Defects** 5](#_Toc531801824)

[**4. Development Time** 5](#_Toc531801825)

[**4.1 Longer Development Time** 5](#_Toc531801826)

[**4.2 Remake** 5](#_Toc531801827)

[**4.3 Learning Experience** 5](#_Toc531801828)

[**5. Summary** 6](#_Toc531801829)

[**6. References** 6](#_Toc531801830)

# **1.Introduction**

This section describes the purpose of the document and its overview, which describes how it’s organized.

## **1.1 Purpose**

The purpose of this document is to describe the Electronic Task Logger by TriNetra. It will briefly describe what the Electronic Task Logger is, while thoroughly explaining why and how the project was implemented. This is to report any deviations from the requirements or any known issues that the development team is still aware of.

This document will also contain TriNetra team member’s views on how the project proceeded, as well as what could have been accomplished if given a larger resource pool, mostly time. This is important if this project is to be revisited or updated.

## **1.2 Overview of Document**

This document is intended to provide information about the Electronic Task Logger and Team TriNetra’s development process. A brief description about the project, a list of its functionalities, and its deployment is provided in the next section. The following section will contain information about the implementation phase of development. This will provide information on errors or issues and how they were corrected, any deviations from the required functionalities, and if any known defects are still in the project. The following section will describe the developmental procedure of TriNetra, which will include their thoughts on if they had been given more time to complete the project, what they might have done differently if the project was to be restarted, and each individual from TriNetra will describe what valuable knowledge was gained from the completion of this project. The last section will be a summary of the project. This will include where it is deployed, who will be using it, and what TriNetra hopes it is used for.

# **2. Overview of Project**

This application was developed for a single user, Dr. Stringfellow. Its purpose is to allow her to electronic log her tasks. This is an improvement on her current notepad and pen technique, as it will track the same data, but allow for quicker and quality of life improvements in searching, viewing, and deleting that data.

## **2.1 Why was this Project Developed?**

The user of this application is Dr. Catherine Stringfellow. She is a professor at Midwestern State University in the McCoy College of Science, Mathematics, and Engineering for the Department of Computer Science. She has numerous tasks to complete and participate in on a daily basis.

She currently documents her tasks by manually writing them in a small notebook, that she has access to when she is in her office. This notebook serves as a data storage device for her so that she can recall information on the tasks she has completed, which may include: the type of task completed, the person or people she was interacting with, the date the task was completed on, the time the task started or was completed on, as well as any personal notes or comments about completing the task.

She is currently looking to improve on this practice by being able to do this task logging on an electronic device. This improvement should include the following features: be able to store the task and the information associated with it, while also being able to be searched quickly for specific tasks and information associated with them, which can be displayed to her for quick access. This will improve searching manually through her notebook, as well as display certain tasks from a filtering process on the same page, which is impossible in her current process. This electronic task logger is a system that will have the features and functionality of her current process of logging tasks, as well as some additional beneficial features which have been stated above.

This application would work by allowing her to select from a dropdown menu the type of task she is currently logging, with the possibility of adding a new task if necessary. She would then be able to select a person, multiple persons, or organizations to be associated with the task if they are necessary, or not be selected if the task was not social. She will then be able to enter a comment associated with this task noting anything of importance and significance she identifies. This information along with the current time and date will be captured and stored to the device. Since this is a stand-alone application, specifically for Dr. Stringfellow, the information will be stored on the device’s local storage. Utilizing the space, a feature to delete by date will most likely be important as years of tasks start to accumulate and should be added. This application should be able to be utilized by Dr. Stringfellow while she is working diligently, and not disrupt her daily manual logging routine. It should enhance and quicken the task of logging itself.

## **2.2 Electronic Task Logger Functionalities**

This section describes the functionalities of the Electronic Task Logger as of the deployed version.

2.2.1 Add Entry: This function will display the user interface that will allow for a new task to be entered and added to the system.

2.2.2 Select/Add Task Type: This function will allow the user to select one of the existing tasks or be able to add a new task if needed to the collection of tasks.

2.2.3 Select/Add Person: This function will allow the user to select one or many of the existing persons or organizations or be able to add a new person or organization if needed to the collection of persons.

