STASH - An Inventory Management System

Mariom Akter Misu
Id: 012163014

A Project

In

The Department

Of

Computer Science and Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

November, 2019

© Mariom Akter Misu, 2019

`

Approval Certificate

This project titled "**STASH- An Inventory Management System**” submitted by **Mariom Akter Misu**, Student ID: **012163014**, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on November, 2019.

Board of Examiners

1.

________________________________ Supervisor

Dr. Mohammad Nurul Huda
Professor & Director-MSCSE
Department of Computer Science & Engineering (CSE)
United International University (UIU)
Dhaka-1212, Bangladesh

2.

________________________________ Examiner

Rubaiya Rahtin khan
Assistant Professor
Department of Computer Science & Engineering (CSE)
United International University (UIU)
Dhaka-1212, Bangladesh

3.

________________________________ Ex-Officio

Dr. Mohammad Nurul Huda
Professor & Director-MSCSE
Department of Computer Science & Engineering (CSE)
United International University (UIU)
Dhaka-1212, Bangladesh

`

Declaration

This is to certify that the work entitled “**STASH- Inventory Management System**" is the outcome of the project carried out by me under the supervision of Dr. Mohammad Nurul Huda, Professor and Director - MSCSE, United International University (UIU), Dhaka, Bangladesh.

Mariom Akter Misu
Department of Computer Science and Engineering
MSCSE Program
Student ID: 012163014
United International University (UIU)
Dhaka-1212, Bangladesh.

In my capacity as supervisor of the candidate’s project, I certify that the above statements are true to the best of my knowledge.

Dr. Mohammad Nurul Huda
Professor & Director-MSCSE
Department of Computer Science & Engineering (CSE)
United International University (UIU)
Dhaka-1212, Bangladesh

`

Abstract

STASH - An Inventory Management System is a web based application where aim is to manage the stock of an organization. With the involvement of technology system the inventory management system refers to the system that mainly processes to manage the stock of organization. To store the details of the purchase, adjustments, bookings, sample issue, sales and generate report based on criteria this system can be used. We are solving different problems, whose are involve in direct sales management and purchase management of this project. This system is important for both warranty and nowarranty item. Unless making any proper inventory control, a bulky store may runout of stock on a valuable item. A good inventory management system improves the accuracy of inventory orders and show the actual quantity of available item. Inventory Management System is also important to lead a more organize warehouse. While recording the stock, an automated inventory management system helps to minimize the errors and maximize service effectiveness and value at the same time.

`

Acknowledgement

I would like to start by expressing my deepest gratitude to the Almighty Allah for giving me the opportunity, ability and the strength to finish the task successfully within the scheduled time.

This Project titled "**STASH- Inventory Management System**" has been prepared to fulfill the requirement of MSCSE degree. I am very much fortunate that I have received sincere guidance, supervision and co-operation from various persons.

I would like to give thanks to my supervisor, **Dr. Mohammad Nurul Huda**, Professor and Director-MSCSE, United International University, for his invaluable support like continuous guidance, encouragement, and patience, and for giving me the opportunity to do this work. His valuable suggestions and strict guidance made it possible to prepare a well-organized project report.

Finally, my special thanks and love to my parents for their support, encouragement, and endless love.

`

Table of Contents

LIST OF TABLE vii

LIST OF FIGURES viii

1. Introduction 1

1.1 Problem Statement 1

1.2 Objectives 2

1.3 Features of Project 3

1.4 Scope of the Application 3

2. System Analysis & Design 5

2.1 STASH Requirement 5

2.2 Users Requirements 6

2.3 Functional Specification for STASH 6

2.3.1 User Management Module 6

2.3.2 Parameter Setup Module 11

2.3.3 Purchase Module 18

2.3.4 Sale Module 22

2.4 System Diagrams 26

2.4.1 Data Flow Diagrams for STASH 26

2.4.2 Use Case Diagram for STASH 30

2.4.3 Sequence Diagram for STASH 32

2.4.4 Class Diagram for STASH 33

2.5 Grant Chart for STASH 35

3. Methodology 37

3.1 Introduction 37

`

3.2 Method to Develop STASH Software 37

3.3 Determine the Condition for Software 38

3.4 Determine the Software Features.. 39

3.5 Interface Identify the Software Development Tools to Be Used............................ 39

3.6 Screenshot of the Graphical User Interface 39

3.7 Table Design.. 42

4. Results 51

5. Conclusion 52

6. References.. 53

7. Appendix A.. 54

Few Line of Source Codes:.. 54

`

LIST OF TABLE

Table 1 : User Group 42

Table 2 : User Info..42

Table 3 : Vendor 43

Table 4 : Customer Info..44

Table 5 : Item Info 45

Table 6 : Purchase Order Info...46

Table 7 : Purchase Order Item...46

Table 8 : Receive Info..47

Table 9 : Item Receive Info 48

Table 10 : Sale Info..48

Table 11 : Sale Item Info 49

`

LIST OF FIGURES

Figure 1 : Zero Level Data Flow Diagram for STASH 27

Figure 2 : First Level Data Flow Diagram for STASH 28

Figure 3 : Second Level Data Flow Diagram for STASH 29

Figure 4 : Use Case Diagram for STASH 31

Figure 5 : Sequence Diagram for Login System 32

Figure 6 : Sequence Diagram for whole process after successful login 32

Figure 7 : Over all Class Diagram for STASH 34

Figure 8 : Grant Chart for STASH 36

Figure 9 : Methodology for STASH 38

Figure 10 : Graphical User Interface (Login Page) 40

Figure 11 : Graphical User Interface (Dashboard) 41

Figure 12 : Graphical User Interface (Report Module) 41

`

Chapter 1

Introduction

STASH – An Inventory Management System is a complete web based application designed on PHP technology using codeIgniter Framework. The main aim of the project is to develop inventory management system in which all the information regarding purchase, receive, booking, sale, adjustment and the stock of the organization will be existing. STASH is an intranet based web application or platform that can be useful for any product oriented organization, which has admin component or settings to manage the software. This application is based on the management of product and inventory reports of an organization. The application contains general organization profile, dynamic parameter setup like warehouse can be added as per user requirements in any time, sales details, Purchase details and see the remaining stock at dashboard that are presented in the organization. The login page is created for protect the management of organization in order to prevent it from any kinds of threads and misuse of the product.

1.1 Problem Statement

After analyzing many existing Inventory Management System we have now the obvious vision of the project to be developed. Before we started to build the application I faced many challenges. We defined our problem statement as:

- To make web based application of Inventory Management System for small and medium business organization.

- STASH is such kinds of system which is more secured and easily manageable.

- To cover all the areas of Inventory Management System like purchase details, item receive details, item receive return details, item adjustment details, item booking details, sample issue details, sample issue return details and sales details.

`

1.2 Objectives

Inventory management system is very useful for any product oriented organization. If we are not aware on our inventory or counting stock regularly, that are setting ourselves up for potential inventory errors and challenges. A good inventory management really act like a matter of fact that it can make or break a business! So we should keep the following benefits in mind

It expands the correctness of inventory orders:

A good inventory management system helps us figure out exactly how much product we need to have on-hand. This will help avoid product shortages and allow us to keep just enough product without having too much in the warehouse.

It leads to a more organized warehouse:

Proper inventory management strategy supports an organized warehouse. If our warehouse is not organized, we will have a hard time managing our inventory. Many companies choose to optimize their warehouses by putting the highest selling items together and in easily accessible places in the warehouse. This, in turn, helps speed up the order fulfillment process and keeps customers happy.

It helps save time and money:

Inventory management can have real-time and monetary benefits. By keeping track of which items we have on-hand or ordered, we save ourselves the effort of having to do an inventory recount to ensure our records re accurate. A good inventory management strategy also helps us to save money that could otherwise be wasted on slow-moving products.

It increases efficiency and productivity:

Inventory management devices, such as barcode scanners and inventory management software, can help drastically improve your efficiency and productivity. These devices will help eliminate manual processes so your employees can focus on other – more important – areas of the business.

It keeps your customers coming back for more:

It is a matter of fact that proper inventory management leads to what we are constantly striving for – repeat customers. If we want our hard-earned customers to come back for our products and services, we need to be able to meet customer demand quickly.

`

Inventory management helps us to meet this demand by allowing the right products on-hand as soon as our customers need them.

1.3 Features of Project

This application is used to show the details about the sales and purchase. It gives the details about the available stock of the product of an organization. Mechanisms are described as below:

Login page: At first when we starts with application, the login page appears. Admin login is strong-minded by the user Id and password that has all the authority to add, update and delete as per the requirement.

Purchase details: It manages the details about the purchase note and purchase order made by the organization along with the price and dates.

Item receive details: It manages the details about the item receive information. It also show the details about the item receive in return.

Item adjustment details: It manages the details about the item adjustment with appropriate remarks.

Sample issue details: It manages the details about the sample issue and the remaining stock of item. It also show the details about the sample issue in return.

Sales details: It show the details about the sales and the remaining stock of the product.

1.4 Scope of the Application

Scope of the project was selected based on time, requirements, and resources for the program. The scope of the STASH can be described as follows:

`

- Admin login new user creation or user registration function.
- User access functionality based on role
- A FIFO (First in first out) warehouse system
- Strictly maintain month closing procedure which will not allow pending task of closing month
- The always make sure that the data stored in it are always attached on its database. Retrieval of data or updating of data is one of the most characteristic of the proposed system.
- Warehouse can be added as per the requirement.
- Report Generation and every report download in excel.

`

Chapter 2

System Analysis & Design

System Analysis refers into the process of examining a situation with the purpose of improving it through better procedures and methods. System Analysis is the process of planning a new System to replace an existing system. But before any planning is done the old system must be thoroughly understood and the requirements fixed. System Analysis, is therefore, the process of accumulation and illustrate facts, diagnosing problems and using the information to re-comment improvements in the System. Or in other words, System Analysis means a detailed illustration or description. Analyzing is necessary before computerized a system under consideration. We need to study it functions currently, the problems, and the requirements that the software meet.

System Analysis is directed with the following objectives in mind:

- Identify the customer’s need.
- Evaluate the system concept for feasibility.
- Perform economic and technical analysis.
- Hardware, software people, database and other system elements allocation of function is prerequisite.
- Establish cost and schedule constraints.

2.1 STASH Requirement

The goal for the system is to manage the purchase and sale management function of the organization. Once it is automated all the functions can be practically managed and the organization can achieve the ruthless advantage. Business requirement are discussed in details:

- Helps to find the specific item and remaining stock.

`

- Details information about the item sales and purchase.
- To know the report and details of sales and purchase in exact date.
- There is proper example of First-In First Out process.
- Strictly maintain month closing process. There is no scope of pending issue.
- Only registered users can login in the system.

2.2 Users Requirements

User requirement are categorized by the user type

Super Admin

- Able to view the Full dashboard.
- Able to manage the user management and parameter data.
- Able to manage the settings of the system.
- Able to view all reports.

Admin

- Able to login the system with user ID and password.
- Able to add, edit, view and delete the entry as per entry.
- Able to approve and reject entry with appropriate rights.

2.3 Functional Specification for STASH

2.3.1 User Management Module

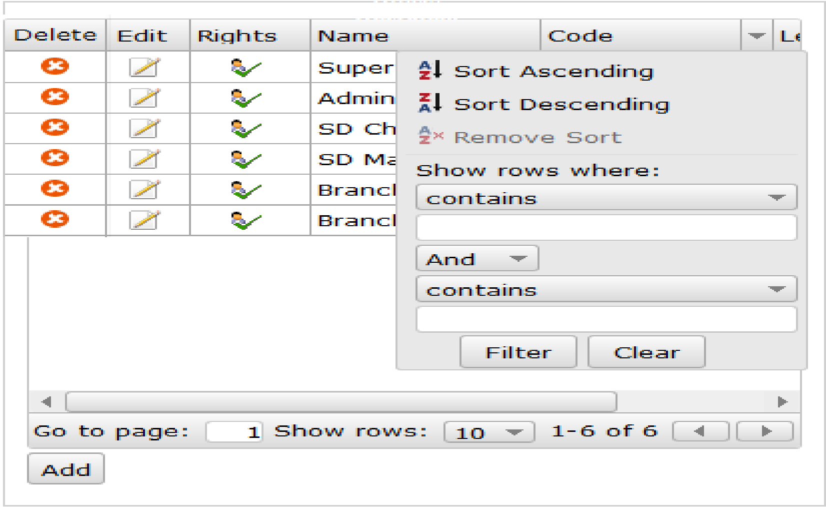
User Group Management (Functions include):

User group will manage and it will used at the time of user creation.

