EDENZ NEW ZEALAND
colleges

Diploma in Software Development

DSE 720 Desktop Software Development

Assessment: Project Report

Total marks: 100

Course Weighting: 100%

Due Date: Wednesday, 24 January, 2017

Student Name(s):……………Annie Rose Deloso…………………………………………

EXECUTIVE SUMMARY

Fixed asset or referred as tangible assets are owned by companies or business establishments that provide services for the benefit of the companies' operation. Such fixed assets are land, buildings, furniture and fixtures, equipment and more. Fixed Asset management system monitors and tracks the condition, location, personnel-in-charge, transaction history, and depreciation value of the assets. Some of the establishments are not yet fully aware of the importance of this system that resulted in losing track of their assets or doubling the cost spending for another asset. Though some adapted to the manual paper-based processing, the efficiency of the data is not accomplished.

With the advancement of technology, automation of the manual processing resolves issues and problems in this aspect. The proposed system, fixed asset management system is designed to easily manage fixed asset transaction history within the span of time. The system then generates a report for the location, condition or status, personnel, and history. For the ingoing of the asset, the Developer uses the Receiving transaction to input and store all the necessary information into the database. Then, the transfer transaction is to record the fixed asset new location, personnel liable for the asset and remarks. While maintenance transaction is for the services the fixed asset received. Lastly, the pull-out transaction is the outgoing process of the fixed asset. The main goal of the system is to track the ingoing (receiving) and outgoing(pull-out) specific and updated location of the fixed asset, the person liable for the fixed asset, maintenance history, depreciating value and the pullout information.

The developer used different technologies to develop this application: Eclipse, HeidiSQL and Xampp, MySQL. Eclipse is an IDE (Integrated Development Environment) ready for Java. Xampp was used to set up to use the MySQL server and HeidiSQL is the database manager software. The developer first identified the problems, formulate the solution, set the scope and limitations for the project and started the design, test, and development of the project. During the development, test cases were created first to ensure efficiency, portability, and usability of each functions of the system feature at different scenarios.

The developed fixed asset management system provides the fundamental transaction flow and execute tasks as required. Though, the system is completed it need more improvements as it lacks security features, filtering, and features to make it more functional, useable and flexible to the future users.

Introduction

Background

Describe why you chose this application and what you were planning to achieve.

I chose this application to incorporate my knowledge in fixed asset management and develop a simple application using Java programming language with MySQL backend. Fixed asset management is a useful system to help establishment track the status, location, personnel liable, transaction history and depreciation value of the asset.

I'm planning to achieve a functional, useable and maintainable system even though it has limited features. Through developing this project, I can use my skills in programming to create an automated solution to an existing problem that most of the companies and other establishment experience. Aside from learning the basic flow of fixed asset management, I can also learn how to implement the solution formulated, by developing it using Java programming language with MySQL database. The main goal is to learn how to analyse the problem, construct a solution and implement it.

Scope

Features:

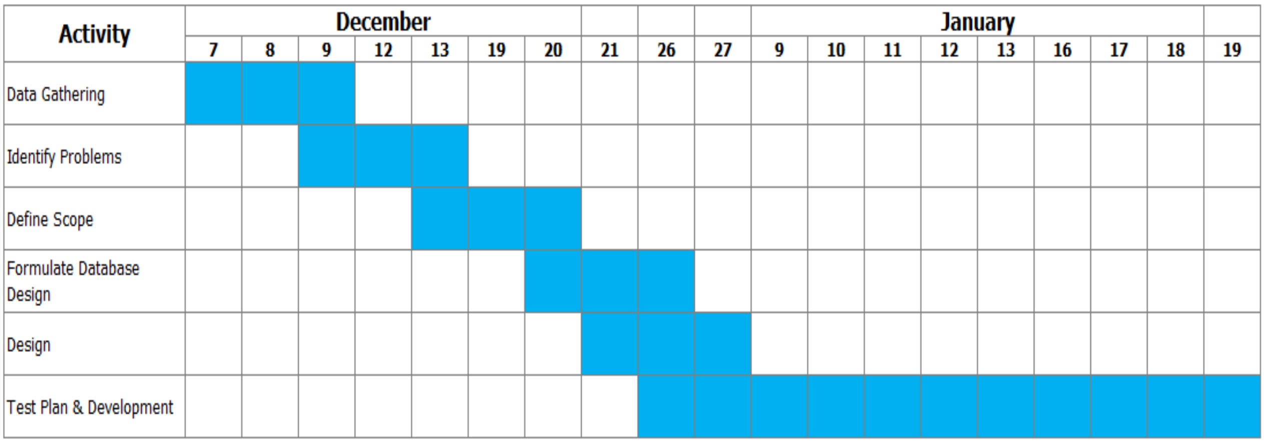
1. The classifications used are computer equipment and software only.
2. Depreciation computation used was the straight line formula.
3. Supplier Information only, no linked information to product supplied.
4. Set and update location, personnel-in-charge and status using transfer and maintenance
5. Single fixed asset entry in transfer, maintenance and pull-out transactions.

Further Improvements:

1. Include all the classifications
2. Restrictions and filtering.
3. Multiple user interaction for transaction verifications.
4. Multiple fixed asset entry in transfer, maintenance and pull-out transactions
5. Additional detailed reports
6. Image uploading

Project Planning and Execution

Project Plan and Gantt Chart



Activity	December										January								
	7	8	9	12	13	19	20	21	26	27	9	10	11	12	13	16	17	18	19
Data Gathering	■	■	■																
Identify Problems			■	■	■														
Define Scope					■	■	■												
Formulate Database Design							■	■	■										
Design								■	■	■									
Test Plan & Development									■	■	■	■	■	■	■	■	■	■	■

The application followed a sequence of activities to accomplish certain tasks. At first, I already decided to create a fixed asset management application, I gathered data about the fixed asset management system and identify some issues and problem that companies or establishments encounter with regards to this system. After identifying the issues and problems, I defined the scope and limitations of the application and then formulated the database design for the system. The last phase or activity I made was the test plan and development. I created test cases first to make sure that the functions works well before implementing it to the actual User Interface.

Risk Management

1. Classification of fixed assets is very complex for the project so I used only two fixed asset classifications namely computer equipment and software. With the two classification I can easily structure and manage the data I need.
2. Fixed asset management software is a complex system that needs verifications from multiple users. I reduced the features of the system into a brief and straightforward transactions.
3. Database structure is also reduced into a brief and concise manner.

Architecture and Design

Application structure

Describe what you learned about how to structure an application, for example, using packages and classes. Did you consider and/or use any design patterns.

I learned the importance creating packages in categorising classes based on its use or function, creating constructors to communicate to other classes for data. I considered Factory design pattern for the interaction of my objects and methods. With this design, I am able to create specific java class that contains the centralized methods with parameters. The methods are accessible to other java classes located in different package to execute specific functions.

Discussion of technologies used

- **Eclipse** – This IDE (Integrated Development Environment) is free and I was able to adapt really well and I am comfortable using it. I tried using IntelliJ IDEA but discontinued it because I'm still not familiar to the environment.
- **MySQL** – This is the database management system that I have been using since then and I want to use this using Java programming language. I consider SQLite but switched back to MySQL because it is more easier for me to manage it.
- **Xampp** – I use Xampp for my local server. I use xampp for portability and it is a cross-platform server.
- **HeidiSQL** – This database manager software is light, easy and flexible to use.

