

A project report on

**RASTAX TECHNOLOGY BASED GATEWAY  
FOR TAX GENERATION**

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

**MASTER OF COMPUTER APPLICATIONS**  
Of



Visvesvaraya Technological University  
Belgaum, Karnataka

By

**SRINIDHI V**

**1CY18MCA64**



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

A project report on

**RASTAX TECHNOLOGY BASED  
GATEWAY FOR TAX GENERATION**

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

**MASTER OF COMPUTER APPLICATIONS**  
Of



Visvesvaraya Technological University  
Belgaum, Karnataka

By

**SRINIDHI V**

**1CY18MCA64**



**CMR INSTITUTE OF TECHNOLOGY**

**132, IT Park Road, Kundalahalli,**

**Bangalore-560037**

**2019-2020**

A project report on

**RASTAX TECHNOLOGY BASED  
GATEWAY FOR TAX GENERATION**

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

**MASTER OF COMPUTER APPLICATIONS**

Of

Visvesvaraya Technological University  
Belgaum, Karnataka

By

**SRINIDHI V**

**1CY18MCA64**

Under the guidance of

**Internal Guide**

**Ms. Varsha Pawar**

Assistant Professor, MCA Department  
CMR Institute of Technology.  
Bangalore

**External Guide**

**Mr. Preetish Mukundhan**

Project Manager  
Sorbix LLC  
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 - 560037**



***CERTIFICATE***

**This is to certify that the project work entitled**

**RASTAX TECHNOLOGY BASED GATEWAY  
FOR TAX GENERATION**

*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  
is a result of the bonafide work carried out by*

**SRINIDHI V  
1CY18MCA64**

*during the academic year 2019-2020.*

Signature of the Guide

**Ms. Varsha Pawar**

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

Signature with date

1.

2.

## DECLARATION

I, **Srinidhi V**, student of 6<sup>th</sup> Semester MCA, **CMR Institute of Technology**, bearing the USN **1CY18MCA64**, hereby declare that the project entitled “**Rastax Technology Based Gateway For Tax Generation**” has been carried out by me under the supervision of External Guide **Mr. Preetish Mukundhan**, Project Manager, Sorbix LLC, Bangalore and Internal Guide **Ms. Varsha Pawar**, 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

Srinidhi V

Date:

(1CY18MCA64)

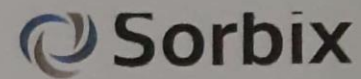
## **ACKNOWLEDGMENT**

I would like to thank all those who are involved in this endeavour 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. Preetish Mukundhan, Project Manager, Sorbix LLC., Bangalore, and to my Internal Project guide Ms. Varsha Pawar, 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. Preetish Mukundhan, Project Manager, Sorbix LLC., 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.

**Srinidhi V**  
**(1CY18MCA64)**



15-06-2020

### PROJECT COMPLETION CERTIFICATE

This is to certify that Mr. Srinidhi V, bearing **(USN NO: 1CY18MCA64)** a bona fide student of CMRIT **College**, pursuing his **Master of Computer Application**, has worked on the project entitled "**Rastax technology based gateway for tax generation**" from Feb 2020 to Jun 2020 at our organization SORBIX LLC-Bangalore, and has successfully completed the internship program.

During the tenure of the project, he was found punctual, hardworking and inquisitive. We wish all the best for his future endeavors.

For, Sorbix LLC

A handwritten signature in blue ink, appearing to read "Preetish mukundhan".