View (Grid)

`

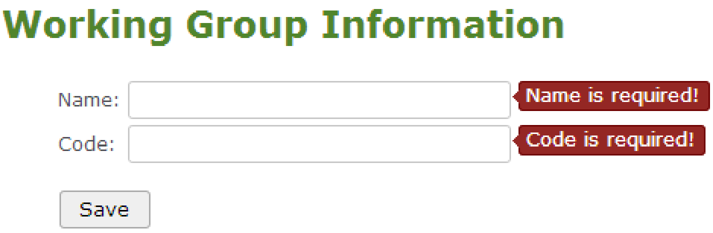
- There will be a grid to view all the user group with grid search
- In the grid there will be three action Edit, Delete and Rights
- There will be no checker option.



Delete	Edit	Rights	Name	Code	Le
✖	✎	✓	Super		
✖	✎	✓	Admin		
✖	✎	✓	SD Ch		
✖	✎	✓	SD Ma		
✖	✎	✓	Brancl		
✖	✎	✓	Brancl		

Add/Edit

- User group can be added/ updated here.



Working Group Information

Name: Name is required!

Code: Code is required!

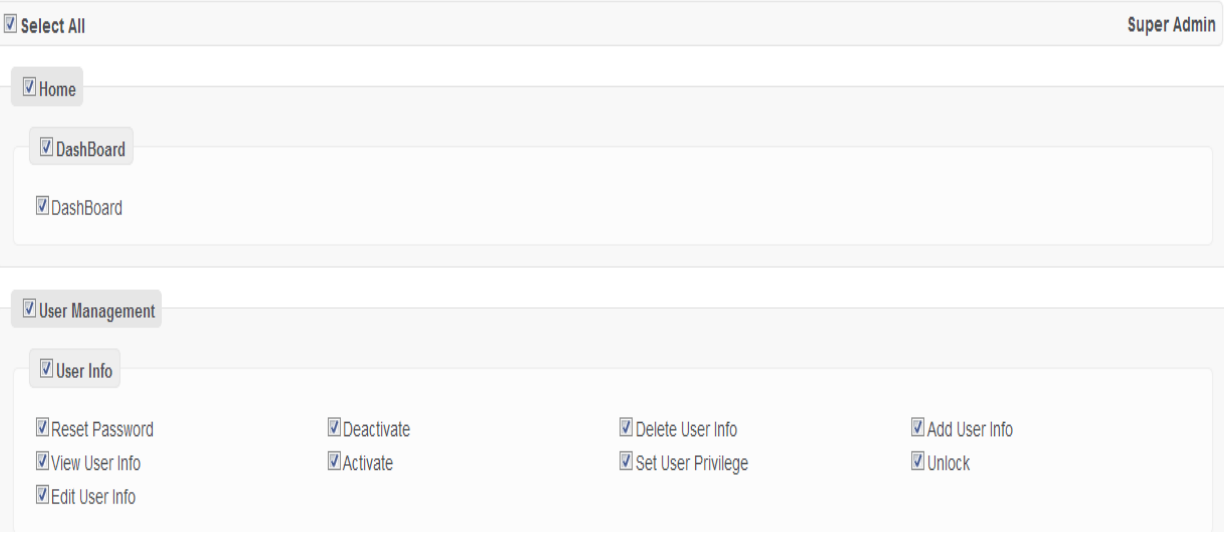
Delete

- There will be an option to delete User Group

Rights

- There will be an option to set group right and the right will automatically set at the time of user create

`



☑ Select All — Super Admin

☑ Home

☑ DashBoard

☑ DashBoard

☑ User Management

☑ User Info

☑ Reset Password ☑ Deactivate ☑ Delete User Info ☑ Add User Info

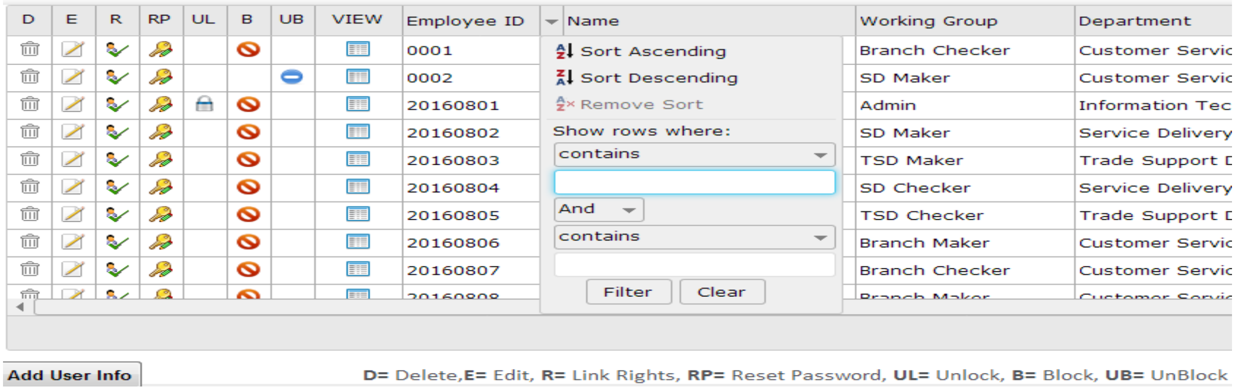
☑ View User Info ☑ Activate ☑ Set User Privilege ☑ Unlock

☑ Edit User Info

User Management (Functions include):

View (Grid)

- There will be a grid to view all the user with grid search
- In the grid there will be seven action Edit, Delete, Rights, Reset Password, Unlock, Block and Unblock
- There will be no checker option.



D	E	R	RP	UL	B	UB	VIEW	Employee ID	Name	Working Group	Department
🗑	✎	✓	🔑		🚫		▤	0001		Branch Checker	Customer Servic
🗑	✎	✓	🔑			⊖	▤	0002		SD Maker	Customer Servic
🗑	✎	✓	🔑	🔒	🚫		▤	20160801		Admin	Information Tec
🗑	✎	✓	🔑		🚫		▤	20160802		SD Maker	Service Delivery
🗑	✎	✓	🔑		🚫		▤	20160803		TSD Maker	Trade Support I
🗑	✎	✓	🔑		🚫		▤	20160804		SD Checker	Service Delivery
🗑	✎	✓	🔑		🚫		▤	20160805		TSD Checker	Trade Support I
🗑	✎	✓	🔑		🚫		▤	20160806		Branch Maker	Customer Servic
🗑	✎	✓	🔑		🚫		▤	20160807		Branch Checker	Customer Servic
🗑	✎	✓	🔑		🚫		▤	20160808		Branch Maker	Customer Servic

D= Delete,**E=** Edit, **R=** Link Rights, **RP=** Reset Password, **UL=** Unlock, **B=** Block, **UB=** UnBlock

Add/Edit

- User can be added/ updated here.

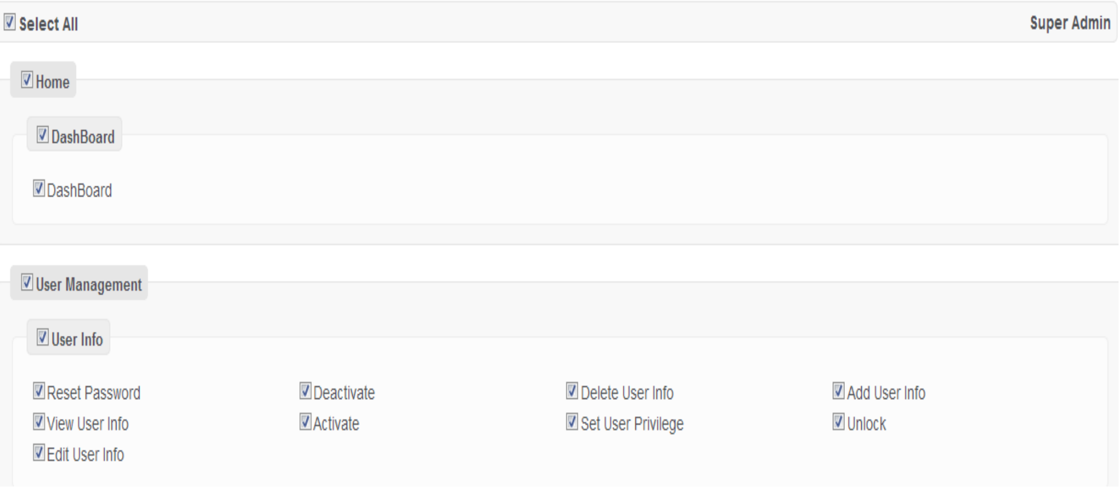
Delete

- There will be an option to delete User

`

Rights

- There will be an option to set user right



☑ Select All Super Admin

☑ Home

☑ DashBoard

☑ DashBoard

☑ User Management

☑ User Info

☑ Reset Password	☑ Deactivate	☑ Delete User Info	☑ Add User Info
☑ View User Info	☑ Activate	☑ Set User Privilege	☑ Unlock
☑ Edit User Info			

Reset Password

- There will be an option to reset the user password and password will be reset by setting.

Unlock

- There will be an option to unlock the user (which user is lock for two wrong login attempts)

Block

- There will be an option to Block the user

Unblock

- There will be an option to Unblock the block user

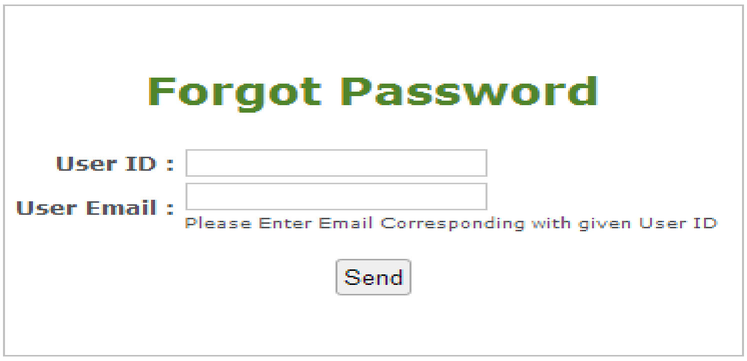
Forget Password

There will be an option to forget pass word and pass word will be sent to email

- A link will be including in login screen (Forgot Password? Click Here)

- By clicking on the link a pass word reset screen will be shown

`



Forgot Password

User ID :

User Email :

Please Enter Email Corresponding with given User ID

User_Login

This section allows user to enter into STASH by adding login details created during user profile creation.

User Access Management

SL	Key Points	Technical Details
1	User ID	Employee ID (7-8 digit numeric or pattern can be set by super admin) would be the user ID for every individual

Password Guideline

Parameters will be manageable through a GUI interface. As an example, current password length 8.In future may need to change 10 or any length that provision will be there.

SL	Key Points	Technical Details
1	Password Type	Any combination of Capital Letters, Small Letters & Numeric
2	Password Length	At least 8 digit and up to 64 digits; Parameters must be changeable.
3	Default Password	System must have a provision to set a predefined default password (a dot (.)) by only User Admin. Once this default password is set by User Admin for any user that user must user default password. When both of the above are true the system will prompt new password fields and the end user must not be able to enter the system until the new password is set.
4	Password Validity	Each and every password will be valid for 30 days max. After setting a password the system will automatically prompt for password change starting from 25th day during login to system and that alert will continue till 30th day. If the password is not changed within 30th day the user ID will be locked and only User Admin can manually activate

		that ID (will follow the default password process). There will be option to set the validity period of password.
5	Password History	System must not allow users to reuse immediate pervious 2 passwords. While changing passwords if new password matches any of immediate previous 4 passwords system will send an alert and will not accept the password and prompt to reassign new password.
6	User Lockout	If any user enter wrong password 3 times the user id will automatically be deactivate. Only user admin can reactivate the ID.
7	User Activation / Deactivation	There must be a provision to activate or deactivate a user id. Once an id is deactivated it will not be accessible.
8	User session timeout	If any user logged into the system and idle for 15 minutes system will forcibly logged out the user (time should be parameterized) There will be global setup from user interface and individual user wise times differ.
9	User login time	System must be capable to handle user login time duration i.e. starting and ending time of system login
10	Audit	• User ID Administration audit portal • List of active users • List of inactive users • Individual user wise creation, date, time, & every maintenance activities on the user.

2.3.2 Parameter Setup Module

Basic Parameters

The basic parameters of the software will be inputted by this module. The common functionalities are:

- View
- Add
- Edit
- Delete
- There will be no checker option.

The basic parameters are:

`

- Department
- Designation
- Blood Group
- Country
- Units of Measure
- HS Code
- Currency
- Bank
- Sector

Department/Division:
Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.
- This Parameter is used for when the user information will entry by this software.

Designation:
Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.
- This Parameter is used for when the user information will entry by this software.

Blood Group:
Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.

`

- This Parameter is used for when the user information will entry by this software.

Country:

Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.
- This Parameter is used for when the product information will entry by this software.

Units of Measure:
Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.
- This Parameter is used for when the product information will entry by this software.

HS Code:
Remarks

- Duplicate Name/Code will not be allowed.
- This Parameter is used for when the product information will entry by this software.

Currency:
Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.
- This Parameter is used for when the purchases note will entry by this software.

