CMSC 447 ⧫ Software Engineering 1 ⧫ Professor Druffel



User Interface Design Document

# Crisis Response Ticket System for Shawn Davis by Team Cloud

**Team Members:** David Foster, Jack McGrann, Julie Nau, Patrick Wheeler

**Last updated:** October 25, 2020

*Crisis Response Ticket System*

User Interface Design Document

**Table of Contents**

[**1**](#_gjdgxs) **Introduction 2**

[**1.1**](#_30j0zll) **Purpose of This Document 2**

[**1.2**](#_1fob9te) **References 2**

[**2**](#_3znysh7) **User Interface Standards 3**

[**3**](#_2et92p0) **User Interface Walkthrough 5**

[**4**](#_tyjcwt) **Data Validation 11**

**5****Open Issues 12**

**6****Appendix A – Agreement Between Customer and Contractor 12**

**7****Appendix B – Team Review Sign-off 13**

**8****Appendix C – Document Contributions 14**

**Document Versioning Control**

|  |  |  |  |
| --- | --- | --- | --- |
| Version Number | Date | Changes from Previous Version | Author |
| 1.0 | 10/15 | Initialized template skeleton and started working on Introduction | Cloud |
|  |  |  |  |

# Introduction

## Purpose of This Document

This document provides a thorough and detailed description of the user interface for the Crisis Response Ticket System. The main purpose of this document is to accurately describe how each user will interact with the system, and how the system will provide the means for each user to interact with it. This document will begin by going over the general layout of the program and the common implementation tools and design methods that will be used universally throughout it. This document will then give diagrams of user screen layouts to provide the reader a top-level view of the system as well as give an in depth walkthrough of how each user will interact with each of these displays in order to simulate the natural life cycle of the system. Finally, the document will end by clarifying exactly which datatypes are being used for each field that the user will be interacting with to ensure all the necessary data values will be input into the system correctly. Using this process, this document will provide a fully comprehensive, descriptive design for what the system will look like from each user's perspective, and how each of these perspectives will interact with one another.

## References

Druffel, K. (2020, September). 2-ProjectDescription-CalltoMission. Retrieved September 18, 2020, from <https://umbc-my.sharepoint.com/:w:/g/personal/kdruffel_umbc_edu/EYVFEiSwyVJHv5VKzjr2OJUBkPo4klqrugJTDvI-cusDCA?e=SbTP27>

Druffel, K. (2020, October). 30-UIDD-TeamName-Template-F20. Retrieved October 15, 2020, from <https://umbc-my.sharepoint.com/:w:/g/personal/kdruffel_umbc_edu/ER0tR47j-JhKvpPWQ2GWXUoB0-UwFNCcE_5440QHCwwwBw?e=eFxU7i>

Duffel, K. (2020, October). 30-uiGradingRubric\_S18. Retrieved October 15, 2020, from <https://umbc-my.sharepoint.com/:w:/g/personal/kdruffel_umbc_edu/EU-A7vL5y2FMkdujUcQjp34BwBpfMCgCwZGBh6E2K0XoiQ?e=IlFJdx>

Druffel, K. (2020). 447-10-13-Design-Diagrams [PowerPoint slides]. Retrieved from <https://umbc-my.sharepoint.com/:p:/g/personal/kdruffel_umbc_edu/EdGKWmw3jh1NiKZV6WqIu6cBjW7F80tdi1nB0dZ_5ZhFxQ?e=ydrvRI>

