**Remarkably Busy Business’s**

**Issue Tracker System**

**USER MANUAL**

# User Manual Title Page

**Version 1**

Release Date: 2024/06/19

Approved by:

Allan Hu \_\_\_\_\_\_\_\_\_\_\_\_\_

Nicolao Baretto \_\_\_\_\_\_\_\_\_\_\_\_\_

Puja Shah \_\_\_\_\_\_\_\_\_\_\_\_\_

Wah Paw Hser \_\_\_\_\_\_\_\_\_\_\_\_\_

# User Manual Version History Page

Version 1 - original 2024/06/19 by Allan Hu, Nicolao Baretto,  
 Puja Shah, Wah Paw Hser

# Table of Contents

[**User Manual Title Page 1**](#_gk0t2zhwfro9)

[**User Manual Version History Page 2**](#_1pbj8w853juo)

[**Table of Contents 3**](#_mzlcp5hbqcyg)

[**Section 1: Introduction 4**](#_dbemoq4rq2ls)

[1.1: Congratulations on Your Purchase! 4](#_omu2dt23761w)

[1.2: Overview 4](#_9tf8ivch6jit)

[1.3: Typographical Conventions 6](#_wbt10ptyddup)

[**Section 2: Installation 7**](#_e3wj1ghn6xpm)

[2.1: Requirements 7](#_iw5l004jq3k0)

[2.2: Back-Up Installation Media 7](#_ib5imshttdj0)

[2.3: Installation Steps 8](#_vd9mfknm3tix)

[2.4: Starting the Application 8](#_zpajbs5nata)

[**Section 3: Introduction to the User Interface: 9**](#_qozsa9m0o509)

[3.1 Menu System 9](#_2unsx11q5nv8)

[3.2 Data Entry 11](#_997b89oa6va1)

[3.3 Prompts 12](#_lglbmyhccjg5)

[3.4 Updating Change Items 15](#_hw6grfdddygx)

[**Section 4: Operating Procedures 15**](#_4z9lem35xu4m)

[4.0: Context 15](#_xhc02l4zffim)

[4.1: General Exceptional Behaviour 16](#_bh0dpviw9fx9)

[4.1.1: Field Requirements Not Met 16](#_dvyeb3a1bbsn)

[4.1.2: Selection Out of Domain 16](#_wc05i74lxib7)

[4.1.3: Save Failure 16](#_24oa06pnh9gr)

[4.2: Operations 16](#_2h2g1hu3xqhx)

[**Section 5: Database Back-Up Procedure 34**](#_vkv7tcp93ru4)

[**Section 6: References 35**](#_aullw8w439f5)

[6.1: Error Messages 35](#_ij5xcbs7n2k9)

[6.2: Data Format 37](#_u30stzk6vqi7)

[6.3: Menu Tree 38](#_w2wum3widjkm)

[**Section 7: Terms Glossary 39**](#_u74orlw16lkb)

# 

# Section 1: Introduction

## **1.**1: Congratulations on Your Purchase**!**

Welcome to the Issue Tracking System User Manual! We at *Remarkably Busy Business* are grateful that you have chosen our system and we are very excited to be working with you. This system will allow users to track bug requests and also new feature requests from customers and store related data, all while being able to track and update information about the requests, customers and statuses.

Our user manual will help you get started and guide you through everything that you need to know about our Issue Tracking System. With the detailed instructions that we have provided, this manual will help you make the most out of your new purchase. We are very confident that you will be satisfied with your experience. If you require any further assistance or should you have any inquiries about our system, you can contact us and our support staff will be happy to assist you!

Enjoy your journey,

*Remarkably Busy Business*

## 1.2: Overview

Our Issue Tracking System will be tracking incoming bug requests and new feature requests from customers and also workers of your department by storing the information about the requests, customers, and the products.

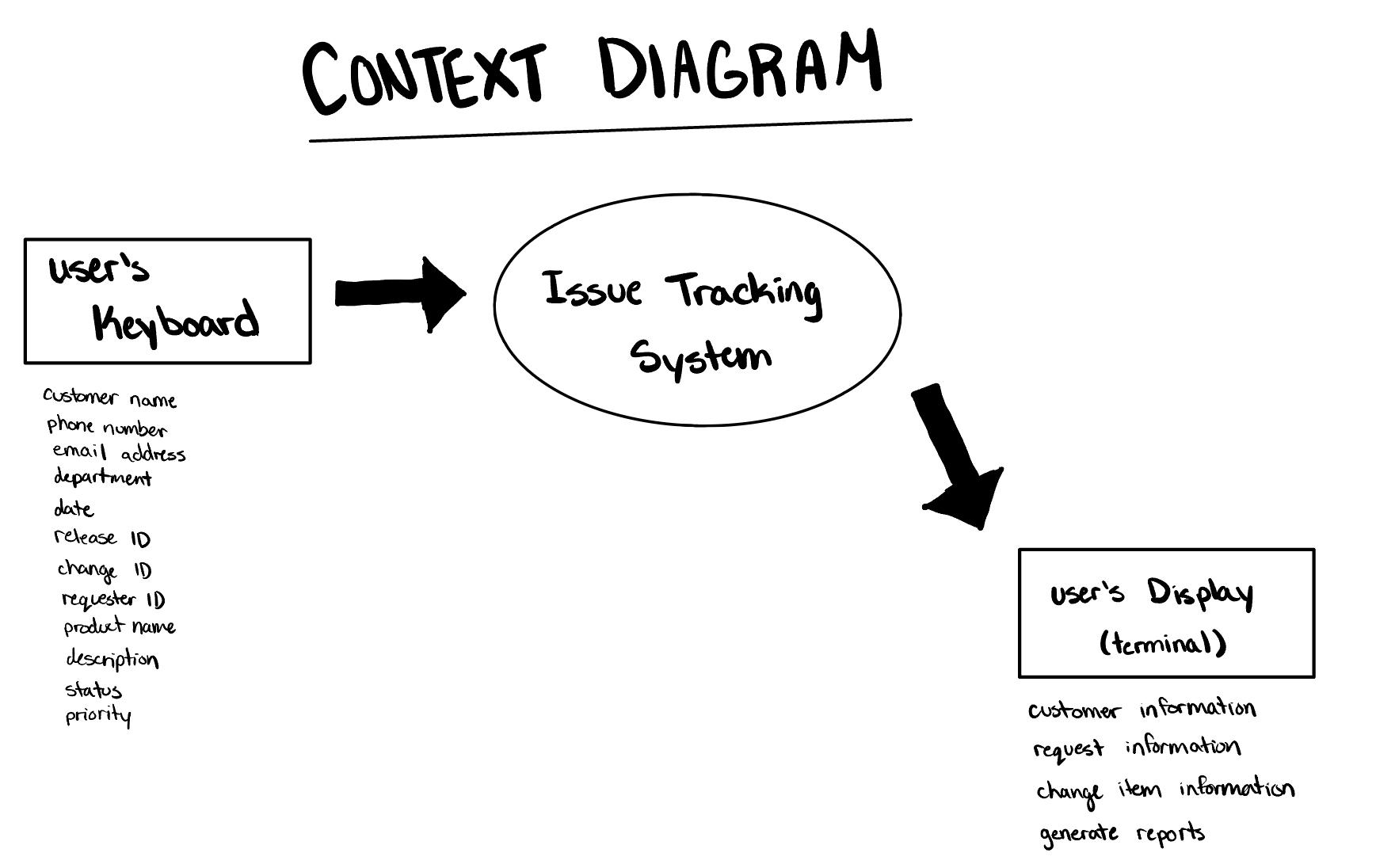
Features that our Issue Tracking System has are:

* Creating a new requester (customer or internal staff)
* Creating a new change request
* Creating a new change item
* Creating a new product
* Registering a new product release
* Updating a change item
* Updating a product release
* Generating a report of all new change items
* Generating a report of all changes related to a product
* Generating a report of all requesters that need to be informed about a change that has been implemented

The advantages of using our Issue Tracking System are:

* The program is user-friendly and it is easy to understand, making:
  + Training new employees much faster
  + Using the system take less time
* The status of each change item make it efficient to see which items have not started being implemented / have not been completed
* Our generative reports will give the user a transparent view of all products, changes, and also customers, which makes it easier to track the information
* The program ensures that change items are not counted more than once, regardless of how many requests a bug or feature has
* The program keeps track of the priority of each change item, making it easier for your team to progress with implementations of high priorities
* The program has a historical record of past requests and requesters, keeping track of:
  + Which customers already exist in the system
  + Which bugs / features have already been implemented / fixed
* The reports that are generated by the program are detailed and provide the user with the relevant information that is related to their report
* The program ensured that the stored data is protected with backup features
* The user can inquire any time about a change item

Below is a context diagram that will help you visualise the interactions between the user and the *Issue Tracking System*.