UI Design

Initial screen designs showing layout and user interaction

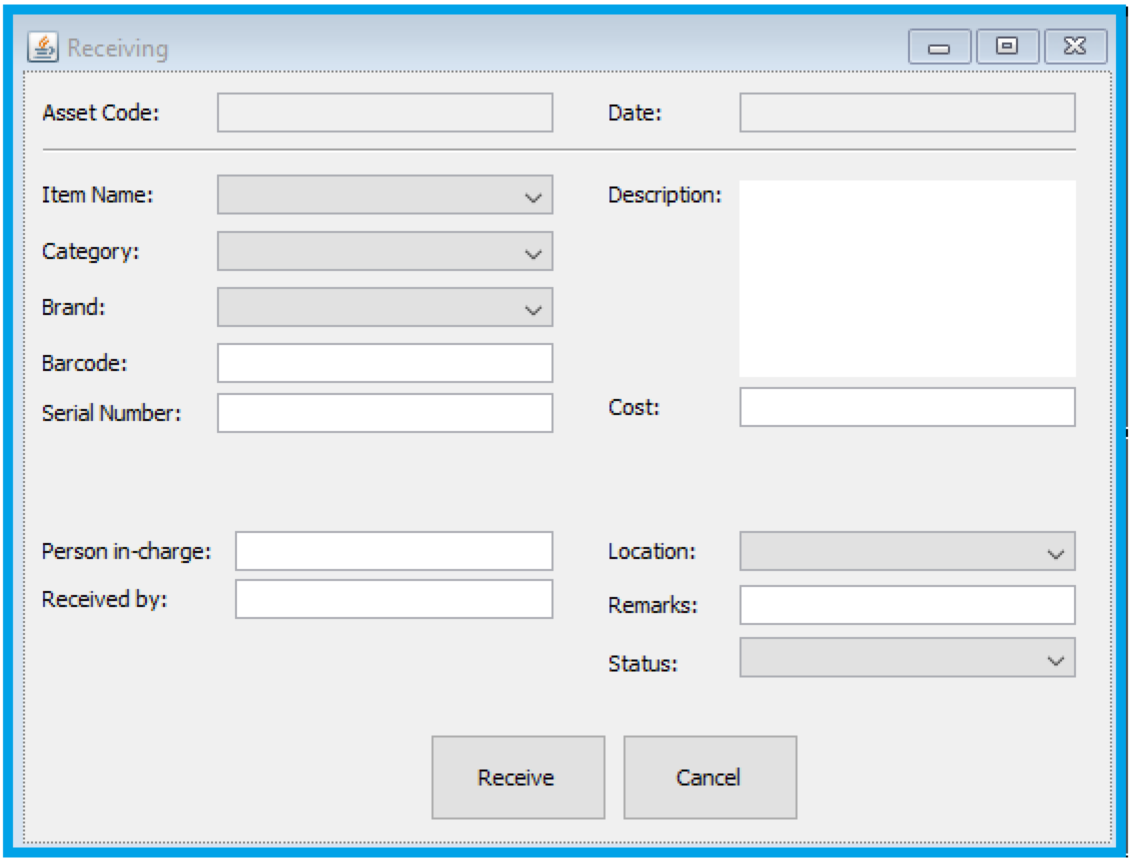


Figure 1. Receving Transaction

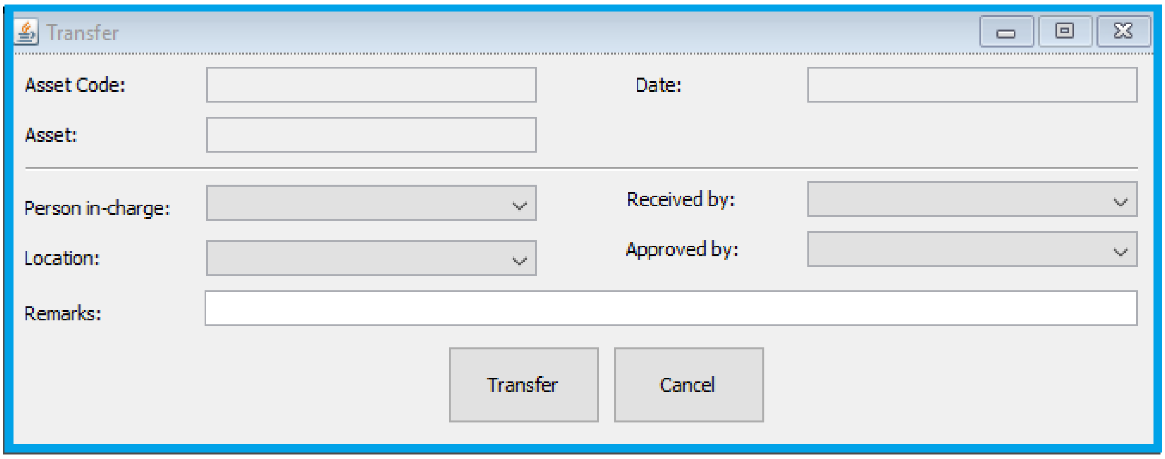


Figure 2. Transfer Transaction

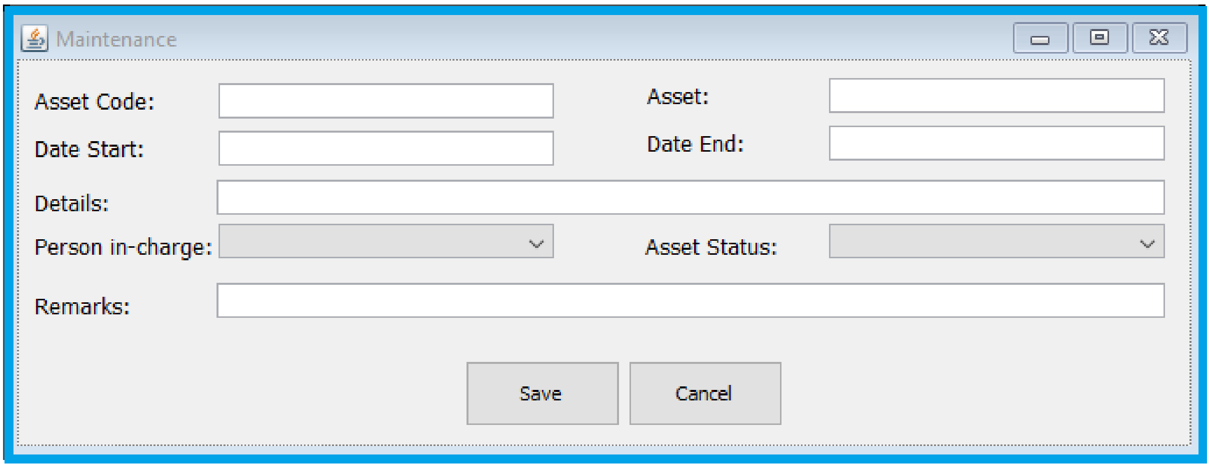


Figure 3. Maintenance Transaction

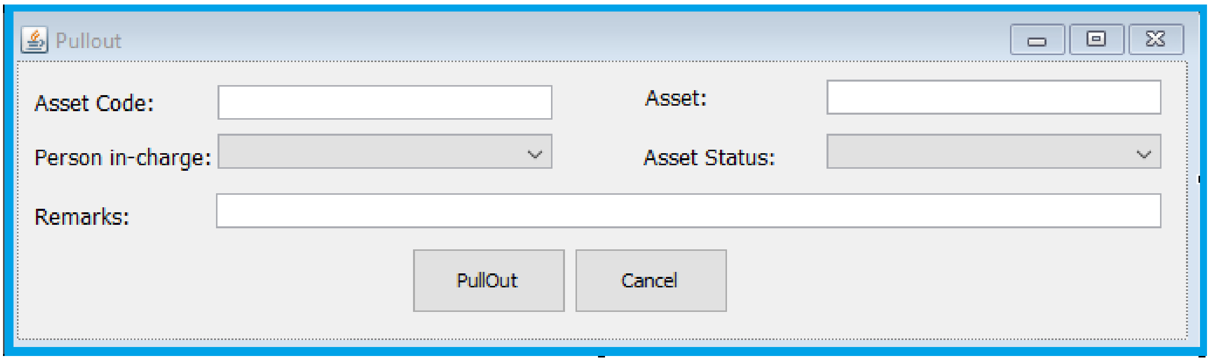
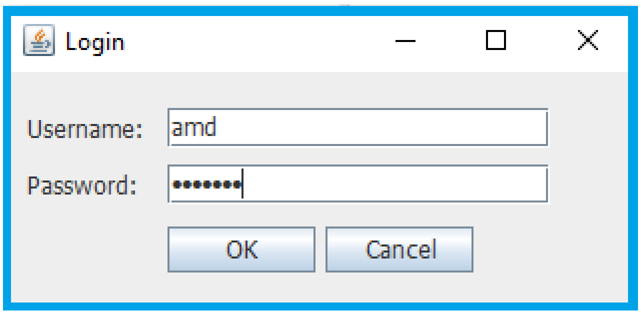