2.2.4 Enter Comment: This function will allow the user to enter input that will be assigned to the current task. This function will be designed to work with handwriting to text recognition software.

2.2.5 Update: This function will update the system with the newly input task specifications.

2.2.6 Search: This function will display the user interface that will allow for a task to be searched and displayed.

2.2.7 Select Filter Start Date: This function will allow the user to input a start date to filter their search.

2.2.8 Select Filter End Date: This function will allow the user to input an end date to filter their search.

2.2.9. Select Filter Task Type: This function will allow the user to select one of the existing tasks to filter their search.

2.2.10 Select Filter Person: This function will allow the user to select one of the existing persons or organizations to filter their search.

2.2.11 Select Filter Keywords: This function will allow the user to provide input that will be utilized to filter their search in the comment attribute. This function will be designed to work with handwriting to text recognition software.

2.2.12 Cancel: This function will allow the user to navigate to the Home Page.

2.2.13 Filtered Search: This function will navigate the user to the View Page, which will have the appropriate filtered data available for viewing.

2.2.14 Delete: This function will delete the current task it is associated with.

2.2.15 Delete All: This function will delete all of the resulting data from the filtered search that is currently available for viewing.

## **2.3 Deployment**

This section will describe the deployment of the Electronic Task Logger. During the testing phase of development, the target environment, a Samsung Galaxy Tab S3, was provided from the user. The application was tested in the target environment. This was followed by a presentation and demonstration to the user, where all the test and demonstration data is deleted at the conclusion. Afterwards the target environment will be delivered with the application installed and ready for use to the user.

# **3. Implementation**

This section describes aspects of the implementation phase of development. It consists of sections that will cover: the problems encountered and solutions applied to them, any deviations from the specifications in the requirements document, and if there are any known defects in the application at the time of deployment.

## **3.1 Problems and Solutions**

During the implementation phase of development, the team had to address a few issues. Each subsection will discuss the following issues and the solutions implemented to address them: JavaScript Date functionality, empty string acceptance, and data loading call stack navigation with delete.

3.1.1 JavaScript Date Functionality

During implementation, the choice to use the Date function in JavaScript was made as it returned an int, which was represented by milliseconds in UNIX Epoch Time. The start date for the Central time zone beings at Wednesday December 31, 1969 at 18:00:00 GMT-0600. We had to adjust some of the millisecond parameters when capturing the current time and setting the time with filters to adjust by subtracting six hours of milliseconds to get the correct time.

3.1.2 Empty String Acceptance

During system testing, an error occurred that we had not anticipated. This error came through a user attempting to use the system, where they did not know the functionalities. A user enters the name of a New Person and attempts to Add Task. This produces an error prompt that there is no Task selected. Then, they dismiss the prompt and select a Task, and then backspace over the New Person. This has the system interpret the New Person as an Empty String and will save such to the Person Set. Providing unintended functionality, as there should not be a Person that is an empty string. This was solved by adding an additional check to the New Person variable where an empty string was checked for and if it was the variable was set to NULL, as a NULL New Person will not be pushed to the Person Set.

3.1.3 Data Loading Call Stack Navigation with Delete

During implementation, an error occurred when navigating from the Results page back to the Search page. Each page when loaded calls a constructor that retrieves the most current data on the device so that is readily available. Once the delete feature was implemented however, when a task was deleted on the Results page and then navigated back to the Search page, the Search page would still have the data available to search and appear if searched. Through logging the data to the console, it was clear it was not there, and to compound issues if the user deleted a task, navigated back to Search, back to Home, then forward to Search, the data would not appear there correctly deleted. The team decided that this instance of the Search page was constructed in the call stack that was used for navigating between pages, so whenever the page was visited, we had the page retrieve the data, but this proved ineffective in addressing the issue. We then attempted to again reload the data when the search button was clicked, but this also proved ineffective. Since the problem was resolved by navigating back to the Home page, or the root page, we decided to address this unintentional functionality by navigating from the Results page back to the Home page. This currently proves to be an effective solution to the error, however, using the tablet’s physical hardware buttons, it is still possible that this error occurs. This will be discussed in the still known defects section.