## 

## 1.3: Typographical Conventions

For this *Issue Tracking System* user manual, any instance of “you” is referring to the user of this program.

This user manual will be following these typographical conventions:

* “#” the hashtag will be used to refer to an arbitrary numeric character.
* “$” will be used to refer to an arbitrary alphanumeric character.
* Ranges of acceptable values will be shown using the notation (X){X} . It reads - there must be at least 1 and a maximum of i+1 characters of type X (where X is either # or $).
* The ENTER key will be denotes as **Enter (↵)**
* A vertical ellipses will denote lines in terminal printouts not included in the instruction manual.
  + Note : When using ellipses as replacements for list outputs, we will display the maximal length of the output (16 listings in most cases). It is possible to have fewer.
* To show console printouts in the program, the user manual will use:  
  “Courier new, unbolded”
* To show user input in the program the user manual will use:  
  **“Courier New, bold”**

# Section 2: Installation

## 2.1: Requirements

Equipment required to operate the issue tracking system:

* A keyboard
* A monitor with at least 1024 x 768 resolution
* At least 16GB of RAM
* At least 1TB SSD storage
* Optimised for Ubuntu 22.04.3 LTS, (Jammy Jellyfish), but could work on other version of Ubuntu as well
* At least two backup floppy disk, one for back-up
* Mouse to open the system, not required to operate the system
* Printer
* 9th generation i5 intel processor or better

## 2.2: Back-Up Installation Media

Before using the issue tracking system, it is recommended to first back-up the distribution disks onto another floppy disk. This is done in order to save your data in the case of your disk being corrupted, damaged or lost. This can be done by following the steps bellow:

1. Open the terminal on your computer by either:
   1. Press Ctrl+Alt+T
   2. Click on the “show applications” icon on the bottom left of your screen and then click on the search bar in the middle of your screen and type in “terminal” in the search bar and press **Enter (↵)** on your keyboard.
2. Type “**c:**” in the terminal and then press **Enter (↵)** on your keyboard
3. Type “**cd\**” in the terminal and then press **Enter (↵)** on your keyboard
4. Type “**md tempITS**” in the terminal and then press **Enter (↵)** on your keyboard
5. Type “**cd tempITS**” in the terminal and then press **Enter (↵)** on your keyboard
6. Type “**copy a:\\*.\* c:**” in the terminal and then press **Enter (↵)** on your keyboard
7. Remove the origin floppy disk and insert your back-up floppy disk
8. Type “**copy \*.\* a:\**” in the terminal and then press **Enter (↵)** on your keyboard
9. Remove the back-up floppy disk and keep it in a safe place
10. Type “**del \*.\***” in the terminal and then press **Enter (↵)** on your keyboard
11. Type “**cd\**” in the terminal and then press **Enter (↵)** on your keyboard
12. Type “**rd tempITS**” in the terminal and then press **Enter (↵)** on your keyboard
13. Insert the original floppy disk back

## 2.3: Installation Steps

Please follow the steps below to install the issue tracking system:

1. Open the terminal on your computer by either:
   1. Press Ctrl+Alt+T
   2. Click on the “show applications” icon on the bottom left of your screen and then click on the search bar in the middle of your screen and type in “terminal” in the search bar and press **Enter (↵)** on your keyboard.
2. Insert the Issue Tracker System into the floppy disk
3. Type “**A:**” in the terminal and then press **Enter (↵)** on your keyboard
4. Type “**ITSsetup.bat**” in the terminal and then press **Enter (↵)** on your keyboard

The data files stored in the Issue Tracker System can be accessed in the program directory C:\IssueTrackerSystem\

In the case of there not being enough space in your SSD, please check whether or not you have at least 1TB on your SSD. If there isn’t, please clear up some space by deleting some applications, please be advised that these applications could be lost forever if deleted.

## 2.4: Starting the Application

In order to start the Issue Tracker system please follow the following steps:

1. Open the terminal on your computer by either:
   1. Press Ctrl+Alt+T
   2. Click on the “show applications” icon on the bottom left of your screen and then click on the search bar in the middle of your screen and type in “terminal” in the search bar and press **Enter (↵)** on your keyboard.
2. Type “**cd IssueTrackerSystem**” in the terminal and then press **Enter (↵)** on your keyboard
3. Type “**make ITS**” in the terminal and then press **Enter (↵)** on your keyboard

# Section 3: Introduction to the User Interface:

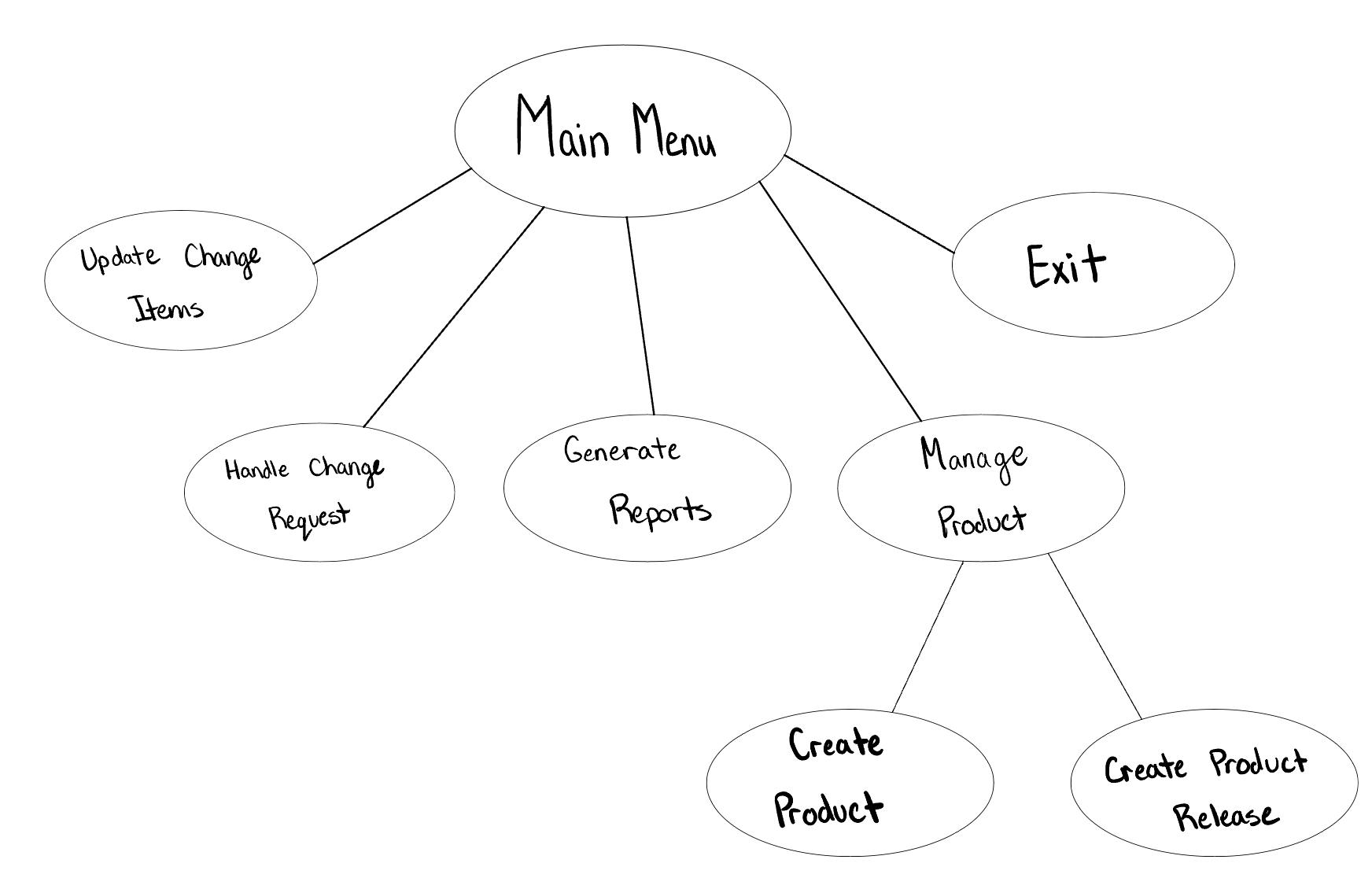
Our *Remarkably Busy Business’s Issue Tracking System* is a command-line interface that uses a terminal to display contents. There will be a single screen, and it will not be cleared after any selection. There is no option for the user to resize or generally edit the appearance of the screen from within our program, but they are free to do so externally, if need be.