Figure 4. Pull-out Transaction

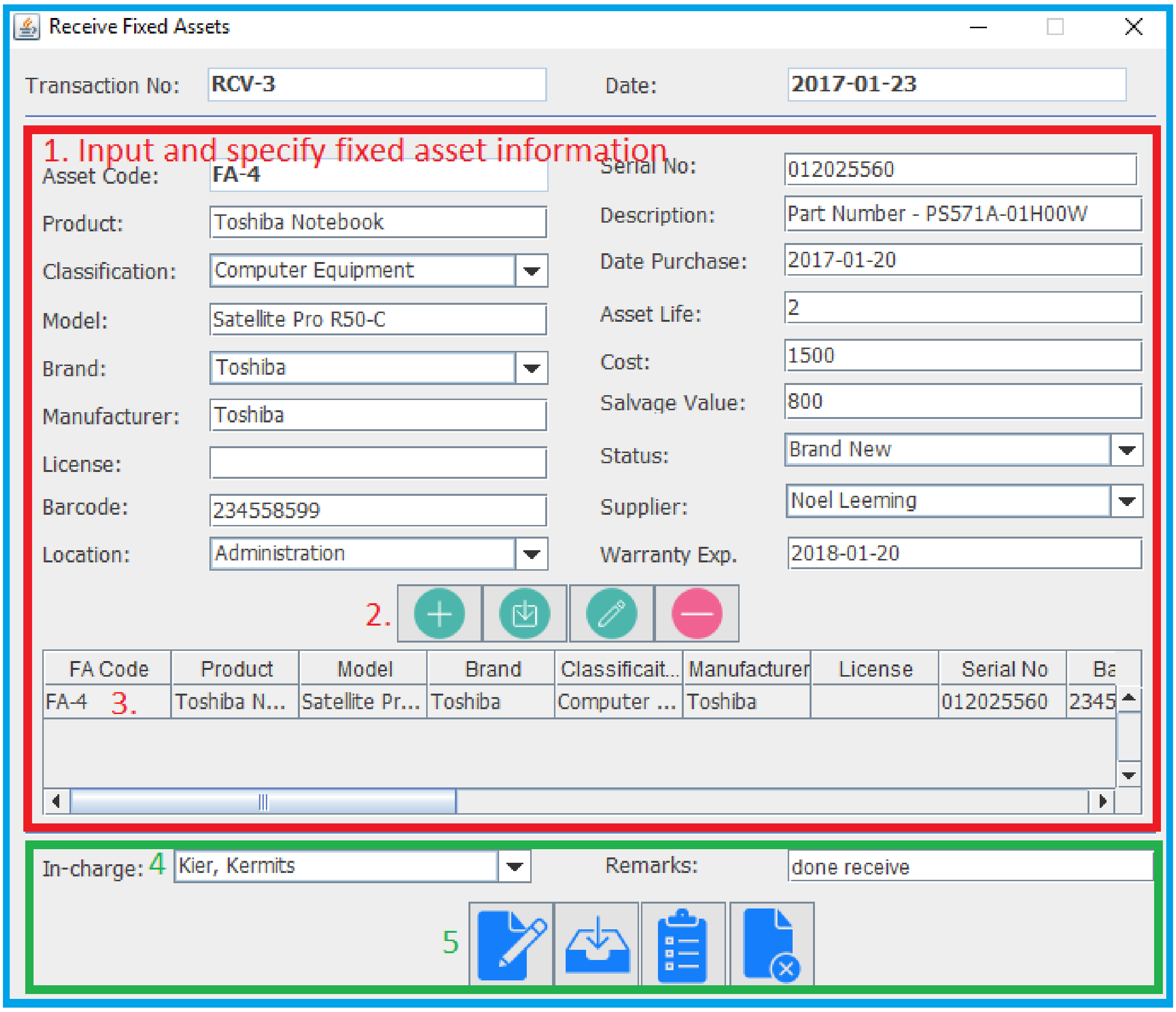
Final designs



Username: amd

Password: •••••••

Figure 1. Login Form



Receive Fixed Assets

Transaction No: RCV-3
Date: 2017-01-23

Asset Code: FA-4
Product: Toshiba Notebook
Classification: Computer Equipment
Model: Satellite Pro R50-C
Brand: Toshiba
Manufacturer: Toshiba
License:
Barcode: 234558599
Location: Administration

Serial No: 012025560
Description: Part Number - PS571A-01H00W
Date Purchase: 2017-01-20
Asset Life: 2
Cost: 1500
Salvage Value: 800
Status: Brand New
Supplier: Noel Leeming
Warranty Exp. 2018-01-20

FA Code	Product	Model	Brand	Classificait...	Manufacturer	License	Serial No	Ba
FA-4	Toshiba N...	Satellite Pr...	Toshiba	Computer ...	Toshiba		012025560	2345

In-charge: Kier, Kermits
Remarks: done receive

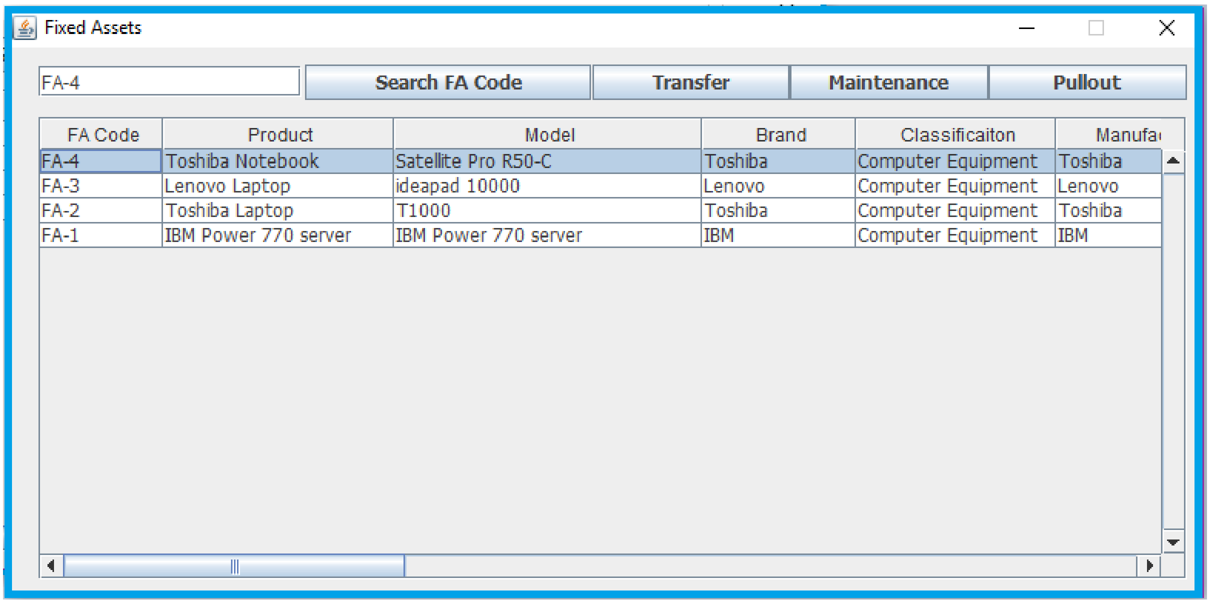
Figure 2. Receiving Transaction

Access the form

To access receiving transaction, login first to the system: username: *amd* password: *passamd*. Then, go *Transaction* menu > select *Receiving* > *Receive Fixed Asset* form will appear

Create Transaction

In this form, multiple fixed asset can be added in a single receiving transaction. In Figure 1.0 with the red highlight, in this section the user can add, edit and delete fixed asset information. Step 1, input and specify all the necessary information except for asset code. Then in step 2, to add fixed asset, click the save button ⤓, then to edit added fixed asset, select the information first in step 3, back to step 1 and click the edit button ✎, lastly to delete the fixed asset, select the information in step 3 and click the delete button ⊖. To receive the fixed asset, specify the other information needed and click the receive button . To view the added information, click the view button . To cancel information, click the button .

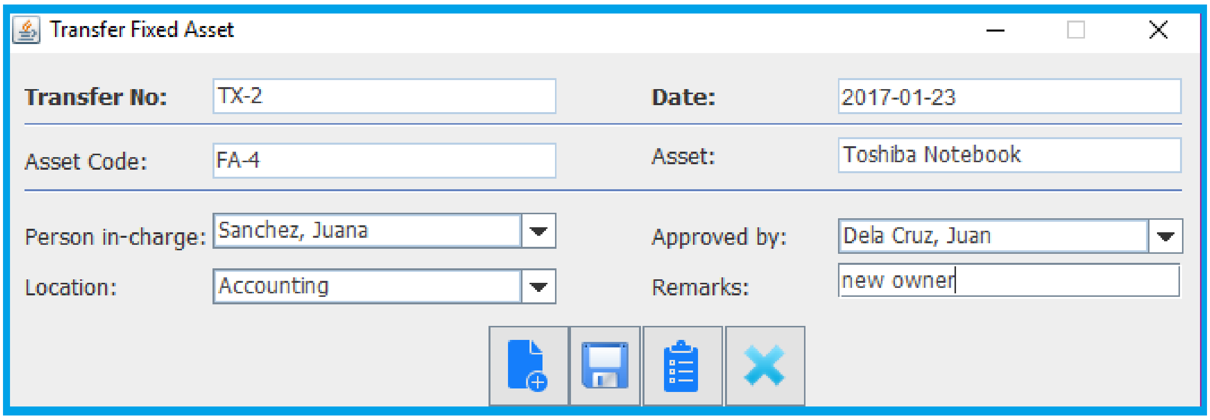


FA-4

FA Code	Product	Model	Brand	Classificaiton	Manufa
FA-4	Toshiba Notebook	Satellite Pro R50-C	Toshiba	Computer Equipment	Toshiba
FA-3	Lenovo Laptop	ideapad 10000	Lenovo	Computer Equipment	Lenovo
FA-2	Toshiba Laptop	T1000	Toshiba	Computer Equipment	Toshiba
FA-1	IBM Power 770 server	IBM Power 770 server	IBM	Computer Equipment	IBM

Figure 3. View Fixed Assets

This form show the received information of the fixed assets.