Preetish mukundhan  
(Team Leader)

<b>S.NO.</b>	<b>Content s</b>	<b>Page No.</b>
<b>1.</b>	Introduction	
	<b>1.1</b> Project Description	1
	<b>1.2</b> Company Profile	3
<b>2.</b>	Literature Survey	
	<b>2.1</b> Existing System and Proposed System	4
	<b>2.2</b> Feasibility Study	5
	<b>2.3</b> Tools and Technologies Used	6
	<b>2.4</b> Hardware and Software Requirements	8
<b>3.</b>	Software Requirement Specification	9
	<b>3.1</b> Functional Requirements	11
	<b>3.2</b> Non- Functional Requirements	13
<b>4.</b>	System Design	
	<b>4.1</b> System Perspective	15
	<b>4.2</b> Context Diagram for rastax integrators	16
<b>5.</b>	Detailed Design	
	<b>5.1</b> Use Case Diagrams	
	<b>5.1.1</b> Use Case of CA clerk	17
	<b>5.1.2</b> Use Case for Accountant	18
	<b>5.1.3</b> Use Case for Client Operations	18
	<b>5.1.4</b> Use Case for Admin	19
	<b>5.2</b> Sequence Diagrams	
	<b>5.2.1</b> Sequence Diagram for Accounting Head	20
	<b>5.2.2</b> Sequence Diagram for data loading by authentication check	21
	<b>5.2.3</b> Sequence for tax generation calling.	22
	<b>5.2.4</b> EMP and Organisation based tax generation	23
	<b>5.3</b> Collaboration Diagram	
	<b>5.3.1</b> Collaboration diagram for RAS TAX integration	24



	<b>5.3.2</b> Collaboration diagram for RAS TAX integration each employee	24
	<b>5.3.3</b> Collaboration diagram for RAS TAX integration in organization	25
	<b>5.3.4</b> Collaboration diagram for product link	25
	<b>5.4</b> Activity Diagrams	
	<b>5.4.1</b> tax generation with RASTAX plug in	26
	<b>5.4.2</b> for organization tax report generation.	27
	<b>5.5</b> Database Design	28
	<b>5.6</b> ER Diagram	31
<b>6.</b>	Implementation	
	<b>6.1</b> Screen Shots	32
<b>7.</b>	Software Testing	49
<b>8.</b>	Conclusion	53
<b>9.</b>	Future Enhancements	54
<b>10.</b>	Bibliography	55
<b>11.</b>	User Manual	56

# CHAPTER 1

## INTRODUCTION

### 1.1 PROJECT DESCRIPTION

This software is developed for chartered accountant associates for tax calculation in the organisation which is located remote area. The technology of RASTAX can work over web based and stand alone application data and generate the tax calculation report. The developers are needed to use the raspberry pi unit for the tax calculation and the raspberry pi will act as intermediate to fetch the data from the main server for the tax calculation and tax value generation will be calculated in the client machine.

#### Why to use RASTAX technology in tax calculation?

Most of the small and medium size organisation use stand alone application for their billing and the report generations. So for the chartered accountant associates machine cannot access these details which are essential for the final tax calculations. So most of the chartered accountant associates are needed to get accounts details by mail or in Excel format which is carried for the tax calculation, by doing so associate accountants do not get the connectivity from the previous year tax statements or there will be a possibility to miss entries needed for current year tax value generation. Now with help of RASTAX technology the users can make integration over the data and even this technology will transfer the locally accessible data for

