CMPT 276 Group 20

Macrohard Corp.

TrackMaster

Release 5

Release Date: 2024/07/31

Approved By:

Saadati Aryan, *Project Manager* ______________________

Bowen Jin, *CEO* ______________________

Sit Kwan Wai, *CFO* ______________________

James Isaac, *Chief of Staff* ______________________

Release History Page

Rel. 1 - 2024/06/07

- Was composed of just the preliminary Requirements Spec.

Rel. 2 - 2024/06/19

- Initial release of the user manual, providing comprehensive guidance on *TrackMaster's* features and functionalities

Rel. 3 - 2024/07/03

- Initial release of the architectural design document as well as modules headers and source code.

Rel. 4 - 2024/07/17

- Initial release of the detailed design document as well as modules implementation code and unit test cases.

Rel. 4 - 2024/07/31

- Initial release of the Integration Report, updated User Manual, enhanced modules, implementation code, and unit test cases.

Release Table Page

Configuration Item:	Version	Number of Each Configuration Item in:			
	Rel. 1	Rel. 2	Rel. 3	Rel. 4	Rel. 5
Documents					
Requirements Spec	1	1	1	1	1
User Manual		1	1	1	2
Architectural Design Document			1	1	1
Detailed Design Document				1	1
Integration Report					1
Modules					
Main.cpp			1	1	2
Backup.h				1	2
Backup.cpp				1	2
Change.h			1	1	2
Change.cpp				1	2
EntityIO.h					1
EntityIO.cpp					1
FileNotOpenException.h				1	1
FileNotOpenException.cpp				1	1
FileOpenFailedException.h				1	1
FileOpenFailedException.cpp				1	1
GlobalEntities.h					1
LifeCycleController.cpp					1

LifeCycleController.h			1
RecordNotFoundException.h		1	1
RecordNotFoundException.cpp		1	1
Request.h	1	1	2
Request.cpp		1	2
Requester.h	1	1	2
Requester.cpp		1	2
PrintController.h	1	1	2
PrintController.cpp		1	2
Product.h	1	1	2
Product.cpp		1	2
ProductRelease.h	1	1	2
ProductRelease.cpp		1	2
ScenarioController.h	1	1	2
ScenarioController.cpp		1	2
testFileOps.cpp		1	2
UI.h	1	1	2
UI.cpp	1	1	2
unitTestMain.cpp	1	1	2

CMPT 276 Group 20

Macrohard Corp.

TrackMaster

Integration Report

Version 1

Written by:

James Isaac ____________________________

Jin Bowen ____________________________

Saadati Aryan ____________________________

Sit Kwan Wai ____________________________

Approved by:

Quality Assurance Dept. ____________________________

Document Version History

Version 1 - Original version released in 2024/07/17 by James Isaac, Jin Bowen, Saadati Aryan, and Sit Kwan Wai

Table of Contents

1) Integration Discussion

At Macrohard, our team utilized a hybrid integration strategy that combined both bottom-up and big-bang integration approaches. Initially, we began with a bottom-up integration strategy, focusing on the low-level modules such as Request, Requester, Change, Product, and ProductRelease. These modules were initially tested in isolation using test drivers to verify their functionality against the specifications outlined in their respective header files. Addressing issues in these isolated lower-level components early on allowed us to resolve these errors before integrating the modules into more complex, higher-level modules. This method proved highly efficient for our team, as it allowed each member to work on different modules simultaneously, thereby reducing overall development time.

Once the low-level modules had undergone testing, we decided to proceed with a big-bang approach for integrating the remaining high-level modules. This strategy was chosen because it allowed us to capitalize on the reliability of the previously tested lower-level modules. By doing so, any new errors that emerged could be attributed to the recently integrated high-level modules. Additionally, the big-bang integration approach simulated the conditions of the final, fully integrated program, which helped in identifying issues that did not become apparent during the isolated testing of individual modules. It also provided an opportunity for white box stress testing, which was not feasible with the individual test drivers used for earlier testing phases. Through this approach, we were able to uncover and address bugs that spanned multiple modules, ensuring that the TrackMaster program functioned exactly as advertised.

2) Test Case Reports

2.1) Normal Test Case

This test case is to simulate a real use case for *TrackMaster*, where a user wishes to first create a change request, then update the change item. To test if these operations were successful, we can inquire about the change item to display all information related to its relevant product and change.

<u>Preconditions:</u>

- Start in the main menu, after the launch of *TrackMaster*.
- The program contains previous products, requesters and change items. For test case purposes, we will use the requester Front Tussell, and the change regarding the product FreddyBear.

