

A project report on

## **HIVE TECHNOLOGY**

Submitted in partial fulfillment of the requirement  
For the award of the degree

### **MASTER OF COMPUTER APPLICATIONS** Of



Visvesvaraya Technological University  
Belgaum, Karnataka

By

**SUBHIKSHA U**

**1CY18MCA66**



**CMR INSTITUTE OF TECHNOLOGY**  
**132, IT Park Road, Kundalahalli, Bangalore-560037**  
**2019-2020**

A project report on

## **HIVE TECHNOLOGY**

Submitted in partial fulfillment of the requirement  
For the award of the degree

### **MASTER OF COMPUTER APPLICATIONS**

Of



Visvesvaraya Technological University  
Belgaum, Karnataka

By

**SUBHIKSHA U**

**1CY18MCA66**



**CMR INSTITUTE OF TECHNOLOGY**  
**132, IT Park Road, Kundalahalli, Bangalore-560037**  
**2019-2020**

A project report on

**HIVE TECHNOLOGY**

Submitted in partial fulfilment of the requirement  
for the award of the degree

**MASTER OF COMPUTER APPLICATIONS**

of

Visvesvaraya Technological University  
Belgaum, Karnataka

By

**SUBHIKSHA U**

**1CY18MCA66**

Under the guidance of

**Internal Guide**

**Ms. Ashwini Patil**

Assistant Professor,

MCA Department

CMR Institute of Technology, Bangalore.

**External Guide**

**Mr. Selva Kumaran**

Project Manager,

SAN IT Solutions PVT.LTD

Bangalore.



**CMR INSTITUTE OF TECHNOLOGY**

**132, IT Park Road, Kundalahalli, Bangalore-560037**

**2019-2020**

**CMR INSTITUTE OF TECHNOLOGY**  
**Department of Master of Computer Applications**  
**Bangalore - 560 037**



***CERTIFICATE***

*This is to certify that the project work entitled*

**SAN IT SOLUTIONS**

*Submitted in partial fulfilment of the requirement  
for the award of the degree of*

*Master of Computer Applications of the  
Visvesvaraya Technological University, Belgaum, Karnataka bonafide work  
carried out by*

**SUBHIKSHAU**  
**1CY18MCA66**

*during the academic year 2019-2020.*

\_\_\_\_\_  
**Signature of the Guide**  
**Ms. Ashwini Patil**  
Assistant Professor,  
MCA

\_\_\_\_\_  
**Signature of the HOD**  
**Ms.Gomathi.T**  
HOD, MCA

\_\_\_\_\_  
**Signature of the Principal**  
**Dr. Sanjay Jain**  
PRINCIPAL, CMRIT

External Viva

Name of the Examiners  
1.  
2.

Signature with date

## DECLARATION

I, **SUBHIKSHA U**, student of 6<sup>th</sup> MCA, **CMR Institute of Technology**, bearing the USN **1CY18MCA66**, hereby declare that the project entitled “**HIVE TECHNOLOGY**” has been carried out by me under the supervision of External Guide **Mr. Selva Kumaran**, Project Manager, and Internal Guide **Ms. Ashwini Patil, Assistant Professor, Dept. of Master of Computer Applications** and submitted in the partial fulfillment of the requirements for the award of the Degree of Master of Computer Applications by the **Visvesvaraya Technological University** during the academic year 2019-2020. The reports has not been submitted to any other University or Institute for the award of any degree or certificate.

Place: Bangalore

SUBHIKSHA U

Date:

(1CY18MCA66)

## **ACKNOWLEDGEMENT**

I would like to thank all those who are involved in this endeavor for their kind cooperation for its successful completion. At the outset, I wish to express my sincere gratitude to all those people who have helped me to complete this project in an efficient manner.

I offer my special thanks to my external project guide Mr. Selva Kumaran Project Manager, San It Solutions Pvt. Ltd., Bangalore, and to my Internal Project guide Ms. Ashwini Patil, Department of MCA, CMRIT, Bangalore without whose help and support throughout this project would not have been this success.

I am thankful to Dr. SANJAY JAIN, Principal, CMRIT, Bangalore for his kind support in all respect during my study. I would like to thank Mr. Selva Kumaran, Project Manager, San IT Solutions Pvt. Ltd., Bangalore, who gave opportunity to do this project at an extreme organization Most of all and more than ever, I would like to thanks my family members for their warmness, support, encouragement, kindness and patience. I am really thankful to all my friends who always advised and motivated me throughout the course.

**SUBHIKSHA U**  
**(1CY18MCA66)**



## **SAN IT Solutions Pvt. Ltd.**

(An ISO 9001-2015 Certified Company)

**Date– Jan 12, 2020**

### **PROJECT COMPLETION LETTER**

We hereby confirm that **Ms. Subhiksha U** of your college CMR Institute of Technology with USN: 1CY18mca66 has successfully completed the Project at San IT Solutions Pvt. Ltd.

From: 13th January to 15th June

The Project based on .Net the title “**Big data joining and version control integration in business transaction**” under the guidance of Mr. Selva Kumaran, Project Guide, San It Solutions Pvt. Ltd, Bangalore 560041.

**For San It Solutions Pvt Ltd**

**HR Manager**

---

<b>S.NO.</b>	<b>Contents</b>	<b>Page No.</b>
<b>1.</b>	Introduction	
	<b>1.1</b> Project Description	1
	<b>1.2</b> Company Profile	4
<b>2.</b>	Literature Survey	5
	<b>2.1</b> Existing System and Proposed System	6
	<b>2.2</b> Feasibility Study	7
	<b>2.3</b> Tools and Technologies Used	8
	<b>2.4</b> Hardware and Software Requirements	10
<b>3.</b>	Software Requirement Specification	11
	<b>3.1</b> Functional Requirements	13
	<b>3.2</b> Non- Functional Requirements	15
<b>4.</b>	System Design	16
	<b>4.1</b> System Perspective	17
	<b>4.2</b> Context Diagram	18
<b>5.</b>	Detailed Design	
	<b>5.1</b> Use Case Diagrams	18
	<b>5.1.2</b> Use Case Diagram for Admin	18
	<b>5.1.3</b> Use Case Diagram to clerk	19
	<b>5.1.4</b> Use Case Diagram to main	20
	<b>5.1.5</b> Use Case Diagram to Clerk2	21
	<b>5.2</b> Sequence Diagrams	
	<b>5.2.1</b> Sequence Diagram for admin	22



	<b>5.2.2</b> Sequence Diagram for clerk	22
	<b>5.2.3</b> Sequence Diagram for clerk2	23
	<b>5.2.4</b> Sequence Diagram for main clerk	24
	<b>5.3</b> Collaboration Diagram	
	<b>5.3.1</b> Diagram for Admin	24
	<b>5.3.2</b> Collaboration for clerk	24
	<b>5.3.3</b> Collaboration for clerk 2	25
	<b>5.3.4</b> Collaboration for main clerk	26
	<b>5.4</b> Activity Diagrams	
	<b>5.3.1</b> Activity Diagram for admin	27
	<b>5.3.2</b> Activity Diagram for user	28
	<b>5.3.3</b> Tables	30
	<b>5.4</b> ER Diagrams	31
	<b>5.4.2</b> ER Diagram for chart flow	31
	<b>5.4.3</b> ER Diagram for main flow	32
<b>6.</b>	Implementation	33
	<b>6.1</b> Screen Shots	36
	<b>6.2</b> Coding	47
<b>7.</b>	Software Testing	48
<b>8.</b>	Conclusion	51
<b>9.</b>	Future Enhancements	52
<b>10.</b>	Bibliography	53
<b>11.</b>	User manual	54

# CHAPTER 1

## INTRODUCTION

### 1.1 PROJECT DESCRIPTION

The HIVE technology is used in the big data application to enhance the performance in the area of data storage and the big data joining. Currently the technology for the data storage and the joining are used by developers to make application more efficient and advantageous in the application. But technology which will support many big data feature in a single platform like HIVE is not introduced.

#### **Features**

CVS for storage:

The big data application will have mass data storage and the needed large space for storing. The HIVE based CVS can identify and merge the same attributes value together and save the total space together.

Indexing in application:

The application of big data will have multiple table and class based integrations. The attributes in the class and table will points the same objects but will be having different names. The feature of indexing in the application can mark the same attributes in the table and class

#### **Domain details**

This technology of the HIVE is implemented in the domain of KPO internal operations. Let us see how the head office of the KPO operates over the different work order received from the branch. If we take example for KPO for bank operations, the different banking activities are divided in the sprints each for managing the cash based transactions, the online, the loan and the details of transaction failed work management. Here the bank do not group under each bank based operation but each work will be given for the separate tracking number and the updates are

created on the basis this tracking number like work completed pending, etc. This work also may include other banking activities and some the works are chartable too.

### **Role of HIVE in the Domain**

The HIVE will be working to find and store the similar data which will be saved in the software banking operations. When the two or more branches of the bank are processing the banking activities there will be module or controls like cheque processing, check the authorisation of the cheque number, the MICR reader, the signature matching etc. The steps needed for these banking activities will be same but the parameters forwarded from the banks will be different. So with help of HIVE the different parameters will be brought together but the similar work and call together

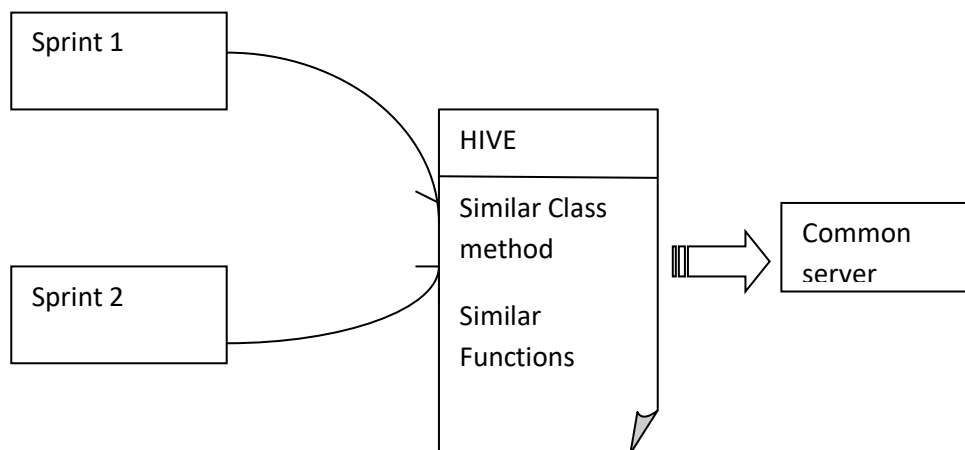


Fig 1.1: HIVE with sprint integrations.

### **MODULE DETAILS**

The application with HIVE is implemented for the KPO units. The KPO who work with large number of business operation the work will be divided, developed and completed in sprints. The HIVE will identify the similar work data and allocate the class methods for the processing. The module name in the KPO

- The client registration
- Clients work progress
- The work manager HIVE server module

- Administration module
- Domain report

#### The client registration

The Details of the client organisation registration is explained in the module. The work detail of each client will be different from the format of the other client who registered for the KPO activities. The detail of the work carried, the user assignment, the work manger, the back log management are also created along with the client registration module.

#### Clients work progress

This module operation will be different with the client, the domain where the KPO operations are used. The application uses the HIVE technology to for boosting the KPO work with more reliability and the processing. The details with the payment related steps are also added in the client work progress module. The cost will be calculated based on the work time and effort spends on each sprints.

#### The work manager HIVE server module

This module has network based interface to connect all the system with the server where the HIVE is implemented. The KPO based project will be carried in more than one system and the HIVE algorithm needed to fetch the data from the different system check for authorization, the error possibility and the building. The developers use separate machine for the HIVE building with better configuration setting. If the work data load is less than the developers will use same system too for the HIVE based building the project.

#### Administration module

The admin works are not linked with HIVE database. The work regarding the KPO work to carried, the client requirement analysis, find and assign the clerk for the 3each works, the payment generation, the payment collection, the fund distribution for the employees, KPO sub work process will be under administration module.

#### Domain report

The report generated based on the different work carried in the KPO units. Normally the domain report do not have HIVE algorithm implemented. But the values generated from the HIVE are collected for the report generation

## 1.2 COMPANY PROFILE

The SANIT is Bangalore based software company works in the E commerce and SEO business domain. The organisation is established in the year 2004 has been in the domain of Microsoft – JAVA android ad PHP domain application development. We have experience team who can find solution for E commerce based business requirements, the technical consulting works etc. We have expertise in the field of SEO and the product branding support.



### IT Consulting

We tailor make solutions which are unique to each clients needs and always deliver ahead of deadlines.



### Our Team

24\*7 Support by an experienced Team.  
Led by Director with Vast knowledge in  
Microsoft/Cisco Products.



### Assortment Of Clients

Experience with both SMB and Enterprise customers  
makes us unique.

Fig 1.2 the services of the SANIT

# **CHAPTER 2**

## **LITERATURE SURVEY**

### **2.1 EXISTING SYSTEM AND PROPOSED SYSTEM**

#### **2.1.1: Existing system**

In the existing system the developers use pool encryption and manual back up of the software application. This can reduce the total memory needed for the big data and cloud computing based data storage. But considering the large requirement of storage space needed for the big data the existing system is not enough and needed to go with more advanced features of big data storage technology and programming logic to reduce the complexity applications.

Limitation of the Existing system

- The traditional big data storage techniques are used for the data storage.
- Merging techniques for similar data are not used to store the data in the SQL server.
- The developers use the help of version control software to merge and store in the SQL.
- The separate data access is needed to get into the backup server data. The developers use separate server for the backup and the access to this server will be limited to the network users and users from the Developers side will not have any access to the server.

#### **2.1.2: Proposed system**

The technology of HIVE is implemented for the domain of KPO. Instead of taking the pooling based encryption the HIVE will find and merge of same code and encrypt. The advantage of HIVE is for the big data server where the update from the user side is bulky and outsized. Another advantage of HIVE is this technology can be used not only for the big data application and also for the business with medium size transaction flow. The branch of hive technology can access the data from the backup system and connect with the user system. So there is not needed separate access for the data accessing for the users. The develops use integrated pooling system to enable this feature.

## **Advantage**

Searching will be based on data structure format, the hive pointer will fast up the searching technique from the SQL server to the system

No extra space needed to store similar data in the HIVE. HIVES locate and compress together for the same data storage.

Developers can avoid third party software tools purchase and usage for the data compression techniques.

The data security of the HIVE server is high since the storage is all connected with SQL server based data storage.

## **2.2 FEASIBLE STUDY**

The existing organisation of KPO has working without HIVE based technology and now the developers got a opportunity to implement the new advances technology and the question regarding the implementation will rise. Whether the client can afford the new HIVE technology based KPO, is there any changes in the existing operation of the KPO after the HIVE also list of software needed for HIVE to serve and HIVE to application etc.

- The technical feasibility study
- The operational feasibility
- The cost feasibility

### **The technical feasibility study**

The list of supporting tools needed for HIVE based IDE development and also the KPO based modules development. The controls for the KPO and the controls for the HIVE a network based application is different so the integration units for the KPO and network oriented platform developers need to select the IDE with domain and network supporting features. Visual studio IDE is standard selection for this project since visual studio supports the network and a three tier application integrations.

### **The operational feasibility**

In this part developers needed to implement HIVE to the application without affecting the any existing operations. Most of the clerks of KPO will be used to the existing application and a new change with different operation style may delay the business progress. So the developers need to implement HIVE in each and every web pages of the software application where no new manner are introduced.

### **The cost feasibility**

The total cost in the new implementation must be under the client budget. The total cost included software purchase for supporting the HIVE, the server hire, maintenance for the server and the technical consultant hired to support. The business developer team are responsible for the cost feasibility study. The check lists made for the application development, the cost for special operations which are added for the architecture design / included for the HIVE are calculated

## **2.3 TOOLS AND TECHNOLOGY**

The tools and technology used in HIVE and KPO application is visual studio based IDE. The 2014 professional edition of visual studio the base IDE used for the software development and integrations. The professional edition has the controls needed for the domain based KPO application development and the HIVE operations too. The list of software tools used for application is

- Front end in ASP
- C#.net for HIVE and KPO
- SQL for server back up
- SVN
- Team viewer
- Class designer

### **Front end in ASP**

Active server page is the front end based development tool. The presentation layer techniques for the designing the HIVE based AJAX tool kit. the master page are created in the front end. AJAX is used to control the post back events occurs while retrieving the data in the HIVE platform.

### **C#.net for HIVE and KPO**



The software platform used for coding the operation in professional edition of visual studio is C#.net. Other language supported for visual studio VB and F#. For the more secure and the network oriented application language of C# will be an ideal choice. The developers created different units of class and methods with C# language for the KPO and HIVE networks for the C# implementations

#### SQL for server back up

The SQL server studio is the software tool used for the back end based storage and SQL query coding. The HIVE application needed SQL code to operate in precompiled format to fetch the data from the SQL and store in the structured form. The store procedure for storing the data in the pre compiled format, the normalization used for HIVE integrations are coded in the SQL server management studio.

#### SVN

The SVN is the software used for the version controlling feature in the KPO. Before the HIVE is implemented in the organisation the developers used the help of SVN software tools for saving the updates. So for the modules which are not included for the HIVE is still needed the help of SVN software application and this software tool is included for the application development kit.

#### Team viewer

The developer's unit are divided in HIVE based unit and KPO as we know. The users work area also can be located in different location or even in two far cities too. So the application uses the software of team view to connect, file transfer regarding the KPO to HIVE and even for the screen sharing with network or admin users

#### Class designer

The software used to mark the list of class used in the software. The developers will create multiple class , write the code for domain or network based operations and connect with the object . So when new updates are made in the application the developers will use the class designer software to view name of the class and how each class are created. In this software the developers can directly create new class by right clicking and create class. That is not needed to go main page of visual studio and create class. And this created class will be auto created in the visual studio page too.