`

Bank:

Remarks

- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.
- This Parameter is used for when the Bill Payment will entry by this software.

Sector:

Remarks

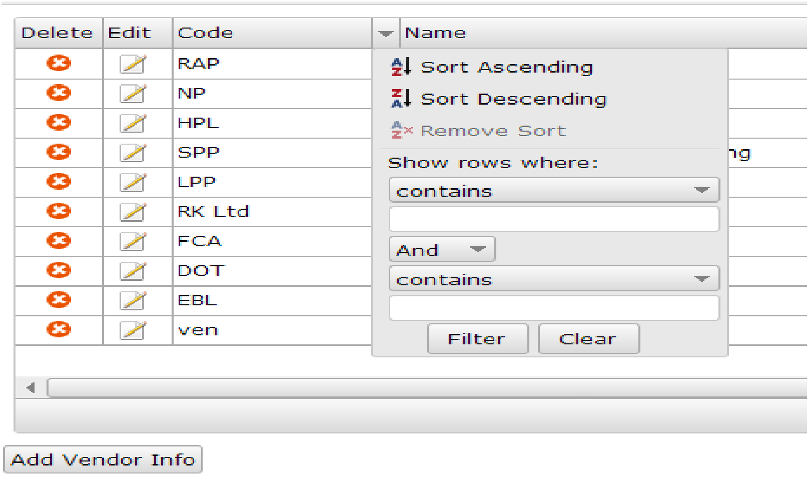
- Duplicate Name and Code will not be allowed.
- Code is sort form of Name. Sometimes Code is uses instead of Name.

Vendor Profile (Functions include):

STASH organizes standard vendor information into a single comprehensive system that provides easy access, review and retrieval.

View Vendor

- There will be a grid to view all the Vendor with grid search
- In the grid there will be four action Edit, Delete, Block and Unblock
- There will be no checker option.



Delete	Edit	Code	Name
✖	✎	RAP	
✖	✎	NP	
✖	✎	HPL	
✖	✎	SPP	ng
✖	✎	LPP	
✖	✎	RK Ltd	
✖	✎	FCA	
✖	✎	DOT	
✖	✎	EBL	
✖	✎	ven	

Add/Edit Vendor

- Vendor can be added/ updated here.

`

<u>Block</u>

- There will be an option to Block the Vendor

<u>Unblock</u>

- There will be an option to Unblock the block Vendor

<u>Delete</u>

- There will be an option to delete Vendor

Courier Company (Functions include):

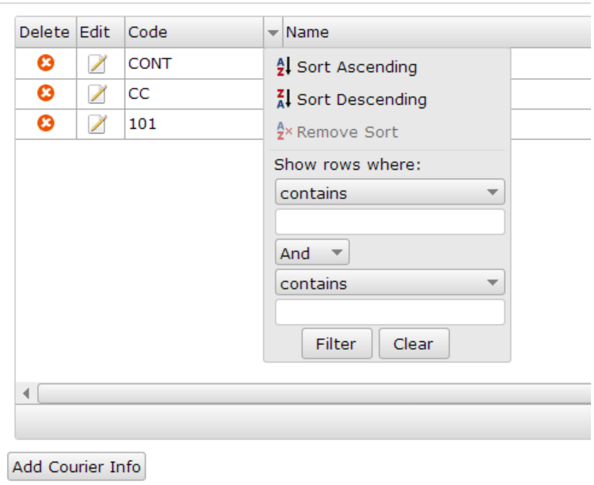
STASH organizes standard courier company information into a single comprehensive system that provides easy access, review and retrieval.

<u>View Courier Company</u>

- There will be a grid to view all the Courier Company with grid search

- In the grid there will be four action Edit, Delete, Block and Unblock

- There will be no checker option.



Delete	Edit	Code	Name
⊗	✎	CONT	
⊗	✎	CC	
⊗	✎	101	

<u>Add/ Edit Courier Company</u>

- Courier Company can be added/ updated here.

<u>Block</u>

There will be an option to Block the Courier Company

<u>Unblock</u>

`

- There will be an option to Unblock the block Courier Company

Delete

- There will be an option to delete Courier Company

Customer (Functions include):

View Customer

- There will be a grid to view all the Customer with grid search
- In the grid there will be four action Edit, Delete, Block and Unblock
- There will be no checker option.

Add/ Edit Customer

- Customer can be added/ updated here.

Block

- There will be an option to Block the Customer

Unblock

- There will be an option to Unblock the block Customer

Delete

- There will be an option to delete Customer

Terms and Conditions (Functions include):

View Terms and Conditions

- There will be a grid to view all the Terms and Conditions with grid search
- In the grid there will be two action Edit and Delete
- There will be no checker option.

Add/ Edit Terms and Conditions

- Terms and Conditions can be added/ updated here

Delete

- There will be an option to delete Terms and Conditions

`

Item Category (Functions include):

View Item Category

- There will be a grid to view all the Item Category with grid search
- In the grid there will be two action Edit and Delete
- There will be no checker option.

Add/ Edit Item Category

- Item Category can be added/ updated here.

Delete

- There will be an option to delete Item Category. In case of items available under a specific item category that category can't be deleted.

Item Sub-Category (Functions include):

View Item Sub-Category

- There will be a grid to view all the Item Sub-Category with grid search
- In the grid there will be two action Edit and Delete
- There will be no checker option.

Add/ Edit Item Sub-Category

- Item Sub-Category can be added/ updated here.

Delete

- There will be an option to delete Item Sub-Category. In case of items available under a specific item Sub-category that Sub-category can't be deleted.

Item (Functions include):

View Item

- There will be a grid to view all the Item with grid search
- In the grid there will be two action Edit and Delete

`

- There will be a View option to view all product details
- There will be no checker option.

<u>Add/ Edit Item</u>

- Item can be added/ updated here.

<u>Delete</u>

- There will be an option to delete Item. In case quantity available under a specific item that item can't be deleted.

2.3.3 Purchase Module

Purchase Note (Functions include):

<u>View Purchase Note</u>

- There will be a grid to view all the Purchase Note with grid search
- In the grid there will be four actions Edit, Delete, send to checker and print.
- After reject by checker, Recommender and approver maker can edit maker's data.

<u>Add/ Edit Purchase Note</u>

- Purchase Note can be added/ updated here.

<u>Remarks</u>

- User can add more rows for Purchase Note Item.

<u>Delete</u>

- There will be an option to delete. After reject by checker, Recommender and approver maker can delete maker's data.

<u>Send to Checker</u>

- There will be an option to send the purchase note for Checker's checked.

<u>Checker (Approve/Reject)</u>

- There will be an option for Checker to approve or reject the purchase note. If it is approved than it will go for recommenders approve. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

<u>Recommender (Approve/Reject)</u>

`

- There will be an option for Recommender to approve or reject the purchase note. If it is approved than it will go for higher authorities approve. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Higher Authorities (Approve/Reject)

- There will be an option for higher authorities to approve or reject the purchase note. If it is approved than it will ready for make purchase order. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Print

- There will be an option for print the purchase note. Print option will come after Approved by higher authorities.

Purchase Order (PO) (Functions include):

STASH generates purchase order based on item reorder points. PO can only be generated after successful approval of the purchase note from higher authority. Option will be available to email purchase order directly to vendor from STASH. At the time of creating the Purchase Order there will be option to view earlier purchases history of particular item.

View Purchase Order

- There will be a grid to view all the Purchase Order with grid search
- In the grid there will be four actions Edit, Delete, send to Approver and Print.
- After reject by approver maker can edit maker’s data.

Add/ Edit Purchase Order

- Purchase Note can be added/ updated here.

Remarks

- Only vendor supplied (Item mapping) items will be enabled with checkbox.
- Partial purchase order will be checked against purchase note.

Terms and Condition:

`

- All the terms and condition will be shown here with checkbox for select the appropriate one and description can be change.

Delete

- There will be an option to delete. After reject by approver maker can delete maker’s data.

Send to Approver

- There will be an option to send the purchase order for Approver’s Approve.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the purchase order. If it is approved than it will go for Print. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Print

- There will be an option for print the purchase order. Print option will come after Approved by Approver. There will be two type of print format (Foreign, Local)

Item Receive

View Item Receive

- There will be a grid to view all the Item Receive with grid search
- In the grid there will be three actions Edit, Delete, send to Approver.
- Edit and delete can be possible before Approve.

Add/ Edit Item Received

- Item Received can be added/ updated here.
- Item Receives against a Purchase Order.
- Provision of partial receives against a Purchase Order.

Delete

- There will be an option to delete. After reject by approver maker can delete maker’s data.

`

Send to Checker

- There will be an option to send the entry for Approver’s Approve/Reject.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the Received. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Item Receive Return

View Item Receive Return

- There will be a grid to view all the Item Receive return with grid search
- In the grid there will be three actions Edit, Delete, send to Approver.
- Edit and delete can be possible before Approve.

Add/ Edit Item Receive Return

- Item Received Return can be added/ updated here.
- Item Receive Return against an Item Received.

Delete

- There will be an option to delete. After reject by approver maker can delete maker’s data.

Send to Approver

- There will be an option to send the entry for Approver’s Approve/Reject.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the Received. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Vendor Payment/Bill Payment

View Vendor Payment

`

- There will be a grid to view all the Vendor Payment with grid search
- In the grid there will be three actions Edit, Delete, send to Approver.
- Edit and delete can be possible before Approve.

Add/ Edit Vendor Payment

- Vendor Payment can be added/ updated here.

Delete

- There will be an option to delete. After reject by approver maker can delete maker's data.

Send to Approver

- There will be an option to send the entry for Approver's Approve/Reject.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the Received. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Set Item Actual Price

- There will be an option to set item actual price (item received reference no. wise).

2.3.4 Sale Module

Booking

View Booking

- There will be a grid to view all the Booking with grid search
- In the grid there will be four actions Edit, Delete, send to Checker and Cancel.
- After reject by approver maker can edit maker's data.

Add/ Edit Booking

- Booking can be added/ updated here.

Remarks

- User can add more rows for Booking Items.

`

Delete

- There will be an option to delete. After reject by approver maker can delete maker’s data.

Send to Checker

- There will be an option to send the Booking for Checker’s checked.

Checker (Approve/Reject)

- There will be an option for Checker to approve or reject the Booking. If it is approved than it will go for Approvers approve. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the Booking. If it is approved than it will ready for sale. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Cancel

- There will be an option to cancel the Booking after Approver Approve.

Sale

View Sale

- There will be a grid to view all the sale with grid search

- In the grid there will be six actions Edit, Delete, send to Checker, Create Challan , Print Invoice and Edit after Verify.

- After reject by approver maker can edit maker’s data.

Add/ Edit Sale

- Sale can be added/ updated here.

Remarks

- User can add more rows for Sale Items.

`

Delete

- There will be an option to delete. After reject by approver maker can delete maker's data.

Send to Checker

- There will be an option to send the Sale for Checker's checked.

Checker (Approve/Reject)

- There will be an option for Checker to approve or reject the Sale. If it is approved than it will go for Approvers approve. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the Sale. If it is approved than it will ready for Create Challan, Invoice and Stock will be updated automatically. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Create Challan

- There will be an option to Create Challan after Approver Approve.

Print Invoice

- There will be an option for print invoice. Print Invoice option will come after Approved by Approver.

Edit After Verify

- There will be an option to edit the verify sale. Edit after verify will only be possible before Create Challan and month closing. Edit after verify option will come after Approved by Approver.

Sample Issue:

View Sample Issue

- There will be a grid to view all the Sample Issue with grid search

`

- In the grid there will be five actions Edit, Delete, send to Checker, Create Challan and Sale.

- Edit and delete can be possible before Approve.

<u>Add/ Edit Sample Issue</u>

- Sample Issue can be added/ updated here.

<u>Remarks</u>

- User can add more rows for Sample Issue Items.

<u>Delete</u>

- There will be an option to delete. After reject by checker and approver maker can delete maker's data.

<u>Send to Checker</u>

- There will be an option to send the Sample Issue for Checker's checked.

<u>Checker (Approve/Reject)</u>

- There will be an option for Checker to approve or reject the Sample Issue. If it is approved than it will go for Approvers approve. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

<u>Approver (Approve/Reject)</u>

- There will be an option for Approver to approve or reject the Sample Issue. If it is approved than it will ready for Create Challan. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

<u>Challan</u>

There will be an option to Create Challan. After Approve Create Challan can be possible.

<u>Sale</u>

- There will be an option to Sale. After Approve Sale can be possible.

Sample Issue Return:

`

View Sample Issue Return

- There will be a grid to view all the Sample Issue return with grid search
- In the grid there will be three actions Edit, Delete, send to Checker.
- Edit and delete can be possible before Approve.

Add/ Edit Sample Issue Return

- Sample Issue Return can be added/ updated here.
- Sample Issue Return against a Sample Issue.