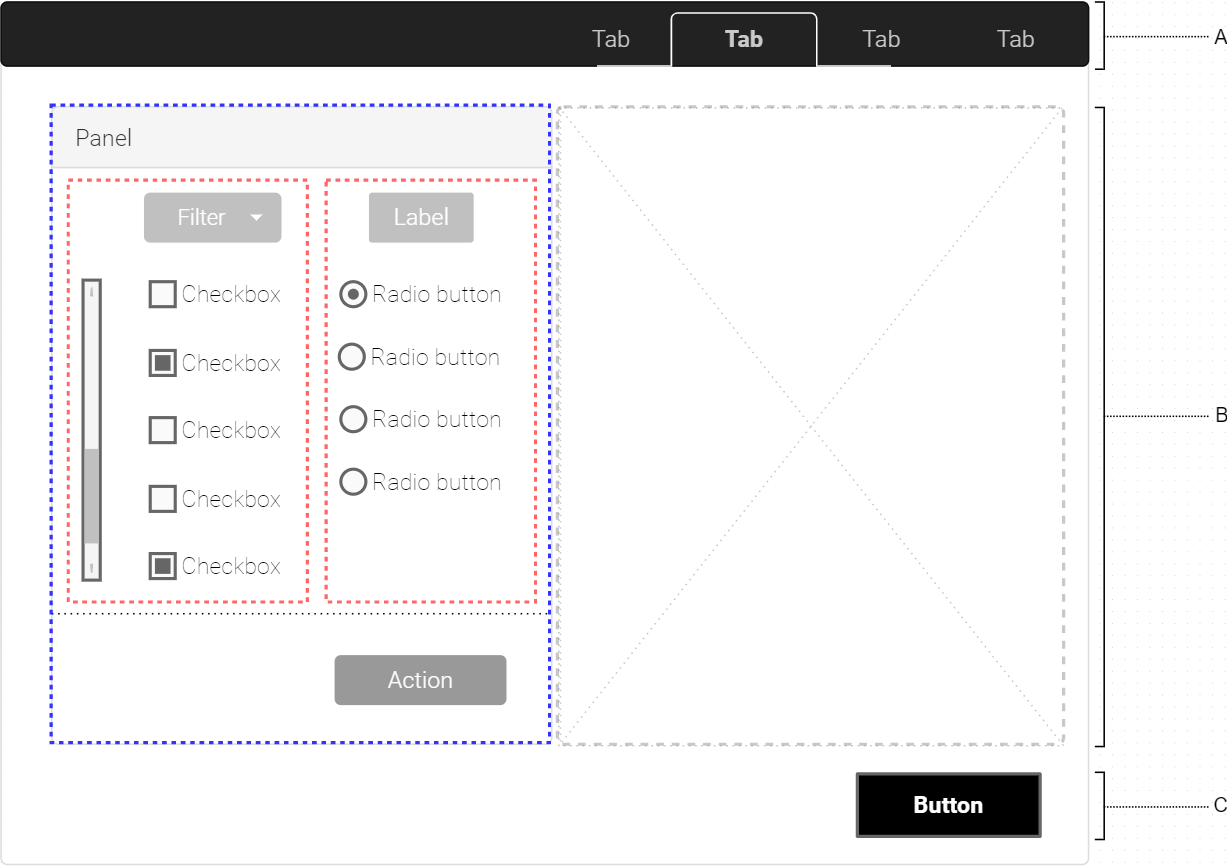
Druffel, K. (2020, September). 447-00-Timeline-20b. Retrieved September 27, 2020, from <https://umbc-my.sharepoint.com/:x:/g/personal/kdruffel_umbc_edu/EU4VoggR0ihEgVqFVYqbfh8BgSj96iz2EQLeNlxRpX0Ebg?e=91dtyj>

# User Interface Standards

Standardizing the user interface design is an essential step in creating a comprehensive, robust, and versatile application. Our application will consist of four separate interfaces as described in the SRS; each role (i.e. Administrator, Ticket Creator, Chief Operating Officer, and Mission Manager) will have their own, discrete variation of the general layout as illustrated in Figure 2.1.

**Figure 2.1**

*Diagram of the prototypical GUI for the Crisis Response Ticket System.* | *(A) Navigation menu with linked pages. (B) A pair of window panes, vertically-split. (C) Example of a primitive component.*



Primitive components provide the basis for building every aspect of the interface. Component types include buttons (see *Figure 2.1-C*), drop-down menus, check and radio selection types, and text forms.

The menu bar provides users with a simple, effective way to navigate different views of the system. These views implement various functionalities defined by the user type. Although *Figure 2.1-A* uses ‘Tab’ as a general placeholder name, an actual implementation uses unique, descriptive labels. As an example, a user belonging to the Call Center Operator group will have available two tabs: one named ‘Ticket’, in which the user creates and populates a new ticket form, and another named ‘History’, in which the user may review information about previously submitted tickets.