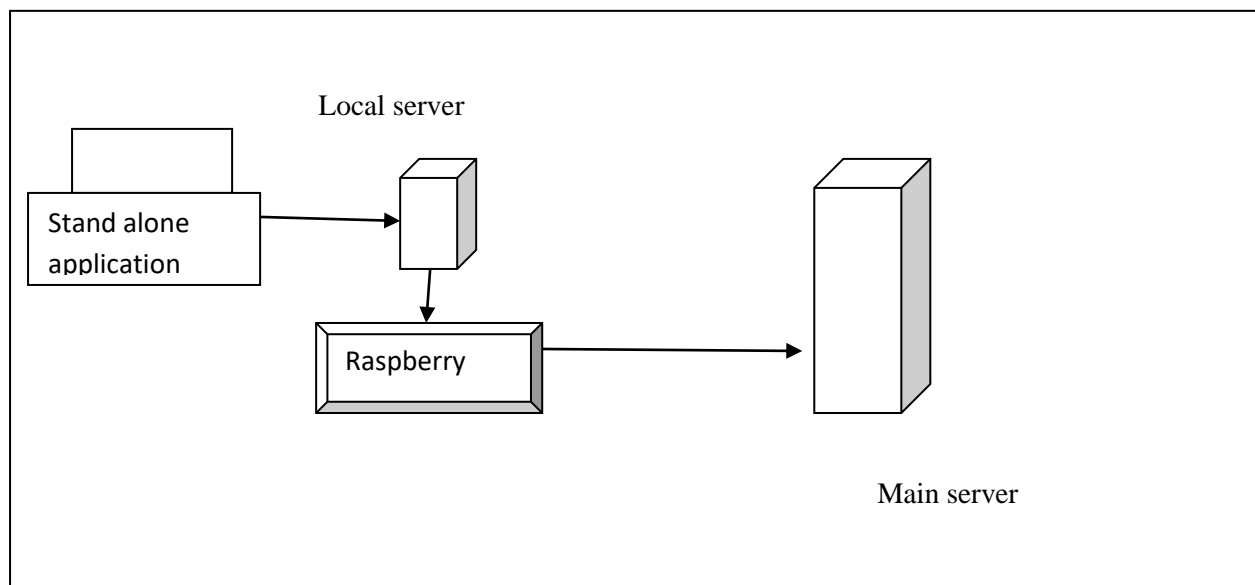


Fig 1.1: The data flow of RASTAX application.

The developers will install the application interaction algorithm in the raspberry pi unit which will fetch the details from the stand alone application and pass to the main server. Here the application will be fetching only the SQL data base and GUI or any other web based data migration steps are not included in the RASTAX applications. The admin developer will install the RASTAX software in the multiple user pages and the details related to the tax calculations are fetched and transferred to the main server which is located in the associate company. Finally after the tax values are generated from the main server the application will return the tax report respective to the company where the data are originated.

Module

- Charted associate module
- Client machine module
- Network server/ data management
- Tax calculation
- Report

### **Charted associate module**

The module where the details of tax, the tax percentage, the tax category the client and employee of the associate and the third party user created and saved. The tax profiles needed for each business category are dissimilar so the same procedure cannot be used for the tax calculation for all. The steps over the tax implementation, surcharge calculation final report collecting and paying tax are managed from this module.

### **Client machine module**

This module has the inputs based on the client business and algorithm will be implemented on the basis of the client nature of business. The developer will schedule the unit for the specific tax calculation algorithm. The client machine module is connected with raspberry pi unit which is directly connected with admin server. The algorithm will identify the nature of the client machine and the tax percentage are gathered depending on the client business.

### **Network server/ data management**

This module is created for the data mining from the SQL and the tax calculations. The complexities in the module are finding the right data from the multiple servers and gather in the cache memory for the exact tax calculations. For example if the users are needed for the

calculating the export shipping company the algorithm needed to search the data for export, the procedure for the tax calculation, the tax refund or exceptions etc.

### **Tax calculation**

In the previous module the parameters for the tax calculations are only gathered but algorithm for calculating and the finding what rate to be paid for the tax is coded in this module. The data are received from the remote SQL server and the tax will be generated in the chartered accountant machine. The report will be send either Email or as hard copy to the client. There is no alternative to be shared in the client GUI machine.

### **Report**

The final report in the tax generation is created in this module. All the reports are auto generated and the application will fetch the details with respect to the each client, the total revenue generated, the auto balance sheet generation and the tax value etc will be generated in the report form.

## **1.2 COMPANY PROFILE**

Sorbix is E commerce and Account based software development and supporting organisation located in Bangalore. The CASE (computer aided software engineering) tools based .NET software application is the speciality of Sorbix in the market domain. The organisation has created product for accounting integration and tax calculation which will support in the various domain. The t square platform will connect SQL, Oracle and JAVA based application interactions with the main application developed in the chartered accountant associate. This platform is incorporate to all like of business application and the periodically updates in the software application will ensure the latest tax percentage for the report generations.