We have attempted to make typical uses of our system as accessible to our users as possible, while also providing plenty of options for less frequent use cases. We hope our interface feels intuitive and enjoyable to use.

## 3.1 Menu System

**3.1.1 Menu Tree**

All major operations to be done using our issue tracking system are represented via this menu tree (also to be found in **Section 6.3**) :



**3.1.2 UI Overview**

Unless it is absolutely necessary for you to enter specific details in order to complete an action, our menu system will provide you with numbered options that you can select from.

This is the first menu you will encounter our **Main Menu** :

Main Menu:

[1] Handle Change Request

[2] Update Change Item

[3] Manage Products

[4] Generate Reports

[0] Exit

You will notice how each option is preceded by a number. Selecting an option is as simple as pressing the desired option number on your keyboard followed by the **Enter (↵)** key. Selecting an option will either begin a series of prompts to help you execute your option, or lead to a sub-menu to further narrow down your options. The sub menu will work in the same way as our main menu (and all other menus) will.

Since each menu option will be a whole number, you will be expected to always input an integer[[1]](#footnote-0) that falls within the range of the options presented to you. If the number you enter is not one of the options, you will encounter the error : **Selected Option is Not Available.** If you enter a non-numeric value, you will see the error : **Incorrect Character Type.** Once you are presented with an error, you will be given the option to retry. To get further details about these errors, please refer to **Section 6.1**.

Once an option has been selected by you, the screen will scroll up to present you with the next content. *The latest content will therefore be the one closest to the lower end of your screen.*

**3.1.3: Going Back : Option 0**

You may notice the last option in our Main Menu above is [0] Exit. This will be an ever-present option in our menus, albeit renamed to “Back,” in the sub-menus. In the main menu, this option helps you exit the system completely, for when you are done interacting with it. When used within sub-menus, it will take you back to its parent menu (the one that appeared before it). If you happen to accidentally select the wrong option, you will always have the option to go back and try again.

**3.1.4**: **Data Lists**

In addition to everything described above, we have a modified version of these menus for when you are required to select an item from the ones pre-existing in our database. Let us say you are in the process of handling a change request, and need to choose a requester for it. We will present you with a list of all requesters in our system 16 at a time:

Handle Change Request:

Select a requester:

[1] RequesterA

[2] RequesterB

.

.

.

[16] RequesterN

[N] Create New Requester

[0] Back [C] Next Page

If the requester you wish to add exists on the list, you should use this in the same way as any other menu. Simply type the number in square brackets associated with the requester of choice, and press the **Enter (↵)** key. If you need to see more, type [C] followed by the **Enter (↵)** key, and you will be presented by the next 16 items in the list (or less than 16, if the list is near its end).

You will notice how there is an extra option called [N] Create New Request. When the items you are selecting from cannot be explicitly created through a main menu selection ,you will be presented with this option if your desired item is not on the list. Selecting this option will lead to a series of prompts to create a new one.

In other cases, where creating the list item is a menu option of its own, you will be required to return to the main menu to create it.

Please refer to the **Menu Tree (Sections 3.1.1 or 6.3)** to see the differences. Items not on the menu tree will not have a dedicated option for creation, and will instead be created through the [N] option.

## 3.2 Data Entry

**3.2.1 Overview**

Certain processes in the system cannot be done via providing options, and require you to input data from your end. Actions that involve adding items to the system as well as modifying them are the ones that require this, while view-only operations will generally be done through menus.

When prompted to enter data such as name of a product when attempting to add a new one, you will simply have to type the information and press **Enter (↵).** The input of all text will always be from the bottom of your screen. In each instance you will be guided as to what kind of data the system is expecting ( the name of a Product, a phone number, etc) alongside the expected requirements. Below is an example of the first prompt upon trying to add a Product to the system:

Create Product:

Define Product Name:

[0] Back

Field Requirements:  
(1-10 alphanumeric characters)

Enter Product name:

You will notice the phrase Field Requirements, succeeded by a description in brackets. The response you enter to the prompt must match these requirements. In this particular case, the product name must be between 1 and 10 characters long, and the characters themselves must be alphanumeric[[2]](#footnote-1).

You will be expected to type the name for the product you wish to enter into the system, and then press **Enter (↵)**.Details on the process for reaching this option as well as what comes after (such as having to confirm the name you entered) will be thoroughly explained in **Section 4.3**.

**3.2.2 Field Requirements Not Met : Errors**

It is especially important to keep field requirements in mind when entering data from your end instead of selecting options. Not following the requirements will lead to error messages being shown. These messages simply attempt to guide you, but can nonetheless be a little jarring and therefore best avoided. After an error appears, you can always try again. Errors that may occur due to not meeting field requirements are:

1. **Incorrect Character Type**
2. **The Value Entered Exceeds the Character Limit**
3. **The Value Entered Is Below the Character Requirement**

Please refer to **Section 6.1** for more details on each error.

## 3.3 Prompts

This is an overview of the types of interactions you will have with the system, and the expected response for each.

**Style 1 : Select From a List**

*Use : Main Menu, Sub-Menus, Selection Lists*

As mentioned above (3.1 and 3.2), the system will display a list as an output for most operations, and these lists will act as menus to help you select the next option.

Lists will follow identical formatting for the most part, with all the sub-menus resembling:

Sub Menu Title:  
Prompt:

[1] Option 1

[2] Option 2

.

.

.

[0] Back

Longer lists will have an additional option for continuing [C], and within them certain lists will allow for creating a new item. These lists will resemble:

Option Title:

Prompt:

COLUMN1 HEADER COLUMN2 HEADER ... COLUMNm HEADER

[1] ItemA Value1 ItemA Value2 ItemA Valuem

.

.

.

[16] ItemN Value1 ItemN Value2 ... ItemN Valuem

[0] Back [C] Next Page [N] Create New Item

The selection remains otherwise identical to the other menus.

**Style 2 : Confirmation**

*Use : Create Entries, Update Status*

When you attempt to create a new entry of a product or product release, or change an item’s status to an immutable status, it is important to ensure there are no mistakes. In order to maintain accuracy in our records, you will be prompted to confirm your entry.

Option Title:

Selection Name:

[0] Back  
  
Field Requirements:  
(n-k “CHARACTER TYPE” characters)

Prompt: **abd**

Would you like to “COMPLETE ACTION” abd?  
[y/n] : **n**

Option Title:

Selection Name:

[0] Back  
  
Field Requirements:  
(N - K “CHARACTER TYPE” characters)

Prompt: **abc**

Would you like to “COMPLETE ACTION” abd?  
[y/n] : **y**

“ACTION COMPLETED”

Here “CHARACTER TYPE” is a placeholder for the type of character expected, and n-k represents the range. “COMPLETE ACTION” and “ACTION COMPLETED” refer to the action you are attempting to perform (eg: complete action is replaced by *create product* & action completed is replaced by  *product created*). As you can see, entering ‘**n,’** allows you to enter the value again by restarting the process.