## 2.4 HARDWARE AND SOFTWARE REQUIREMENTS.

### Hardware Requirements

RAM ( <b>without HIVE in KPO</b> )	2 GB
Hard Disk ( <b>without HIVE KPO</b> )	250 GB
Server ( <b>without HIVE KPO</b> )	IIS, HTTP caching server
Processor ( <b>without HIVE KPO</b> )	Pentium 4

RAM ( <b>HIVE</b> )	128 GB
Hard Disk ( <b>HIVE</b> )	16 TB
Server ( <b>HIVE</b> )	IIS, TFS

## Software requirements

Test ( <b>HIVE and KPO</b> )	MTM
Front end ( <b>HIVE and KPO</b> )	ASP.NET
Middleware ( <b>HIVE and KPO</b> )	C#.NET
IDE ( <b>HIVE and KPO</b> )	Visual studio
Back end ( <b>HIVE and KPO</b> )	SQL server 2008 R2
Processor ( <b>HIVE and KPO</b> )	Pentium 4

# CHAPTER 3

## SOFTWARE REQUIREMENTS SPECIFICATION

### 3.1 USERS TYPES

The users of the application are developed on the basis of user access in the module. This application of HIVE and KPO has the technical supporting team in HIVE based operations and the non-technical teams in the KPO based. The profile of these users to each module is as follows

- KPO clerk
- The Network user
- Client Side users
- Admin panel

KPO clerk

Module	Users profile
Client Module	View the request received from the client, create order details
Work manager	Permission to update and delete the work manager profile
Server network	Not permitted to work
Administration module	Permission denied

The Network user

Module	Users profile
Client Module	Server accessing permission. IP address, the team viewer access are permitted
Work manager	NA

Server network	Complete access to the main server and HIVE based local server data access
Administration module	Read the only mode for network based admin instructions

#### Client Side users

<b>Module</b>	<b>Users profile</b>
Client Module	Complete access in the client based operations. Create order, update the requirement , delete the order etc.
Work manager	View the update created by the clerk. Not permitted to direct update of work manager ,but permitted to make update requirement.
Server network	Not permitted
Administration module	Not permitted to client , only for employee within the organisation

#### Admin panel

<b>Module</b>	<b>Users profile</b>
Client Module	Read instruction, update delete not given for the admin users
Work manager	Complete access given to the work manager
Server network	Admin network user bases operations only
Administration module	Complete access given

## 3.2 FUNCTIONAL REQUIREMENTS

The functional requirement number 1:

The functional requirement name: Auto gen work order Reference number

Description: different work order must have unique reference number

Input: Client name,

Process: Fetch the client name pass to the store procedure to generate reference number

Output: display the generated reference number.

The functional requirement number 2:

The functional requirement name generates HIVE stack ID for the work order

Description: allocate the HIVE memory to the created work order reference ID, different work order will have save attribute and will be saved in the HIVE stack

Input: NA

Process: call algorithm (data structure) to create the HIVE stack

Output HIVE stack generated.

The functional requirement number 3:

The functional requirement name: find work order with same attribute.

Description: link the HIVE stack with multiple work order by keeping the reference ID as pointer

Input: NA

Process: call algorithm module to find similar attribute create reference ID and pass to HIVE based data set and node will be loaded with other attributes of the application.

Output: HIVE based integration

The functional requirement number 4:

The functional requirement name: Work break down work in HIVE

Description: User clicks the work break down and assigned to different employees. The HIVE based data must remain same but accesses different machine, and updates made by the user must be access in the server and the HIVE database.

Input: Check box active break down

Process: create work order updates

Output: reference for the sub work order created

The functional requirement number 5: cost for the work order

The functional requirement name: calculation within the HIVE based data and generates the result

Description: fetch the data from the SQL, the HIVE and use for the intermediate class to generate the output.

Input: work order number

Process: calculate the down time needed, client budget, total fee created.

Output: generate amount payable to the selected work order.

### **3.3 NON FUNCTIONAL REQUIREMENTS**

The requirement collected made to improve the KPO – HIVE implementation. The last study before the developing the application and this non-functional requirement has the steps to be taken to improve the software in the areas of

#### **Security**

The security of the data access is high since the HIVE working in server base integrations, the access to the server is only given with user ID and password and HIVE do not store the data in cookies since the size of data saved in large and not can be included in the cache memory. HIVE can support distributed cache where the data are saved in cache memory but if the developers avoid storing the data in the D distributed cache then security can be high.

### **Flexibility**

The KPO work with different client and their requirements and there the work profile are different. Where the HIVE needed to find the common interoperability in these work profile and find save in the single HIVE stack. The HIVE flexibility is stored in the steps of finding and storing the same location and use for the calculation needed for the KPO applications.

### **Reliability**

The relativity stands for how sharp is the data access from the HIVE. When the client makes request to access the data the application needed to fetch the data from single HIVE stack and also the data from the different SQL where HIVE is not implemented. A developer has programmed to give high reliability in finding and storing the data for the KPO based operations.

### **Data integrate ability**

Writing the code to generate unique number, connect with HIVE stack with pointer. If the user makes more updates with respect to client's requirement the complexity in connecting with HIVE application will become challenging. So to improve the data integrity of the application the developers programmed HIVE with objects which holds the update stack pointers, so when the new updates or table is connected the object will have the point to link with the newly added stack and thereby improving the software integrity.



# CHAPTER 4

## SYSTEM DESIGN

### 4.1 SYSTEM PERSPECTIVE

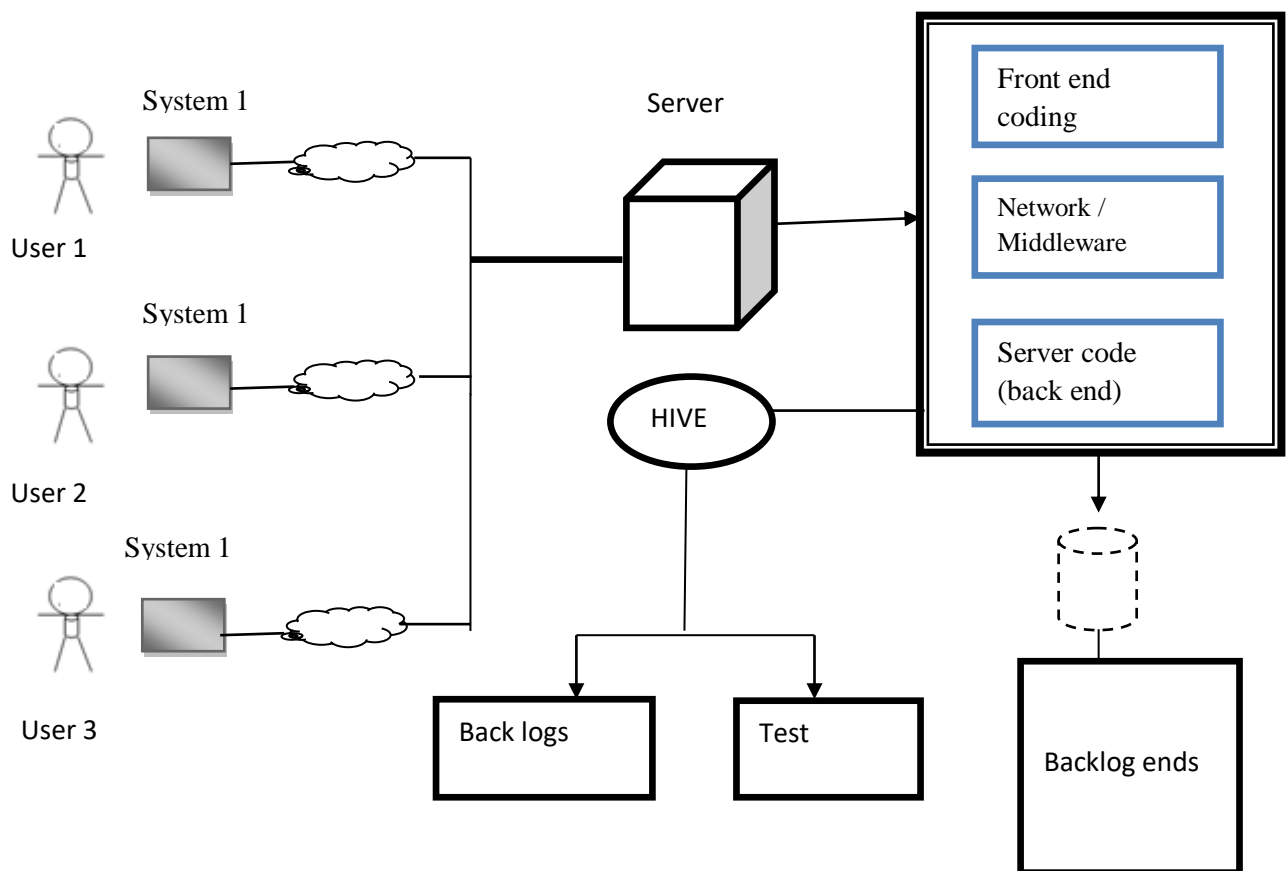


Fig 4.1: architecture diagram for HIVE

4.2 Context Diagram

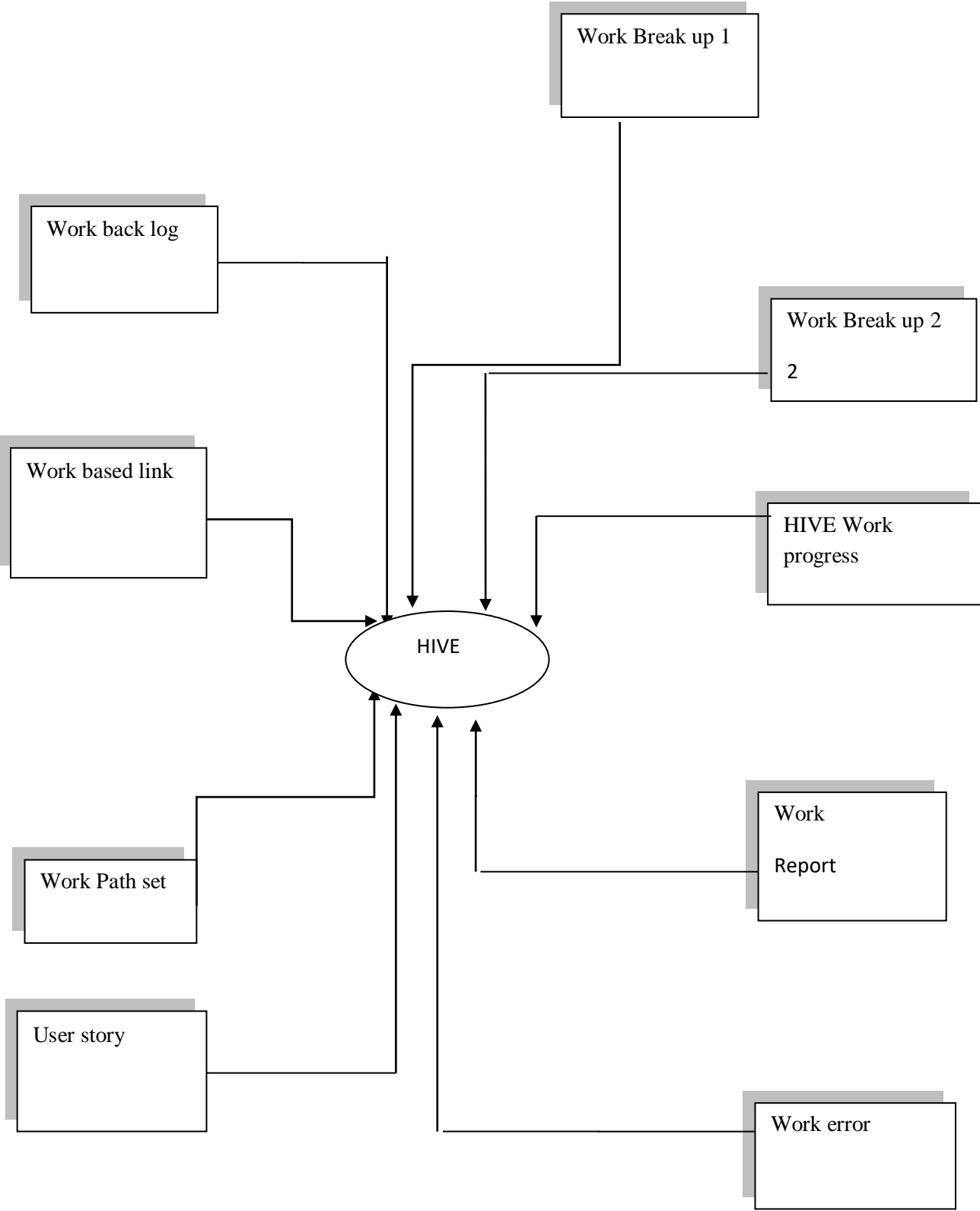


Fig 4.2 Context Diagram

# CHAPTER 5

## DETAILED DESIGN

### 5.1 USE CASE DIAGRAM

#### 5.1.1 Use case Diagram for Admin

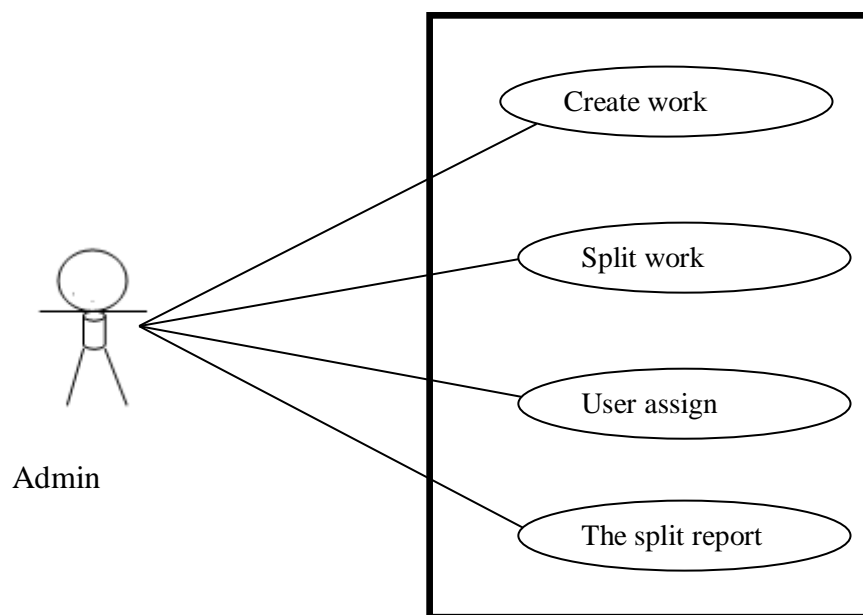


Fig 5.1.1 Admin Use Case Diagram for work divides and assign the user in the KPO unit

The admin has to create a new product details such as the title of the product, the description regarding the product work profile, the risk rank the work has to be assigned. The admin can also link the excel sheet or other document with the parent product.

### 5.1.2 Use case Diagram to the Clerk

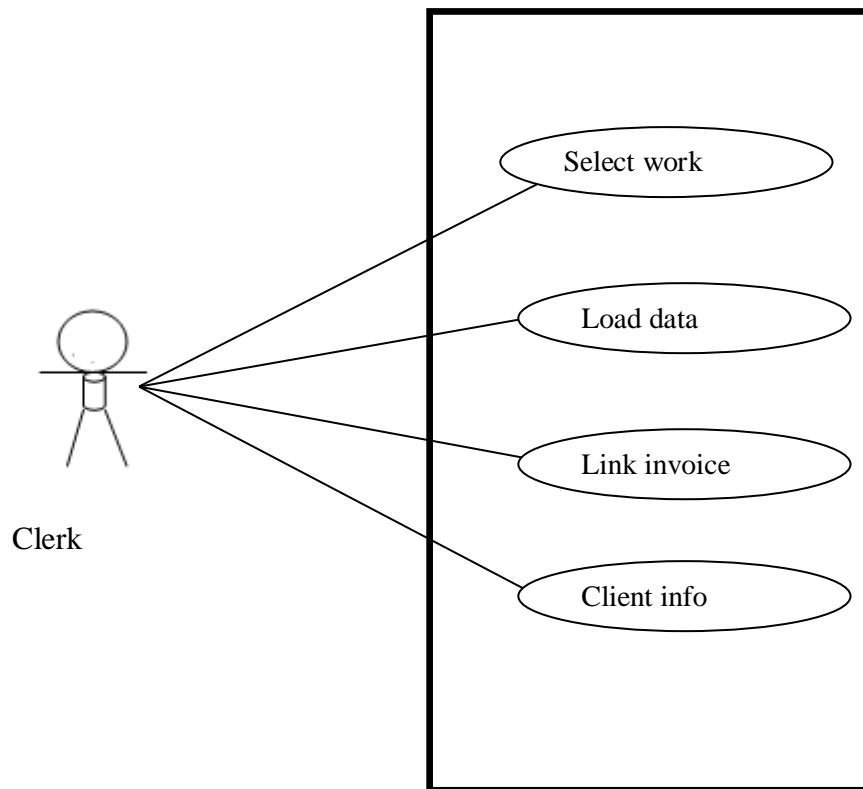


Fig 5.1.2: use case for the clerk in client and data load

The project status is been approved by the admin and the client the next work status is to create the sprint area and load data, each work under the work. The store board is the section where the requirement given by the client