Start the program by clicking on the *TrackMaster* icon. You will be presented with the main menu:

```
=== MAIN MENU ===
1) Create
2) Update
3) Inquire
4) Print
5) Backup
0) Quit TrackMaster
ENTER selection [0-5]: 1
```

Step 1: ENTER **`<1>`** to enter the Create menu.

```
=== Create Menu ===
1) Create Request
2) Create Requester
3) Create Product
4) Create Product Release
0) Main Menu
ENTER selection [0-5]: 1
```

Step 2: Once in the create sub-menu, ENTER **`<1>`** to create a request.

```
Is this request coming from a customer or an employee?
ENTER a selection [c - customer/e - employee]
OR ENTER <0> to abort and exit to the main menu: c
```

Step 3: ENTER **`<c>`** to indicate the request coming from a customer.

```
=== Select Customer ===
  Name                   Email
1)Front Tussel           ft@sfu.ca
2)Mark Zuck              markzuck@metuh.com
3)Jeff Bozos             jeffb@amoozon.com
4)New Customer
ENTER selection [1-4] OR <0> to abort and exit to the main menu: 1
```

Step 4: ENTER **<1>** to select the customer ‘Front Tussell’

```
ENTER the DATE of the request (YYYY-MM-DD) OR ENTER <0> to abort and exit
to the main menu: 2024-06-29
```

Step 5: ENTER the date of the request in the requested format..

```
=== Select Product ===
1)freddy
2)cheese
3)goober
4)asdasdad
5)asdadasd
ENTER selection [1-7] OR <0> to abort and exit to the main menu: 2
```

Step 6: ENTER **<2>** to choose the product ‘FreddyBear’.

```
=== Select Product Release ===
1) 1.0
2) 1.1
3) 1.2
4) 1.3
ENTER selection [1-4] OR <0> to abort and exit to the main menu: 4
```

Step 7: ENTER **<4>** to choose the release ‘1.3’.

```
=== Select a Priority ===
1) Lowest
2) Low
3) Medium
4) High
5) Highest
ENTER selection [1-5] OR <0> to abort and exit to the main menu: 5
```

Step 8: ENTER **<5>** to choose the priority ‘Highest’.

```
=== Select Change Item ===
   Description                       ChangeID
1) Freddy’s voice is squeaky         1234
2) New Change
ENTER selection [1-2] OR <0> to abort and exit to the main menu: 2
```

Step 9: ENTER **<2>** to create a new change.

```
ENTER a description for the new change (30 char. max): Freddy bites too
much.
```

Step 10: ENTER the description in the requested format.

```
Select a product release that the change is anticipated to be completed
for.
=== Select Product Release ===
1) 1.0
2) 1.1
3) 1.2
4) 1.3
5) Skip
ENTER selection [1-4] OR <0> to abort and exit to the main menu: 4
```

Step 11: ENTER **<4>** to choose the release ‘1.3’.

```
=== New Request Information===
Requester (customer): Front Tussell - ft@sfu.ca
Product: FreddyBear
Release: 1.3
Priority: 5 - Highest
Change ID: 1234
ENTER <1> to confirm OR <0> to abort and exit to main menu: 1
```

Step 12: ENTER **<1>** to confirm, and then you will be redirected to the main menu.

```
=== MAIN MENU ===
1) Create
2) Update
3) Inquire
4) Print
5) Backup
0) Quit TrackMaster
ENTER selection [0-5]: 2
```

Step 13: ENTER **<2>** to enter the Update menu.

```
=== Update Menu ===
1) Assess new change items
2) Update a change item
0) Main Menu
ENTER selection [0-2]: 1
```

Step 14: ENTER **<1>** to assess new change items.

```
=== Select Change Item ===
   Product        Description                    Status
1) FreddyBear     Freddy’s voice is squeaky      Open
2) FreddyBear     Freddy Bites too much          Open
ENTER selection [1-2] OR ENTER <0> to abort and exit to main menu: 2
```

Step 15: ENTER **<2>** to select the recently added change item.

```
=== Select Status ===
1) Assessed
2) Canceled
ENTER selection [1-2] OR ENTER <0> to abort and exit to main menu: 1
```

Step 16: ENTER **<1>** to select ‘Assessed’.

```
ENTER a new description for the change [max 30 characters, leave blank to
skip] OR <0> to abort and exit to main menu:
```