Screen space is further partitioned into any number of window panes which organize components into functionally separate sections. *Figure 2.1-B* illustrates what this might look like. On the right is an empty, uninitialized example pane. To its left and outlined in blue is a pane labeled ‘Panel’ which, itself, contains further component groupings to define and isolate specialized actions.

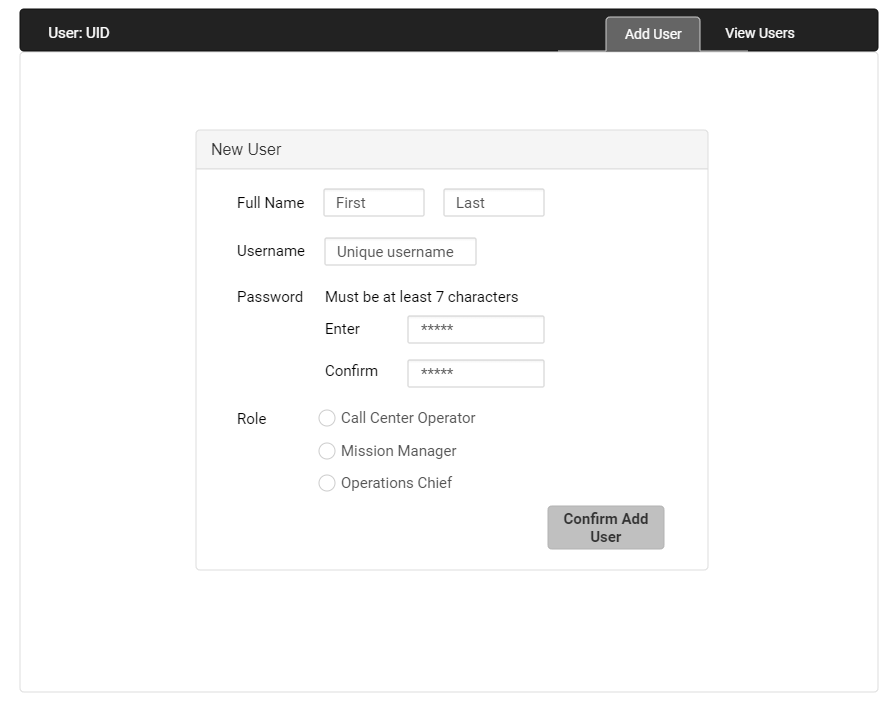
Thoughtful use and organization of components will provide mostly passive error handling. A dialogue box will pop up to review and confirm the submission of any data that will enter the database. Undefined or errant actions, however, will prompt the display of an error message.