### 5.1.3 Use case Diagram to the main clerk

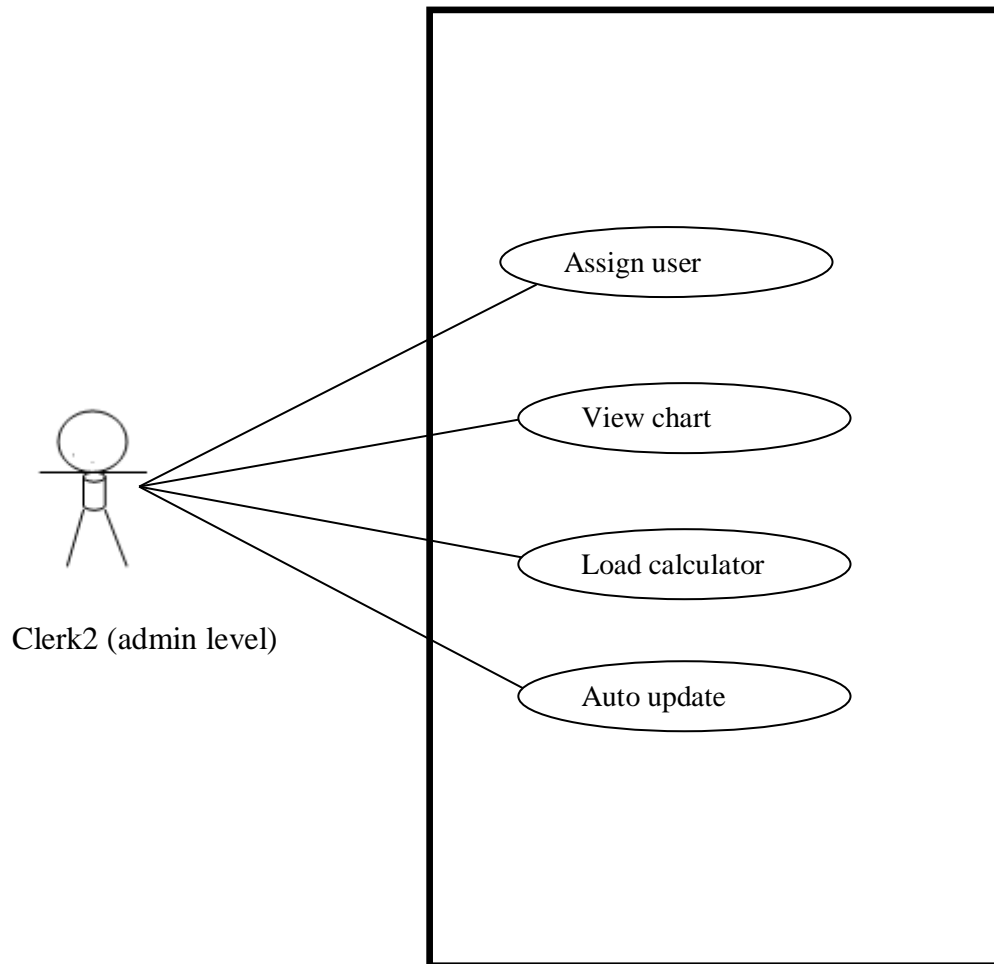


Fig 5.1.3: Use case Diagram for main clerk

### 5.1.4 Use case Diagram to the clerk 2

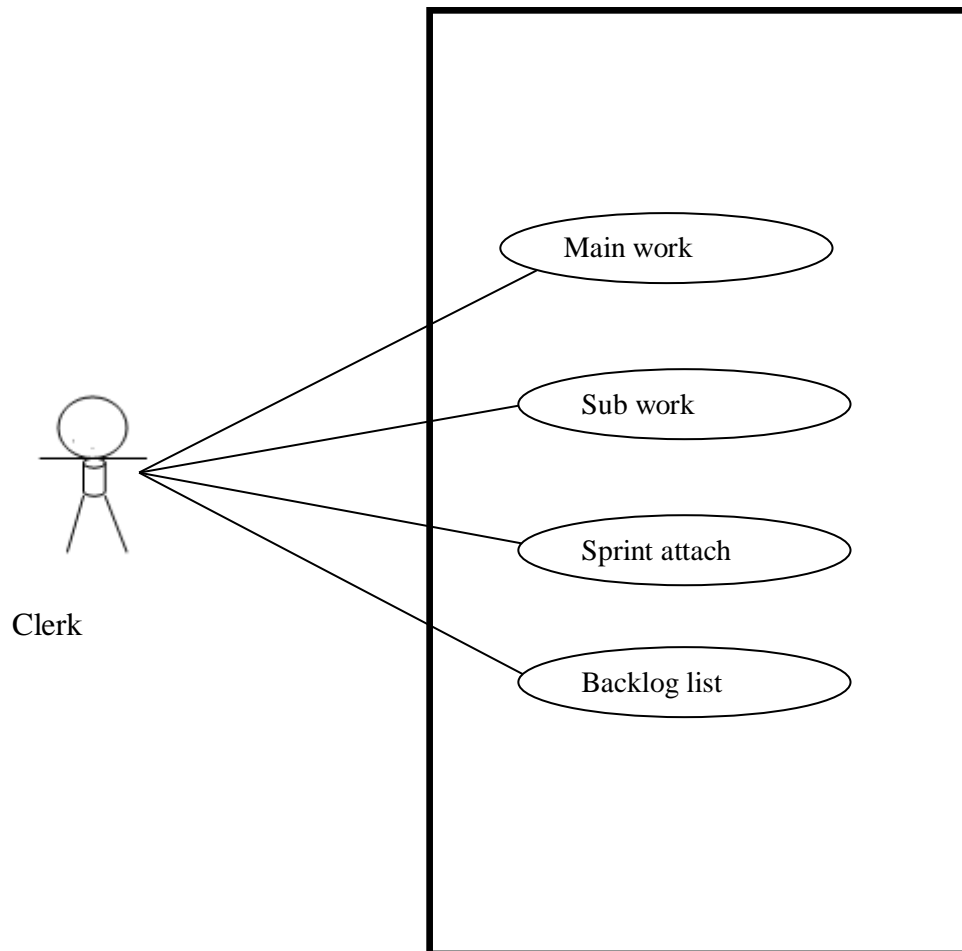


Fig 5.1.4: Use case Diagram for Clerk 2

These users are capable of the managing the current sprint status. The updating each sprint status, arrainging the sprint list according to the high priority sprint on top and low at the down. The feedback gathered from the client will be created as new feedback from where the error or issues will be handled to resolve the situation from the second time occurrence.

## 5.2 Sequence Diagram

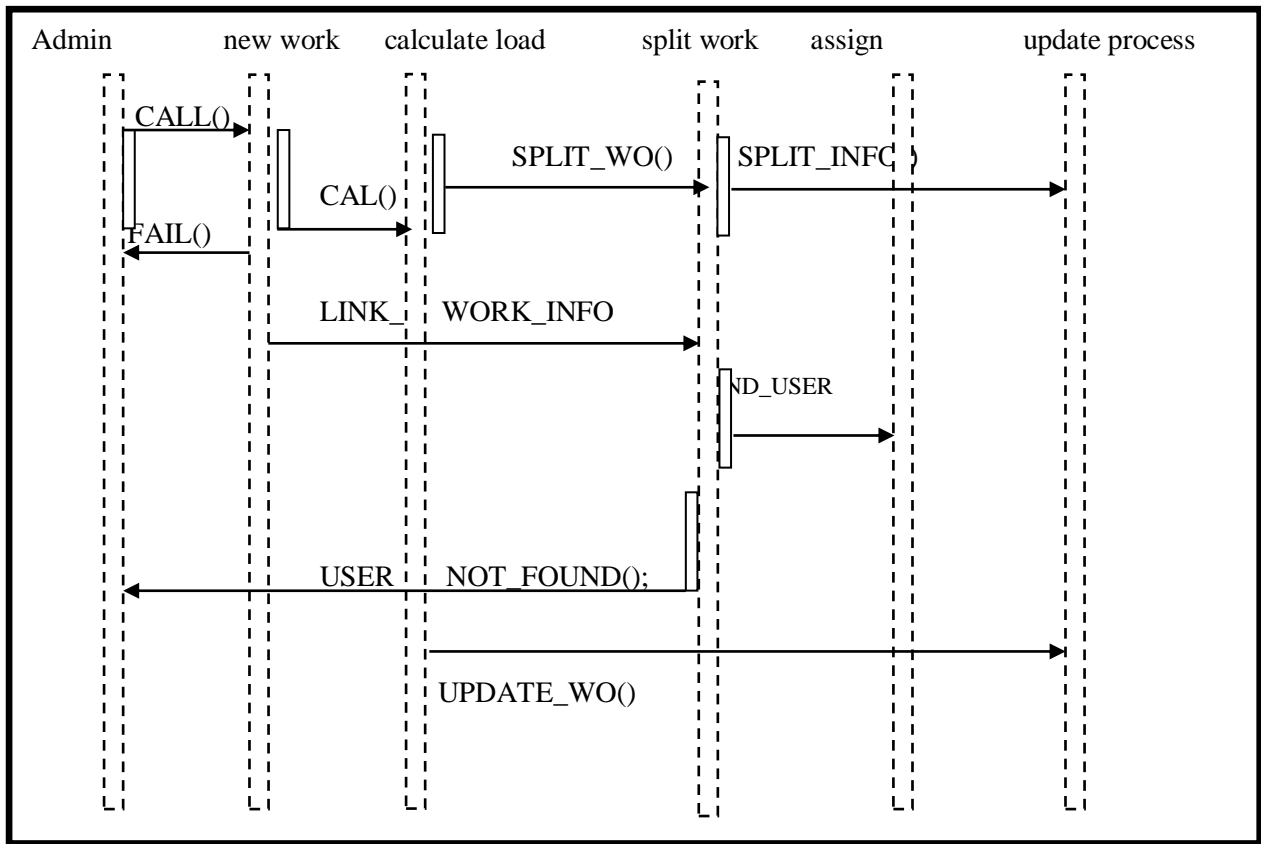


Fig 5.2.1: Sequence Diagram for work integration and divide the work based on work load calculated

The admin is allowed to do changes to the auto generated the fields. If the transaction is for the first time the user has to enter all the values and store in the data base for the next entry.

### 5.2.2 Sequence Diagram for Clerk

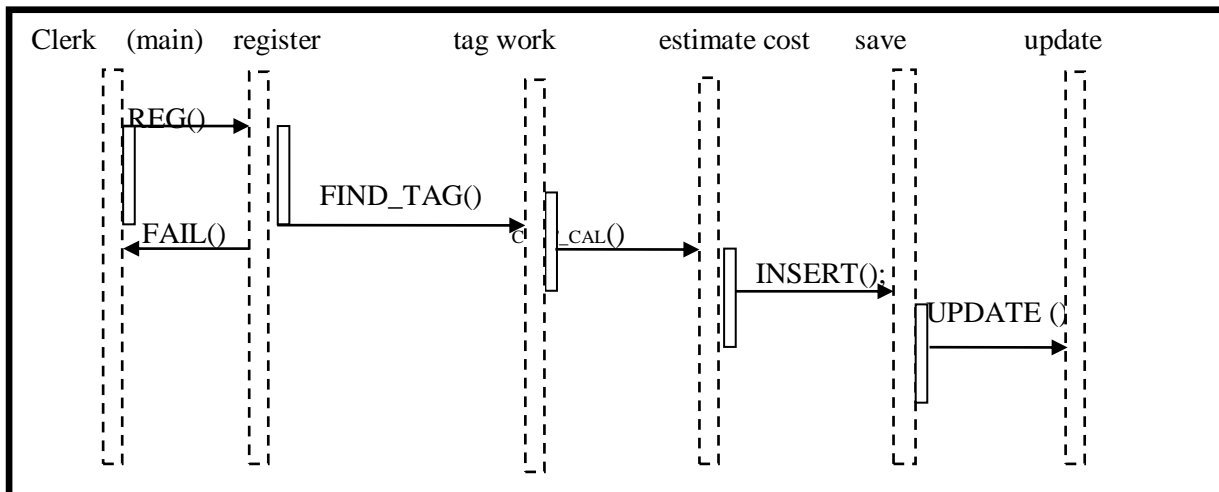


Fig 5.2.2: Sequence Diagram for Clerk in tagging the users and the calculating the estimated cost

When a client gets a new work details he has to study the effect of the sprint (the information collection, the risk can occur) etc. the story board is the customer requirement over the existing problem.

### 5.2.3 Sequence Diagram for Clerk2

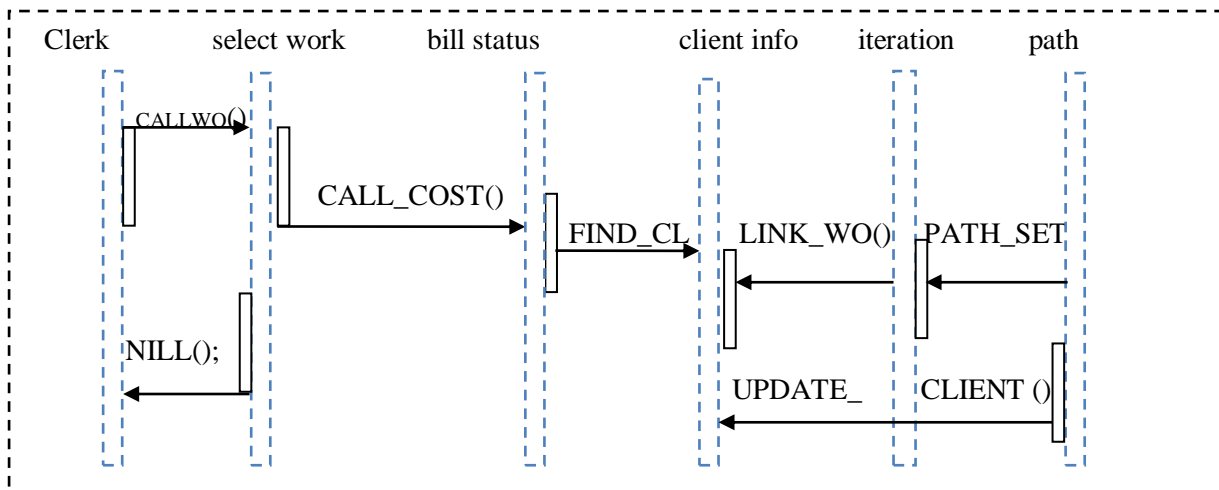


Fig 5.2.3: Sequence Diagram for creating the iteration (work flow path) for the selected work

The work of clerk 2 will deals with the customer feedback and handling the client info or technical issues has occurred during the sprint activities.



### 5.2.4 Sequence Diagram for Main Clerk

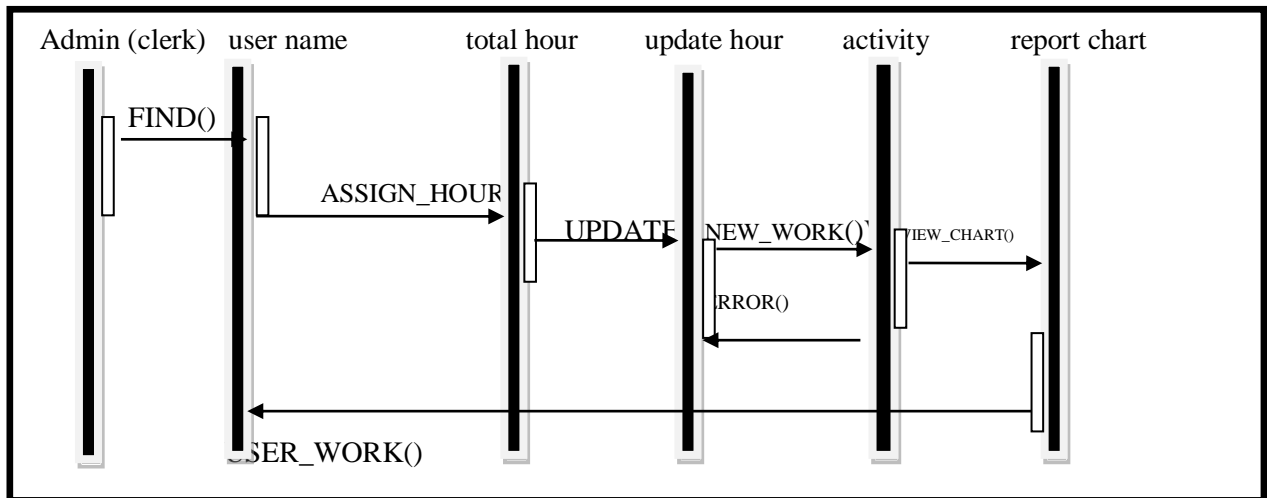


Fig 5.2.4: Sequence Diagram for Main Clerk in finding the total hour and cost estimation for the selected path.

Depending on the user story or the task in the work the admin the main clerk will create the sprints. The main clerk will create new users to the work and also enter the working capacity of each user towards the project. The main clerk can assign the work task to each user depending in the working capacity of the user.