Step 17: Enter a new description or leave the field blank to indicate no changed description and ENTER.

```
Select a product release that the change is anticipated to be completed
for.
=== Select Product Release ===
1) 1.0
2) 1.1
3) 1.2
4) 1.3
5) Skip
ENTER Selection [1-4] OR <0> to abort and exit to main menu: 4
```

Step 18: ENTER **<4>** to choose the release ‘1.3’.

```
=== Assessed Change Item Information ===
Product: FreddyBear
Description: Freddy bites too much
Anticipated release: 1.3
Status: Assessed
Change ID: 1246
ENTER <1> character to confirm OR 0 to abort and exit to main menu: 1
```

Step 19: ENTER **<1>** to confirm the assessed change item, and then you will be redirected to the main menu.

```
=== MAIN MENU ===
1) Create
2) Update
3) Inquire
4) Print
5) Backup
0) Quit TrackMaster
ENTER selection [0-5]: 2
```

Step 20: ENTER **<2>** to enter the Update menu.

```
=== Update Menu ===
1) Assess new change items
2) Update a change item
0) Main Menu
ENTER selection [0-2]: 2
```

Step 21: ENTER **<2>** to update a change item.

```
=== Select Product ===
1) TakTak
2) FreddyBear
3) Cheeses
4) Rudolph
5) New Product
ENTER selection [1-5] OR <0> to abort and exit to the main menu: 2
```

Step 22: ENTER **<2>** to choose ‘FreddyBear’.

```
=== Select Change Item ===
  Description                       Status            ChangeID
1)Freddy’s voice is squeaky         Assessed          1235
2)Freddy Bites too much             Assessed          1246
ENTER selection [1-2] OR <0> to abort and exit to main menu: 2
```

Step 23: ENTER **<2>** to select the recently added change item.

```
=== Select Status ===
1) In Progress
2) Done
3) Canceled
4) Skip
ENTER selection [1-4] OR <0> to abort and exit to main menu: 1
```

Step 24: ENTER **<1>** to choose ‘In Progress’.

```
ENTER a new description for the change (leave blank to skip) OR <0> to
abort and exit to main menu [max 30 characters]:
```

Step 25: Leave the field blank to indicate no changed description and ENTER.

```
Select a product release that the change is anticipated to be completed
for.
=== Select Product Release ===
1) 1.0
2) 1.1
3) 1.2
4) 1.3
5) Skip
ENTER Selection [1-4] OR <0> to abort and exit to main menu: 4
```

Step 26: ENTER **<4>** to choose the release ‘1.3’.

```
=== Updated Change Information ===
Product: FreddyBear
Description: Freddy bites too much
Anticipated release: 1.3
Status: In Progress
Change ID: 1246
ENTER <1> character to confirm OR 0 to abort and exit to main menu: 1
```

Step 27: ENTER **<1>** to confirm the updated change item, and then you will be redirected to the main menu.

```
=== MAIN MENU ===
1) Create
2) Update
3) Inquire
4) Print
5) Backup
0) Quit TrackMaster
ENTER selection [0-5]: 3
```

Step 28: ENTER **<3>** to enter the Inquire menu.

```
=== Inquire Menu ===
Select the product you wish to inquire a change item from:
1) TakTak
2) FreddyBear
3) Cheeses
4) Rudolph
0) Exit
ENTER Selection [0-4]: 2
```

Step 29: ENTER **<2>** to select ‘FreddyBear’.

```
Select a change item to inquire
=== Select Change Item ===
  Description                        Status            ChangeID
1)Freddy’s voice is squeaky          Assessed          1235
2)Freddy Bites too much              In Progress       1246
ENTER selection [1-2] OR <0> to abort and exit to main menu: 2
```