The other scenario in which you will be asked for confirmation is when you attempt to change the status of a change item to ‘Done,’ or ‘Cancelled’. Since both these states are irreversible, we want to confirm the update. It will appear like:

Update Change Item:

Update Status:

Confirm change to “NEW STATUS”?

[y/n] : **y**

Status Updated

Here, “NEW STATUS” is a placeholder for a selection of ‘Done’ or ‘Cancelled,’ from a menu that appears right before this question. **y** will update the status, while **n** will take you back to the status selection menu **(**refer **Section 4.2.3.6**).

**Style 3 : Regulated Data Entry**

*Use : Create, Update, filter*

When adding details such as product name, requester name, change item description, and so forth, you will be allowed to enter what you wish, but with some restrictions on length and type of data. As elaborated in Sec 3.2.2, you will be required to input data in the correct format. The interface will resemble:

Menu Option Name:

Selection Name:

[0] Back  
  
Field Requirements:  
(n-k “CHARACTER TYPE” characters)

Prompt:

Here, N and K represent the lower and upper limits, respectively, on the number of characters. “CHARACTER TYPE” is a placeholder for the type of character that will be mentioned. Please refer to the **Terms Glossary** (**Section 7**) for clarification on character types.

## 3.4 Updating Change Items

The user does not have the ability to modify anything but a Change Item, and that too will have restrictions. A user can only update the release ID, change the priority, alter the description, or update the status. No other fields can be altered.

Once a product, product release, change request, requester, or change item has been created, it can never be deleted. Even if requests are for items marked ‘cancelled’ or for products that have been discontinued, neither the request nor the product will be deleted.

We provide options to generate reports that only show what is most relevant, but the system will forever store all the information that has been entered in it.

# Section 4: Operating Procedures

## 4.0: Context

For brevity and for minimal redundancy, information relevant in every step or many steps of the operating procedures will be written within this sub section.

For all example printouts showing a list of items from the database the entries of these lists and their data are placeholders for data in the user’s database. The first line of placeholder entries demonstrates the maximum length of each field.

A “selection,” is made by typing a menu item’s index number and pressing **↵**. Throughout the operating procedures making a selection will be referred to as selecting “*i”*, where *i* is the index character of some menu option. In list prompts, making a selection in [1, *n*], means selecting “*i*” where *i* is any integer value in the closed interval [1, *n*], where *n* is an integer.

A prompt to “enter,” information is completed by typing a response in the terminal and pressing **↵**. Throughout the instructions responding to these prompts will be referred to as “entering *j*,” where *j* is some piece of information.

Whenever the user selects “0” the program will rollback to the previous menu.

“0” is not allowed to be entered into any entry field as it is reserved for the back button.

In some menus in addition to being able to select “0” to go back the user may be able to select “00” to go fully back to the main menu.  
 Whenever there are too many elements to fit all of them on screen at once in a list menu there will be a menu option “[C] Next Page” otherwise this option will not be printed as part of the menu. Whenever the user selects (“C” | “c”) in a list menu, the program will print more options and their indexes.

## 4.1: General Exceptional Behaviour

### 4.1.1: Field Requirements Not Met

In any prompt expecting (n - k alphanumeric characters), or (n - k numeric characters), garbage in will produce exceptional behaviour. Entering too few characters will print the error described in 6.1.3 or too many characters will print the error described in 6.1.4. Entering characters outside of the domain will print the error described in 6.1.2. In either case the program will recover and be able to return to the menu prompting for input where the exception was raised.

### 4.1.2: Selection Out of Domain

In any selection if the user selects an index that does not relate to any menu item the program will print the error described in 6.1.1 and return to the menu prompting the selection. In any true of false not selecting (“Y” | “y”) | (“N” | “n”) will print the error described in 6.1.1

### 4.1.3: Save Failure

If an object cannot be saved then the program cannot continue. The program will print the error described in 6.1.5. The user can return to the main menu or return to the menu that raised the exception.

## 4.2: Operations

[4.2.1: Main Menu 18](#_qvj7cna3mtxw)

[4.2.2.1: Begin main processes 18](#_217jpgr7vy82)

[4.2.2: Handle Change Request 18](#_gpnarto4e9s6)

[4.2.2.1: Start handling a change request 18](#_szuddljyroce)

[4.2.2.2: Choose, or create a new requester for this change request 18](#_ocnhgadtux25)

[4.2.2.3: Collect information for new requester 19](#_g0i7co2d9mip)

[4.2.2.4: Finalise a new requester 20](#_fb91qxhk0au1)

[4.2.2.5: Collect information about change request 21](#_zauki3r7pvti)

[4.2.2.7: Creating a change item 23](#_kjvcp5tlz2i5)

[4.2.3: Update Change Item 23](#_kh01htkim8my)

[4.2.3.1: Start updating a change item 23](#_47nh1sctiy2i)

[4.2.3.2: Filter by product 23](#_1zymp2vf531b)

[4.2.3.3: Filter by status 24](#_kpkdqb478feb)

[4.2.3.4: Select change item to update 24](#_896hae5msewp)

[4.2.3.5: Update Menu 25](#_kkblx0e3hza3)

[4.2.3.7: Updating a description 26](#_l0kwb1s1a64q)

[4.2.3.8: Updating a Priority 27](#_mlhkf7sd3o68)

[4.2.3.9: Updating a change item’s release 27](#_vwxns5g7lhce)

[4.2.4: Manage Products 28](#_nb7r8nyvpbpl)

[4.2.4.1: Start managing products 28](#_sxwby745vqhq)

[4.2.4.2: Choose whether to create a product or a release of a product 28](#_h0zyb0b80da6)

[4.2.4.3: Adding a Product 28](#_vhqokddysc36)

[4.2.4.4: Adding a product release 29](#_z3tl51jhqixt)

[4.2.3: Generate Reports 30](#_a8na37s766qa)

[4.2.5.1: Start generating reports. 30](#_6evxxfcfcz0z)

[4.2.5.2: Choose report type 30](#_5q2fqan9u0ry)

[4.2.5.3: Report all requesters who need to know about a change 31](#_b3fvshav1wtw)

[4.2.5.4: Report all unresolved change items of a product 32](#_eimgo5bprncf)

[4.2.5.5: Query specific change item 33](#_81flv5npql8u)

The user manual is designed in a stepwise manner to emulate the behaviour of the program. If you have a specific use case you need help with, please feel free to use the table below to know which step you can jump to and where to go from there.

| I want to… | Go to Section… |
| --- | --- |
| 1. Enter a bug report or feature request into the system. | 4.2.2.1 |
| 1. Update a change item. | 4.2.3.1 |
| 1. Add a new product to the system. | 4.2.4.1  => 4.2.4.3 |
| 1. Add a new product release to the system. | 4.2.4.1 => 4.2.4.4 |
| 1. View a report of all requesters for a particular change item. | 4.2.5.1 => 4.2.5.3 |
| 1. View a report of all change items whose status is not “Done” or “Cancelled,” for a product. | 4.2.5.1 => 4.2.5.4 |
| 1. View all change items in the system, and select to query | 4.2.5.1 => 4.2.5.5 |

#### 4.2.1: Main Menu

Refer to 4.1.2 for Exceptional behaviour in this menu.

##### 4.2.2.1: Begin main processes

When launched the program will print the main menu. From here the user can access the main features of the program.

Main Menu:

[1] Handle Change Request

[2] Update Change Item

[3] Manage Products

[4] Generate Reports

[0] Exit

If the user selects “1” the program begins the process described in 4.2.2.

If the user selects “2” the program begins the process described in 4.2.3

If the user selects “3” the program begins the process described in 4.2.4

If the user selects “4” the program begins the process described in 4.2.5

#### 4.2.2: Handle Change Request

##### 4.2.2.1: Start handling a change request

Selecting ‘1’ from the main menu begins the process to create a change request, and if needed a requester, or a change item. The program will begin the process described in section 4.2.2.2

##### 4.2.2.2: Choose, or create a new requester for this change request

Refer to section 4.1.2 for exceptional behaviour in this section.

The user is prompted to select an existing user from a list or choose to create a new user.