### 5.3 Collaboration Diagram

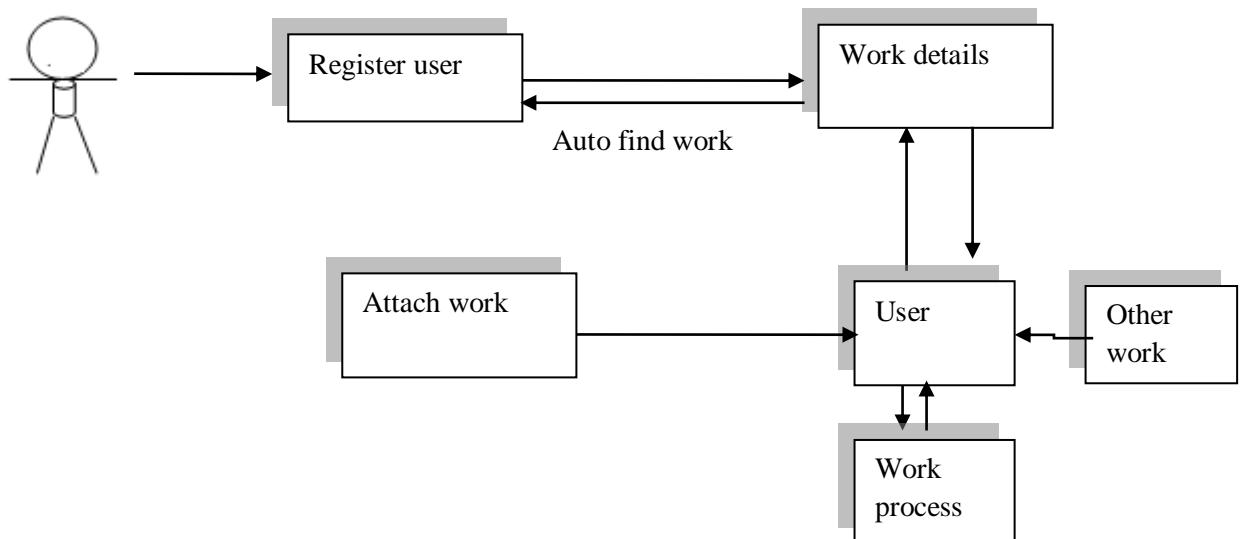


Fig 5.3.1 Collaboration Diagram for Admin

### 5.3.2 Collaboration Diagram for Clerk

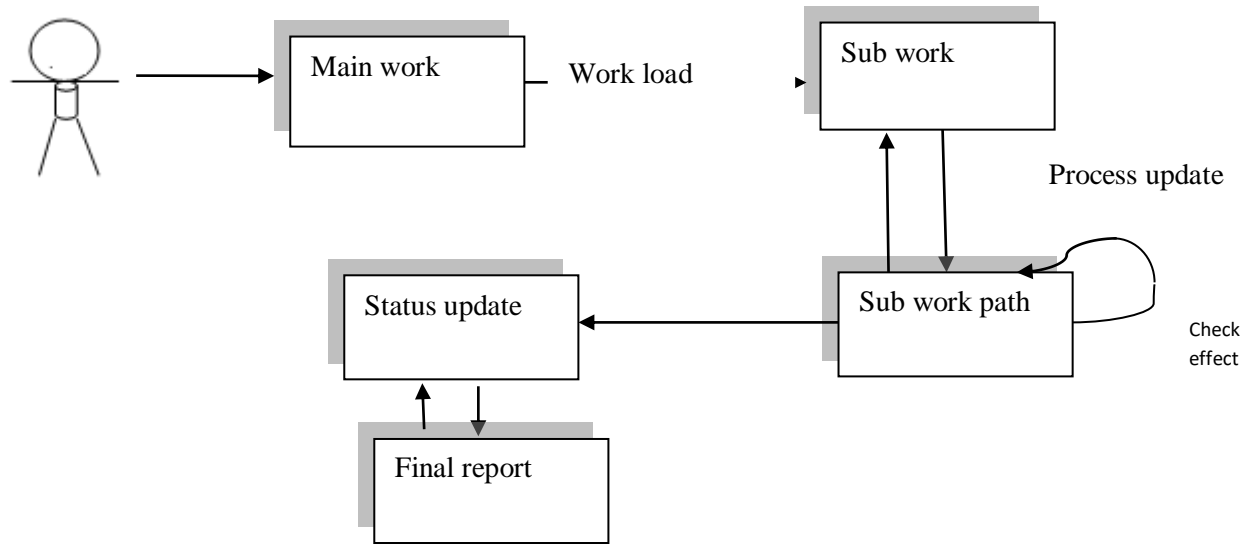


Fig 5.3.2 Collaboration Diagram for Clerk

### 5.3.3 Collaboration Diagram for Clerk 2

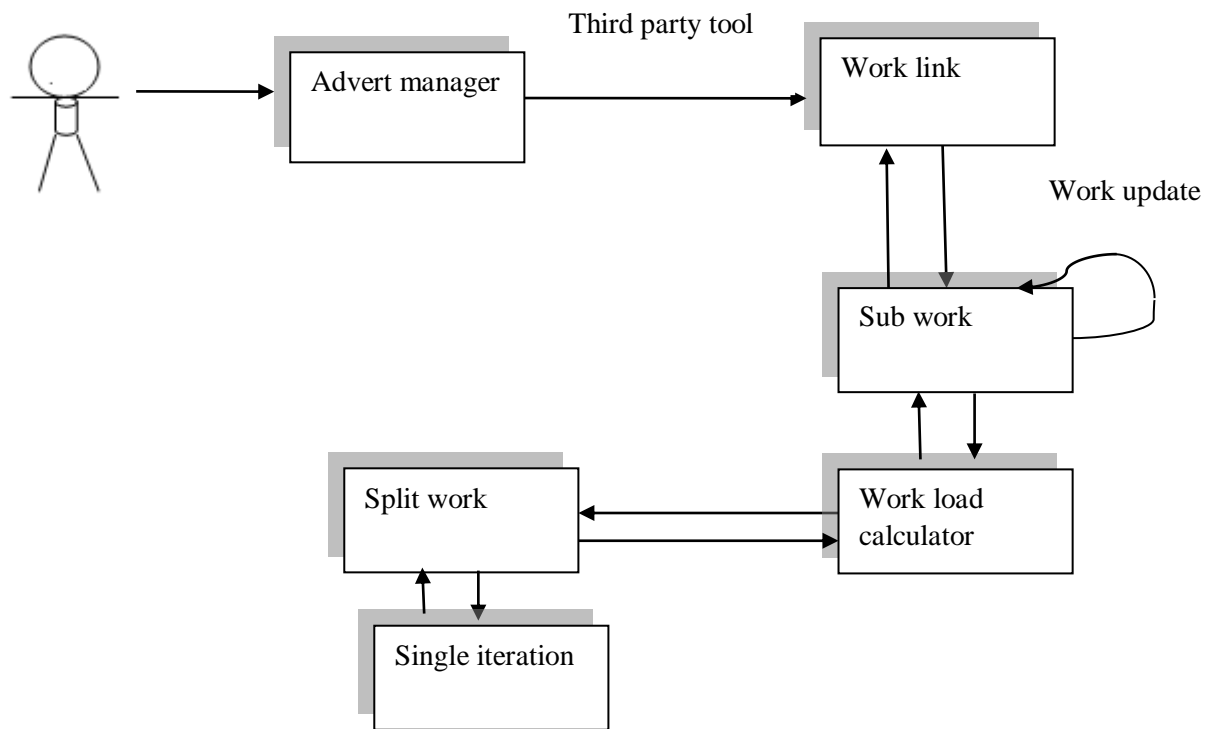


Fig 5.3.3 Collaboration Diagram for Clerk

### 5.3.4 Collaboration Diagram for main clerk

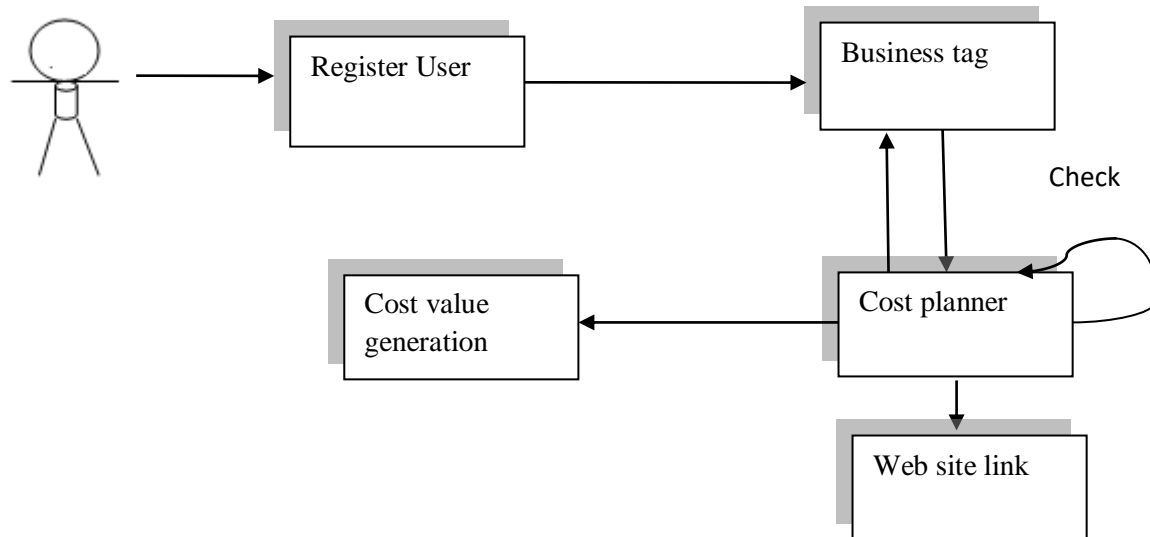


Fig 5.3.4 Collaboration Diagram for Main clerk

### 5.3 Activity Diagram

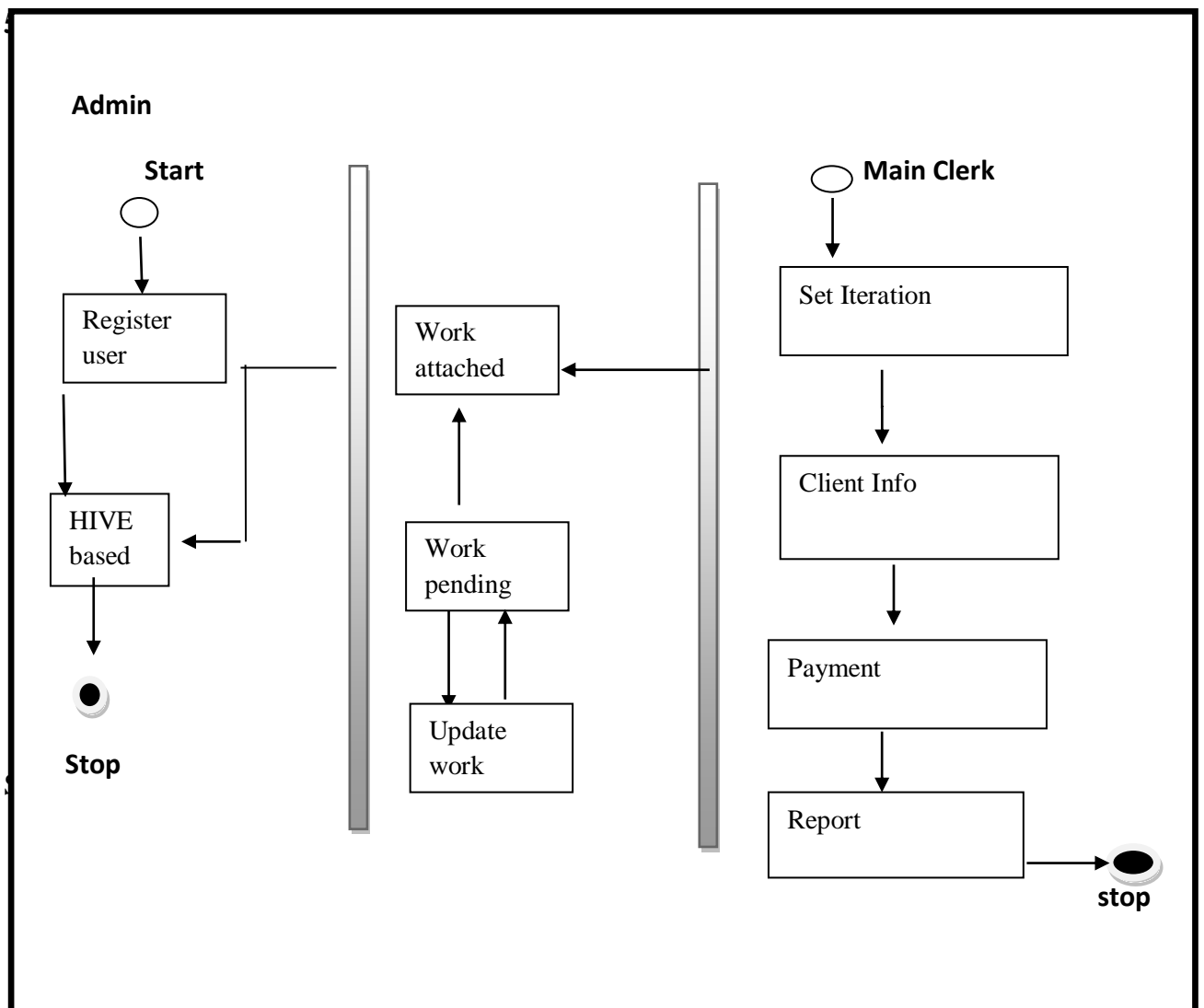


Fig 5.4.1: Activity diagram for Admin, the details of HIVE matching and updates in work are described in the picture.

### 5.4.2 Activity Diagram for Clerk

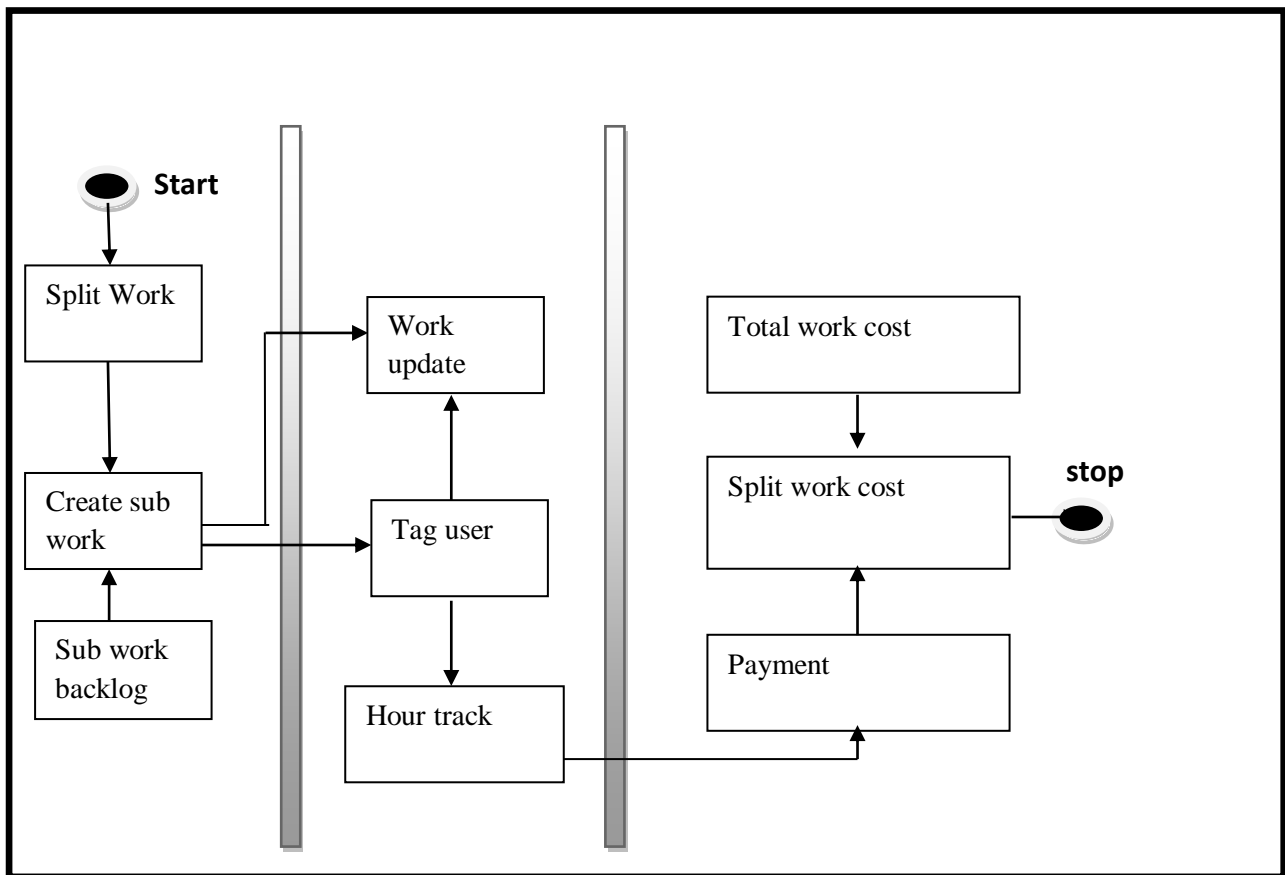


Fig 5.4.2: Activity diagram for divide and manage the work flow.

## 5.5 Database design

### Tables

Table name DBO.HIVE\_KPO\_\_SPRINT\_BACKLOG\_MANEGEMENT

Serial	Column Name	Data type (size)	Constraints
1	HIVE_KPO__SPRINT_BRANCH	INT	PRIMARYKEY,
2	HIVE_KPO_PR_BACKLOG	VARCHAR(25)	NA
3	HIVE_KPO_PR_FREEDBACK	VARCHAR(25)	NA
4	HIVE_KPO_PR_BUGS	VARCHAR(25)	NA

	Column Name	Data Type	Allow Nulls
▶	HIVE_KPO__SPRINT_BRANCH	int	<input type="checkbox"/>
	HIVE_KPO_PR_BACKLOG	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_PR_FREEDBACK	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_PR_BUGS	varchar(25)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table name DBO.HIVE\_KPO\_BR\_SPRINT\_OTHER

Serial	Column Name	Data type (size)	Constraints
1	HIVE_KPO_TELE_SPRINT_BRANCH	INT	PRIMARYKEY,
2	HIVE_KPO_NON_T_TECH_ID	VARCHAR(25)	NA
3	HIVE_KPO_NON_TEAM	VARCHAR(25)	NA
4	HIVE_KPO_NON_T_TEAM_SIZE	INT	NA
5	HIVE_KPO_NON_TTEAM_ST	DATE	NA
6	HIVE_KPO_NON_TTEAM_END	DATE	

	Column Name	Data Type	Allow Nulls
▶	HIVE_KPO_TELE_SPRINT_BRANCH	int	<input type="checkbox"/>
	HIVE_KPO_NON_T_TECH_ID	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_NON_TEAM	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_NON_T_TEAM_SIZE	int	<input checked="" type="checkbox"/>
	HIVE_KPO_NON_TTEAM_ST	date	<input checked="" type="checkbox"/>
	HIVE_KPO_NON_TTEAM_END	date	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table name DBO.HIVE\_KPO\_BRANCH\_SPRINT

Serial	Column Name	Data type (size)	Constraints
1	HIVE_KPO_SPRINT_BRANCH	INT	PRIMARYKEY,
2	HIVE_KPO_START_DATE	VARCHAR(25)	NA
3	HIVE_KPO_END_DATE	VARCHAR(25)	NA
4	HIVE_KPO_DEV	VARCHAR(25)	NA

	Column Name	Data Type	Allow Nulls
▶	HIVE_KPO_SPRINT_BRANCH	int	<input type="checkbox"/>
	HIVE_KPO_START_DATE	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_END_DATE	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_DEV	varchar(25)	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

Table name DBO.HIVE\_KPO\_SPRINT\_PRIMARY

Serial	Column Name	Data type (size)	Constraints
1	HIVE_KPO_TELE_SPRINT_BRANCH	INT	PRIMARYKEY,
2	HIVE_KPO_TECH_ID	VARCHAR(25)	NA
3	HIVE_KPO_DEV_TEAM	VARCHAR(25)	NA
4	HIVE_KPO_DEV_TEAM_SIZE	INT	NA
5	HIVE_KPO_DEV_TEAM_ST	DATE	NA
6	HIVE_KPO_DEV_TEAM_END	DATE	NA

	Column Name	Data Type	Allow Nulls
▶	HIVE_KPO_TELE_SPRINT_BRANCH	int	<input type="checkbox"/>
	HIVE_KPO_TECH_ID	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_DEV_TEAM	varchar(25)	<input checked="" type="checkbox"/>
	HIVE_KPO_DEV_TEAM_SIZE	int	<input checked="" type="checkbox"/>
	HIVE_KPO_DEV_TEAM_ST	date	<input checked="" type="checkbox"/>
	HIVE_KPO_DEV_TEAM_END	date	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