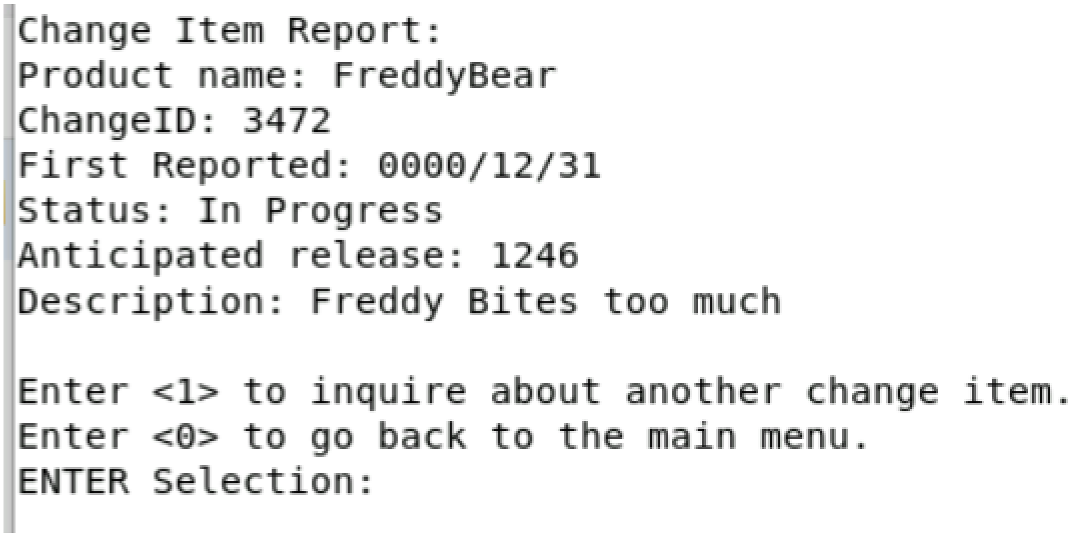
Step 30: ENTER **<2>** to inquire about the recently created and updated change item.

```
Change Item Report:

Product name: FreddyBear
ChangeID: 1246
First Reported: 2024-06-29
Statues: In Progress
Priority: 1
Anticipated release: 1.3
Description: Freddy Bites too much

Enter <1> to inquire about another change item.
Enter <0> to and go back to the main menu.
ENTER Selection: 0
```

Step 31: ENTER **<0>** to go back to the Main Menu.

Expected Results:
If the change item is created and updated successfully, the above screen should appear. This also reflects the success of inquiring about a change item.

Actual Results (Tested on Release 5 of *TrackMaster)*:
The date first reported, change ID and anticipated release ID are different than expected, the format has also changed.



```
Change Item Report:
Product name: FreddyBear
ChangeID: 3472
First Reported: 0000/12/31
Status: In Progress
Anticipated release: 1246
Description: Freddy Bites too much

Enter <1> to inquire about another change item.
Enter <0> to go back to the main menu.
ENTER Selection:
```

This indicates a fail.

2.2) Stress Test Case

This test case tests *TrackMaster* for using invalid input.

Preconditions:

- Start in the main menu, after the launch of *TrackMaster*.
- The program does not contain any previous data.

Start the program by clicking on the *TrackMaster* icon. You will be presented with the main menu:

```
=== MAIN MENU ===
1) Create
2) Update
3) Inquire
4) Print
5) Backup
0) Quit TrackMaster
ENTER selection [0-5]: 1
```

Step 1: ENTER **<1>** to enter the Create menu.

```
=== Create Menu ===
1) Create Request
2) Create Requester
3) Create Product
4) Create Product Release
0) Main Menu
ENTER selection [0-5]: 2
```

Step 2: Once in the create sub-menu, ENTER **<2>** to create a requester.

```
ENTER the EMAIL ADDRESS of the requester (Length: max 30)
OR ENTER <0> to abort and go back to the main menu: ExampleEmail@Email.ca
```

Step 3: ENTER an appropriate email for the requester as shown above.

```
ENTER the PHONE NUMBER of the requester (Length: Exactly 10)
OR ENTER <0> to abort and exit to the main menu: 911
```

Step 4: ENTER the phone number ‘911’ for the requester.

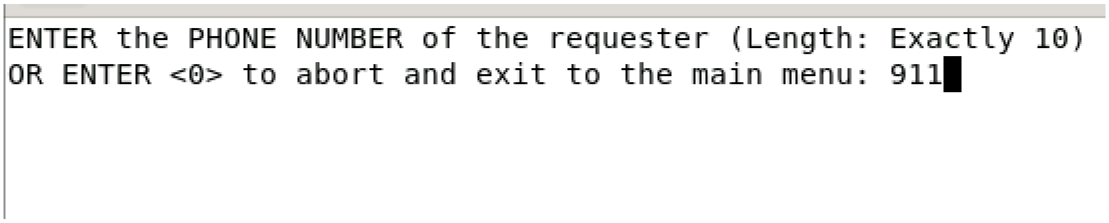
```
Error: Input is invalid. Re-enter input.
Enter 0 to abort and return to the main menu.
```

Step 5: ENTER **<0>** to go back to the main menu.