In the entries of the list the requester’s name is printed in a 30 width field, the requester’s phone number is printed in a 15 width field and the requester’s email is written in a 24 width field. All fields are separated by 2 spaces.

Handle Change Request:

Select Requester:

NAME: Phone: Email:

[1] REQUESTERNAMEREQUESTERNAMEREQU PHONENUMBERPHON REQUESTEREMAILREQUESTERE

[2] Name +1 101-101-1010 mail@mail.ca

.

.

.

[16] More Elaborate Name is Big +1 604-604-6044 sphere@cube.com

[0] Back [C] next page [N] Create New Requester

If the user makes a selection in [1,16] the program will begin the process described in section 4.2.2.5, otherwise if the user selects (“N” | “n”) the program will begin the process described in section 4.2.2.3.

##### 4.2.2.3: Collect information for new requester

Refer to sections 4.1.1, and 4.1.2 for exceptional behaviour in this step.

The user is prompted to enter the name of the change requester.

Handle Change Request:

Get Requester Name:

[0] Back

Field Requirements:  
(1-30 alphanumeric characters)

Enter Requester’s Name:

Once the user enters (**$**)({**$**}29) the program will continue to the next menu where the user is prompted to enter the requester’s phone number.

Handle Change Request:

Get Requester Phone Number:  
[0] Back

Field Requirements:  
(exactly 10 numeric characters)

Enter Requester’s Phone Number:

Once the user enters **##########**, the program will continue to the next menu where the user is prompted for the requester’s email.

Handle Change Request:

Get Requester Email:

[0] Back

Field Requirements:  
(1-24 alphanumeric characters)

Enter Requester’s Email:

Once the user enters (**$**)({**$**}23), the program will continue to the next menu where the user is prompted to select whether or not to enter a department.

Handle Change Request:

Get Department?:

[1] No Department

[2] Enter Department

[0] Back

If the user selects “1” the program will begin the processes described in section 4.2.2.4, if the user selects “2” the program will print the following menu prompting the user for the name of the department.

Handle Change Request:

Get Department:

[0] Back

Field Requirements

(1-12 alphanumeric characters)

Enter Requester’s Department:

Because in this prototype a department object is not maintained, in this version departments are entered instead of selected each time a requester object is being made.

Once the user enters (**$**)({**$**}11) the program will begin the process described in section 4.2.2.4.

##### 4.2.2.4: Finalise a new requester

Refer to sections 4.1.2 and 4.1.3 for exceptional behaviour in this section.

(“ALL CAPS”) text in the menu will be replaced with information provided by the user in section 4.2.2.3. If the requester does not have a department, the department line will be omitted from the printout.

Handle Change Request:

Confirm Requester Information:

Name: “REQUESTER NAME”

Phone: “PHONE NUMBER”

EMAIL: “EMAIL”

Department: “DEPARTMENT”

Would you like to add the requester?

[y/n] :

If the user selects (“Y” | “y”) they program will try to save the new requester, print the message (Requester Added), and continue to the process described in section 4.2.2.5, otherwise if the user selects (“N” | “n”) they will return to section 4.2.2.2 restarting the requester selection process.

ADDITIONAL EXCEPTIONAL BEHAVIOUR: If the user tries to create a requester object that is identical to an existing requester object they will not be allowed. The program will print the error described in 6.1.7. The user can return to the main menu or the program can begin the process described in section 4.2.2.5 using the equivalent requester object, already saved in the database.

##### 4.2.2.5: Collect information about change request

Refer to sections 4.1.1 and 4.1.2 for exceptional behaviour in this section.

The user is prompted to select the product that the change request is for.

Handle Change Request:

Select Product Needing Change:

Product Name

[1] ProductA

[2] ProductB

.

.

.

[16] ProductN

[0] Back [C] Next Page

If the user selects an index in [1,16] the program will continue to the next menu where the user is prompted to select the release that the change is requested for.

Only releases of the product selected in this section will be listed.

Handle Change Request:

Select Release Needing Change:

Release Name

[1] ReleaseA

[2] RelaseB

.

.

.

[16] ReleaseN

[0] Back [C] Next Page

If the user selects an index in [1,16] the program will continue to the next menu, prompting the user to enter the date.

Handle Change Request:

Get Date:

[0] Back

Field Requirements:

(YYYY-MM-DD: Y, M, D, are numeric characters, include dashes)

Enter Date:

If the user enters (**####**)(“**-**”)(**##**)(“**-**”)(**##**) the program continues to the next menu prompting the user to enter a description of the change requested.

Handle Change Request:

Get Description:

[0] Back

Field Requirements:

(1-30 alphanumeric characters)

Enter Description:

Once the user enters (**$**)({**$**}29) the program will continue to the next menu where the user will be prompted to select an existing change item that already describes the change being requested or allows them to create a new change item if no existing change item is a suitable match.

The program lists descriptions of the change items in a 30 width field, followed by the change ID in a 6 width field, all fields are separated by 2 spaces.

Only change items of the product selected in this section will be shown.

(“DESCRIPTION”) wil be replaced with the description the user entered in this section.

Handle Change Request:

Does a Change Item described by “DESCRIPTION” Exist?:

Description ID

[1] DESCRIPTIONDESCRIPTIONDESCIRPT CHANGE

[2] DescriptionA 123

.

.

.

[16] DescriptionN 123456

[0] Back [C] Next Page [N] Create New Change Item

If the user selects an index in [1,16] the program will attempt to save a change request. The request handling process will complete, the program will print the message (Change Request Added), and will return to the main menu (process described in section 4.2.1). Otherwise if the user selects (“N” “n”) to create a new change item program will begin the process described in section 4.2.2.7.

##### 4.2.2.7: Creating a change item

The user is prompted to select a priority for the change item.

Refer to 4.1.2 and 4.1.3 for exceptional behaviour in this step.

Handle Change Request:

Enter Priority of Change:

[1] Lowest

[2] Low

[3] Middle

[4] High

[5] Highest

[0] Back

If the user makes a selection in [1,5] the program will create a change item. It will attempt to save the change item and the change request and will print the messages (Change Item Added), and (Change Request Added). The request handling process will complete and the program will return to the main menu.

#### 4.2.3: Update Change Item

##### 4.2.3.1: Start updating a change item

Selecting ‘2’ from the main menu begins the process to update change items. The program begins the process described in section 4.2.3.2.

##### 4.2.3.2: Filter by product

Refer to 4.1.2 for exceptional behaviour in this section.

The User selects how they will find the change item they want to update.

Update Change Item:

[1] View Items for All Products

[2] View Items for Product

If the user selects “1” the program will begin the process described in section 4.2.3.3 otherwise, if the user selects “2” they will be prompted to select which product to filter.

Update Change Item:

Show Change Items From What Product?:

Product Name:

[1] ProductA

[2] ProductB

.

.

.

[16] ProductN

[0] Back [C] Next Page

If the user selects an index in [1,16] the program will begin the process described in section 4.2.3.3.

##### 4.2.3.3: Filter by status

Refer to section 4.1.2 for exceptional behaviour in this section.

The user specifies which change items they want to have shown.

Update Change Items:

Show What Change Items?:

[1] New Change Items

[2] Reviewed Change Items

[3] In Progress Change Items

[4] All Not Done or Cancelled

[5] All

[0] Back

If the user selects an index in [1,5] the program will continue to section 4.2.3.4

##### 4.2.3.4: Select change item to update

Refer to section 4.1.2 for exceptional behaviour in this section.

Change Items are listed for the user to select. The items printed in this step are dependent on the selections in sections 4.2.3.3 and 4.2.3.4 If the user selected a product in section 4.2.3.3 only change items from that product will be shown. Only items with a status matching the status selected in section 4.2.3.4 will be shown.

The program lists some information about each change item. It shows the change ID in a 6 width field, followed by the description in a 30 width field, followed by the status in an 11 width field, followed by the product name in a 10 width field, all fields are separated by 2 spaces.

Update Change Items:

Select Item to Update:

ID Description Status Product Name

[1] ###### DESCRIPTIONDESCRIPTIONDESCRIPT STATUSSTATU PRODUCTNAM

[2] 23456 the movie ants IN PROGRESS ants

.