Delete

- There will be an option to delete. After reject by Checker/approver maker can delete maker's data.

Send to Checker

- There will be an option to send the Sample Issue Return for Checker's checked.

Checker (Approve/Reject)

- There will be an option for Checker to approve or reject the Sample Issue Return. If it is approved than it will go for Approvers approve. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

Approver (Approve/Reject)

- There will be an option for Approver to approve or reject the Sample Issue Return. If it is approved than it will update stock. If it is reject than it will go to maker for edit or delete. A reject reason is needed at the time of rejection.

2.4 System Diagrams

2.4.1 Data Flow Diagrams for STASH

Data Flow Diagrams show the flow of data from external entities into the system, showed how the data moved from one process to another process, as well as its logical storage

`

Figure 1 shows a zero level data flow diagram for STASH. It is an abstraction view, represent the project such as STASH a single process with its relationship to external entities like user management, parameter setup, purchase, sale and report. It represent the STASH as a single bubble and incoming and outgoing arrows indicate the input and output data.

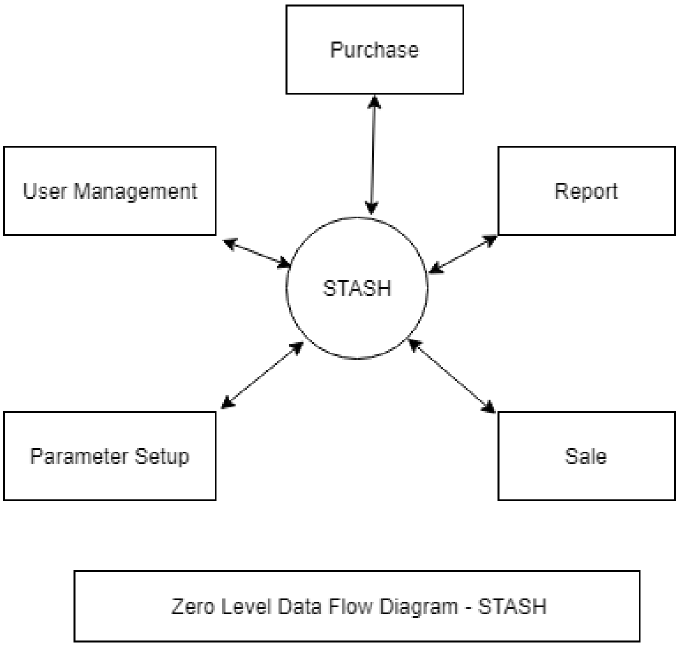


Figure 1 : Zero Level Data Flow Diagram for STASH

`

Figure 2 shows a first level data flow diagram for STASH. It is deeper than zero level data flow diagram, represent the project such as STASH a single process and highlight the external entities like user management , parameter set up, purchase, sale, report with their functions like display all user information, manage parameter data, display purchase and receive information, display sale information and generate stock report.

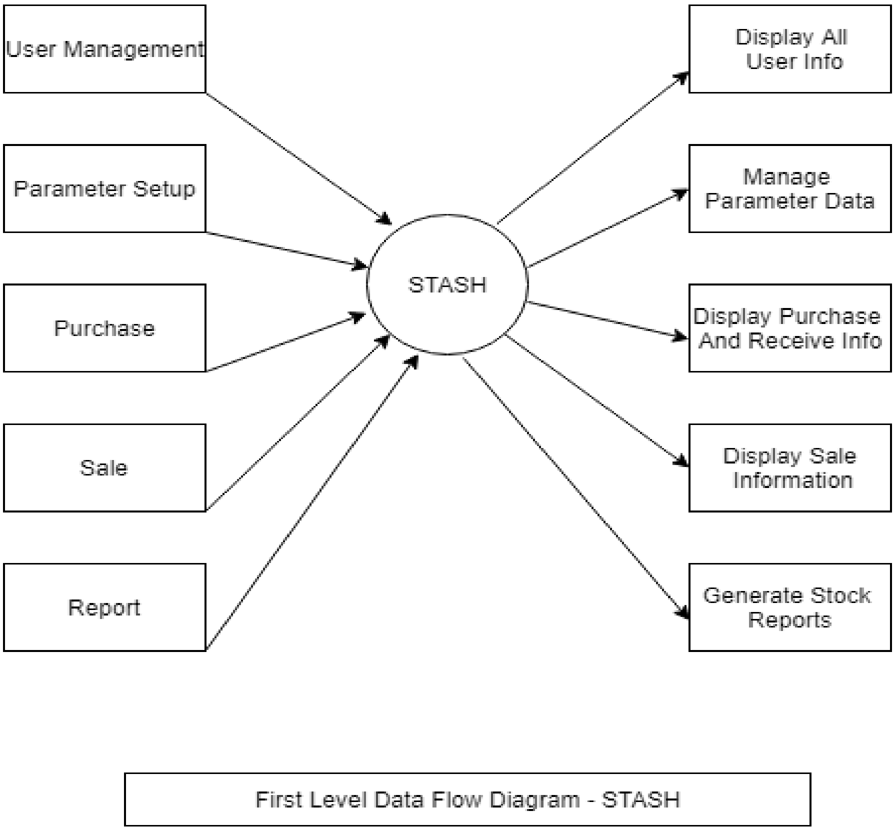


Figure 2 : First Level Data Flow Diagram for STASH

`

Figure 3 shows a second level data flow diagram for STASH. It is deeper than first level data flow diagram, represent the project such as STASH a single process and highlight main functions like manage system user, manage roles of user, manage purchase info, manage item receive info, manage item adjustment, manage booking information, manage sample issue information, manage sample issue return information, manage sale information, report generation, manage login report, manage user permission. It can be used to plan the necessary detail about the system functioning.

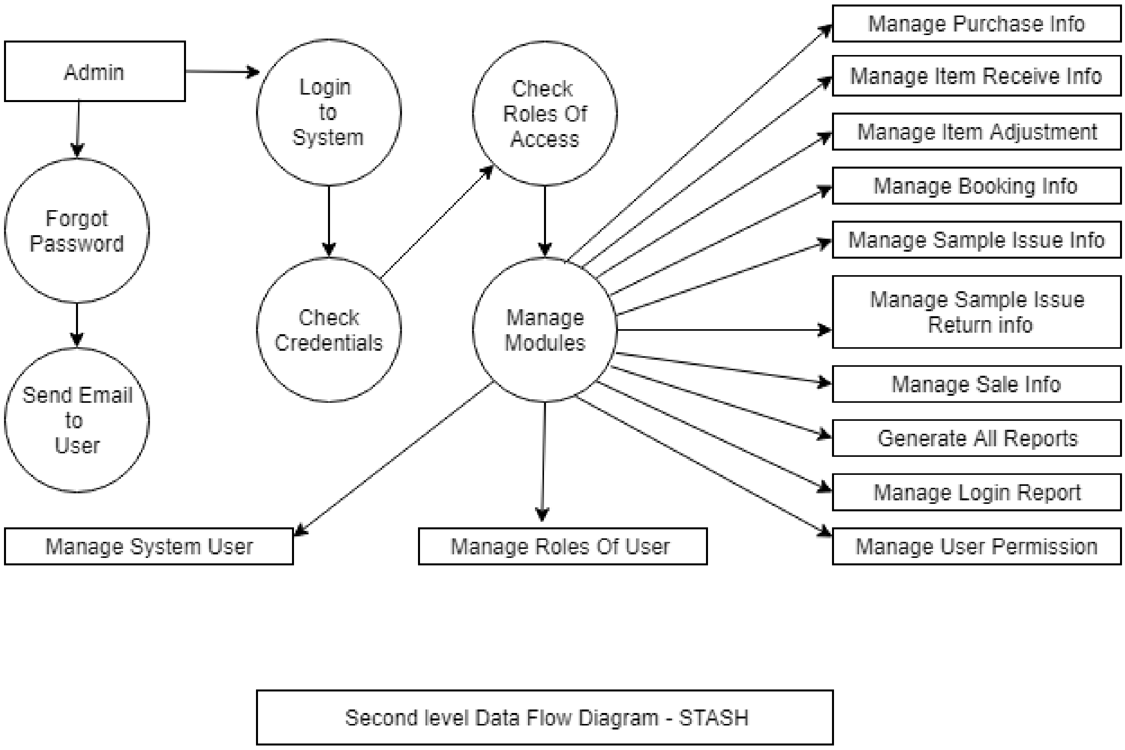


Figure 3 : Second Level Data Flow Diagram for STASH

`

2.4.2 Use Case Diagram for STASH

In this use case diagram aimed to present a graphical representation of the functionality which is provided by system in terms of actors and their goals. It is focused to show what system functions are performed for which actors.

In STASH two kinds of actors are present. One is Admin and another is User. They manages different links. Here manage means operate action like add, edit, delete, view and print and so on. In case of Users action depends on their given permission.

In Home Module Admin see the full dashboard, User see dashboard based on their permission. In User Management Module Only Admin see and manage sub modules like user group. User info, settings.

In Parameter module only Admin manage parameter setup, manage parameter data entry. Admin and users both manage vendor information, manage courier information, manage customers, manage terms and conditions, manage item category, manage item sub category, manage item, manage VAT & AIT and manage services.

In Purchase module Admin and users both manage purchase note, manage purchase order, manage item receive, manage item receive return and manage item adjustment.

In Sale module Admin and users both manage booking information, manage sale information, manage challan information, manage sample issue information, manage sample issue return information, manage cheque register like cheque deposite and cheque reconcile and manage month closing.

In Report module Admin and users both see item stock report, item sale report, item warranty report, item receive report, purchase order report, item list report, customer report and see vendor report.