## **3.2 Deviations**

This section will discuss the deviations from the requirements specifications. We have two deviations from the required functionalities: navigation and the many person search. The navigation as described in the State Transition Diagram and Navigation Diagram in the Design Documentation, have the Add Task page navigating to the Home page only on the cancel button, and the Results page navigating to the Search Page. With the problem with the stack call as mentioned above, the team decided since we have a Home page, once the user would finish a task, they should be redirected back to the Home page. So, navigation was changed during implementation. The original functionality required that the user be able to search by one or many persons in the person drop down menu, however, the team only had the time to implement a single person search. Perhaps with more time for development, this deviation could be addressed.

## **3.3 Known Defects**

This section will discuss any known defects in the application. As mention above in 3.1.3, data that was deleted was still showing up during navigation from the Results page to the Search page after deletion. The target environment for this application, Samsung Galaxy Tab S3, has a touch screen button for navigation in the bottom right of the screen. This button was not anticipated when correcting the navigation implementation when navigating from the Results page to the Home page after the user has finished a task. This button may be selected by the user to navigate back to the Search Page from the Results page after a deletion of a task, which will produce the data loading call stack navigation with delete error described above. This defect was found at the time of deployment and after testing, which was much too late in the development cycle to correct properly by the deadline.

# **4. Development Time**

This section will discuss the team’s views on what could have been accomplished with more time, what would be done differently if we had to remake the application again, and what was learned during the development process for this application.

## **4.1 Longer Development Time**

One of the areas the team would like to see an improvement on is the testing phase. Since starting the implementation phase was pushed back due to other priorities, the testing phase was cut short. This is even shown in that there is a known defect in the application involving the could have delete functionality, where the team just could not find a proper solution to the issue. This could in turn lessen our deviations from the specifications for navigation. The team would also have liked to be able to implement the multiple person search functionality as it was a functionality that was not able to implement in the final build. In addition, the team would have liked to have been able to implement a scalable user interface so that the application would look more pleasing on other mobile devices and orientations of the current device. The team would have liked to have added the functionality to have the drop-down menus be sorted when populated for ease of access to the user as well.

## **4.2 Remake**

If the team was allowed a complete remake of the application, the development process model would remain the same. However, the team would attempt to adhere to the original schedule and start implementation earlier, allowing for a much longer testing phase. The addition of an Edit Button on the Results page was brought up during implementation and is something the team would like to add to the could have functionality in a replication of the application. Finally, the team realized when the delete all functionality the user may accidently delete their entire data set. A data recovery method would like to be discussed with the client, perhaps a backup data file, or have the data file be associated with the android or apple account in some way where it would be retrievable.

## **4.3 Learning Experience**

The team was able to gain valuable knowledge, wisdom, and experience throughout the development process of this application. Throughout the team was able to learn and experience how theoretical practices and models are utilized during development is the most important experience to have come out of this development process. The team learned that planning and scheduling are important as time is one of the major constraints for software development. Putting into practice and attempting to adhere to the schedule decided up, where deadlines were approached, and phases were incomplete was difficult to accept, but something the team realized will happen in many futures projects unfortunately. In the decentralized democratic team structure, it was often difficult assigning or delegating tasks that no team member volunteered for as there was not authority leader, but the team was able to overcome and learn from this obstacle. This allowed the team to learn to trust in each other and become more accustomed to working in a development team environment. The team was also able to improve on their technical writing skills with all the documentation that accompanied a software engineering project. During implementation, the team understood the importance and value of design diagrams and appreciated the effort put into that phase of development, which allowed a greater focus to explore and learn the implementation languages utilized, namely: Ionic Framework, Cordova Plug-ins, HTML, CSS, and JavaScript.

# **5. Summary**

This section will summarize and conclude the software development of this application. The team was able to follow the Scrum Model for its development process model and adhere to the short scrums required by quickly discussing, assigning, and reporting during them. Software development proceeded by accomplishing the following phases: requirements and specifications analysis, project plan with management and scheduling, design, implementation, testing, and deliverables for deployment. Team TriNetra was able to complete most phases with great success and understand the importance of each during the development and deployment of the Electronic Task Logger.

# **6. References**

This section will contain a list of references utilized in order to produce this document.

[1] Stringfellow, C. *DeliveryFinalReport,* Dec 2018. (https://d2l.mwsu.edu/d2l/le/content/57393/viewContent/589227/View?ou=57393)