Transfer No: TX-2

Date: 2017-01-23

Asset Code: FA-4

Asset: Toshiba Notebook

Person in-charge: Sanchez, Juana

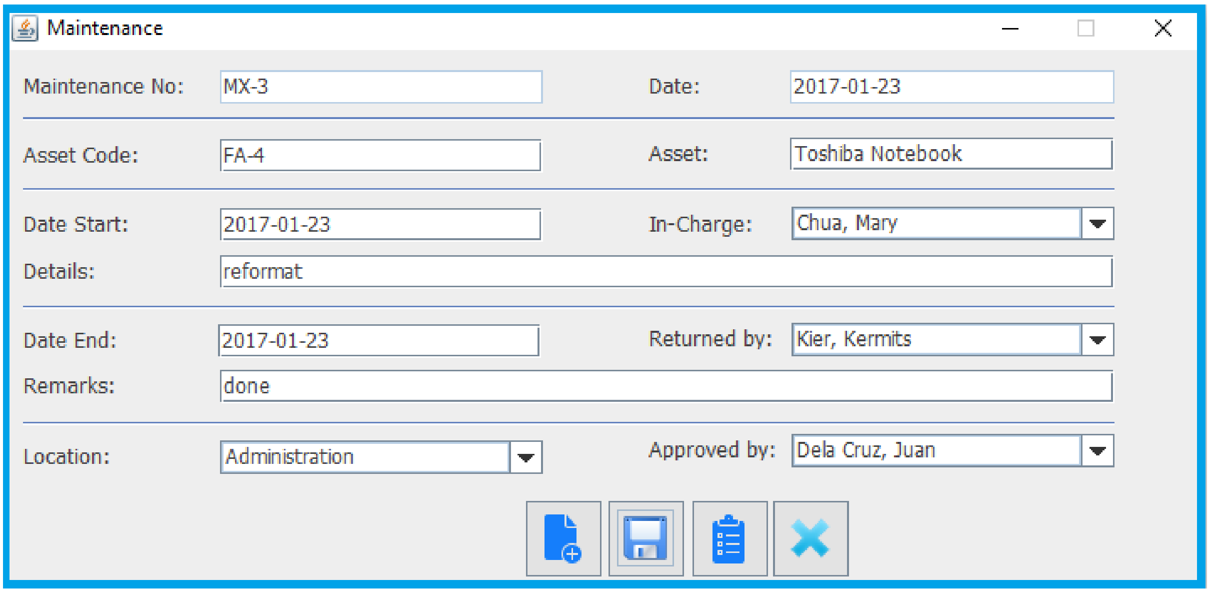
Approved by: Dela Cruz, Juan

Location: Accounting

Remarks: new owner

Figure 4. Transfer Form

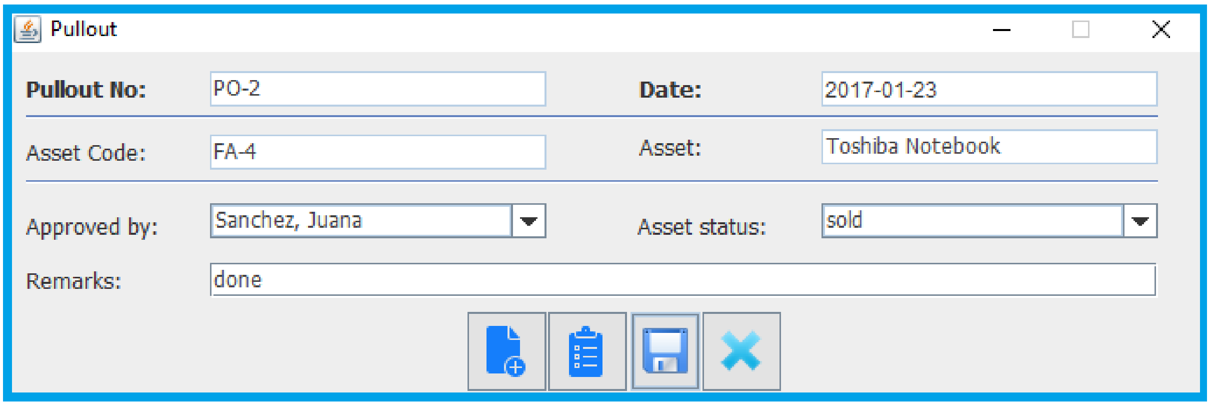
The figures shows the form for transferring the fixed asset to another location. The user need to click the new button to generate the new transfer number of the transaction. Then, click save button to save the transfer information, the user can then view the update of the fixed asset. Lastly, the user can cancel transaction.


Maintenance

Maintenance No:	MX-3	Date:	2017-01-23
Asset Code:	FA-4	Asset:	Toshiba Notebook
Date Start:	2017-01-23	In-Charge:	Chua, Mary
Details:	reformat		
Date End:	2017-01-23	Returned by:	Kier, Kermits
Remarks:	done		
Location:	Administration	Approved by:	Dela Cruz, Juan

Figure 5. Maintenance Form

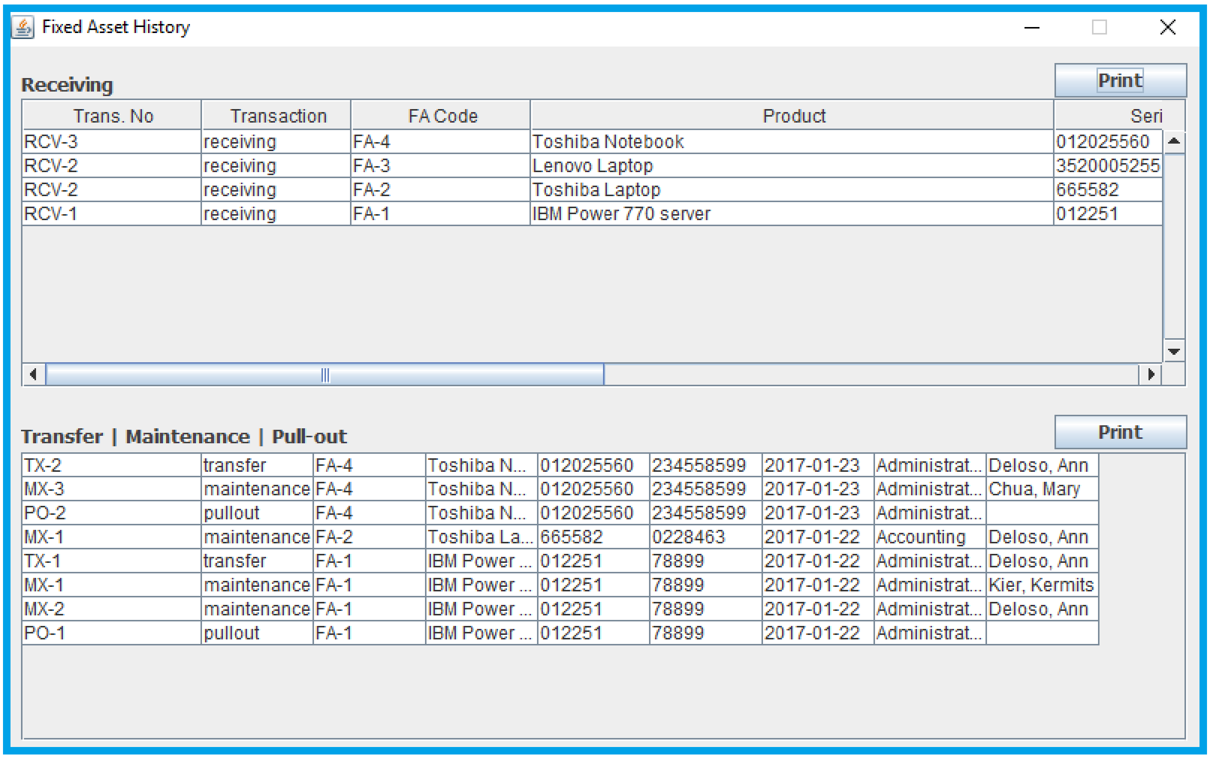
The figures shows the form for maintenance information the fixed asset. The user need to click the new button to generate the new maintenance number of the transaction. Then, input and specify all the necessary information, and click save button to save the maintenance information, the user can then view the update of the fixed asset. Lastly, the user can cancel transaction.


Pullout

Pullout No:	PO-2	**Date:**	2017-01-23
Asset Code:	FA-4	Asset:	Toshiba Notebook
Approved by:	Sanchez, Juana	Asset status:	sold
Remarks:	done		

Figure 6. Pull-out Form

The figure shows pull-out form for the out-going of the fixed asset into the company or establishment. The fixed asset added in this transaction is already disposed or sold by the company or establishment. The user need to click the new button to generate the new pull-out number of the transaction. Then, click save button to save the pull-out information, the user can then view the update of the fixed asset. Lastly, the user can cancel transaction.



Receiving

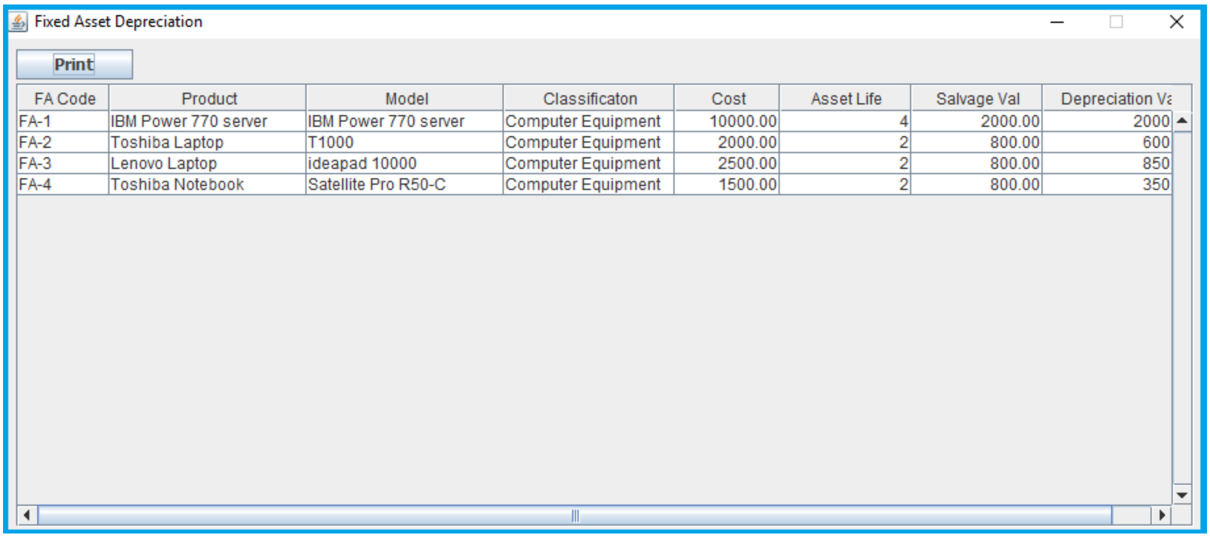
Trans. No	Transaction	FA Code	Product	Seri
RCV-3	receiving	FA-4	Toshiba Notebook	012025560
RCV-2	receiving	FA-3	Lenovo Laptop	3520005255
RCV-2	receiving	FA-2	Toshiba Laptop	665582
RCV-1	receiving	FA-1	IBM Power 770 server	012251