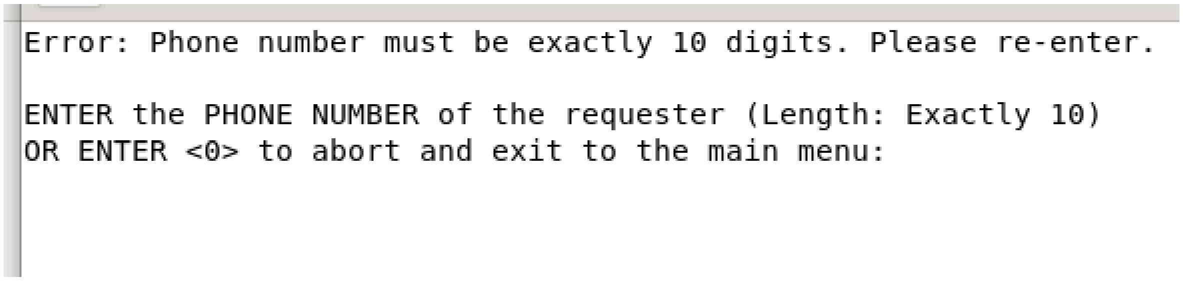
Expected Results:

The above input error will be shown, as the phone number length must be exactly 10.

Actual Results (Tested on Release 5 of *TrackMaster)*:



```
ENTER the PHONE NUMBER of the requester (Length: Exactly 10)
OR ENTER <0> to abort and exit to the main menu: 911
```



```
Error: Phone number must be exactly 10 digits. Please re-enter.

ENTER the PHONE NUMBER of the requester (Length: Exactly 10)
OR ENTER <0> to abort and exit to the main menu:
```

This indicates a pass, as the program continuously prompts the user for a phone number of exactly 10 digits.

2.3) Performance Test Case

This test case tests *TrackMaster* for its response time for inquiring on a change item.

Preconditions:

- Start in the main menu, after the launch of *TrackMaster*.
- The program contains previous products, requesters and change items. For test case purposes, we will use the requester Front Tussell, the change regarding the product FreddyBear and change item with description ‘Freddy’s voice is squeaky.’
- A stopwatch is required to measure response time.
- There are 20 other change items in the database.

Start the program by clicking on the *TrackMaster* icon. You will be presented with the main menu:

```
=== MAIN MENU ===
1) Create
2) Update
3) Inquire
4) Print
5) Backup
0) Quit TrackMaster
ENTER selection [0-5]: 3
```

Step 1: ENTER **<3>** to enter the Inquire menu.

```
=== Inquire Menu ===
Select the product you wish to inquire a change item from:
1) TakTak
2) FreddyBear
3) Cheeses
4) Rudolph
0) Exit
ENTER Selection [0-4]: 2
```

Step 2: ENTER **<2>** to choose ‘FreddyBear’.

```
=== Select Change Item ===
  Description                       Status            ChangeID
1)Freddy’s voice is squeaky         Assessed          1235
ENTER selection [1] OR <0> to abort and exit to main menu: 1
```

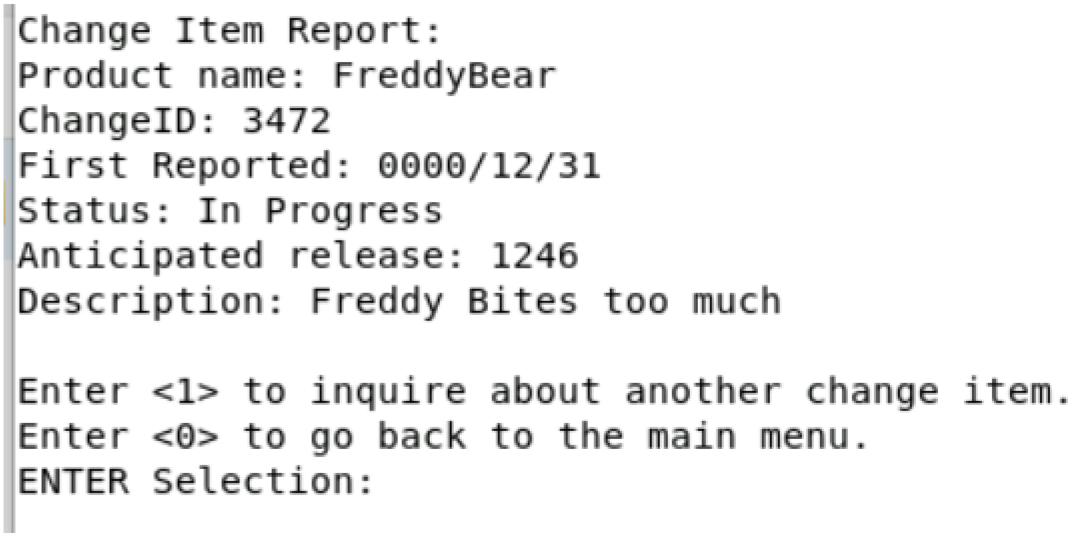
Step 3: ENTER **<1>** to select the change item. Press ‘start’ on the stopwatch at the same time.

```
Change Item Report:

Product name: FreddyBear
ChangeID: 1235
First Reported: 2024-06-29
Statues: Assessed
Priority: 1
Anticipated release: 1.3
Description: Freddy Bites too much

Enter <1> to inquire about another change item.
Enter <0> to and go back to the main menu.
ENTER Selection: 0
```