.

.

[16] 999999 sadly full of bugs NEW myWalls

[0] Back [00] Back to Main Menu [C] Next Page

If the user selects an index on [1,16], choosing a change item, they will continue to the next menu where they are prompted to select the field of that change item they wish to update.

##### 4.2.3.5: Update Menu

Refer to section 4.1.2 for exceptional behaviour in this section.

(“ALL CAPS”) text in the menu will be replaced with the corresponding information of the change item most recently selected in this section 4.2.3.4.

Update Change Item:

Change Item: “CHANGEID”

Product: “PRODUCT NAME”

[1] Status: “STATUS”  
[2] Description: “DESCRIPTION”

[3] Priority: “PRIORITY

[4] Release: “RELEASE ID”

Release Date: “RELEASE DATE”

[0] Back [00] Back to Main Menu

If the user selects “1” the program begins the process described in section 4.2.3.6.

If the user selects “2” the program begins the process described in section 4.2.3.7.

If the user selects “3” the program begins the process described in section 4.2.2.8.

If the user selects “4” the program begins the process described in section 4.2.2.9.

ADDITIONAL EXCEPTIONAL BEHAVIOUR: If the user tries to change the status of a change item that is done or cancelled; the user selects “1” while the status of the change item is “done” or “cancelled.” they will not be allowed to the change status process will not start and the user will receive the error described in 6.16

**Section 4.2.3.6: Updating a status**

Refer to 4.1.2 for Exceptional behaviour in this section.

The user will be prompted to select a change item’s new status.

Update Change Item:

Update Status:

[1] Reviewed

[2] In Progress

[3] Done

[4] Cancelled

[0] Back

If the user selects (“1” | “2”) the program will update the status, print the message (Priority Updated), and will begin the process described in section 4.2.3.5. If the user selects (“3” | “4”) the program will get the user to confirm their choice in the following menu.

(“NEW STATUS”) is replaced with “done” if the user selected “3,” or “cancelled” if the user selected “4.”

Update Change Item:

Update Status:

Confirm change to “NEW STATUS”?

[y/n] :

Selecting (“N” | “n”) is equivalent to choosing back. If the user selects (“Y” | “y”), then if the status is being updated to “done,” the program will update the status and will begin the process described in section 4.2.3.9. Otherwise, if the status is being changed to “cancelled” the program will begin the process described in section 4.2.3.5. In both cases after updating the status the program will print (Status Updated)

##### 4.2.3.7: Updating a description

Refer to 4.1.1 for exceptional behaviour in this menu.

The user will be prompted to enter a new description.

Update Change Request:

Update Description:

[0] Back

Field Requirements:

(1-30 alphanumeric characters)

Enter Description:

Once the user enters (**$**)({**$**}29) the program will update the description, print the message (Description Updated), and begin the process described in section 2.4.3.5.

##### 4.2.3.8: Updating a Priority

Refer to 4.1.2 for exceptions in this step.

The user will be prompted to select a new priority.

Update Change Item:

Update Priority:

[1] Lowest

[2] Low

[3] Middle

[4] High

[5] Highest

[0] Back

If the user selects an index in [1,5] the change item’s priority will be updated to the priority corresponding to the index, the program will print (Priority Updated) and the program will begin the process described in section 4.2.3.5.

##### 4.2.3.9: Updating a change item’s release

Refer to section 4.1.2 for exceptional behaviour in this section.

The user will be prompted to select the release of the product that the change item is to be completed and released in.

Only releases of the product associated with the change item will be shown.

Update Change Item:

Select Change Item’s Release:

Release Name:

[1] ReleaseA

[2 ] RelaseB

.

.

.

[16] ReleaseN

[0] Back [C] Next Page

If the user selects an index in [1,16] the program will update the release and consequently the release date of the change item, print the message (Change Item Release Updated), and begin the process described in section 4.2.3.5.

#### 4.2.4: Manage Products

##### 4.2.4.1:Start managing products

Selecting “3” from the main menu begins the process to manage products. The program will begin the process described in 4.2.4.2.

##### 4.2.4.2: Choose whether to create a product or a release of a product

Refer to 4.1.2 for exceptional behaviour in this step.

Prompt the user to select how they want to manage products.

Manage Products:

[1] Create Product

[2] Create Product Release

[0] Back

If the user selects “1” the program will begin the process described in section 4.2.4.3.

If the user selects “2” the program will begin the process described in section 4.2.4.4.

##### 4.2.4.3: Adding a Product

Refer to 4.1.1, 4.1.2, and 4.1.3 for exceptional behaviour in this step.

The user will give a name for, and create a new product.

The user is prompted to enter a name for the new product.

Add a Product:

Define Product Name:

[0] Back  
  
Field Requirements:  
(1-10 alphanumeric characters)

Enter Product name:

Once the user enters (**$**)({**$**}9) the program will continue to the next menu and the user will be prompted to confirm the creation of a new product.

ADDITIONAL EXCEPTIONAL BEHAVIOUR: if there is already an existing product in the system with the name as the one the user entered, then the program will not let the user create that product. The program will print the error message described in 6.1.8 and begin the process described in section 4.2.4.2.

(“PRODUCT NAME”) will be replaced with the name the user entered in this section.

Would you like to create the product “PRODUCT NAME”?  
[y/n] :

If the user selects (“Y” | “y”) the program will attempt to save the new product, then will print (Product Added) and return to begin the process described in section 4.2.4.2. If the user selects (“N” | “n”) the program will restart the process described in this section

##### 4.2.4.4: Adding a product release

Refer to 4.1.1, 4.1.2, and 4.1.3 for exceptional behaviour in this step.

The user will select a product and define the qualities of a new release for that product.

The user is prompted to select a product.

Add a Product Release:

Add a Release for Which Product?:

Product Name:

[1] ProductA

[2] ProductB

.

.

.

[16] ProductN

[0] Back [C] Next Page

If the user selects an index in [1,16] the program will continue to the next menu prompting the user for the name of the new product release.

Add a Product Release:

Define Release Name:

[0] Back  
  
Field Requirements:  
(1-8 alphanumeric characters)

Enter Release name:

Once the user enters (**$**)({**$**}7) the program will continue and prompt for a release date for this new release.

ADDITIONAL EXCEPTIONAL BEHAVIOUR: if there is already an existing product release in the system with the same name the user entered then the program will not let the user create this release. The program will print the error message described in 6.1.9 and begin the process in 4.2.4.2.

Add a Product Release:

Get Release Date:

[0] Back

Field Requirements:

(YYYY-MM-DD: Y, M, D, are numeric characters, include dashes)

Enter Date:

Once the user enters (**####**)(“**-**”)(**##**)(“**-**”)(**##**) the program will continue and the user will be prompted to confirm the creation of a new release.

(“PRODUCT NAME”) will be replaced with the name of the product selected in this step. (“PRODUCT RELEASE NAME”) will be replaced with the name the user entered in this step. (“RELEASE DATE”) will be replaced with the date the user entered in this step.

Would you like to create the “PRODUCT NAME” release “PRODUCT RELEASE NAME” with the release date: “RELEASE DATE”?  
[y/n] :

If the user selects (“Y” | “y”) the program will attempt to save the new product, then will print (Release Added) and begin the process in section 4.2.4.2. If the user selects (“N” | “n”) the program will restart the process described in this section.

#### 4.2.3: Generate Reports

##### **4.2.5.1:** Start generating reports.

Selecting “4” from the main menu begins the process to print a report to the screen. The program will begin the process described in section 4.2.5.2

##### 4.2.5.2: Choose report type

Refer to 4.1.2 for exceptional behaviour in this section.

The user is prompted to select what they want to have reported.

Generate Reports:

Select Report Type:

[1] All Requesters That Have Requested Some Change

[2] All Change Items of a Product Not Done or Cancelled

[3] View Change Item

[0] Back

If the user selects “1” the program begins the process described in section 4.2.5.3.

If the user selects “2” the program begins the process described in section 4.2.5.5.

If the user selects “3” the program begins the process described in section 4.2.5.6.