**ER diagram for chart flow**

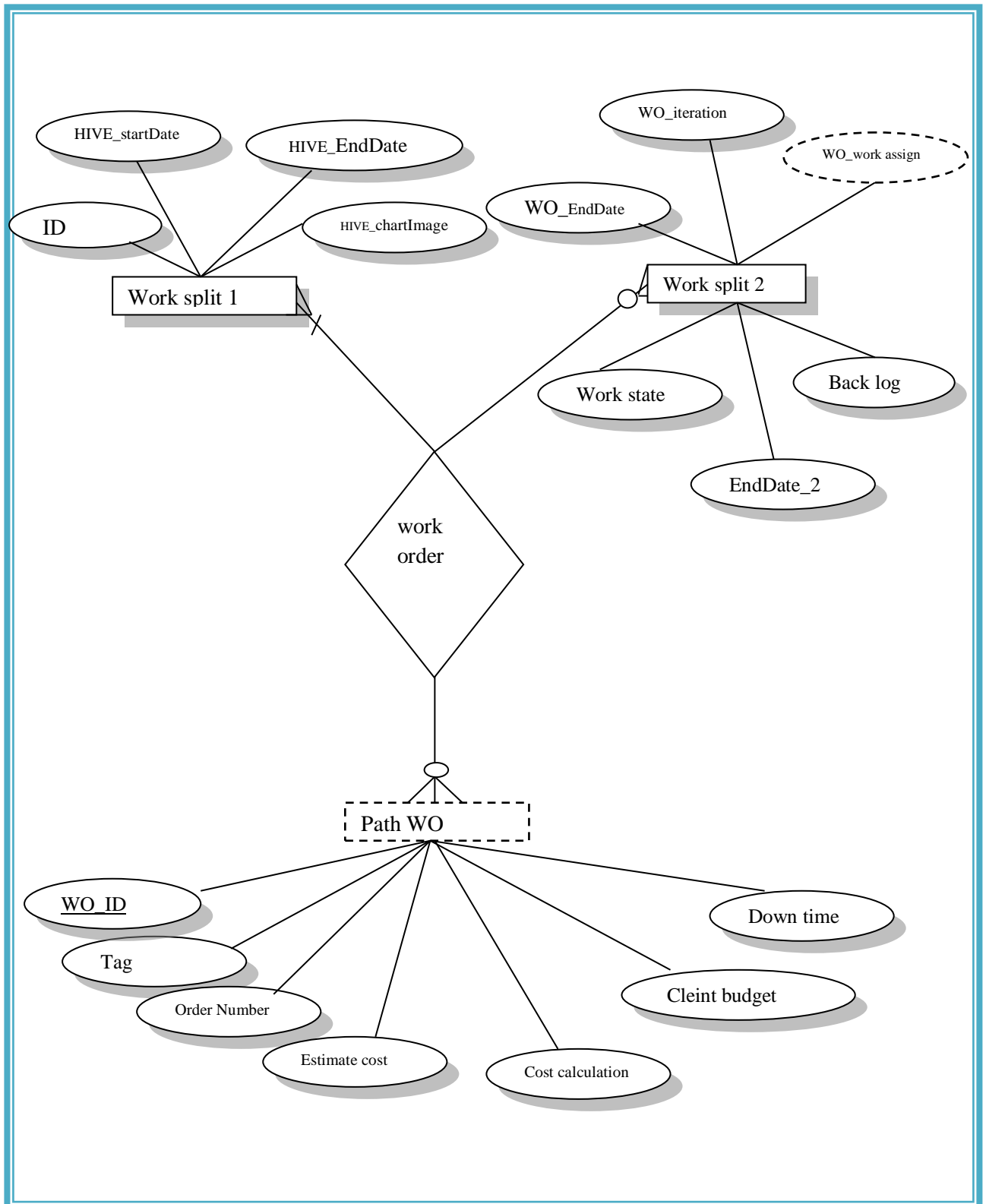


Fig 5.1 ER Diagram for char flow



### ER diagram for product back log

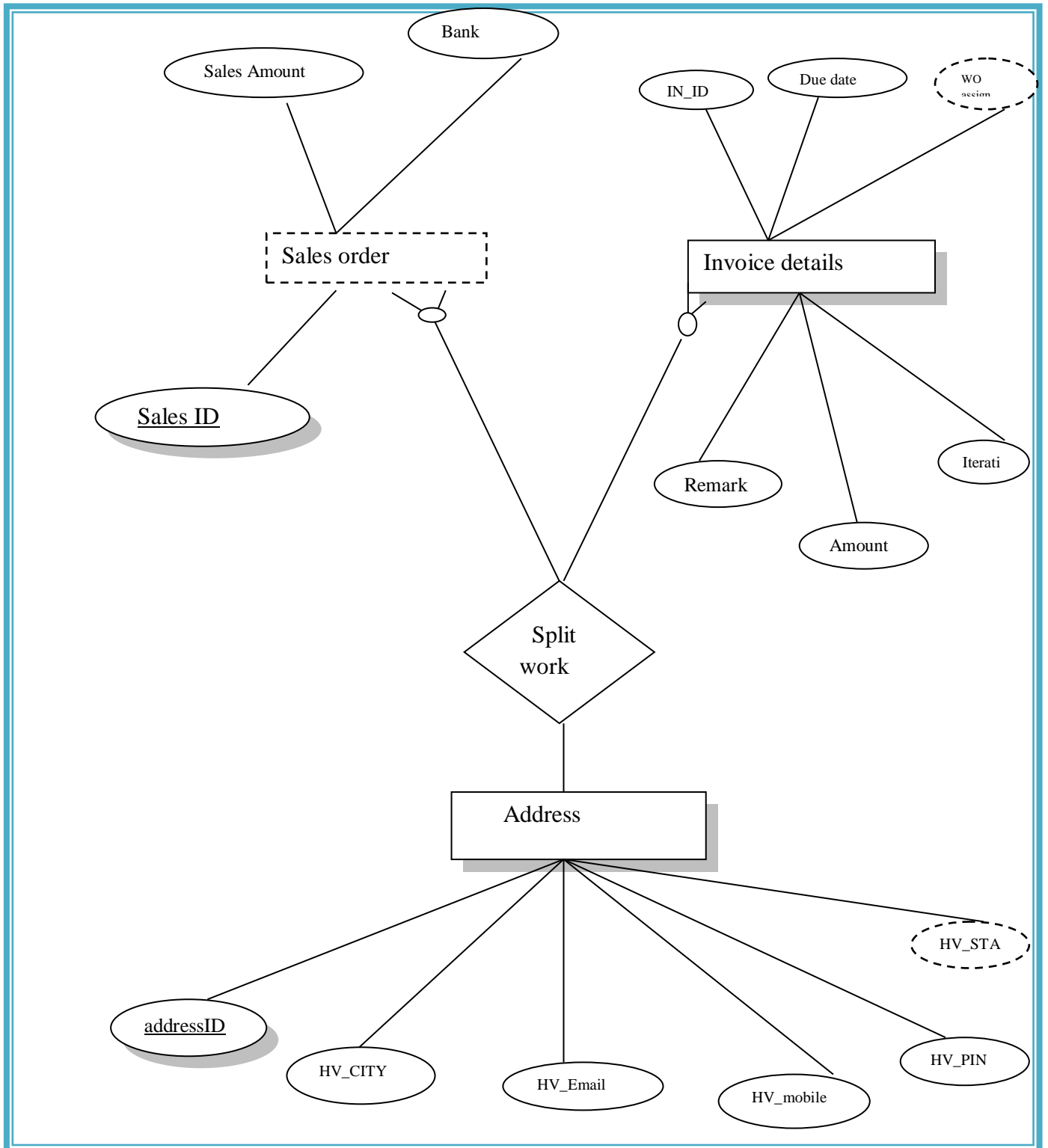


Fig 5.2 ER Diagram for product log

# CHAPTER 6

## IMPLEMENTATION

### 6.1 SCREENSHOT

#### Login page



Fig 6.1 Log in page

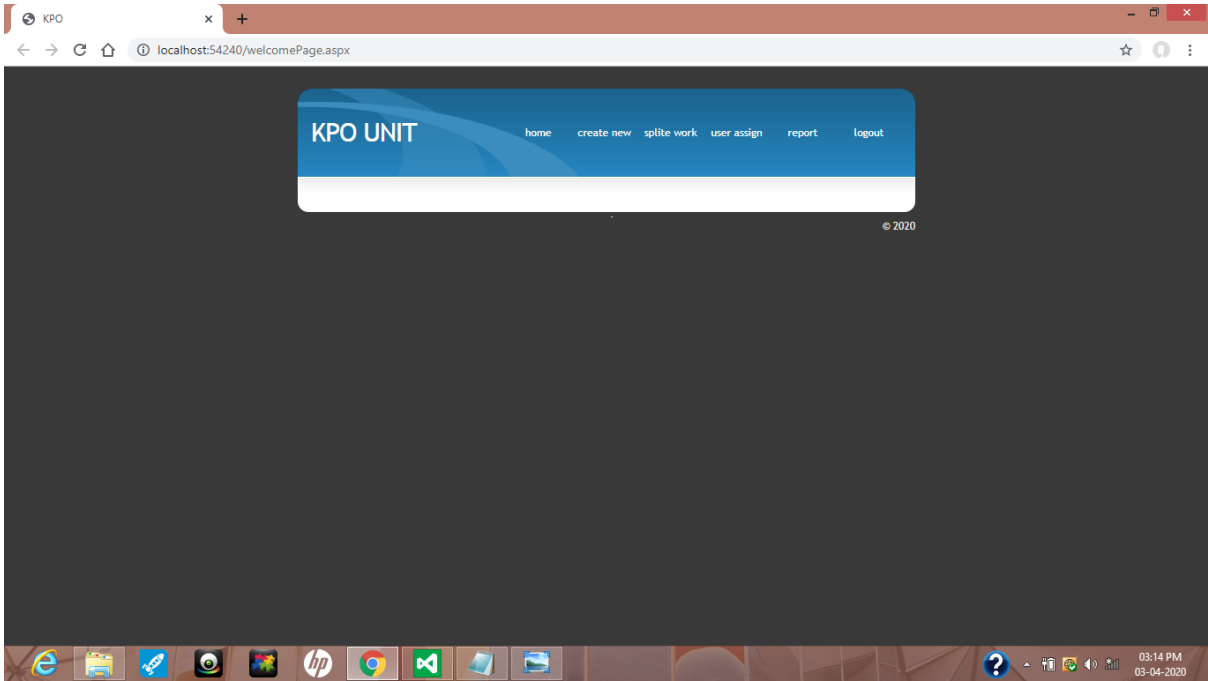


Fig 6.2: welcome page.

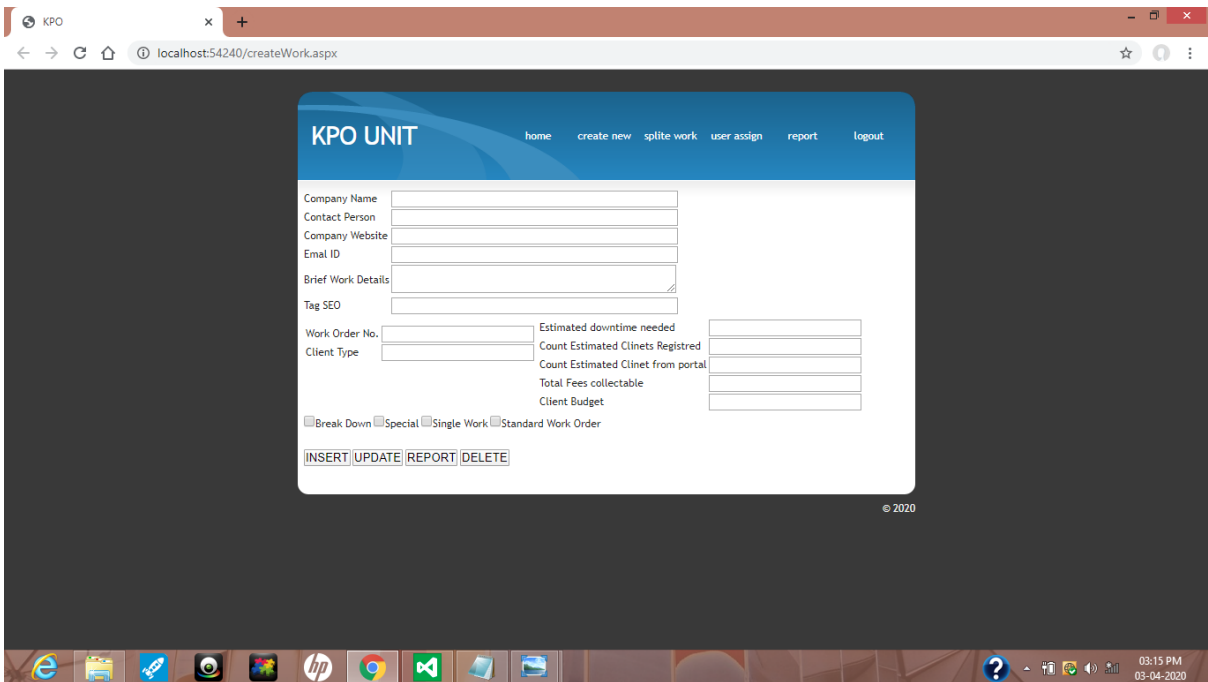


Fig 6.3 page to create work order

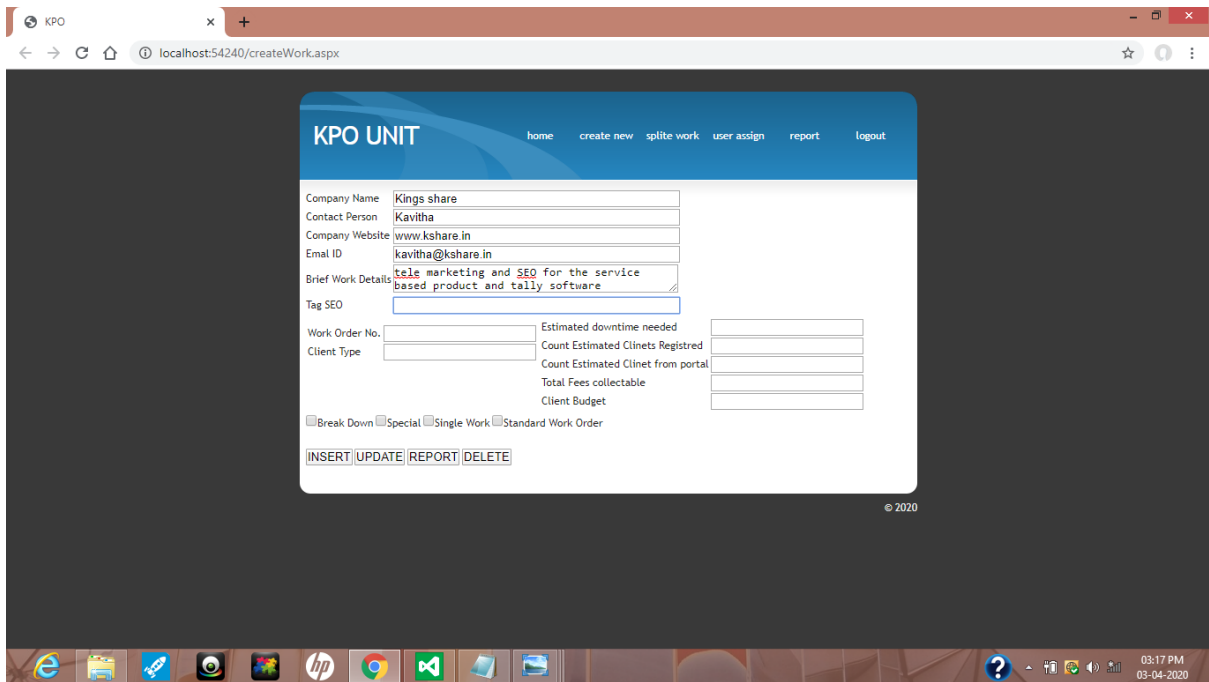


Fig 6.4: fields added with values

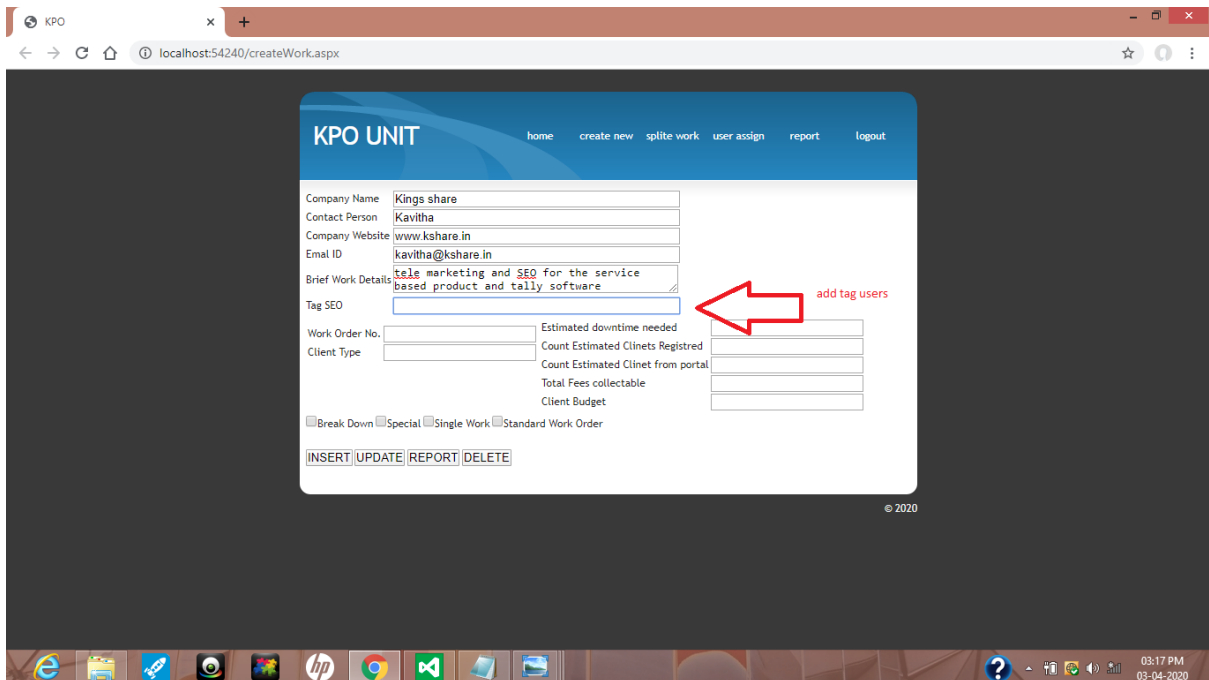


Fig 6.5: enter the tag user's form. To target the users for the created work

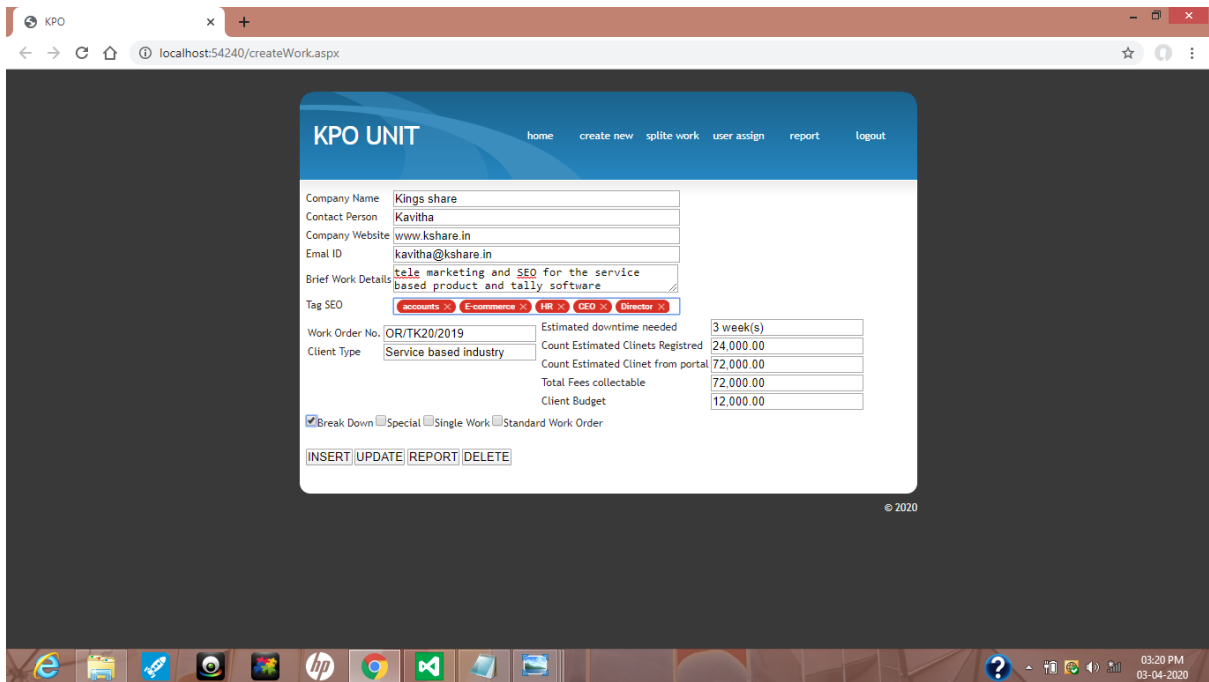


Fig 6.6: This is the place where the HIVE is implemented

The auto generated value will be exclusives only for this page and the rest of auto filled data will be saved in HIVE stack. These HIVE based data like customer address, tag will be loaded in the stack and will be attached with the cost estimated part.