Transfer | Maintenance | Pull-out

TX-2	transfer	FA-4	Toshiba N...	012025560	234558599	2017-01-23	Administrat...	Deloso, Ann
MX-3	maintenance	FA-4	Toshiba N...	012025560	234558599	2017-01-23	Administrat...	Chua, Mary
PO-2	pullout	FA-4	Toshiba N...	012025560	234558599	2017-01-23	Administrat...	
MX-1	maintenance	FA-2	Toshiba La...	665582	0228463	2017-01-22	Accounting	Deloso, Ann
TX-1	transfer	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	Deloso, Ann
MX-1	maintenance	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	Kier, Kermits
MX-2	maintenance	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	Deloso, Ann
PO-1	pullout	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	

Figure 7. Transaction History View

This form shows the summary transaction history report of the fixed asset.



FA Code	Product	Model	Classificaton	Cost	Asset Life	Salvage Val	Depreciation Va
FA-1	IBM Power 770 server	IBM Power 770 server	Computer Equipment	10000.00	4	2000.00	2000
FA-2	Toshiba Laptop	T1000	Computer Equipment	2000.00	2	800.00	600
FA-3	Lenovo Laptop	ideapad 10000	Computer Equipment	2500.00	2	800.00	850
FA-4	Toshiba Notebook	Satellite Pro R50-C	Computer Equipment	1500.00	2	800.00	350

Figure 8. Fixed Asset Yearly Depreciation Value

This form shows the calculated yearly depreciation value of the fixed asset. It shows some of the important information of the fixed asset and then depreciation value (which uses the straight line depreciation calculation).

Reasons for your final choices

The design was created in more user friendly, and presentable manner. Users can easily familiarize the fields and buttons in the form. In *receiving* transaction, the design is to cater the multiple fixed asset entry. The user can add one or more fixed asset information, the user ca also edit the information added or delete.

T*ransfer, maintenance* and *pull-out* transaction on the other hand, allow one fixed asset only in every transaction. The reports is simplified to highlight the important fixed asset transaction information and show the yearly depreciation value of the fixed asset.

The transaction of the fixed asset management system was simplified and follows straightforward transaction flow. In this method, the concept of data relationships and integrity is considered in minimal yet functional manner.

Implementation

Database schema

Show the database schema used

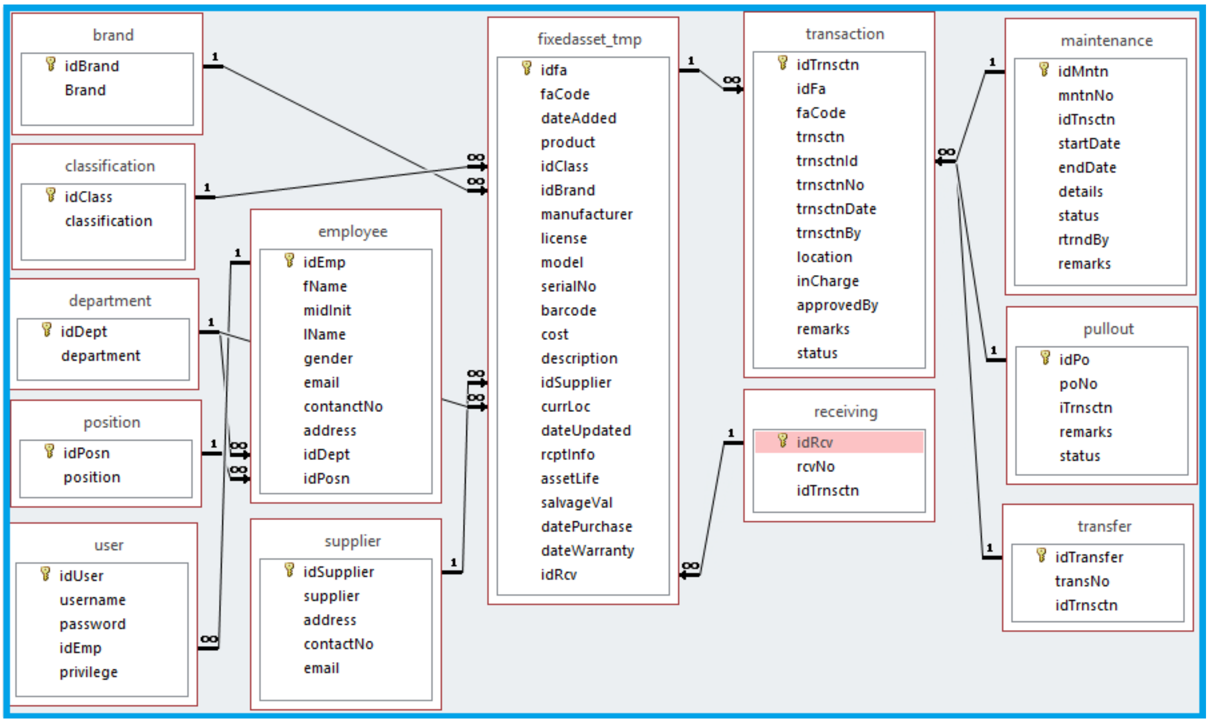


Figure 1. Database *dbfams*

The *fixedasset_tmp* is the table used to store fixed asset informations, the *fixedasset_tmp* contains the *brand*, *classification*, *location(department)*, and *supplier* information and it also connects to *receiving* table. Every fixed asset information has the *receiving* unique identificatio. The *transaction* table stores all the transaction information of the fixed asset and is connected to tables *maintenance*, *transfer* and *pull-out*.

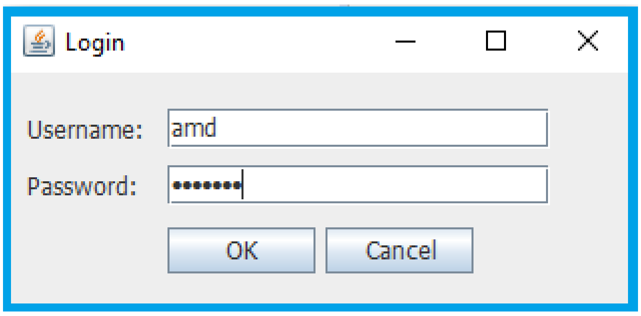
Technologies

Details of programming languages, IDEs, database technologies, etc. that are used and reasons for your choice.

- **JAVA** – I chose this programming language to learn and explore the language and be able to adapt and implement the ideas formulated to create an application.
- **Eclipse** – I use this IDE (Integrated Development Environment) because I was able to explore it and I'm familiar to the tools and environment.
- **MySQL** – I chose this database technology because I want to continue learning it and incorporate it into JAVA programming language.
- **XAMPP** – I have been using this for years and I am most comfortable working on for my Local Server.
- **Heidi SQL** – This database manager software is very handy and I love the portability and efficiency of this software.