# User Interface Walkthrough

Administrator

**Figure 3.1**

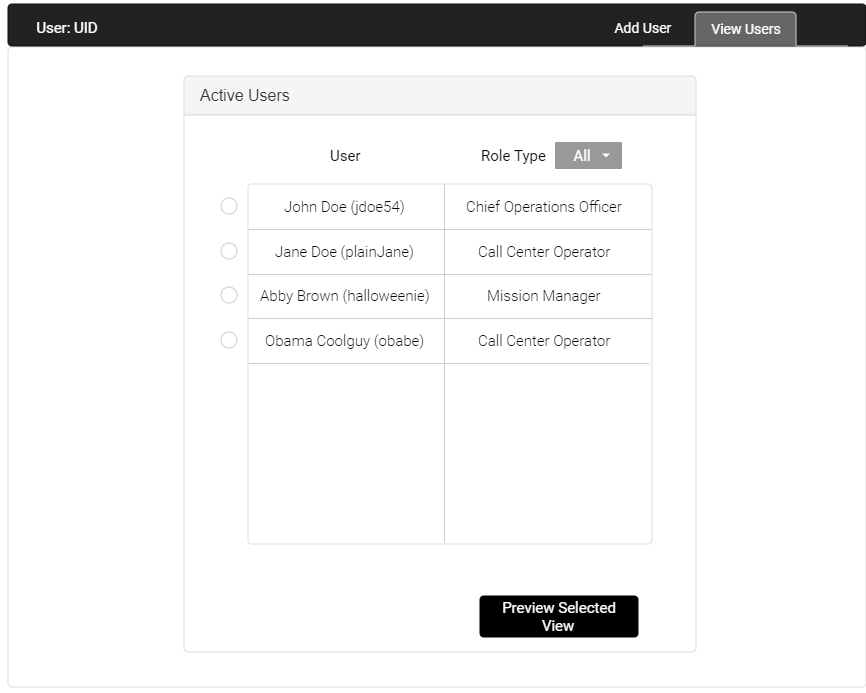
*Administrator: Add User Tab*



This view allows an administrator to add a new user to the system. All visible fields must be completed for the user to appear in the system. Role selection corresponds to system use cases and grants access accordingly.

**Figure 3.2**

*Administrator View All Users*

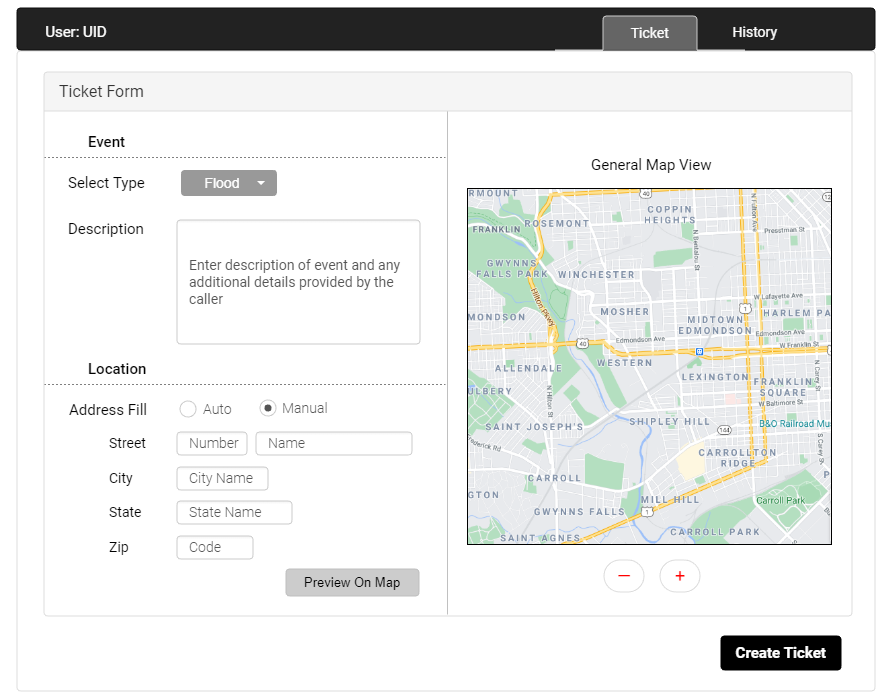


The administrator is also granted access to preview account views--from here, the administrator can verify the successful creation of any user account and its operational status.