Targeted users are from the account and e commerce back ground which will post the advert in the users belongs to accountant, e commerce back ground.

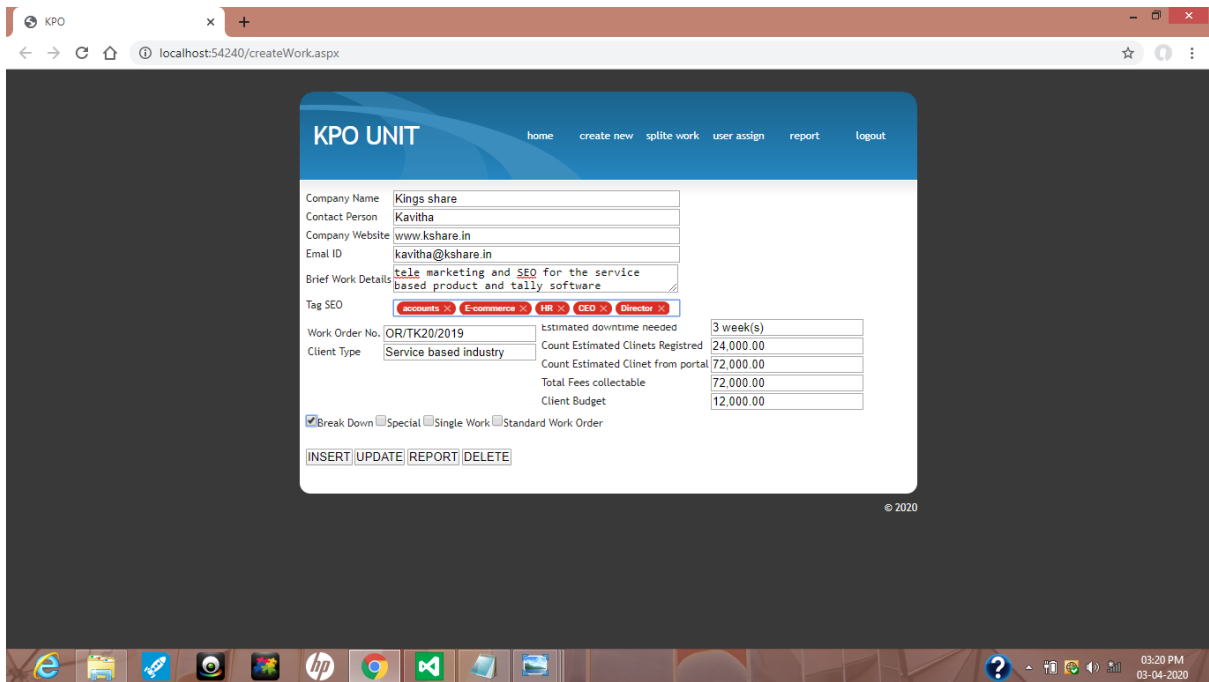


Fig 6.7: the second panel details like work order no to client budget will be saved in the common hive data base. Select the break down for the work data. This feature will invoke the HIVE and the entire data will be saved in the stack. When further updates in the sprint occurs all the further data will be linked with the main parent work.

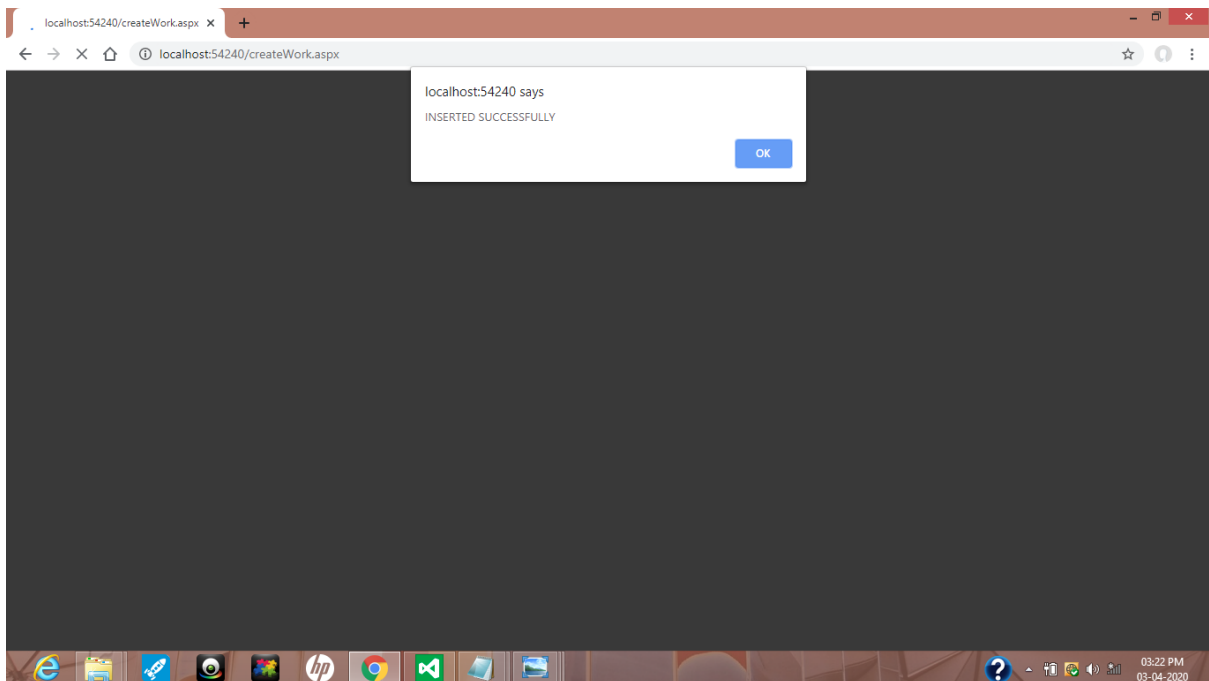


Fig 6.8: insert successfully.

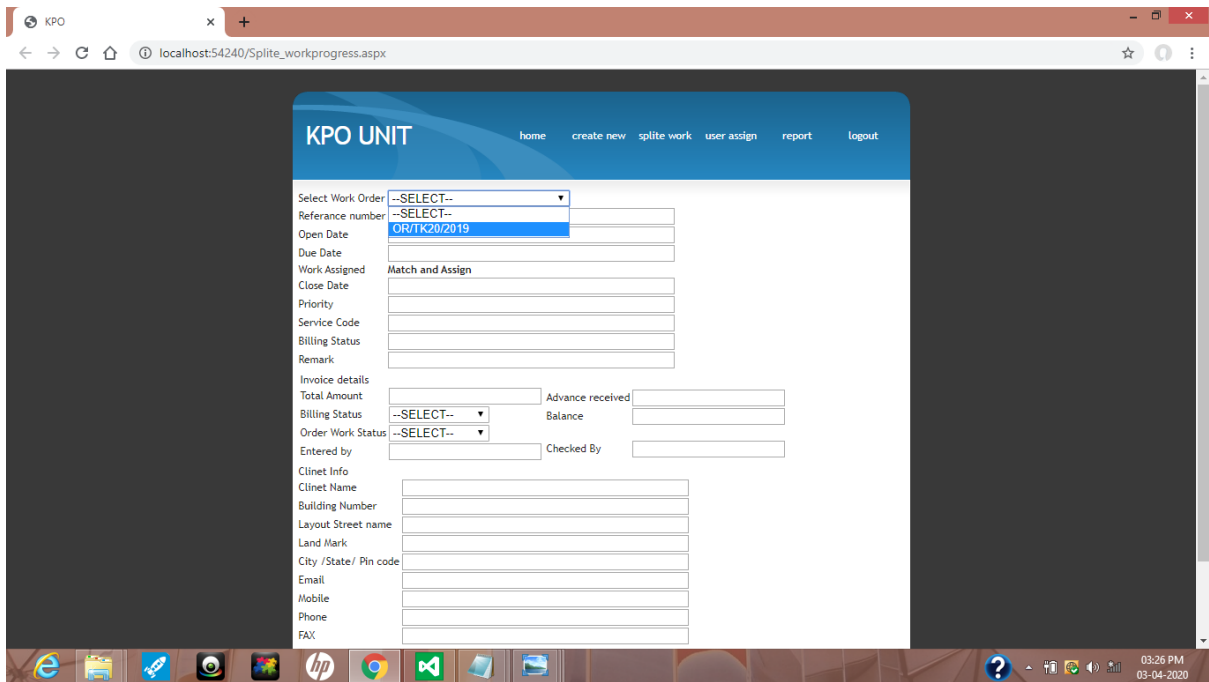


Fig 6.9: select the work order

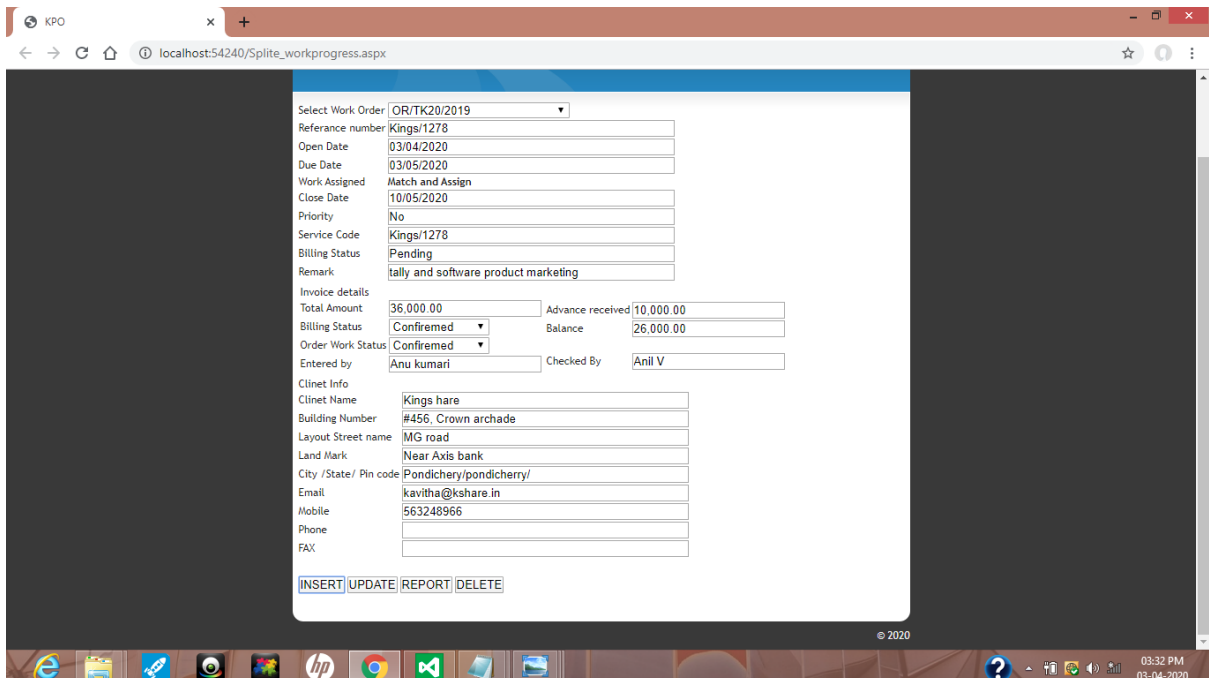


Fig 6.10: the data from the different HIVE stack will be fetched and loaded.

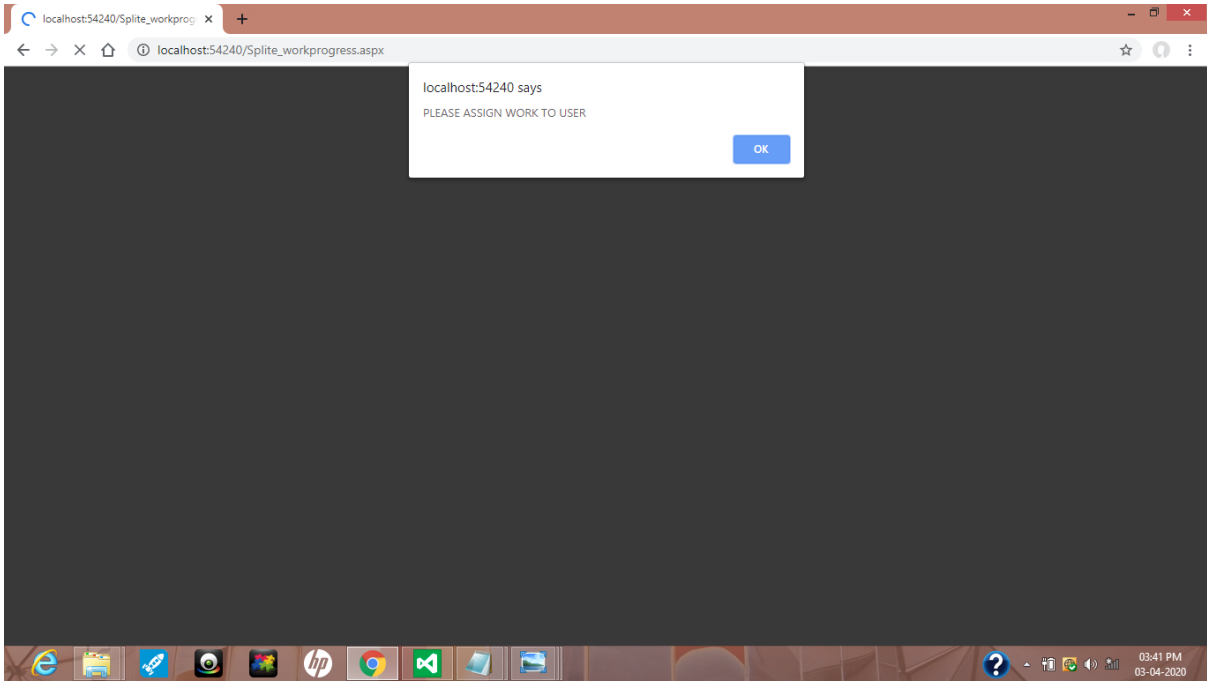


Fig 6.11: saved in SQL server.



Fig 6.12: the employee work updates



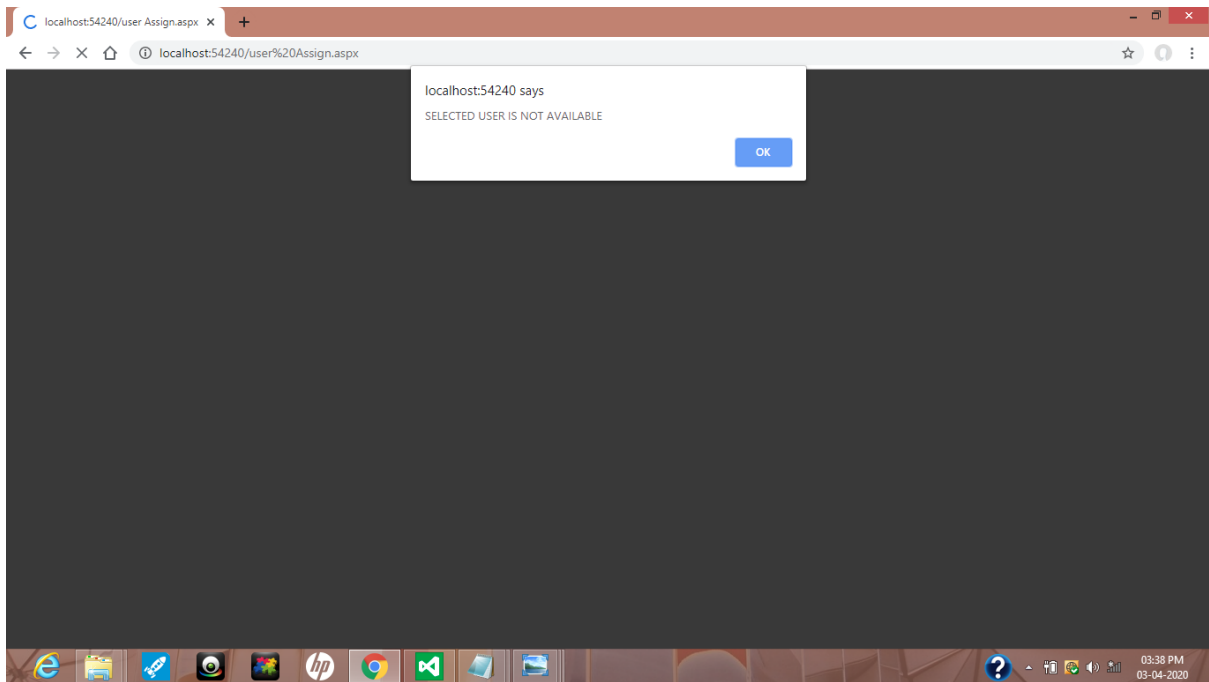


Fig 6.13 if the user balance work less than the assigned work then pop up message for with not available is loaded.