Screen shots

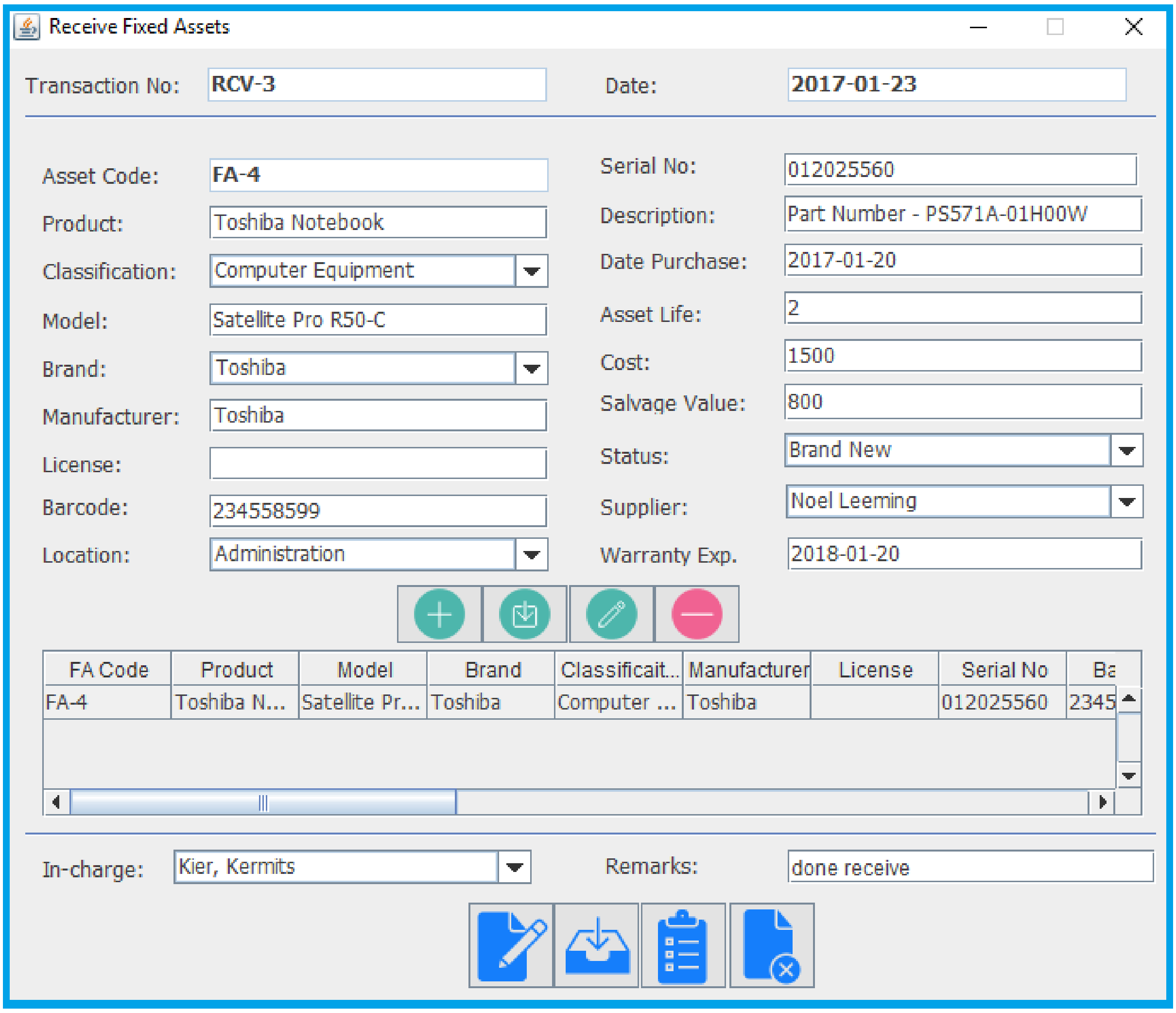
Screen shots of working app



Username: amd

Password: •••••••

Figure 1. Login Form



Transaction No: **RCV-3**

Date: **2017-01-23**

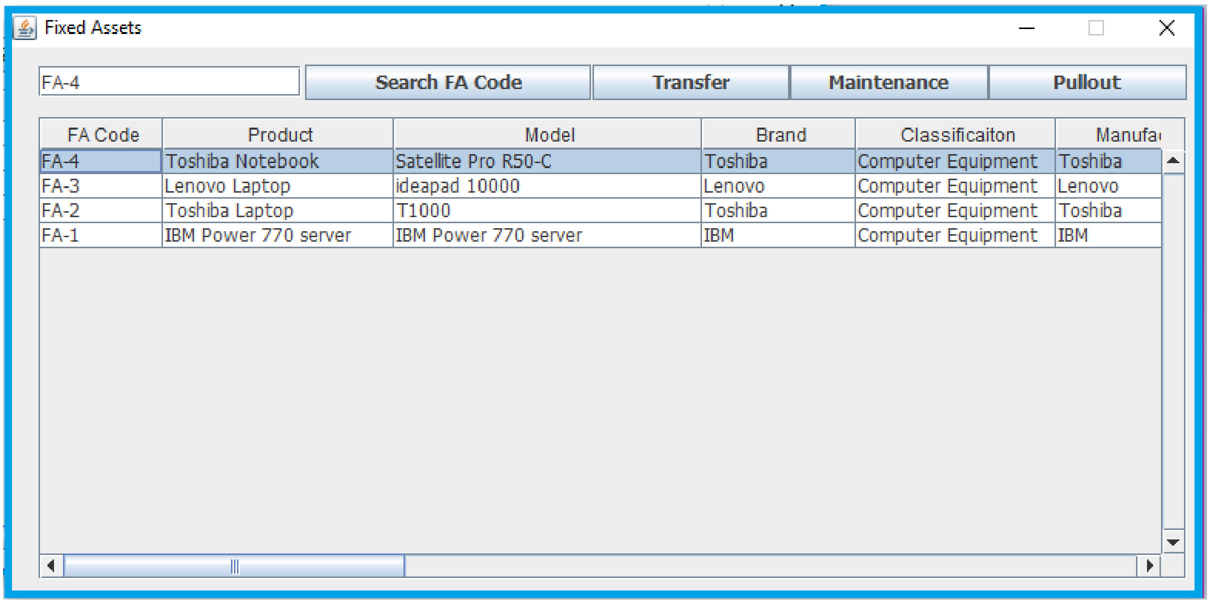
Asset Code:	**FA-4**
Product:	Toshiba Notebook
Classification:	Computer Equipment
Model:	Satellite Pro R50-C
Brand:	Toshiba
Manufacturer:	Toshiba
License:	
Barcode:	234558599
Location:	Administration
Serial No:	012025560
Description:	Part Number - PS571A-01H00W
Date Purchase:	2017-01-20
Asset Life:	2
Cost:	1500
Salvage Value:	800
Status:	Brand New
Supplier:	Noel Leeming
Warranty Exp.	2018-01-20

FA Code	Product	Model	Brand	Classificait...	Manufacturer	License	Serial No	Ba
FA-4	Toshiba N...	Satellite Pr...	Toshiba	Computer ...	Toshiba		012025560	2345

In-charge: Kier, Kermits

Remarks: done receive

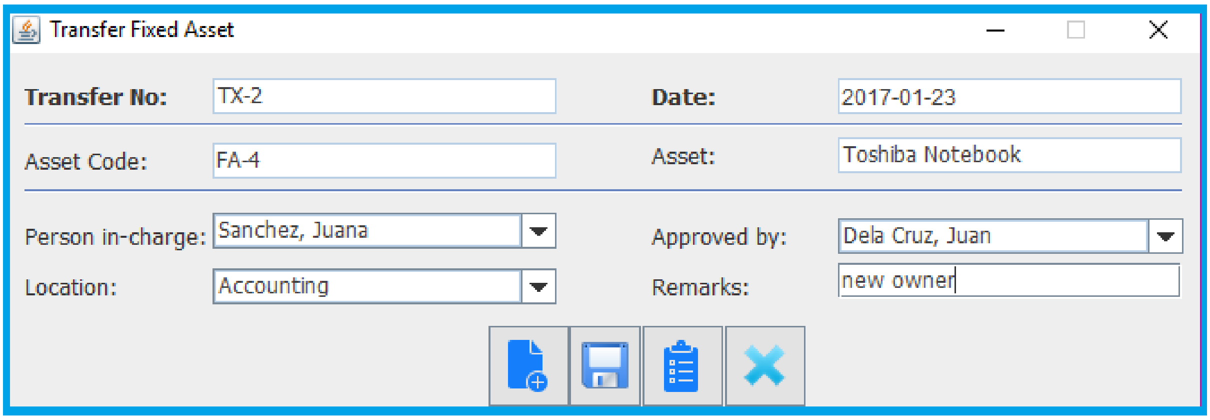
Figure 2. Receiving Transaction



FA-4

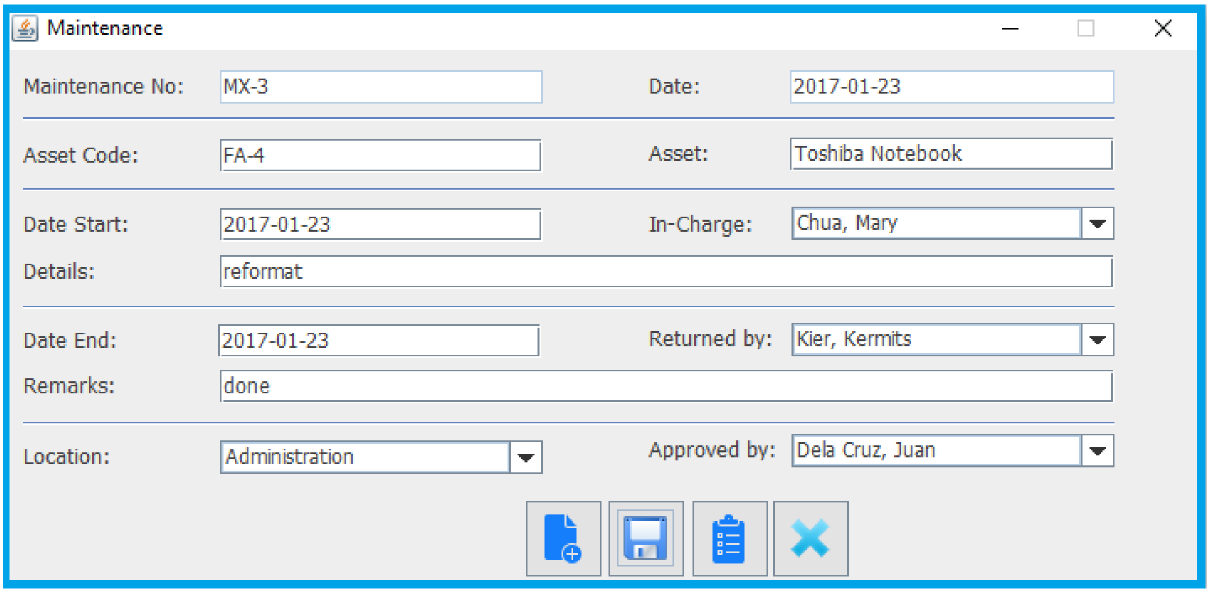
FA Code	Product	Model	Brand	Classificaiton	Manufa
FA-4	Toshiba Notebook	Satellite Pro R50-C	Toshiba	Computer Equipment	Toshiba
FA-3	Lenovo Laptop	ideapad 10000	Lenovo	Computer Equipment	Lenovo
FA-2	Toshiba Laptop	T1000	Toshiba	Computer Equipment	Toshiba
FA-1	IBM Power 770 server	IBM Power 770 server	IBM	Computer Equipment	IBM