##### 4.2.5.3: Report all requesters who need to know about a change

Refer to 4.1.2 for exceptional behaviour in this menu.

The user will select a product, a change item of that product, and confirm their choice, to see a report of all users who have requested that selected change item.

The user is prompted to select a product from a list.

Generate Reports:

Select a Change Item From What Product?:

Product Name:

[1] PRODUCTNAM

[2] ProductA

.

.

.

[16] ProductM

[0] Back [00] Back to Main Menu [C] Next Page

Once the user makes a selection in [1,16] the user is prompted to select a change item from a list. Only change items of the selected product will be shown.

The program lists some information about each change item. it shows the change ID in a 6 width field, followed by the release ID in an 8 width field. Followed by the description in a 30 width field, followed by the status in an 11 width field, all fields are separated by 2 spaces.

Generate Reports:

Generate Report for Which Change Item?:

ID Description Status Release

[1] ###### DESCRIPTIONDESCRIPTIONDESCRIPT STATUSSTATU VersionA

[2] 23456 the movie ants IN PROGRESS Ver2

.

.

.

[16] 999999 sadly full of bugs NEW myWalls

[0] Back [00] Back to Main Menu [C] Next Page

When the user selects an index in [1,16] the program will show more details about that change and in the following menu will ask the user to confirm that they want to create a report for that change.

(“ALL CAPS”) text in the menu will be replaced with the attributes of the change item selected in this section.

Generate Reports:

Change Item: “CHANGEID”

Product: “PRODUCT NAME”

Status: “STATUS”  
Description: “DESCRIPTION”

Priority: “PRIORITY

Release: “RELEASE ID”

Release Date: “RELEASE DATE”

Create a Report for Change Item “ChangeID”?

[y/n]:

If the user selects (“N” | “n”) the program will restart the process described in this section. If the user selects (“Y” | “y”), the program will produce the requested report to the screen.

(“PRODUCT”) will be replaced by the name of the product selected in this section. (“RELEASE”) will be replaced by the release ID attribute of the change item selected in the section, and (“RELEASE DATE”) will be replaced with the release date of that release.

Seventeen lines of change requesters will be printed. three lines are reserved for the release information, attribute headings, and menu options. Each requester will have their name, phone number, and email displayed. Name will be printed in a 30 width field, phone number will be printed in a 15 width field, and email will be printed in a 24 width field, all fields will be separated by 2 spaces.

List entries are placeholders for change requesters in the user’s database.

Release Date: “RELEASE DATE” Product: “PRODUCT” Release: “RELEASE”

Name Phone Email

REQUESTERNAMEREQUESTERNAMEREQU PHONENUMBERPHON REQUEASTEREMAILREQUESTER

Little Critter +1 123-456-7890 little@critter.ca

.

.

.

My name is eighteen +1 818-181-8181 eightteen@yahoo.gov

[00] Main Menu [C] Next Page [N] New Report

If the user selects (“N” | “n”) to create a new report the program will begin the process described in section 4.2.5.2.

##### 4.2.5.4: Report all unresolved change items of a product

Refer to 4.1.2 for exceptional behaviour in this menu.

The user will select a product to see a report of all change items associated to that product that are “in progress,” “new,” or “reviewed.”

The user is prompted to select a product from a list.

Generate Reports:

Generate Report for What Product?:

Product Name:

[1] PRODUCTNAM

[2] ProductA

.

.

.

[16] ProductM

[0] Back [00] Back to Main Menu [C] Next Page

Once the user makes a selection in [1,16] the program will produce the desired report for that product.

(“PRODUCT”) will be replaced by the name of the product selected by the user in this step.

Seventeen lines of change items will be printed, three lines are reserved for product information, attribute headings, and menu options. The program will print some information for each change item. The change ID will be printed in a 6 width field, the description will be printed in a 30 width field, the status will be printed in an 11 width field, the priority will be printed in a 1 width field, and the release will be printed in an 8 width field, all fields will be separated by 2 spaces.

Product: “PRODUCT”

ID Description Status Priority Release

###### DESCRIPTIONDESCRIPTIONDESCRIPT STATUSSTATU # RELEASER

23456 the movie ants IN PROGRESS 5 sample

.

.

.

999999 sadly full of bugs NEW 1 zoom

[0] Back [00] Back to Main Menu [C] Next Page [N] New Report

If the user selects (“N” | “n”) to create a new report the program will restart the process described in section 4.2.5.4.

##### 4.2.5.5: Query specific change item

Refer to 4.1.2 for exceptional behaviour in this step

The user is shown all change items and selects one to see all information for that change item.

The user is prompted to select a change item from a list.

The program lists some information about each change item. it shows the change ID in a 6 width field, followed by the product name in a 10 width field. Followed by the description in a 30 width field, followed by the status in an 11 width field, all fields are separated by 2 spaces.

Generate Reports:

Select Item to View:

ID Description Status Product Name

[1] ###### DESCRIPTIONDESCRIPTIONDESCRIPT STATUSSTATU PRODUCTNAM

[2] 23456 the movie ants IN PROGRESS ants

.

.

.

[16] 999999 sadly full of bugs NEW myWalls

[0] Back [00] Back to Main Menu [C] Next Page

If the user selects an index in [1,16] the program will print a detailed report of that change item to screen.

(“ALL CAPS”) text in the menu will be replaced with the corresponding information of the change item selected in this section.

View Change Item:

Change Item: “CHANGEID”

Product: “PRODUCT NAME”

Status: “STATUS”  
Description: “DESCRIPTION”

Priority: “PRIORITY

Release: “RELEASE ID”

Release Date: “RELEASE DATE”

[0] Back [00] Back to Main Menu

# Section 5: Database Back-Up Procedure

It is recommended that you backup your files once every couple of weeks in order to protect your files from being damaged, corrupted, or lost. You can back-up these files by following the steps below.

1. Open the terminal on your computer by either:
   1. Press Ctrl+Alt+T
   2. Click on the “show applications” icon on the bottom left of your screen and then click on the search bar in the middle of your screen and type in “terminal” in the search bar and press **Enter (↵)** on your keyboard.
2. Type “**c:**” in the terminal and then press **Enter (↵)** on your keyboard
3. Type “**cd IssueTrackerSystem**” in the terminal and then press **Enter (↵)** on your keyboard
4. Place an empty floppy disk into the floppy disk drive
5. Type “**copy datafiles a:\**” and then press **Enter (↵)** on your keyboard

You can transfer these files back to your hard drive by following the steps below.

1. Open the terminal on your computer by either:
   1. Press Ctrl+Alt+T
   2. Click on the “show applications” icon on the bottom left of your screen and then click on the search bar in the middle of your screen and type in “terminal” in the search bar and press **Enter (↵)** on your keyboard.
2. Type “**c:**” in the terminal and then press **Enter (↵)** on your keyboard
3. Type “**cd IssueTrackerSystem**” in the terminal and then press **Enter (↵)** on your keyboard
4. Place an empty floppy disk into the floppy disk drive
5. Type “**copy a:\\*.\* datafiles**” and then press **Enter (↵)** on your keyboard

# 

# Section 6: References

## 6.1: Error Messages

**6.1.1 Selected Option is Not Available**

The following error message occurs when the user enters an option that is not presented to them by the system:

The option that you have entered does not exist in the system.

The previous options are then presented to the user to re-enter their choice.

**6.1.2 Incorrect Character Type**

The following error message occurs when the user enters the incorrect data format for a prompt that was given to them (Please refer to **Section 6.2** to see the correct data formats):

The value that you have entered does not meet the character type requirements for the given field.

Enter Y to try again. Enter N to return to the Main Menu.

**6.1.3 The Value Entered Exceeds the Character Limit**

The following error message occurs when the user enters a value that exceeds the boundaries / character limit of the requirement field (Please refer to **Section 6.2** to see the correct data formats):

The value that you have entered exceeds the bounds set by the data format requirements.

Enter Y to try again. Enter N to return to the Main Menu.

**6.1.4 The Value Entered Is Below the Character Requirement**

The following error message occurs when the user enters a value that does not meet the boundaries of the requirement field (Please refer to **Section 6.2** to see the correct data formats):

The value that you have entered is below the bounds set by the data format requirements.