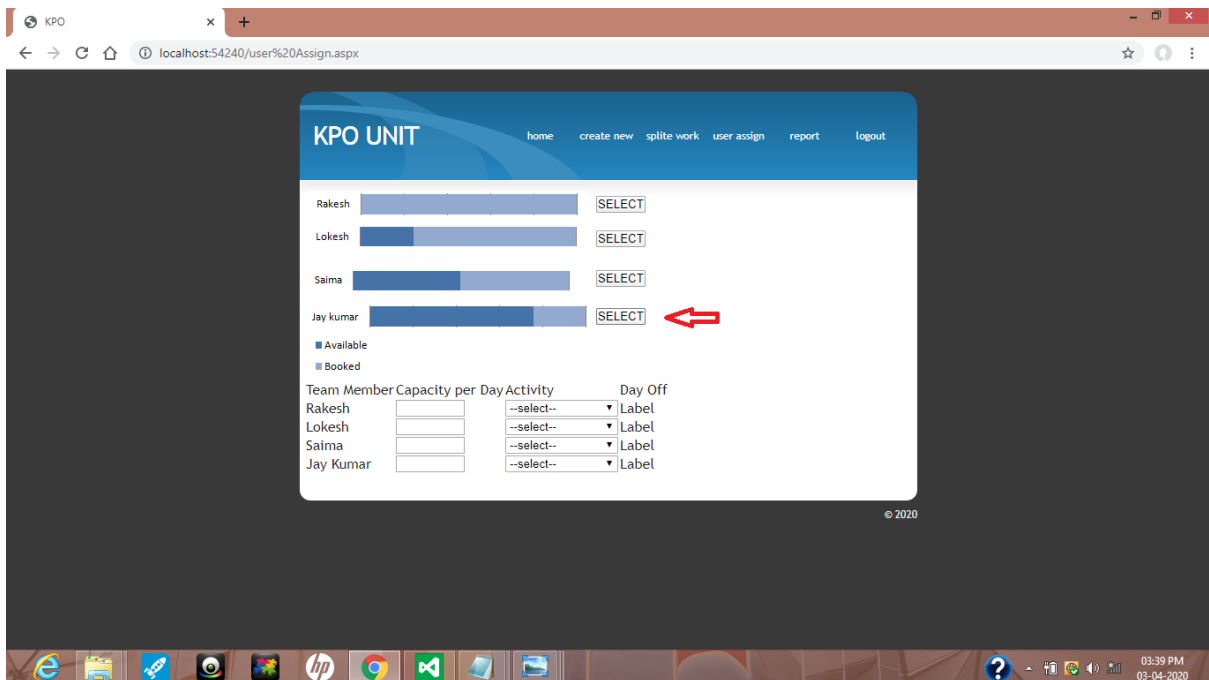


Fig 6.14: the user selects the available user.

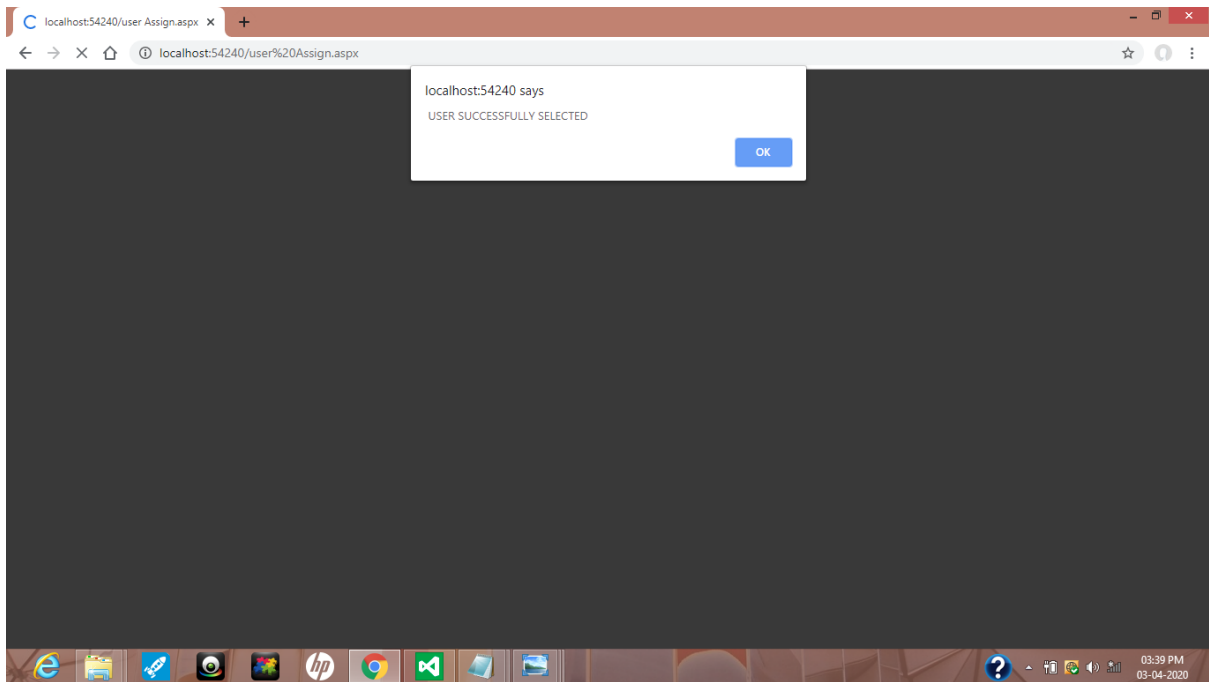


Fig 6.15: user selected

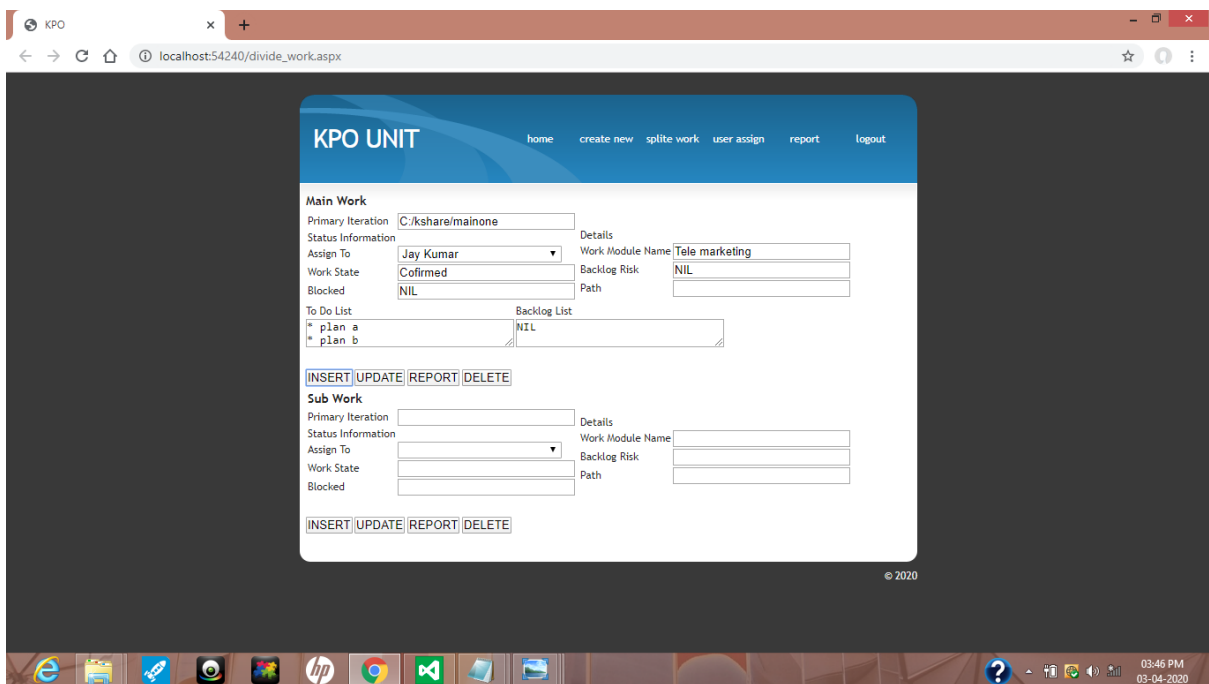


Fig 6.16: sub work. The parent work is loaded in HIVE and divided work in the stack details.

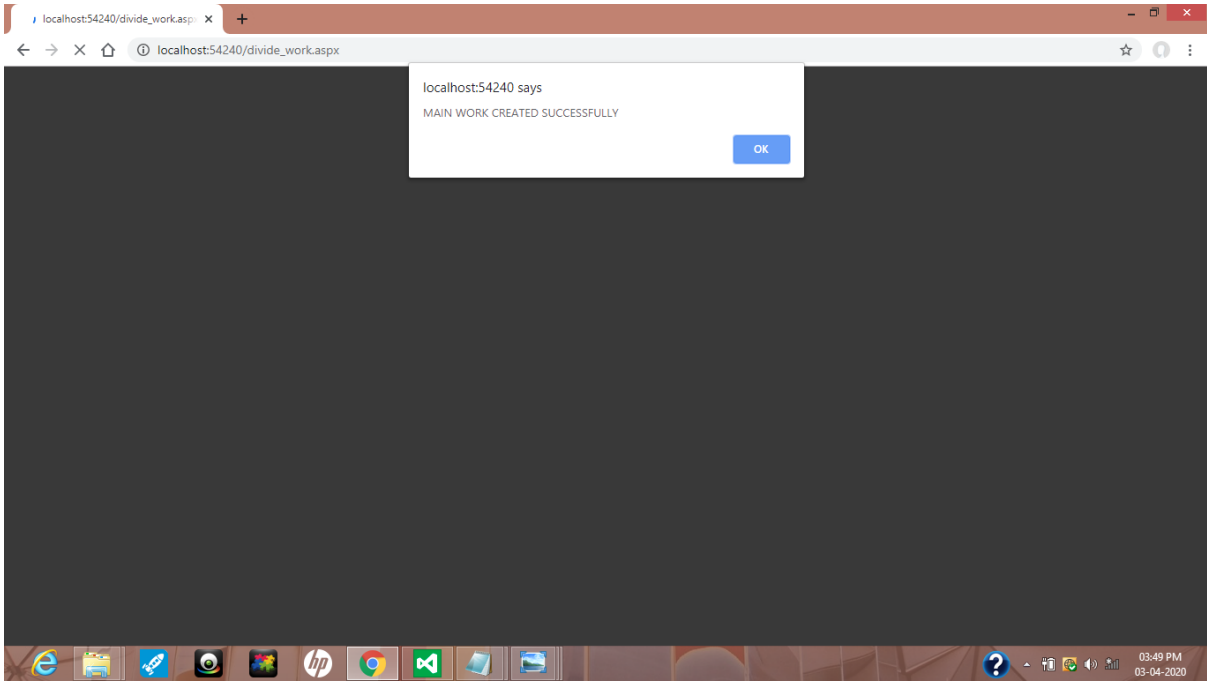


Fig 6.17: create successfully.

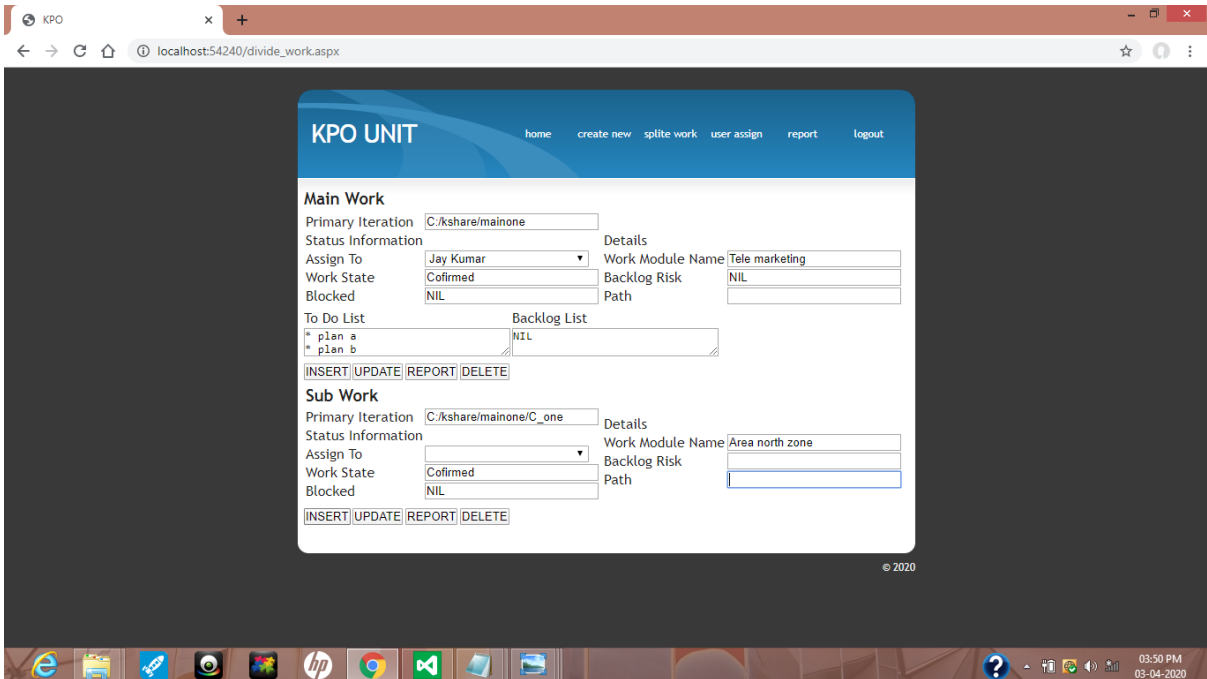


Fig 6.18: the HIVE data is linked with stack data (sub work)

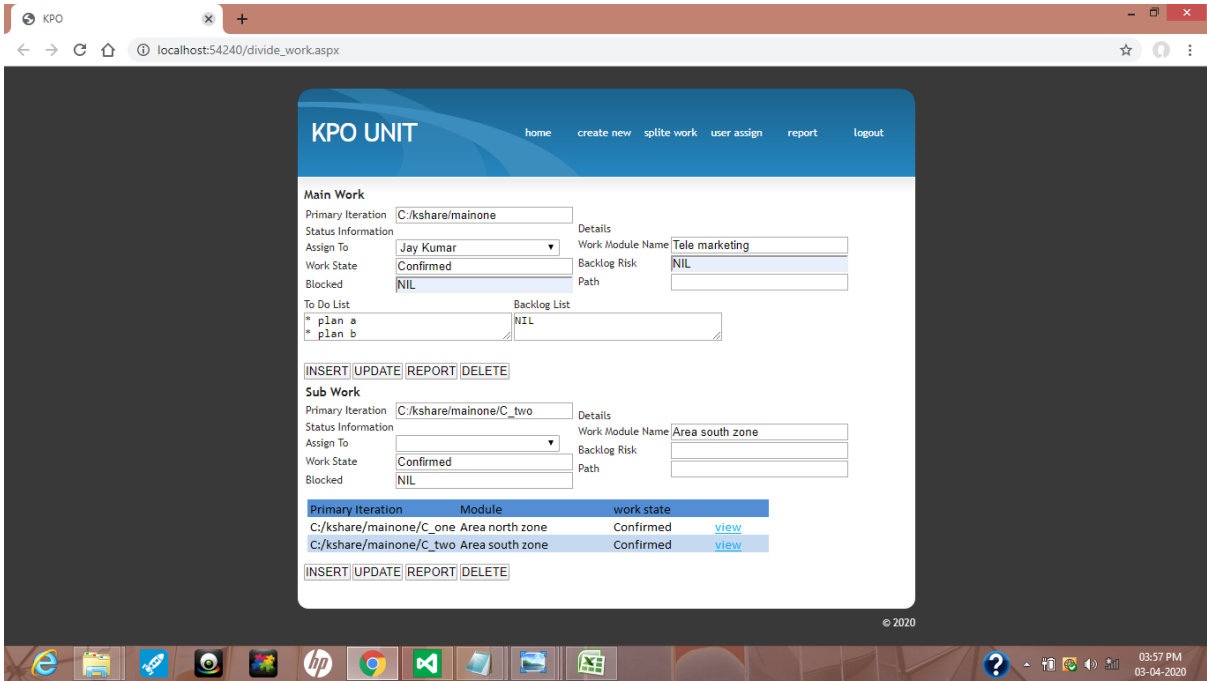


Fig 6.19: displayed in the grid view.