Call Center Operator

**Figure 3.3**

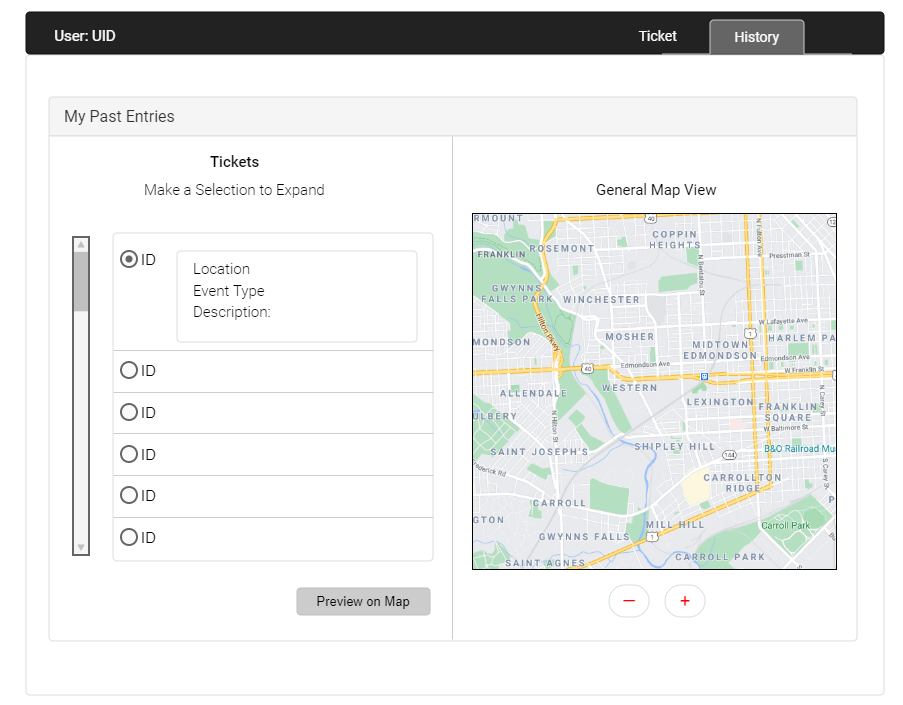
*Call Center Operator Ticket Form*



The call center operator completes the ticket form. They are responsible for obtaining the type and location of the event and providing an additional description of the call. If the caller isn’t sure of the exact address, the map allows the operator to pinpoint streets and provide a good estimation. When confirmed, the operator will hit the *Create Ticket* button to push the information to the Chief Operations Officer.

**Figure 3.4**

*Call Center Operator History View*

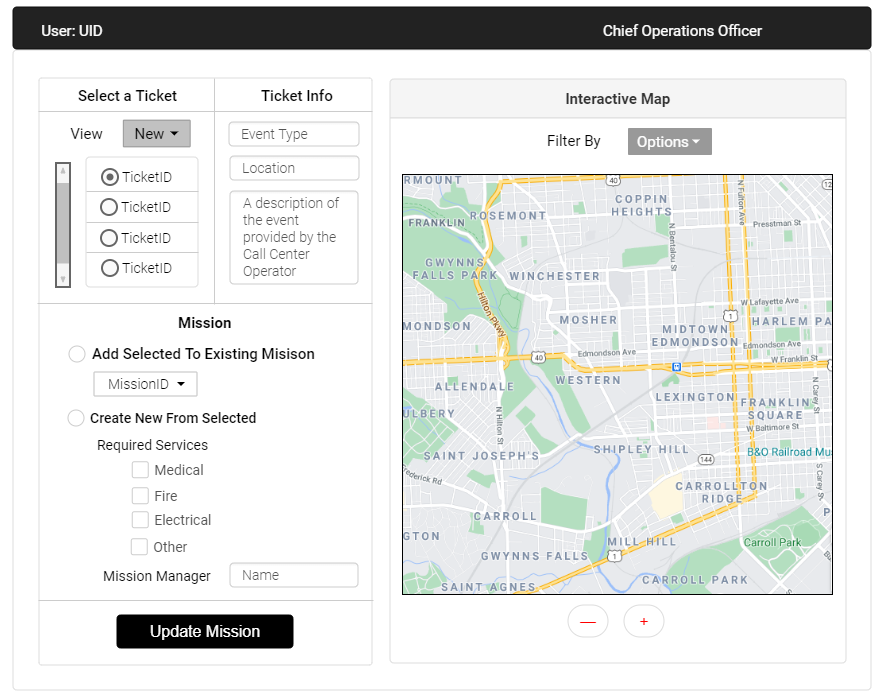


The history view provides the operator with a list of their completed ticket transactions which may be important in identifying and resolving any unanticipated errors down the line.

Chief Operations Officer

**Figure 3.5**

*Chief Operations Officer Window.*



This view allows the operator to view new, or active, tickets and their associated values. From here, the user will either add the ticket to an existing mission, or assign it to a new mission. If the latter, the chief will specify what services will be required to handle it so the mission manager can allocate resources accordingly. To aid in ticket-mission grouping, the chief may find it useful to view ticket information on the interactive map. The map view can be filtered by event type, location, ticket status, and more.