Figure 3. View Fixed Assets



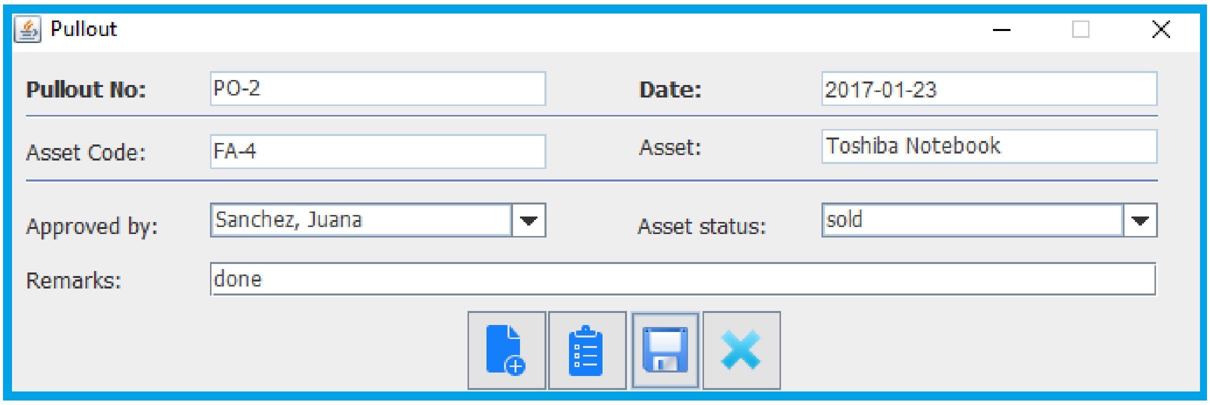
Transfer No:	TX-2	**Date:**	2017-01-23
Asset Code:	FA-4	Asset:	Toshiba Notebook
Person in-charge:	Sanchez, Juana	Approved by:	Dela Cruz, Juan
Location:	Accounting	Remarks:	new owner

Figure 4. Transfer Form



Maintenance No:	MX-3	Date:	2017-01-23
Asset Code:	FA-4	Asset:	Toshiba Notebook
Date Start:	2017-01-23	In-Charge:	Chua, Mary
Details:	reformat		
Date End:	2017-01-23	Returned by:	Kier, Kermits
Remarks:	done		
Location:	Administration	Approved by:	Dela Cruz, Juan

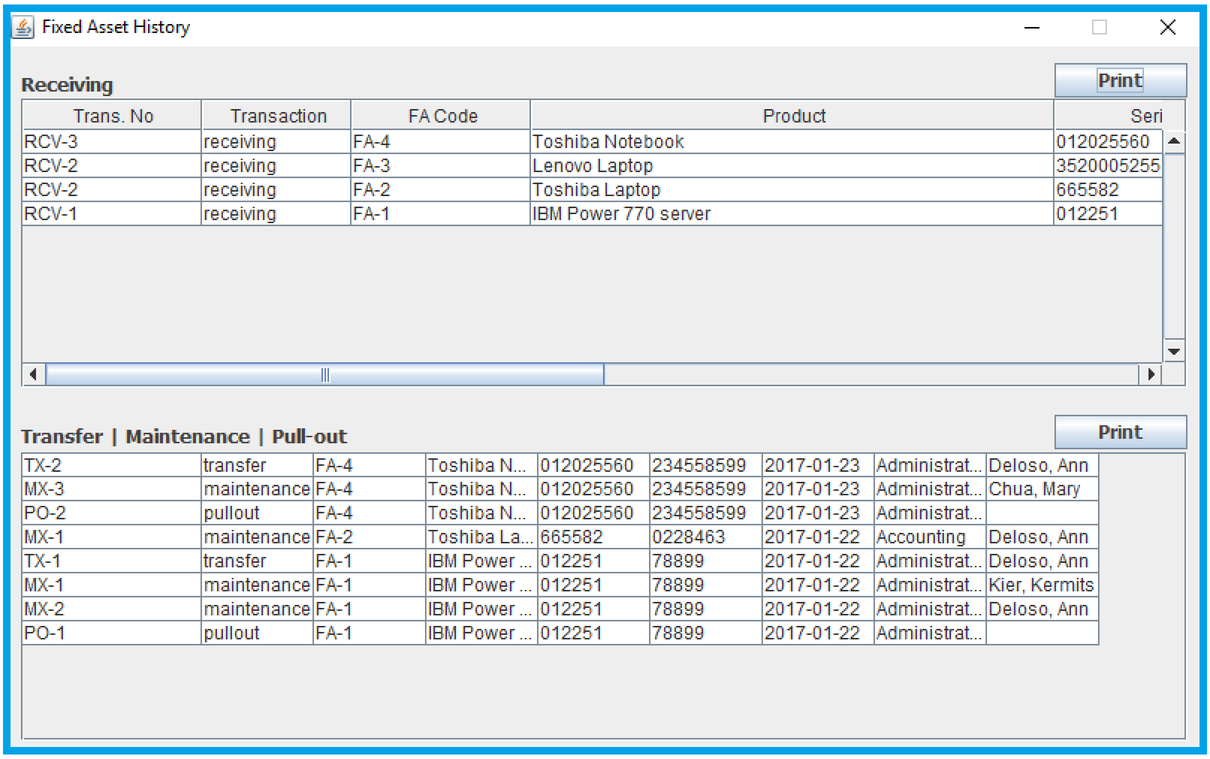
Figure 5. Maintenance Form



Pullout

Pullout No:	PO-2	**Date:**	2017-01-23
Asset Code:	FA-4	Asset:	Toshiba Notebook
Approved by:	Sanchez, Juana	Asset status:	sold
Remarks:	done		

Figure 6. Pullout Form



Fixed Asset History

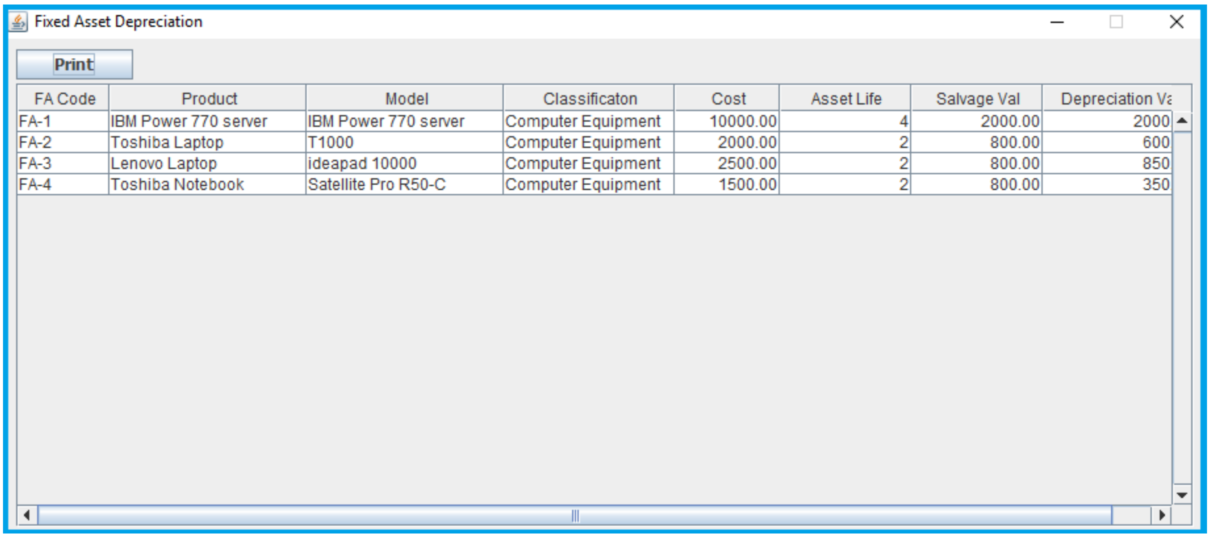
Receiving

Trans. No	Transaction	FA Code	Product	Seri
RCV-3	receiving	FA-4	Toshiba Notebook	012025560
RCV-2	receiving	FA-3	Lenovo Laptop	3520005255
RCV-2	receiving	FA-2	Toshiba Laptop	665582
RCV-1	receiving	FA-1	IBM Power 770 server	012251

Transfer | Maintenance | Pull-out

TX-2	transfer	FA-4	Toshiba N...	012025560	234558599	2017-01-23	Administrat...	Deloso, Ann
MX-3	maintenance	FA-4	Toshiba N...	012025560	234558599	2017-01-23	Administrat...	Chua, Mary
PO-2	pullout	FA-4	Toshiba N...	012025560	234558599	2017-01-23	Administrat...	
MX-1	maintenance	FA-2	Toshiba La...	665582	0228463	2017-01-22	Accounting	Deloso, Ann
TX-1	transfer	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	Deloso, Ann
MX-1	maintenance	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	Kier, Kermits
MX-2	maintenance	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	Deloso, Ann
PO-1	pullout	FA-1	IBM Power ...	012251	78899	2017-01-22	Administrat...	