Enter Y to try again. Enter N to return to the Main Menu.

**6.1.5 Save Failure**

The following error message occurs when an item cannot save, likely due to a hardware failure:

The file could not be saved at this time.

Enter Y to try again. Enter N to return to the Main Menu.

**6.1.6 Cannot Edit Status Out of ‘Done’ or ‘Cancelled’**

The following error message occurs when the user tries update the status of an item that is in the ‘Done’ or ‘Cancelled’ state:

The change item’s status cannot be changed out of its status.

Enter N to return to the Main Menu.

**6.1.7 The Requester Already Exists**

The following error message occurs when the information of a new requester that is being entered by the user matches the information of an already existing requester:

The requester already exists.

Enter Y to continue creating a change request. Enter N to return to the main menu.

The user can choose to continue creating a change request with this requester object.

**6.1.8 The Product Already Exists**

The following error message occurs when a user is creating a new product and the product already exists:

The product already exists.

Enter Y to try again. Enter N to return to the main menu.

**6.1.9 The Product Release Already Exists**

The following error message occurs when a user is creating a new product release and the ID matches an already existing one:

The product release already exists.

Enter Y to try again. Enter N to return to the main menu.

## 6.2: Data Format

Please refer to **Section 1.3** for the typological conventions.

Product Name: ($){$}9

Release ID: ($){$}7

Change ID: (#){#}5

Email Address: ($){$}23

Department: ($){$}11

Description: ($){$}29

Priority: (“1” | “2” | “3” | “4” | “5”)

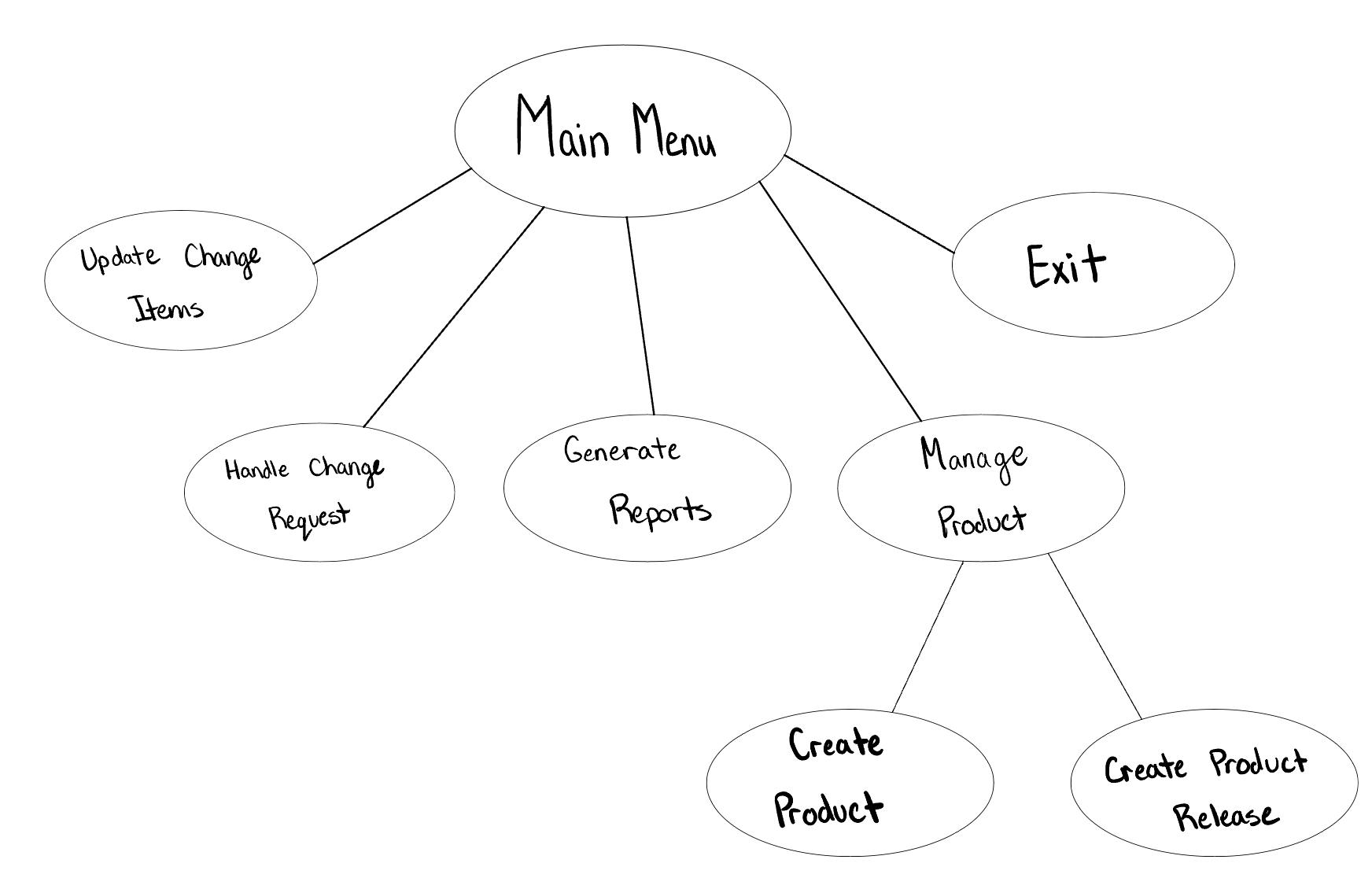
Phone Number: **##########**

Dates: (**####**)(“**-**”)(**##**)(“**-**”)(**##**)

Requester Name: ($){$}29

## 6.3: Menu Tree

Tree Menu:



Main Menu:

[1] Handle Change Request

[2] Update Change Item

[3] Manage Products

[4] Generate Reports

[0] Exit

Entering ‘**1**’ in the main menu would not display any submenus to the user, and instead it will guide the user through prompts to carry out the process of creating a change request, including creating a requester (conditional). Although at any time throughout this process, the user may enter ‘**0**’ to go back to the main menu.

[0] Back

Entering ‘**2**’ in the main menu would bring up lists for the user to select the intended change item to update. After the change item is selected, it will bring up:

Update Change Item:

[1] Status  
[2] Description

[3] Priority

[4] Product Release

[0] Back [00] Back to Main Menu

Entering ‘**3**’ in the main menu would bring up the submenus:

Manage Products:

[1] Create Product

[2] Create Product Release

[0] Back

Entering ‘**4**’ in the main menu would bring up the submenus:

Generate Reports:

Select Report Type:

[1] All Requesters That Have Requested Some Change

[2] All Change Items of a Release Not Done or Cancelled

[3] View Change Item

[0] Back

Entering ‘**0**’ in the main menu would exit the program.

# Section 7: Terms Glossary

**Alphanumeric**

A type of character that is either an alphabet or a number.

**Character**

A single alphabet or number. Eg: a,b,c,..,x,y,x,A,B,C,...,X,Y,Z,1,2,3,….

**Command-Line**

The line on a terminal where you type in the command and press enter to have it executed.

**Floppy Disk**

A type of disk storage that has a magnetic storage component enclosed in a fabric-lined plastic.

**GB**

Short for Gigabyte - a measurement of the number of bytes, typically used to describe memory capacity.

**Integer**

A positive or negative whole number (e.g:.-3,-2,-1,0,1,2,3….)

**Numeric**

A type of character that is a number.

**SSD**

Short for Solid-State Drive, it is a storage drive that tends to be light, durable, and energy-efficient, and generally a part of a laptop’s internal hardware.

**TB**

Short for Terabyte - a measurement of the number of bytes, typically used to describe memory capacity.

**Terminal**

A text interface to the computer

**Ubuntu**

A Linux distribution (Linux is an open source operating system).

**Whole Number**

Any non-negative integer. (e.g: 0,1,2,3,.....)

1. Refer to Section 7: Terms Glossary for the definitions of all technical terms in field requirements [↑](#footnote-ref-0)
2. Refer to Section 7: Terms Glossary for the definitions of all technical terms in field requirements [↑](#footnote-ref-1)