`

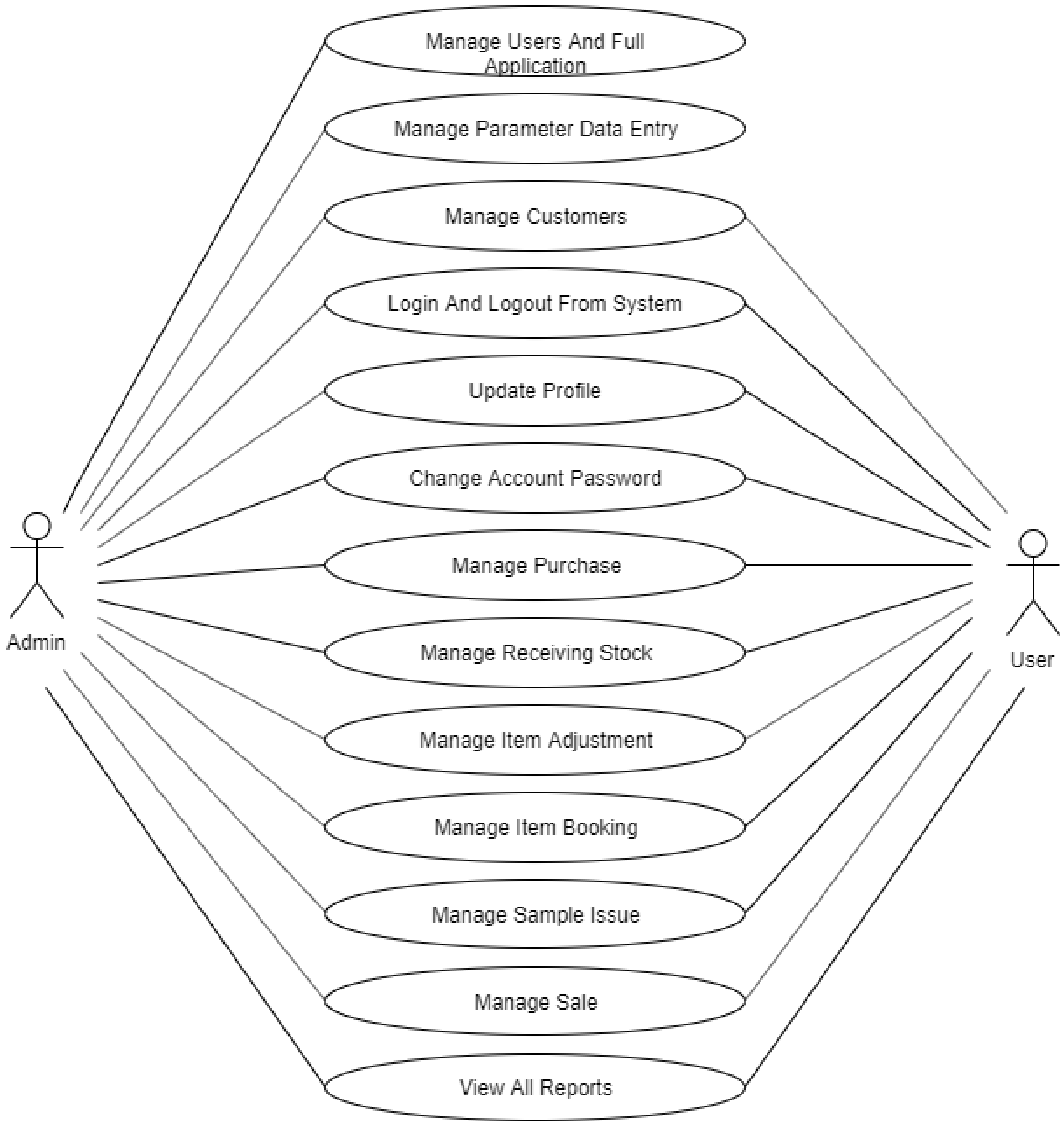


Figure 4 : Use Case Diagram for STASH

`

2.4.3 Sequence Diagram for STASH

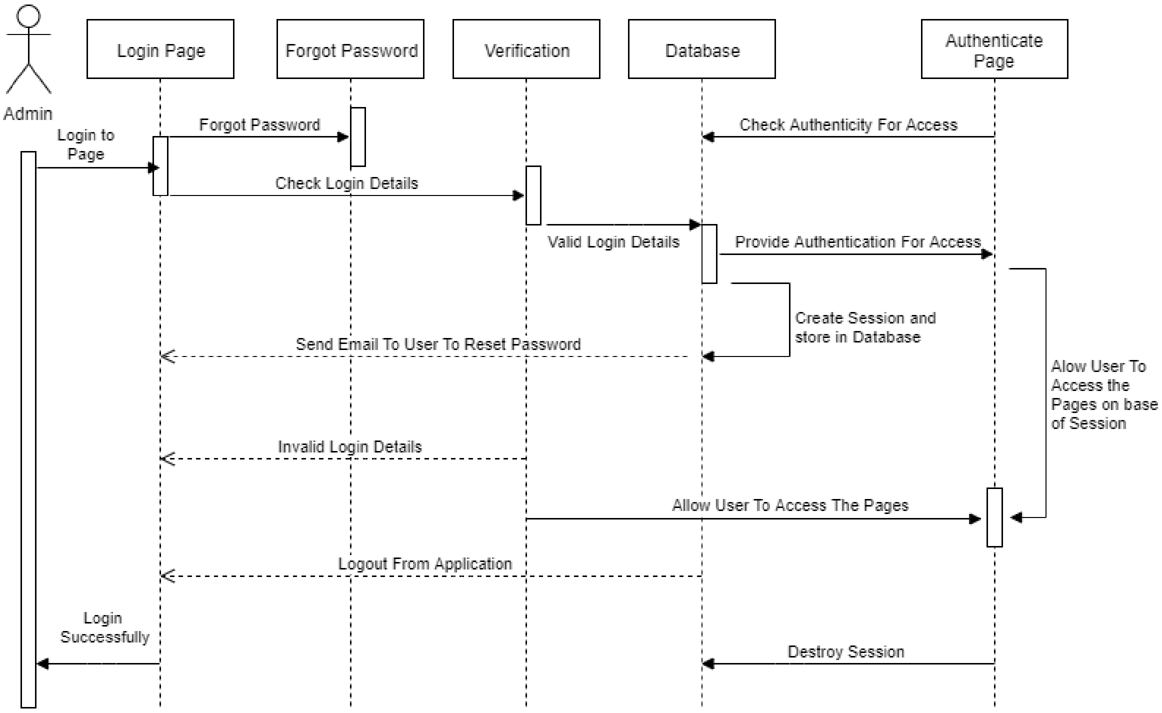

Admin
Login Page
Forgot Password
Verification
Database
Authenticate Page
Login to Page
Forgot Password
Check Authenticity For Access
Check Login Details
Valid Login Details
Provide Authentication For Access
Create Session and store in Database
Send Email To User To Reset Password
Alow User To Access the Pages on base of Session
Invalid Login Details
Allow User To Access The Pages
Logout From Application
Login Successfully
Destroy Session

Figure 5 : Sequence Diagram for Login System

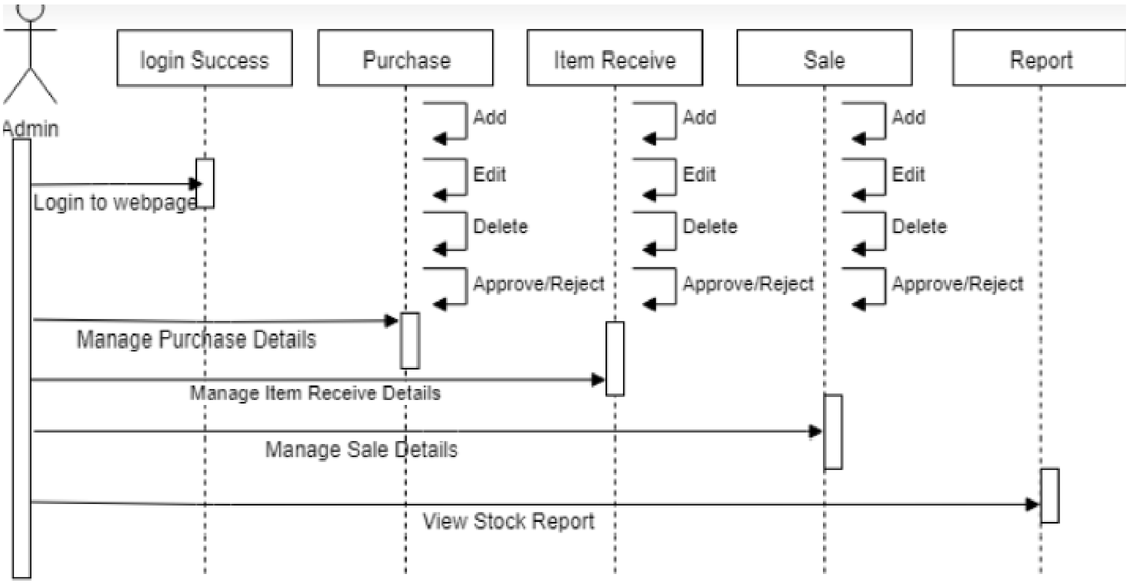


Figure 6 : Sequence Diagram for whole process after successful login

`

2.4.4 Class Diagram for STASH

Class diagram is a diagram that describe the structure of a system by showing the system classes, their attributes, methods and the relationship among objects. A class notation consists of class name, class attributes and methods. Class name in first portion, class attributes with data type in second portion and methods with return type in third portion. In STASH project there is eleven class.

In user class attributes are id, employee id, name, gender, user group, department, designation, joining date, date of birth, phone number, emergency contact number, first name, last name. Attributes are: add, edit, delete, view, unlock, active, deactivate, reset password.

In role class attributes are id, title, description. Attributes are: add, edit, delete, view, assign.

In permission class attributes are id, role id, title, module, description. Attributes are: check permission, uncheck permission.

In purchaseNote class attributes are id, reference, date, department, item, quantity, unit price and amount. Attributes are: add, edit, delete, view, sendToChecker, checkerApproval and print.

In purchaseOrder class attributes are id, purchaseNote, reference, date, vendor, item, quantity, unit price and amount. Attributes are: add, edit, delete, view, sendToChecker, checkerApproval and print.

In itemReceive class attributes are id, purchaseOrder, reference, date, receivedBy, receivedDate, warehouse, item, quantity, unit price and amount. Attributes are: add, edit, delete, view, sendToChecker, checkerApproval and print.

In itemAdjustment class attributes are id, reference, date, remarks, item and quantity. Attributes are: add, edit, delete, view, sendToChecker and checkerApproval.

In itemSale class attributes are id, option, reference, customer, saleBy, saleDate, vendor, item, commission, vat, discount, advance, quantity, unit price and amount. Attributes are: add, edit, delete, view, sendToChecker, checkerApproval and print.

In itemBooking class attributes are id, reference, date, bookingType, item, quantity, customer and remarks. Attributes are: add, edit, delete, view, sendToChecker, checkerApproval and cancel.

`

In sampleIssue class attributes are id, reference, date, item, quantity and customer. Attributes are: add, edit, delete, view, sendToChecker and checkerApproval.

In sampleIssueReturn class attributes are id, reference, sample id, date, item, quantity and customer. Attributes are: add, edit, delete, view, sendToChecker and checkerApproval

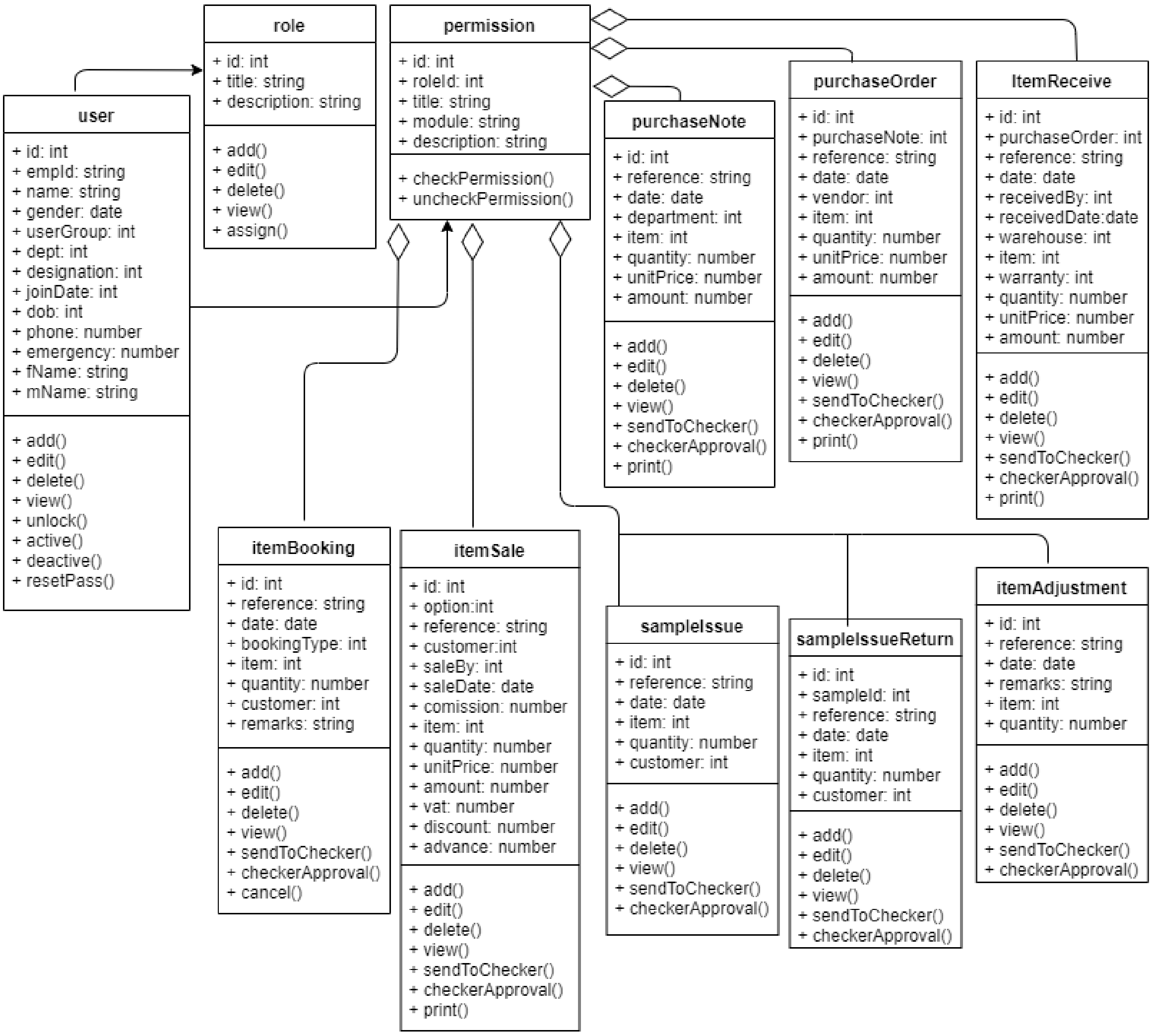
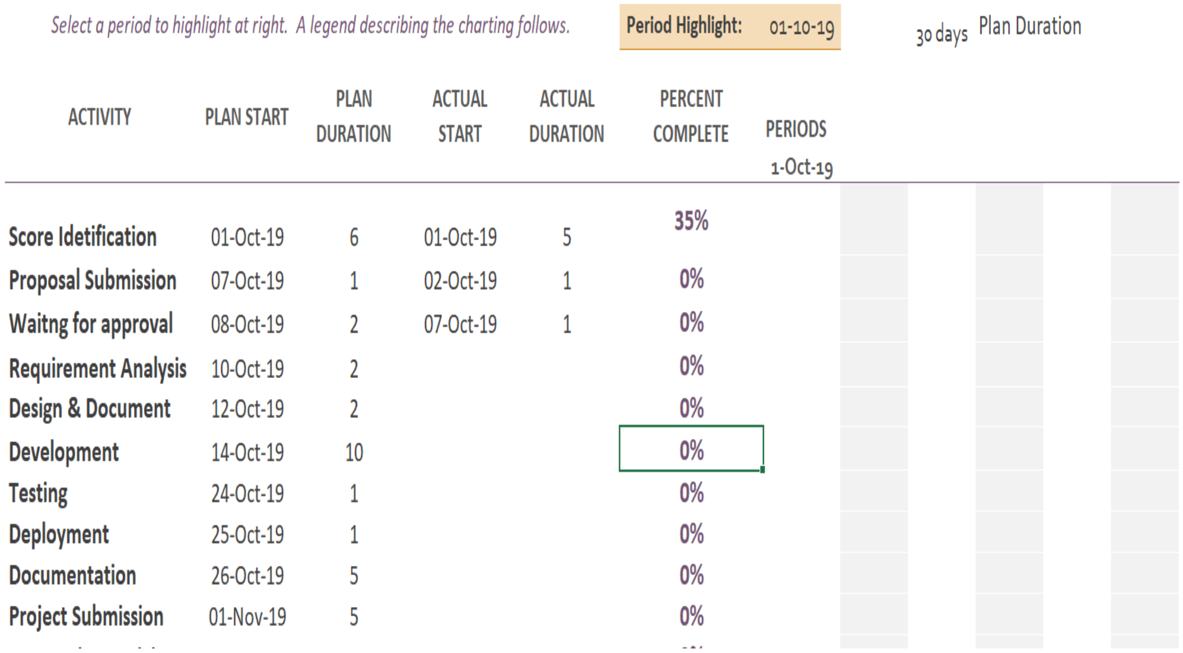