Mission Manager

**Figure 3.6**

*Mission Manager Window*



The Mission Manager is responsible for allocating resources for, and updating the status of, their given mission. This view allows a manager to keep track of the tickets, people, and vehicles associated with their particular mission. They have the option to add agents and vehicles according to need and will ultimately be responsible for updating the mission’s status.

***NOTE****: The mission manager will be sent a list of required resources from the chief operations officer. Although not pictured, this is a vital step helping the mission manager determine the types and quantities of agents and vehicles they will add to the mission.*

# Data Validation

The primary form of data visualization for this system will be the Call Center Operator user’s input of “event” data fields into a “ticket”. For the purposes of our demonstration, the “event” data will come from a prewritten excel sheet. The Call Center Operator user will use the data from the event to create ticket objects. Each ticket requires the CCO user to enter the following information:

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Field Name** | **Data Type** | **Format** | **Description** |
| Ticket\_ID | Integer | Six digit number. Auto-generated by the system | Primary key for ticket |
| Comment | String | Long form text entered in text box | Description of the event. |
| Mission\_Status | String | Auto-generated by system | Ticket status. Always starts as “New” |
| Event\_Type | Integer | Chosen from menu | Type of event (flood, fire, power outage, etc) |
| Street\_Number | Integer | Entered in a text box. Must be a valid number. | Street number of event location |
| Street\_Name | String | Entered in a text box. Must be a valid street. | Street name of event location |
| City | String | Entered in a text box. Must be a valid city. | City of event location |
| State | String | Entered in a text box. Two letter abbreviation. Must be a valid state. | State of event location |
| Zip\_Code | Integer | Five digit number entered in the text box. Must be valid code. | Zip code of event location |
| Map\_X\_Coordinates | Double | Auto-generated by mapping API | Horizontal map coordinates of event location |
| Map\_Y\_Coordinates | Double | Auto-generated by mapping API | Vertical map coordinates of event location. |

# Open Issues

|  |  |  |
| --- | --- | --- |
| Issue # | Issue Description | Target Resolution Date |
| 1 | Map size in UI: How big do we make the map? Do we add scalability sliders? Conversation of feature vs simplicity. | 11/3/2020 |
|  |  |  |

# Appendix A – Agreement Between Customer and Contractor

By signing this document you agree that the information presented in this document is true to the best of your knowledge and that the actions and that this represents the wishes of both parties.

If there are any changes to be made to the document, an audit report of the changes will be available to both parties to review with weekly meetings to discuss changes. Backups of previous versions will be available if a change needs to be rolled back to the previous week’s version.

|  |  |  |
| --- | --- | --- |
| **Typed Name** | **Electronic Signature** | **Date Signed** |
| David F | David Foster | 10/17/2020 |
| Jack M |  |  |
| Julie N | Julia Nau | 10/24/2020 |
| Patrick W | Patrick Wheeler | 10/24/2020 |
| Shawn D |  |  |

## Customer comments written here:

# Appendix B – Team Review Sign-off

By signing below, all team members have reviewed the document and agree on its content and format. The comment section is to be used for minor unresolved issues.

|  |  |  |  |
| --- | --- | --- | --- |
| **Typed Name** | **Electronic Signature** | **Date Signed** | **Comments** |
| David F | David Foster | 10/18/2020 |  |
| Jack M | John McGrann | 10/24/2020 |  |
| Julie N | Julia Nau | 10/24/2020 |  |
| Patrick W | Patrick Wheeler | 10/18/2020 |  |

# Appendix C – Document Contributions

As Requirements leader, Patrick W. is in charge of this document.

|  |  |  |
| --- | --- | --- |
| **Team Member** | **Contribution** | **Estimated Percent Work** |
| David F | Cover page, Sections 4, 6, 7, 8, 9 | 25% |
| Jack M | Section 1 | 20% |
| Julie N | Team Support | 10% |
| Patrick W | Sections 2, 3 | 45% |