Step 4: Press ‘stop’ on the stop watch the moment it is displayed. This will give the response time of searching for the change item.

<u>Expected Results:</u>
The system outputs what is specified on step 4, as well as a 0.1 seconds response time.

<u>Actual Results (Tested on Release 5 of *TrackMaster)*:</u>
During the black box testing on step 4, the program outputs:



```
Change Item Report:
Product name: FreddyBear
ChangeID: 3472
First Reported: 0000/12/31
Status: In Progress
Anticipated release: 1246
Description: Freddy Bites too much

Enter <1> to inquire about another change item.
Enter <0> to go back to the main menu.
ENTER Selection:
```

This is not formatted to the user manual, the details regarding the specified change item requested is listed. By timing with a stopwatch, the time output is 0.14 seconds. As the output is different, it indicates a fail.

3) Project Summary

3.1) Issues Encountered and Recommended Changes

Throughout the implementation phase of the project, our team faced several challenges, primarily stemming from coordination issues. Initially, we failed to employ a reliable version control system to manage code written separately by members, instead opting to exchange code via Discord. This led to multiple versions and distinct changes of the same files, which resulted in extensive dedication of time to manually merge the new changes. As a result, we often missed new changes that caused bugs that were difficult to trace. To fix this issue, we began managing source code control through GitHub. By adopting GitHub, we centralized the codebase, allowing all team members to access and contribute to a single, unified repository. This eliminated the confusion caused by multiple versions of the same files and reduced the time spent on manual merges. Through pull requests, we were able to remotely review and discuss changes on call in real time before integrating them into the main branch, thus reducing the need to set meetings constantly. Additionally, the version history and branching capabilities of GitHub allowed us to experiment with new features without disrupting the main codebase. This change led to a more improved efficiency throughout the rest of the implementation phase.

Furthermore, during the start of the implementation phase, there was also a lot of redundancy throughout the program, spanning across multiple modules. In many of the primary modules[1], there were numerous methods that performed the same functionality, leading to a situation where any change or improvement to one of these modules must be replicated across all the rest. Consequently, this increased the risk of inconsistencies and errors, in addition to a needless duplication of effort. This redundancy oversight ended up hampering our overall efficiency, and complicated the primary modules. However, we were able to solve this by including a separate module named EntityIO. This helper module contained the template implementations for these methods, which minimized the redundancy throughout the primary modules. By centralizing these common functionalities, we simplified the maintenance/debugging process, as any modification in the EntityIO module would automatically propagate to all dependent primary modules. Ultimately, this made our debugging and enhancement process significantly more manageable.

[1] Refers to modules Change, Request, Requester, Product, and ProductRelease

3.2) Positive Outcomes

During our implementation phase, several key practices proved to be extremely beneficial. A major advantage that helped with efficiency was the extensive planning that took place within our team before we began coding. This extends to before the implementation phase, as *TrackMaster's* previous document releases essentially served as a blueprint for the team. As we ensured to clearly outline the scope and objectives of the program, these previous documents exponentially sped up the implementation phase, allowing us to write the majority of the code within a day. Furthermore, given that *TrackMaster* is a database-oriented program, a major success for the team was having little to no issues regarding file access. Since our program requires constant retrieval or writing of records, studying the training module from Simba Technologies before we began coding these functionalities was also critical. Additionally, incorporating test drivers for the low-level modules helped catch errors early on in development, allowing each team member to debug and improve their own modules simultaneously.

We believe that the practice of extensive preparation before implementation and testing is not exclusive to only database-oriented projects. In fact, it can be applied broadly across various types of projects, whether they are real-time systems or graphical applications. Although each project will have its own unique objectives and considerations, this significantly enhanced the efficiency of our software development process.