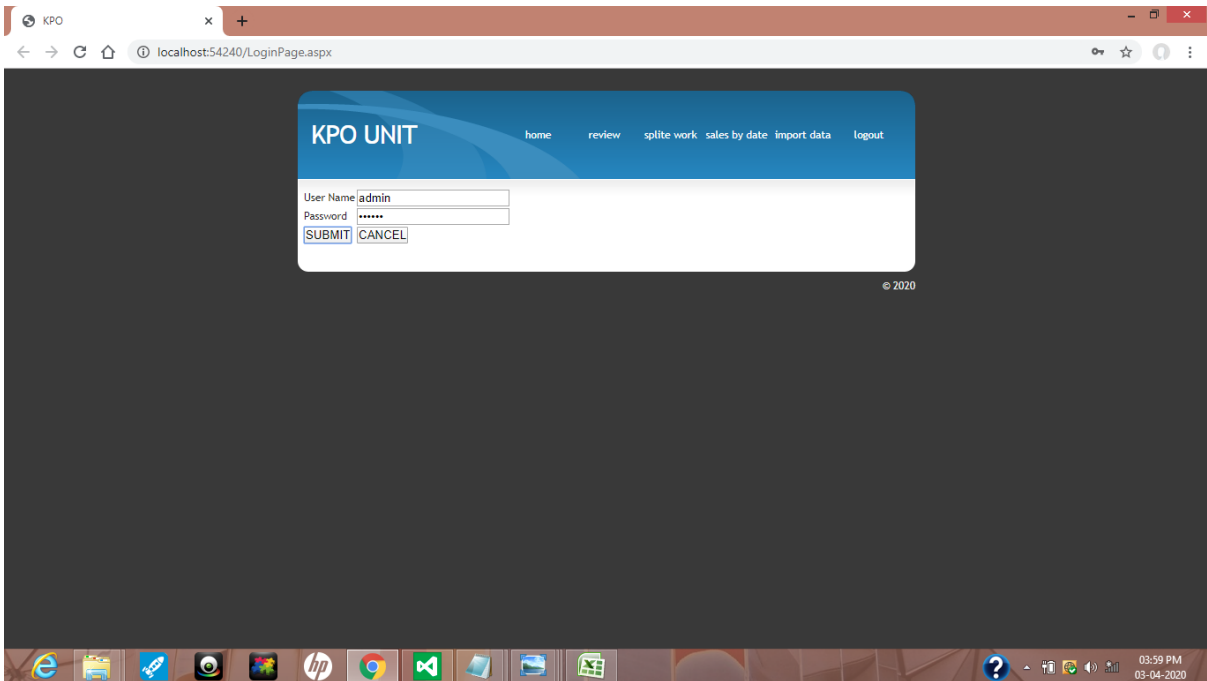


Fig 6.20: admin log in

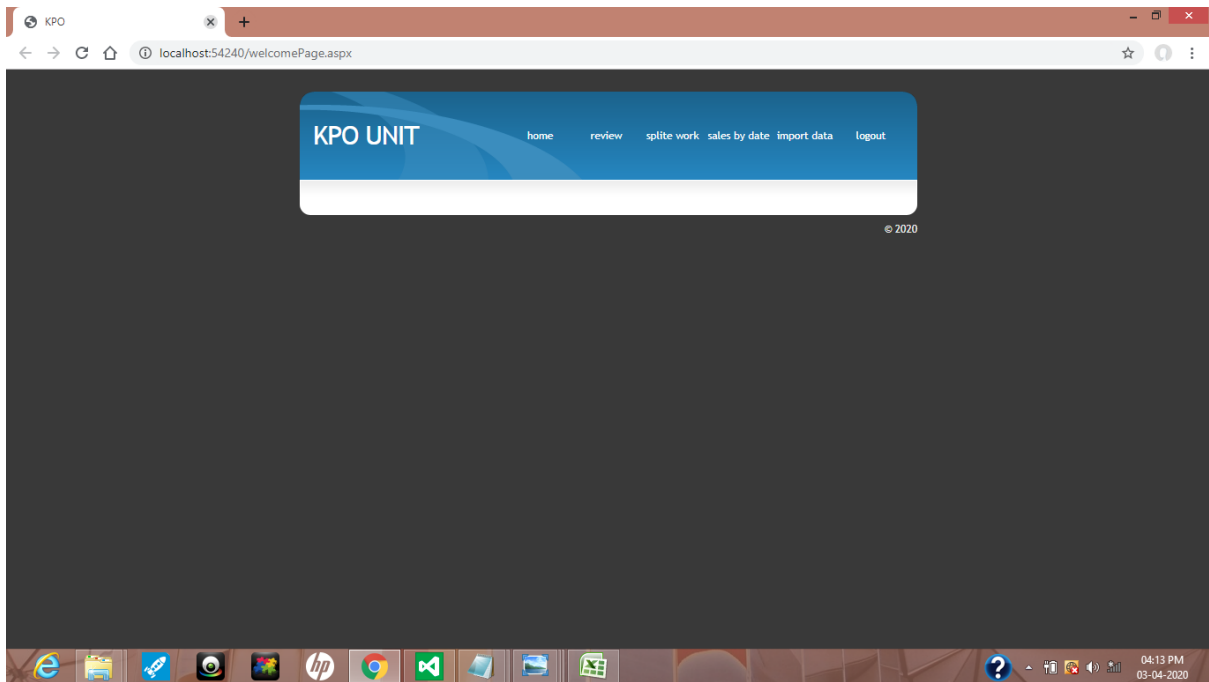


Fig 6.21 admin welcome page

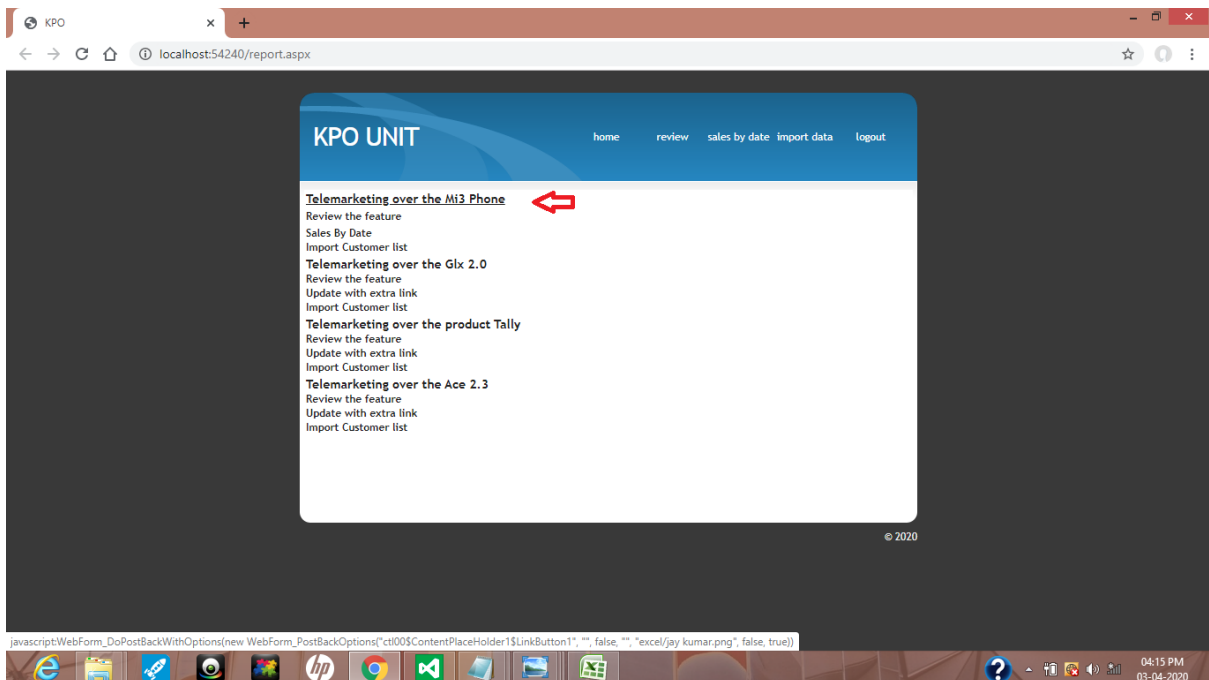


Fig 6.22: enter data loaded in the page (including HIVE and stack data)

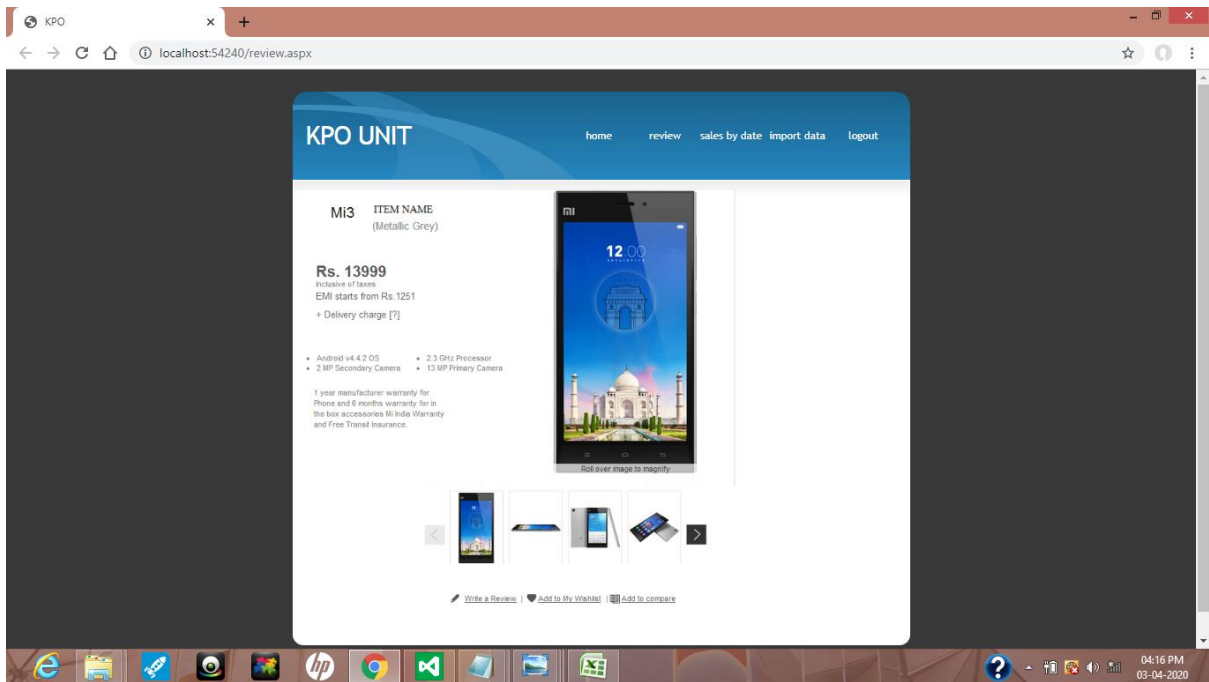


Fig 6.24: the data in the HIVE stack displayed. (Product info)

The screenshot shows a web browser window with the address bar displaying 'localhost:54240/salesBy%20date.aspx'. The page content includes a navigation menu with 'home', 'review', 'sales by date', 'import data', and 'logout'. The main content area displays a table with the following data:

Total	WMI Miles			WMI Tax (\$)		
	Exempt	Out-of-State	Taxable	Rate / Mile	Total	
513	9	0	326	0.726	23.67	🗑️
	0	0	114	0.1266	14.43	
	0	0	62	0.1447	8.97	
904	2	0	292	0.726	21.20	🗑️
	2	358	250	0.1638	40.95	
0	0	0	0	0.0000	0.00	🗑️
1,644	0	244	68	0.726	4.94	🗑️
	1	1,100	230	0.1638	37.67	
3,016	0	1,093	0	0.726	0.00	🗑️
	6	1,161	756	0.1638	123.83	
910	1	0	108	0.726	7.84	🗑️
	12	0	789	0.1638	129.24	
1,855	0	693	0	0.726	0.00	🗑️
	0	1,162	0	0.1638	0.00	
1,885	3	0	160	0.726	11.62	🗑️
	9	28	1,685	0.1638	276.00	
2,109	6	0	160	0.726	11.62	🗑️
<b>25,114</b>	<b>116</b>	<b>12,618</b>	<b>12,474</b>		<b>\$1,702.52</b>	

The browser's taskbar at the bottom shows various application icons and the system clock indicating 04:16 PM on 03-04-2020.

Fig 6.25: stack data for cost generates

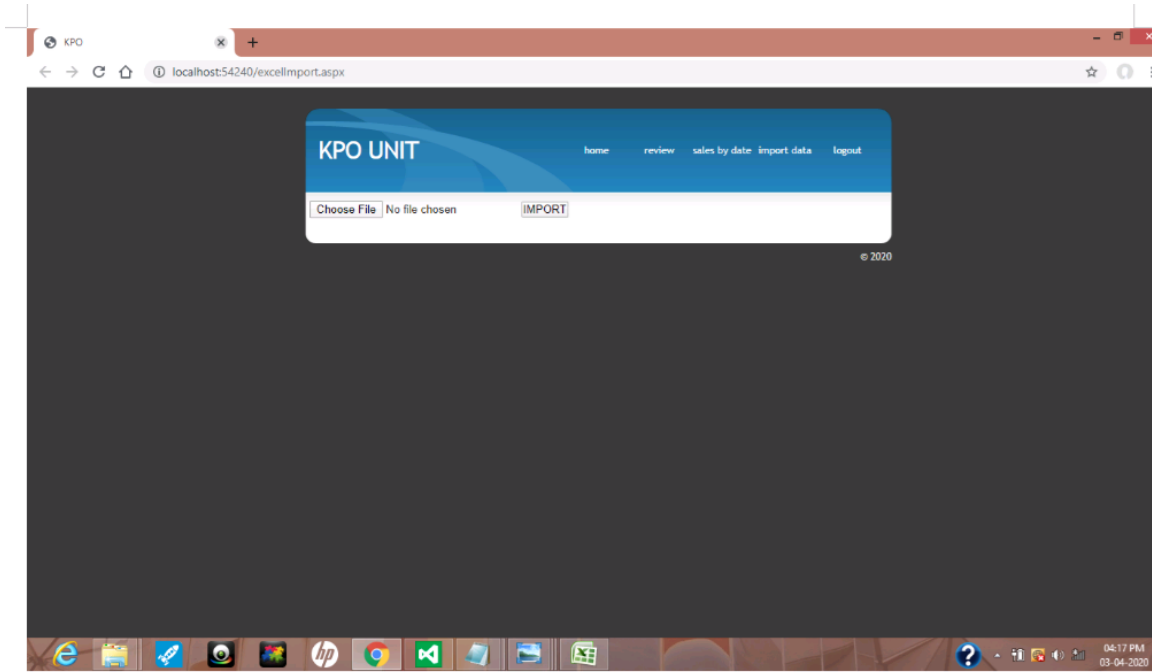


Fig 6.26: manual update in the HIVE stack

## 6.2 CODING

```
private void call_gv()
{
    SqlConnection con = new SqlConnection(strcon);
    SqlCommand cmd = new SqlCommand("select *from dbo.gen", con);
    con.Open();
    SqlDataReader dr = cmd.ExecuteReader();
    GridView1.DataSource = dr;
    GridView1.DataBind();

    con.Close();
}

private void call_gen()
{
    SqlConnection con = new SqlConnection(strcon);
    SqlCommand cmd = new SqlCommand("insert into
dbo.gen(Org_reg,org_name,sale_unit,org_group,email_id,valid)values('"+txt_gen_1.Text+"','"+
Text_gen_2.Text+"','"+txt_gen_3.Text+"','"+txt_gen_4.Text+"','"+txt_gen_5.Text+"','"+txt_gen_6.Text+
"')", con);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
}

private void call_pnt()
{
    //

    SqlConnection con = new SqlConnection(strcon);
    SqlCommand cmd = new SqlCommand("insert into
dbo.plant(Org_reg,plantUnit,tax_cat,tax_cls)values('"+txt_gen_1.Text+"','"+txt_gen_7.Text+"','"+
txt_gen_8.Text+"','"+txt_gen_9.Text+"')", con);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
}

private void call_qt()
{
    SqlConnection con = new SqlConnection(strcon);
    SqlCommand cmd = new SqlCommand("insert into
dbo.qt(Org_reg,MinOrder,delyOrder,DeliveryUnit,C_code,C_serv,C_rate,C_remark)values('"+
txt_gen_1.Text+"','"+txt_qu_1.Text+"','"+txt_qu_2.Text+"','"+txt_qu_3.Text+"','"+txt_qu_4.Text+"','"+
txt_qu_5.Text+"','"+txt_qu_6.Text+"','"+txt_qu_7.Text+"')", con);
    con.Open();
    cmd.ExecuteNonQuery();
    con.Close();
}

private void call_bil()
```



# CHAPTER 7

## SOFTWARE TESTING

### 7.1 TEST CASES

#### 7.2.1 Sign Up/Sign in Test

Test Case ID	Test Cases Description	Purpose	Input	Output	Result
1.	Register	If new User client has to register	Successful	Successful	Pass
2.	Login	If already Registered	Successful	Successful	Pass
3.	Login With User name and password	Check for the network connection before login	Credential should be network failure	Unsuccessful	Pass
4.	Login with password	If password is wrong	Wrong password	Unsuccessful	Pass
5.	Login with user name	If login fails	Wrong User name	Unsuccessful	Pass
6.	Register data for login	Login Successful after recorded page	Successful	Successful	Pass
7	Log Out	Sign Out of the page	Successful	Successful	Pass

### 7.2.2 Profile Test

<b>Test Case Id</b>	<b>Test Case Description</b>	<b>Purpose</b>	<b>Input</b>	<b>Output</b>	<b>Result</b>
1.	Update	User should Update the profile with name/address/contact/number	Update	Successful	pass
2.	Edit	User can change their profile information and update	Edited Successful	Successful	Pass
3.	Delete	User can delete his profile if h is no longer wants to use	Deleted Successful	Successful	Pass

### 7.2.3 Configuration Test

<b>Test Case Id</b>	<b>Test Case Description</b>	<b>Purpose</b>	<b>Input</b>	<b>Output</b>	<b>Result</b>
1.	Account Conformation	Account linked With Social apps for confirmation	Confirmation is Successful	Successful	pass
2.	URL	For tracking the order	URL is correct	Successful	Pass
3.	Email Setting	Recorded for the tracking of the multiple	Successful	Successful	Pass

## **CHAPTER 8**

### **CONCLUSION**

The KPO or Knowledge Process Outsourcing is works on technical and non-technical support for the third party business processing. It describes the outsourcing of core information related business activities which are competitively important form an integral part of a company value chain. KPO requires advanced analytical and technical skills as well as a high degree of specialist expertise, value creation, potential for cost reductions, and shortage of skilled labour. The clients can approach the KPO explain their requirements, the KPO units will create the work order and process it and finish it. One of the subtle thing happens in the KPO unit is there can be similar work created for the different customers in the same unit. So the HIVE implementation is an Ideal so that while process or storing the data in the same SQL stack. For example, if two organisation with health care approached the KPO the work over the patient detail transfer, the medicine purchase transaction entry will save same form with same attributes. But these values are needed to save in different SQL server and instead of saving in stack HIVE can concatenate and store together.

# **CHAPTER 9**

## **FUTURE ENHANCEMENT**

The technology of HVE can be implemented in the domain of MOM based networks. The case study over the HIVE in MOM is carried and this can bring better operation performed in the network domain works. Unlike the Application based data implementation the Network domain is extremely different and need to support more components based on the MOM and SOA architectures.

### Advantage

Network traffic in the MOM and SOA can be reduced to a single server access.

Prevent DNS spoofing efficiently since HIVE work in server side storage.

Limits the heap over flow since the similar data in the network are saved together

Communal storage can prevent the authorised access of objects in the network or server with cross site scripting.

# APPENDIX A

## BIBLIOGRAPHY

### Reference Books:

- Software Engineering 9<sup>th</sup> edition, by Ian Somerville
- Object Oriented Modeling and Design with Uml, edition 2, by James Rumbaugh.

### Reference Websites:

<https://en.wikipedia.org/wiki/sanit>

<https://www.tutorialpoint.com/vba/>

# **APPENDIX B**

## **User Manual**

### **Install the software**

Install the software which is necessary to run the project, includes following

1. VISUAL STUDIO-1.5
2. SQL Server manager 2008 R2
3. IIS server
4. .Net frame work 2.0

### **Path Setting**

**Set the following path after the installation of visual studio software**

MyComputer>Properties>Advanced>SystemSettings>EnvironmentVariables>SystemVariables>

### **Database Connection**

First create a user in the SQL server 2008 R2 database

My SQL5.5 Edition>Go to database Home Page>Login>Administration>Database User

**Next go to MySQL developer and provide the connection**

Connection >Right Click>New Database Connection

Username: Login001

Password: Pass001

Hostname: local host

SID: XE