Figure 7 : Over all Class Diagram for STASH

2.5 Grant Chart for STASH

I decide that creating a timeline of my project is essential. Timeline creation is started with score identification. Score identification started from 1st October 2019 and end at 6th October 2019 where plan duration was 6 days but it completed at 5 days with prepare project proposal. After score identification I submit my project proposal at 7th October 2019 and waiting for approval at 8th October 2019. Then I start requirement analysis from 10th October 2019 with colleting a number of related website and visit, then interview to the concerned people and experiences regarding the concepts. Then visited some organization in Dhaka and analyze its importance and try to note the weakness that were found. From 12th October 2019 I start to design a system with different logic flow and PHP as language which will be suitable for the small and medium organization and then document it by 14th October 2019. From 14th October 2019 I start to develop the system. While developing the system I follow some performance characteristics like user friendliness, user satisfaction where system meet user all kind of logical expectations, reduce response time, error handling where without halting any system operation response to user errors and the undesired situations, the system should be able to avoid fatal behavior and robustness where without human intervention the system should recover from undesired events. At 24th October 2019 development process was completed then I started to test it and after testing it was deployed at 25th October 2019. Then 26th October 2019 to 1st November 2019, 5 days was for documentation. At 1st November 2019 I submit the project. Then 23th November 2019 was my final defense date.

`



Select a period to highlight at right. A legend describing the charting follows.

Period Highlight: 01-10-19

30 days Plan Duration

ACTIVITY	PLAN START	PLAN DURATION	ACTUAL START	ACTUAL DURATION	PERCENT COMPLETE	PERIODS 1-Oct-19
Score Idetification	01-Oct-19	6	01-Oct-19	5	35%	
Proposal Submission	07-Oct-19	1	02-Oct-19	1	0%	
Waitng for approval	08-Oct-19	2	07-Oct-19	1	0%	
Requirement Analysis	10-Oct-19	2			0%	
Design & Document	12-Oct-19	2			0%	
Development	14-Oct-19	10			0%	
Testing	24-Oct-19	1			0%	
Deployment	25-Oct-19	1			0%	
Documentation	26-Oct-19	5			0%	
Project Submission	01-Nov-19	5			0%	

Figure 8 : Grant Chart for STASH

`

Chapter 3

Methodology

3.1 Introduction

In this section, the method and software development tool used to develop the STASH- An Inventory Management System will be discussed. The method used to develop the software will be explained in details.

3.2 Method to Develop STASH Software

First, a condition will be set on the system based on the usage of the software. Then, the features of the system are determined based on the needs. Next, a data flow diagram is created to find out the flow of data from server to user. The software development tools is prefer based on the limitation and features. Then, the software flow chart is designed based on the limitation and features. Next, the conceptual design of the Graphical User Interface is created to device how the software will looks like when it is done. Next the Graphical User Interface will be built with the full function follow by the formation of database. The software will be debugged up to it can run smoothly. Lastly, a survey form is created to gather data on user satisfaction and the software will be tested by 10 persons and then fill up the survey form. Figure 9: Methodology for STASH- An Inventory Management System.

`

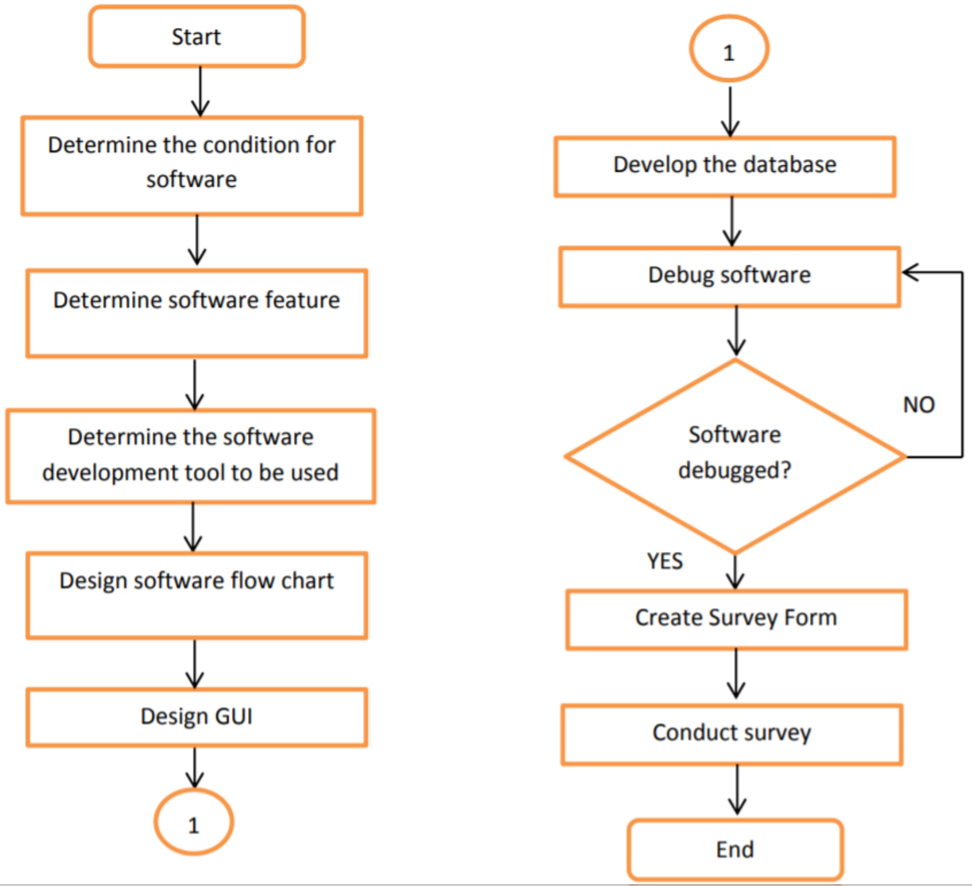

Start
Determine the condition for software
Determine software feature
Determine the software development tool to be used
Design software flow chart
Design GUI
1
1
Develop the database
Debug software
Software debugged?
NO
YES
Create Survey Form
Conduct survey
End

Figure 9 : Methodology for STASH

3.3 Determine the Condition for Software

A condition is set to act as the boundary of the system so that it will only function within the boundary based on the motion of the software. According to the objective of this system, the software will be used within any product oriented organization. Hence, it is assume that the software will be used locally, within the company and there is one or more storage area since the company is small or medium in size. Next the software will be made available only on the server computer to register the check in/ out of the item and check the status of items to prevent more than one access at the same time which will causes data to overload.

`

3.4 Determine the Software Features

The features of the software are determined based on the expected outcome and the result of the research done in chapter 1 Overview. First, the software have to be user friendly so that it is easily understand and can be used by the user with a small training. Next, the basic feature of the system is to register item for both item in and item out. User can add in description to help user to identify the item. The database will be updated in real time. In addition, admin will able to know the item information and status at any time from the server computer. The database is MySQL database that will be hosted by a free software XAMPP. Moreover, there is a dashboard to see the overall condition and the item quantity of the certain item and is lower than the 5 so that the user can restock accordingly.

3.5 Interface Identify the Software Development Tools to Be Used

A research have been done on the current Inventory Management System software in the market. It can be concluded that the language used to develop the system is normally separated into two group PHP and JAVA programming. Most of the software is web based which means that they are developed using PHP language that runs on web platform. There are a few of the software is developed using JAVA which can be website based, windows based or android based. JavaScript can be added in PHP to run on website and JAVA language can be used to develop Android apps or even to make windows based apps.

3.6 Screenshot of the Graphical User Interface

Figure 10 shows the graphical representation of STASH login page. It is a simple login form. Here two input box are present, one is for input User Id and another is for input password. If any user input user Id and password correctly then user can successfully logged in with this system. If any user cannot properly input his/her user Id and password then system show the error message. When user try with wrong user id and password consecutively three times then system automatically lock this user. There is option named forgot password, it is applicable when any user forgot their password they can recover their password by clicking click here link.

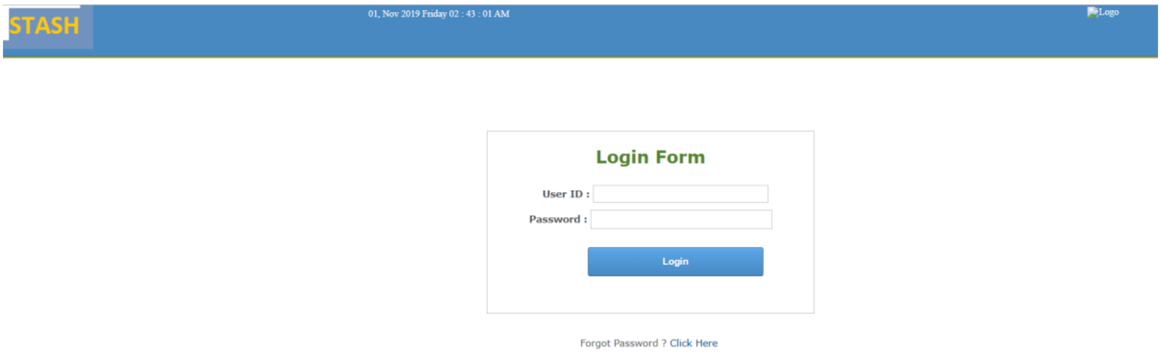

STASH
01, Nov 2019 Friday 02 : 43 : 01 AM
Logo
Login Form
User ID :
Password :
Login
Forgot Password ? Click Here

Figure 10 : Graphical User Interface (Login Page)

Figure 11 shows a complete dashboard of the STASH. Here we see the eight slot named item short fall, purchase note verify pending, work order verify pending, item receive verify pending, item sale verify pending, booking verify pending, sample issue verify pending and month closing history.

In item short fall slot we see summary of item information like item name with threshold value and most importantly we see the available stock of individual item. This is a very essential part of this project. Item short fall summary report also can download in excel format.

In purchase note verify pending slot we can see summary of purchase note not verified entries with purchase note reference number, department name, status means verify pending stage and pending days named aging.

In work order verify pending slot we can see summary of purchase order not verified entries with purchase order reference number, vendor name and pending days named aging.

In item receive verify pending slot we can see summary of item receive not verified entries with item receive reference number, vendor name and pending days named aging.