Figure 7. Transaction History View



FA Code	Product	Model	Classificaton	Cost	Asset Life	Salvage Val	Depreciation Va
FA-1	IBM Power 770 server	IBM Power 770 server	Computer Equipment	10000.00	4	2000.00	2000
FA-2	Toshiba Laptop	T1000	Computer Equipment	2000.00	2	800.00	600
FA-3	Lenovo Laptop	ideapad 10000	Computer Equipment	2500.00	2	800.00	850
FA-4	Toshiba Notebook	Satellite Pro R50-C	Computer Equipment	1500.00	2	800.00	350

Figure 8. Fixed Asset Yearly Depreciation Value

Test Driven Development

Test planning

Transaction Menu (**TestFcnTransMenu)**

Receiving		
Test Cases Methods	**Scenario**	**Expected Result**
testAddAssetInfoToDb	Allow new fixed asset information to be added into *fixedasset_tmp* table.	Successful
testEditExistAssetInfoToDb	Allow edit of an existing fixed asset information in *fixedasset_tmp* table	Successful
testDeleteExistAssetInfoToDb	Allow removing of an existing fixed asset information from *fixedasset_tmp* table	Successful
testAddRcvngInfotoEmptyTbl	Allow new transaction information to be added into an empty *receiving* table.	Successful
testAddFullRcvngInfoToDb	Allow all transaction information to be added into *receiving* table.	Successful
testRestrictAddFullRcvngInfoToDb	Restricted to add new information because of the specified number of data allowed into *receiving* table.	Restriction successful.
testRestrictAddExistRcvngInfoToDb	Restricted to add existing *mntnNo* into maintenance table.	Restriction successful.
testCancelRcvTrans	Allow cancellation of the receiving transaction. Delete all transaction information added in *fixedasset_tmp*, *receiving* and *transaction* tables.	Successful
Transfer		
testAddTransferInfotoEmptyTbl	Allow new transaction information to be added into an empty *transfer* table.	Successful
testAddLastTransferInfoToDb	Allow all transaction information to be added into *transfer* table.	Successful
testAddFullTransferInfoToDb	Allow all transfer information to be added into *transfer* table.	Successful
testRestrictAddFullTransferInfoToDb	Restricted to add new information because of the specified number of data	Failed to add new information.

	allowed into *transfer* table.	
testRestrictAddExistTransferInfoToDb	Restricted to add existing *mntnNo* into maintenance table.	Restriction successful.
testCancelTransferTrans	Allow cancellation of the transfer transaction. Delete all transaction information added in *transfer* and *transaction* tables.	Successful
Maintenance		
testAddMntnInfotoEmptyTbl	Allow new transaction information to be added into an empty *maintenance* table.	Successful
testAddLastMntnInfoToDb	Allow last transaction information to be added into *maintenance* table.	Successful
testAddFullMntnInfoToDb	Allow all maintenance information to be added into maintenance table.	Successful
testRestrictAddFullMntnInfoToDb	Restricted to add new information because of the specified number of data allowed into *maintenance* table.	Failed to add new information.
testRestrictAddExistMntnInfoToDb	Restricted to add existing *mntnNo* into *maintenance* table.	Restriction successful.
testCancelMntnTrans	Allow cancellation of the maintenance transaction. Delete all transaction information added in *maintenance* and *transaction* tables.	Successful
Pullout		
testAddPoInfotoEmptyTbl	Allow new transaction information to be added into an empty *pullout* table.	Successful
testAddLastMntnInfoToDb	Allow last transaction information to be added into *pullout* table.	Successful
testAddFullPoInfoToDb	Allow the all pullout information to be added into *pullout* table.	Successful
testRestrictAddFullPoInfoToDb	Restricted to add new information because of the specified number of data allowed into *maintenance* table.	Failed to add new information.
testRestrictAddExistPoInfoToDb	Restricted to add existing *poNo* into *pullout* table.	Restriction successful.

`testCancelPoTrans`	Allow cancellation of the pullout transaction. Delete all transaction information added in *pullout* and *transaction* tables.	Successful
Other Methods		
testGetMaxIDIfNull	Test if *maximum id* can be generated with an empty table. Applied to all transactions.	Successful, null id will be incremented by 1.
testGetTransNoIfNull	Test if transaction number can be generated with an empty table. Applied to all transactions.	Successful, null id will be incremented by 1.

Test results Success adding new information into the table.

- Success adding full information into the table.
- Success in restricting the addition of new data into the specified table.
- Success in restricting the addition of new data into full table in database.
- Success in restricting existed data to be added into the specified table.
- Success in getting value in a null table
- Success in deleting information from the table.
- Success in cancellation of transaction.
- The system cannot do multiple entries in Transfer, Maintenance and Pull-out transactions.
- The system do not have enough filtering and restrictions for data entry

Reflection

I have learned the importance of functionality testing of the code before implementing it to the User Interface. Test driven development helped me locate bugs in my code and I was able to refactor and improve my code to make it more useable and functional. Test driven development improves the time to test the functionality of the system.

Fixed asset management system has complex transactions and the functionality of the system needs to be tested in every scenario to ensure usability and flexibility. The application design also needs to be reconsidered as it lacks of the full accessibility.

Test driven development is a significant part of application's development because it gives an opportunity to save more time further as soon as the testing is done at the first stage. It also assists to find bugs, refactor and reconsider its functionality or the system's requirements.

Conclusion

Reflection

Creating software on a specified and minimal way with test driven development is very important in developing a usable, functional, flexible and maintainable system. With the help of test driven development, system development prevents it from heavy backlogs. The first phase of the development lifecycle is very crucial in formulating the design of the project. The developer must consider all the possible scenarios that the project requires. The system must match the requirements specified.

Learning new thing is fun but first, it needs to be beneficial in a way that it does not slows down the progress of the application development. The first main problem I encountered was, I tried using a new IDE (IntelliJ) and backend (SQLite) in the middle of my project development and I encountered problems that resulted to the delays. I switched back to Eclipse and was able to recover smoothly. With time constraints together with limited knowledge into other aspects of Java programming and Eclipse, I reduced the features of the system.

For the further improvements of the system, the transaction forms needs to be interactive. The system needs more filtering and restrictions in some areas. Multiple users at the same transaction needs to be considered. Notification for pending transaction, filtering of personnel from their designated position and multiple entries of fixed asset in a single transaction in Transfer, Maintenance and Pull-out Form. Lastly, ensure the system flow match to the business flow

Overall, I learned how to create an application using the Java programming language with MySQL database. I was able to explore different design patterns and other methodologies. Also, the vital part of the system development which is test driven development helped me a lot to create a functional, usable, flexible and maintainable functions to implement in the system.

Summary

Fixed asset management system is a monitoring of the establishments assets. This is often disregarded in most of the business establishments which can create a negative impact on the business when not given attention at an early stage. Fixed Asset Management helps the personnel to trace transaction history, location, personnel liable, and status and depreciation value of the fixed asset. The fixed asset management application caters straightforward transaction flow from receiving, transfer, maintenance and pull-out of the assets. The application can provide summary report of the transaction history and the depreciation value of the fixed assets.

The developer did not include the complex classification of the fixed asset to avoid confusion and to focus on the basic concept of the system. Verifications from other users are not included. The modification of the fixed asset was not considered and the database structure was reduced as well.

The developer will consider more of the design patterns, methodologies and test driven development in the future.

References

Gartner.(March 2008).**Top Tips For Fixed Asset Management.November19, 2010.**http://www.assetware.co.uk/top-tips-for-fixed-asset-management.htm

Harlick.(March 2008). **Top Tips For Fixed Asset Management.November19,2010.**http://www.prlog.org/10058313-top-tips-for-fixed-asset-management-assetware-offers-free-guide-in-response-to-research-findings.html

Fixed Asset. 2016. http://www.accountingtools.com/definition-fixed-asset

Factory Design Pattern. 2017**.** https://www.tutorialspoint.com/design_pattern/factory_pattern.htm

Straight Line Depreciation. 2016. http://www.accountingtools.com/straight-line-depreciation

What is the proper classification of fixed assets?. 2016.http://www.accountingtools.com/questions-and-answers/what-is-the-proper-classification-of-fixed-assets.html

Introduction to Test Driven Development (TDD). Ambysoft Inc.2002-2013. http://agiledata.org/essays/tdd.html