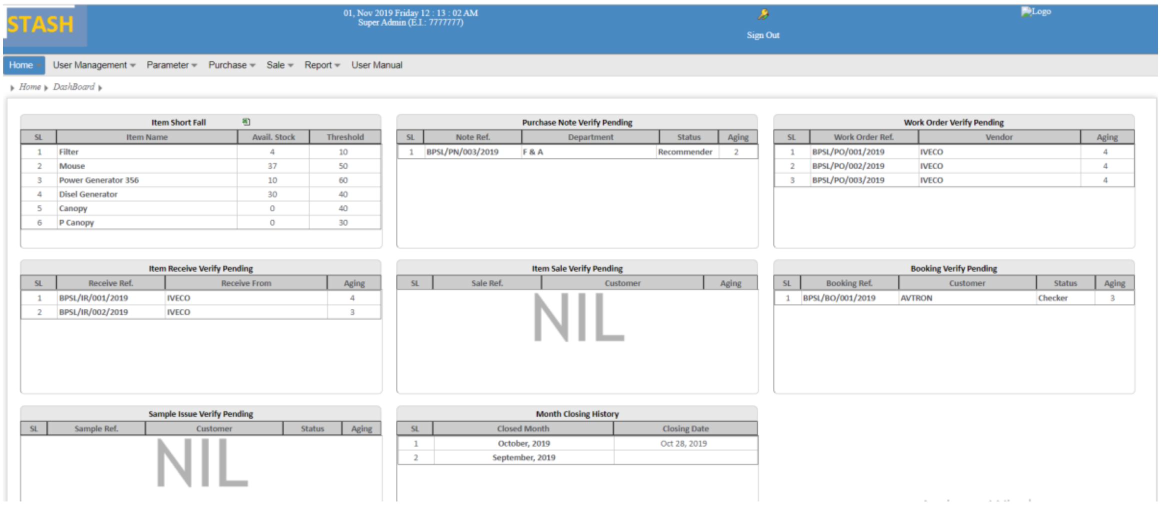
In item sale verify pending slot we can see summary of item sale not verified entries with item sale reference number, customer name and pending days named aging.

`

In booking verify pending slot we can see summary of booking not verified entries with booking reference number, customer name, status means verify pending stage and pending days named aging.

In sample issue verify pending slot we can see summary of sample issue not verified entries with sample issue reference number, customer name, status means verify pending stage and pending days named aging.

In month closing history slot we can see summary of closed month with year and closing date.



Item Short Fall

SL	Item Name	Avail. Stock	Threshold
1	Filter	4	10
2	Mouse	37	50
3	Power Generator 356	10	60
4	Disel Generator	30	40
5	Canopy	0	40
6	P Canopy	0	30

Purchase Note Verify Pending

SL	Note Ref.	Department	Status	Aging
1	BPSL/PN/003/2019	F & A	Recommender	2

Work Order Verify Pending

SL	Work Order Ref.	Vendor	Aging
1	BPSL/PO/001/2019	IVECO	4
2	BPSL/PO/002/2019	IVECO	4
3	BPSL/PO/003/2019	IVECO	4

Item Receive Verify Pending

SL	Receive Ref.	Receive From	Aging
1	BPSL/IR/001/2019	IVECO	4
2	BPSL/IR/002/2019	IVECO	3

Item Sale Verify Pending

SL	Sale Ref.	Customer	Aging

NIL

Booking Verify Pending

SL	Booking Ref.	Customer	Status	Aging
1	BPSL/BO/001/2019	AVTRON	Checker	3

Sample Issue Verify Pending

SL	Sample Ref.	Customer	Status	Aging

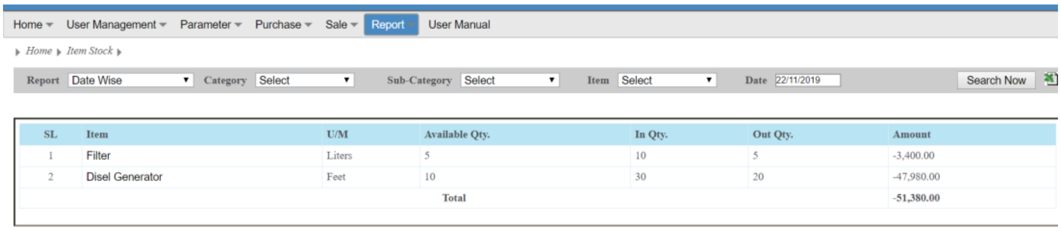
NIL

Month Closing History

SL	Closed Month	Closing Date
1	October, 2019	Oct 28, 2019
2	September, 2019	

Figure 11 : Graphical User Interface (Dashboard)

In Report module Admin and users both see item stock report, item sale report, item warranty report, item receive report, purchase order report, item list report, customer report and see vendor report. Figure 12 shows item stock report with date wise



Report: Date Wise
Category: Select
Sub-Category: Select
Item: Select
Date: 22/11/2019

SL	Item	U/M	Available Qty.	In Qty.	Out Qty.	Amount
1	Filter	Liters	5	10	5	-3,400.00
2	Disel Generator	Feet	10	30	20	-47,980.00
			Total			-51,380.00

Figure 12 : Graphical User Interface (Report Module)

`

3.7 Table Design

Table 1 : User Group

Caption	Input Type	Input Length	Mandatory	Validation
Name	Text	100	Yes	Duplicate Name will be check
Code	Text	50	Yes	Duplicate Code will be check

Table 2 : User Info

Caption	Input Type	Input Length	Mandatory	Validation
User ID	Text	100	Yes	Duplicate User ID will be check and User ID Length will come from settings
Password	Text	64	Yes	Password Validity Period, Password Length and Default Password Type will come from settings
Employee ID	Text	100	No	
Name	Text	250	Yes	
Gender	Drop down		Yes	
User Group	Drop down (Value comes from User Group data table)		Yes	
Department/ Division	Drop down (Value comes from Department data table)		Yes	
Designation	Drop down (Value comes from Designation data table)		No	
Date of joining	Date Picker	10	No	Date format will be check
Date of Birth	Date Picker	10	No	Date format will be check
Blood Group	Drop down (Value comes from Blood Group data table)			
Phone	Text	150	No	
Email	Text	150	No	Email format will be check
Emergency	Text	150	Yes	

`

Contact Number				
Relation with the Contact person	Text		Yes	
Father's Name	Text	150	Yes	
Mother's Name	Text	150	Yes	
Spouse Name	Text	150		
Send Email	Check Box			To Send Email for User Id and Password
Present Address	Text area		No	
Permanent Address	Text area		No	
Remarks	Text area		No	
Job History	Text area		No	
Reference	Text	150		

Table 3 : Vendor

Caption	**Input Type**	**Input Length**	**Mandatory**	**Validation**
Name	Text	100	Yes	Duplicate Name will be check
Type of Vendor	Drop down (Foreign, Local)		Yes	
Address	Text area		No	
Phone	Text	100	Yes	
Email	Text	100	Yes	Email format will be check
FAX	Text	100	No	
Website	Text	100	No	
Company Type	Drop down (Public Limited, Private Limited, Manufacturer, Authorized Dealer, Trader , Partnership, Proprietary)		No	
Present customers List	File Upload (Single) (10	250	No	

`

	MB)			
Details of various certifications	File Upload (Multi)	250	No	
Number of years of experience	Text	50	No	
Sales turnover for the previous 3 years (copy of Audited balance sheets for 3 years to be enclosed)	File Upload (Multi)	250	No	
Details of recognition/ awards if any won in the last three years.	File Upload (Single)	250	No	
Technical collaborations (if any)	File Upload (Multi)	250	No	
Details of manufacturing / service facilities and capacities: (Enclose details)	File Upload (Single)	250	No	
Any other information which you would like to highlight.	Text	250	No	
VAT Registration no	Text	50	No	
VAT Registration certificate	File Upload (Single)	250	No	
Income tax certificate	File Upload (Single)	250	No	
Solvency Certificate	File Upload (Single)	250	No	
Contact Person Name 1st	Text	100	Yes	
Contact Person Phone 1st	Text	100	Yes	
Contact Person Email	Text	100	No	Email format will be check
Contact Person Name 2nd	Text	100	Yes	
Contact Person Phone 2nd	Text	100	Yes	
Contact Person Email	Text	100	No	Email format will be check
Enlisted Status	Check Box		Yes	

Table 4 : Customer Info

Caption	Input Type	Input Length	Mandatory	Validation
Code	Text	50	Yes	Duplicate Code will be check
Name	Text	150	Yes	
Sector	Drop down		Yes	

`

Address	Text area		No	
Phone	Text	100	Yes	
Email	Text	100	Yes	Email format will be check
FAX	Text	100	No	
Website	Text	100	No	
Contact Person Name	Text	100	No	
Contact Person Phone	Text	100	No	
Contact Person Email	Text	100	No	Email format will be check
Status	Check Box			
Remarks	Text area			

Table 5 : Item Info

Caption	**Input Type**	**Input Length**	**Mandatory**	**Validation**
Code	Text	100	Yes	Duplicate Name will be check
Name	Text	200	Yes	Duplicate Name will be check
Category	Drop down (Value comes from Item Category data table)		Yes	
Sub-Category	Drop down (Value comes from Item Sub-Category data table)		Yes	
Warranty Able	Check Box			
Multi Serial Info. **	Check Box			Fixed information will be taken at the time of Item Received (Serial Number, Engine Serial Number, Alternator Serial Number)
Meas. Unit	Drop down (Value comes from Units of Measure data table)		Yes	
Threshold	Text	11	No	Number will be checked
Brand	Text	100	No	
Model	Text	100	No	
HS Code	Drop down (Value comes from HS Code data table)			
Vendor	Drop down with Multi select (Value comes from Vendor data table)	250	Yes	Multiple vendor will select for a product

Description	Text area			Description will be used for Purchase Order
Technical Specification	Text area			Technical Specification will be used for Quotation
Remarks	Text area			

Table 6 : Purchase Order Info

Caption	Input Type	Input Length	Mandatory	Validation
Purchase Note	Search		Yes	
Vendor	Drop down (Value comes from Vendor data table)		Yes	Purchase Notes item wise vendor will come. (Item mapping)
Reference No.	Text	100	Yes	Auto Reference No. will be generate (PO/BPSL/00/17)
Date	Date Picker	10	Yes	Date format will be checked
Vendor Reference No.	Text	100	No	
Vendor Reference Date	Date Picker	10	No	Date format will be checked
Deliver to	Text area		No	
Beneficiary/To	Text area		No	(Vendor Name and Address together)
Total	Text	11	Yes	Numeric value will be check

Table 7 : Purchase Order Item

Caption	Input Type	Input Length	Mandatory	Validation
Item	Drop down (Value comes from Item data table)		Yes	
HS Code	Drop down (Value comes from Item data table)			Purchase Note currency is not BDT (Foreign)
Quantity	Text	10	Yes	Numeric value will be check
Unit Price	Text	64,4	Yes	Numeric value will be check

`

Table 8 : Receive Info

Caption	Input Type		Input Length	Mandatory	Validation
Purchase Order	Search			Yes	
Reference No.	Text		100	Yes	Auto Reference No. will be generate (BPSL/IR/ 00/17)
Date	Date Picker		10	Yes	Date format will be checked
Challan No.	Text		100	Yes	
Challan Date	Date Picker		10	Yes	Date format will be checked
VAT Challan No.	Text		100	No	
VAT Challan Date	Date Picker		10	No	Date format will be checked
GRN No.	Text		100	No	Auto GRN No. will be generate
GRN Date	Date Picker		10	No	Date format will be checked
LC No.	Text		100	No	
LC Date	Date Picker		10	No	Date format will be checked
LC Amount	Text		11	No	Numeric value will be check
Port of Shipment	Text		100	No	
Currency	Label				It will come from PO
BDT Rate	Text		11		Numeric value will be check
Received By	Drop down (Value comes from User data table)			Yes	
Received Date	Date Picker		10	Yes	Date format will be checked
Destination Port Name	Text		11		
Date of Arrival at Port	Date Picker		10	No	Date format will be checked
Date of Release From The Port	Date Picker		10	No	Date format will be checked
Date of Arrival At Warehouse	Date Picker		10	No	Date format will be checked

Warehouse Location	Drop down List			Yes	
Remarks	Text area				

Table 9 : Item Receive Info

Caption	Input Type	Input Length	Mandatory	Validation
Item	Drop down (Value comes from Item data table)		Yes	
Serial No	Text	100	Yes	
Model List	Drop down List		Yes	If Generator then Yes
Engine No	Text	100	No	
Alternator Serial No	Text	100	No	
Quantity	Text	10	Yes	Numeric value will be check
Unit Price	Text	64,4	Yes	Numeric value will be check
Total Price	Text	64,4	Yes	Auto calculate
Total (BDT)	Text	64,4	Yes	Auto calculate

Table 10 : Sale Info

Caption	Input Type	Input Length	Mandatory	Validation
Sale Option	Drop down (Value Booking, Sample and Direct)		Yes	
Booking /Sample Issue Reference No.	Search			
Reference No.	Text	100	Yes	Auto Reference No. will be generate (BPSL / SEL/00/17)
Customer	Search		Yes	There will be an option to add Customer
Sale By	Drop down (Value comes from User		Yes	

	data table)			
Sale Date	Date Picker	10	Yes	Date format will be checked
Work Order No.	Text	100	Yes	
Work Order Date	Date Picker	10	Yes	Date format will be checked
Currency	Drop down (Value comes from Currency data table)		Yes	
Rate (BDT)	Text	11	Yes	
Item Send Option	Drop down (Value Courier and Beacon Own)		Yes	
Courier Company	Drop down (Value comes from Courier data table)		No	
Courier Ref. No.	Text	100	No	
Weight	Text	100	No	
Probable Delivery Date	Date Picker	10	Yes	Date format will be checked
Indenting Sale	Checkbox		No	If Checked then Commission Amount is needed. This type of sale will not under FIFO method only commission amount will be added in sale
Commission Amount	Text	11	Yes	Numeric value will be check.
VAT Inclusive Status	Checkbox		No	
AIT Inclusive Status	Checkbox		No	
VAT	Text	11	No	Numeric value will be check.
TAX	Text	11	No	Numeric value will be check.
Discount	Text	11	No	Numeric value will be check.
Advance	Text	11	No	Numeric value will be check.
Remarks	Text area			

Table 11 : Sale Item Info

Caption	**Input Type**	**Input Length**	**Mandatory**	**Validation**

`

Item	Drop down (Value comes from Item data table)		Yes	
Quantity	Text	10	Yes	Numeric value will be check
Serial No	Search		Yes	If the selected item has warranty.
Warranty	Drop down (Value comes from Warranty data table)		Yes	
Total Price	Text	64,4	Yes	Auto calculate
Total (BDT)	Text	64,4	Yes	Auto calculate

Chapter 4

Results

The current ratio of the company shows reduced investment every year marginally, it helps to save time and money, improves ordering accuracy, increase efficiency and productivity, present more balanced warehouse. This is also indicated that the company is recovering sales info quickly and debtor turnover & inventory turnover is good.

`

Chapter 5

Conclusion

To conclude that in inventory management system the stores department is concerned with the receiving of materials, storing and issuing to the production department. Regarding purchase department, it will purchase item from local market or export from foreign market. The sale procedure of the company move for localization, this greatly influence on easy procurement of materials in time and also reduces ordering and carrying cost of company. It will helps to increase the profit of the company. In summary

- Accuracy on output
- This project has been launched to replace the conventional manual application process.
- Such a platform has a potential to revolutionize by maximizing service effectiveness and & value and minimizing cost & time.

`

References

[1] S. Purohit, “Inventory Control System for calculation and ordering” Overall UML, 2016.

[2] A, Maharjan, “Final Year Project on Inventory Management System,” pp. 80–87, Sep. 2016.

[3] FKP, “– OOICHOONKHENG,” CD–9728, Sep. 2017.

`

Appendix A

Few Line of Source Codes:

```
<? Php if (! defined ('BASEPATH')) exit ('No direct script access allowed');

Class purchase_note extends CI_Controller {

Function __construct ()

   {

      parent::__construct ();



                  $this->output->set_header ('Last-Modified:'.gmdate ('D, d M Y H: i:
s’).'GMT');

                  $this->output->set_header('Cache-Control: no-store, no-cache, must-
revalidate');

                  $this->output->set_header ('Cache-Control: post-check=0, pre-check=0',
false);

                  $this->output->set_header ('Pragma: no-cache');

                  $this->load->model ('purchase_note_model', '', TRUE);

                  $this->load->model ('common_model', '', TRUE);

                  $this->load->model ('item_model', '', TRUE);

         }



         Function view ($menu_group, $menu_cat)

         {
```

`

```
                $data = array (
                                        'menu_group'=> $menu_group,
                                        'menu_cat'=> $menu_cat,
                                        'Pages'=> 'purchase_note/pages/grid',
                                        'per_page' => $this->config->item ('per_pagess')
                                );
                $this->load->view ('grid_layout', $data);
        }
        Function grid ()
        {
                $this->load->model ('purchase_note_model', '', TRUE);
                $pagenum = $this->input->get ('pagenum');
                $pagesize = $this->input->get ('pagesize');
                $start = $pagenum * $pagesize;

                $result=$this->purchase_note_model->get_grid_data ($this->input->get ('filterscount'), $this->input->get ('sortdatafield'), $this->input->get ('sortorder'), $pagesize, $start);

                $data [] = array (
                  'TotalRows' => $result ['TotalRows'],
                  'Rows' => $result ['Rows']
```

```
`

                );

                Echo json_encode ($data);

        }

        Function from ($add_edit='add', $id=NULL, $editrow=NULL)

        {



                $result=array ();

                $item_info=array ();



          If ($add_edit == 'edit') {

                        $result = $this->purchase_note_model->get_pn_info ($id);

                        $item_info   =   $this->purchase_note_model->get_pn_info_details
($result->id);



                }



                $data = array (

                          'Option' => '',

                          'add_edit' => $add_edit,

                          'Result' => $result,

                          'item_info' => $item_info,
```

`

```
                'department_list' => $this->purchase_note_model->get_parameter_data ('ref_division', 'name', array ('sts' => 1)),

                'currency_list' => $this->purchase_note_model->get_parameter_data ('ref_currency', 'name', array ('sts' => 1)),

                'item_category_list' => $this->purchase_note_model->get_parameter_data ('item_category', 'name', array ('sts' => 1,’id! =' => 1)),

                'item_sub_category_list' => $this->purchase_note_model->get_parameter_data ('item_sub_category', 'name', array ('sts' => 1)),

                'item_list' => $this->purchase_note_model->get_parameter_data ('item', 'name', array ('sts' => 1)),

                'Id' => $id,

                'Pages'=> 'purchase_note/pages/form',

                'Editrow' => $editrow

                );

            $this->load->view ('purchase_note/form_layout', $data);

        }


        Function add_edit_action ($add_edit=NULL, $edit_id=NULL)

        {

            $text=array ();

            If ($this->session->userdata ['user'] ['login_status']) {
```

```
`

            $id=$this->purchase_note_model->add_edit_action      ($add_edit,
$edit_id);

        }

        Else {

            $text [] ="Session out, login required";

        }



        $Message='';

        If (count ($text) <=0) {

            $Message='OK';

            $row=$this->purchase_note_model->get_add_action_data ($id);

        } else {

            For ($i=0; $i<count ($text); $i++)

            {

                If ($i>0) {$Message. =',';}

                $Message. =$text [$i];

            }

            $row [] ='';

        }



        $var =array ();
```

`

```
            $var ['Message'] =$Message;

            $var ['row_info'] =$row;

            Echo json_encode ($var);

      }

      Function get_sub_cat () {

        $data=array ();

        $result = $this->item_model->get_sub_cat_by_category ($this->input->post ('category_id'));

        Foreach ($result as $value) {

          $data [] = array (

            'Value' => $value->id,

            'Label' => $value->name

          );

        }

             Echo json_encode ($data);

      }

      Function get_item () {

        $data=array ();

        $result = $this->purchase_note_model->getitemBySubCategory ($this->input->post ('category_id'), $this->input->post ('sub_category_id'));

      Foreach ($result as $value) {

          $data [] = array (
```

`

```
                    'Value' => $value->id,

                    'Label' => $value->name

                );

            }

                     Echo json_encode ($data);

        }

    Function getItemCategory ( ) {

        $result = $this->purchase_note_model->getitemcategory ();

               $status = 'ok';

        $data   = array (

            'Status' => $status,

            'List' => $result

        );

        Echo json_encode ($data);

    }

    Function ajaxgetitemSubCat ($catid = NULL, $id = NULL, $idserial = NULL) {

            $this->load->helper ('form');

        $result    =    $this->purchase_note_model->getiSubcategoryByCategory    ($catid,
$idserial);

        Echo json_encode ($result);

    }
```

`

```
   Function ajaxgetitem ($catid = NULL, $subcatid = NULL, $id = NULL, $idserial =
NULL) {
       $this->load->helper (‘form’);
       $res = $this->purchase_note_model->getitemBySubCategory ($catid, $subcatid,
$idserial);
       Echo json_encode ($res);
   }
          Function detail ($id=NULL)
          {
                   $add_edit='view';
                   $result=array ();
                            $this->load->model ('purchase_note_model', ", TRUE);
                            $result=$this->purchase_note_model->get_pn_info ($id);
                            $result2=$this->purchase_note_model->get_pn_info_details ($id);
                            $data = array (
                                       'Result'=>$result,
                                       'result2'=>$result2,
                                       'Pages'=> 'purchase_note/pages/details'
                              );
                   $this->load->view ('purchase_note/form_layout', $data);
          }
          Function get_pn_detail ($id=NULL, $type=NULL)
```

`

```
        {
                $add_edit='view';
                $result=array ();
                //$id= $this->input->post ('id');
                        $this->load->model ('purchase_note_model', ", TRUE);
                        $result=$this->purchase_note_model->get_pn_info ($id);
                        $result2=$this->purchase_note_model->get_pn_info_details ($id);
                        $data = array (
                                  'Result'=>$result,
                                  'result2'=>$result2,
                                  'Type'=>$type,
                                  'Pages'=> 'purchase_note/pages/pn_details'
                         );
                $this->load->view ('purchase_note/form_layout', $data);
        }
        Function approve_reject_action ($id=NULL, $type=NULL)
        {
                $text=array ();
                If ($this->session->userdata ['user']['login_status']){
                        $id=$this->purchase_note_model->approve_reject_action      ($id,
$type);
                }
`

        Else {

            $text [] ="Session out, login required";

        }

        $Message=";

        If (count ($text) <=0) {

            $Message='OK';

            $row=1;

        } else {

            For ($i=0; $i<count ($text); $i++)

            {

                If ($i>0) {$Messaderge. =',';}

                $Message. =$text [$i];

            }

            $row [] =";

        }

        $var =array ();

        $var['Message']=$Message;

        $var ['row_info'] =1;

        Echo json_encode ($var);

    }
```

`

```
    Function show_reject_message ()

    {

        $results=$this->purchase_note_model->show_reject_message      ($this-
>input->post ('id'));

        $STR='<table   class="service_style"   style="border-collapse:   collapse;
width: 580px ;">

            <Thead>

                <TR>

                    <Th class="center-align">SL</Th>

                    <Th class="center-align">Reject Message</Th>

                    <Th class="center-align">Status By</Th>

                    <Th class="center-align">Status Date</Th>

                </TR>

            </Thead>

            <Tbody>';

        $i=0;

        Foreach ($results as $result) {

            $i++;

            $str. ='<tr>

                    <td  style="border:  1px  solid  #CCC;  margin-left:
5px;" class="center-align">'.$i.'</td>

                    <td  style="border:  1px  solid  #CCC;  margin-left:
5px;" class="center-align">'. $result->reject_message.'</td>
```

`

```
                                        <td style="border: 1px solid #CCC; margin-left:
5px;" class="center-align">'. $result->e_by_name.'</td>

                                        <td style="border: 1px solid #CCC; margin-left:
5px;" class="center-align">'. $result->e_dt.'</td>

                                </TR>';

                   }

          $str. ='</tbody>

                   </table>';

                Echo $STR;

        }

        Function print_purchase_note ($id) {

                If ($id! =" ){

                        $result=array ();

                        $result2 =array ();

                        $this->load->model ('purchase_note_model', '', TRUE);

                        $result=$this->purchase_note_model->get_pn_info ($id);

                        $result2=$this->purchase_note_model->get_pn_info_details_pdf
($id);



                        $STR="Select * from upr_config where id=1";

                        $project_sett=$this->db->query ($STR) ->row ();

                        $data ['result'] = $result;
```

```
`

                          $data ['result2'] = $result2;

                     $data ['project_sett'] = $project_sett;

                          $this->load->view         ('purchase_note/pages/pdf_purchase_note',
$data);

            } else {

                    Echo 'Not Found.';

            }

  }

  Function delete_action ($d_v=NULL)

   {

      $r=$this->purchase_note_model->check_event_status                   ($this->input->post
('deleteEventId'), 0);

      If ($r==0) {

         $jTableResult = array ();

         $jTableResult ['status'] = "error";

         $jTableResult ['errorMsgs'] = ‘Sorry!!! Entry Already Deleted';

         Echo json_encode ($jTableResult);

      }

      Else

      {

         $id=$this->purchase_note_model->delete_action ();

         $jTableResult = array ();
```

```
`

      If ($id>0) {

      $jTableResult ['status'] = "success";

      $jTableResult ['errorMsgs'] = 0;

                  } else {

                  $jTableResult ['status'] = "error";

      $jTableResult ['errorMsgs'] = ‘Sorry!!! Entry Already Deleted';

                  }

      Echo json_encode ($jTableResult);

    }

  }

  Function send_to_checker ($d_v=NULL)

  {

    $r=$this->purchase_note_model->check_event_status ($this->input->post ('id'),
1,"event_sts='1' or event_sts='4' or event_sts='6' or event_sts='8'");

    If ($r==0) {

      $jTableResult = array ();

      $jTableResult ['status'] = "error";

      $jTableResult ['errorMsgs'] = ‘Sorry!!! Entry Already Send';

      Echo json_encode ($jTableResult);

    }

    Else

    {
```

```
`

            $id=$this->purchase_note_model->send_to_checker ();

            $jTableResult = array ();

            $jTableResult ['status'] = "success";

            $jTableResult ['errorMsgs'] = 0;

            Echo json_encode ($jTableResult);

        }


    }


}

?>
```