## **CHAPTER 2**

### **LITERATURE SURVEY**

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

In the existing system the application does not have the direct communication with organization where the taxes are to be calculated. This limitation is created because of lack in the network property and limitation in the intermediate tool for the source and destination machine. The application is coded to import data manually from the SQL server and the tax calculation and report generated. This needed third party tool like team viewer and the virtual machine software for the application communication.

##### **Limitation of the existing system**

- Version control software are needed for taking backup of the original data. Because the tax calculation and the updates are made in the user passé system. If the original content is updated the users needed to roll back entire details form the system where the data is saved.
- The user system will not be updated with the latest tax details which are created in the associate or accountant. These taxes calculated values are forwarded to the client either by email or hard copy.
- The updates made in the client organization after the data forwarded to associate will not available for tax calculation. The new updates are needed to forward to the chartered accountant again and the new calculation will affect the entire calculation created by the accountant.
- Lack of security since details are passed offline mode. There will be no track where the data are viewed or any changes are made.

##### **Proposed system**

In the proposed system the application uses RASTAX technology for the direct interaction of chartered accountant and the client machine with mini raspberry pi CPU. This software will fetch the required data from the client machine and forward to the chartered accountant for the further tax calculation and the latest updates from the client machine as well the last updates made in associates will be available in the client machine. This technology can easily integrate with web to stand alone application and also ASP with JAVA based application too.

## Advantage of the proposed system

- No separate tool for version control is not used in the application. Application will create duplicate of the original data and the further updates and calculation will be calculated in the duplicated data. This will keep the original data safe and without alternated.
- The client will be updated with the latest data from the RASTAX application. The application algorithm of RASTAX will fetch the latest data each time the user login from the client machine.
- This dynamic reporting will mutually interacts with server where the latest tax details are stored. So if the government updates the tax rate or new tax rule are released this updates will be made available for all the organisation registered under RASTAX application
- The data access is given only for the registered users to ensure there is no unauthorised user access to the application data base and in the application data interactions.

## 2.2 FEASIBLE STUDY

The feasibility of the RASTAX software implementation is needed to substantiate over the client machine feasibility as well in the client machine feasibility study too. If the system is feasible only in the one side then this could lead to project failure or client give up the usage of the RASTAX application.

The feasibility of the RASTAX is carried in the areas of

- Technical feasibility
- Operational feasibility
- Cost feasibility

### **Technical feasibility**

The software tool needed for the RASTAX operation and the visual studio interaction are checked in this feasibility study. The RASTAX is coded in the ASP.NET but this module is needed to executed in Linux environment. So when developers gather supporting tools needed for the Linux ASP communication developers are needed to make sure the selected tools must suitable for the visual studio IDE. The developers are not suppose to select the

tools which are not effectively used but those are not supporting the Linux or raspberry pi units.

### **Operational feasibility**

The feasibility over the types of the operation to be carried in the tax calculations and the raspberry pi interactions with server and the module. Each organisation has different tax sections and different method over the tax calculations. So in the operational feasibility the developers are needed to analysis the client business domain and then import the tax rules and percentage with respect to the clients business needed. The developer needed to update the server with latest tax and this SQL server access with the raspberry pi unit, the authentication for the raspberry pi to SQL access etc.

### **Cost feasibility**

The total cost needed for the RASTAX implementation and the check list of supporting tools purchase and implementation are calculated in this cost feasibility study. The cost of purchasing the raspberry pi cpu, the cost of OS if needed or list of raspberry with Linux based free OS , the functioning capacity with Linux etc are brought under study in the cost feasibility study.

## **2.3 TOOLS AND TECHNOLOGY**

The application is developed in the Microsoft IDE named visual studio. This IDE has the supporting tools for the front end designing the platform for the writing the tax rate and tax amount calculation and back end for storing data in the SQL server. The supporting tools for the RASTAX implementation are also included in the session.

### Application IDE

- ASP.NET
- C#.net
- SQL

### RASTAX application

- Team viewer
- Team build
- VM ware

## **ASP.NET**

ASP.NET is the platform where the front end designing, the master page for the XML, the html operation in the tax generation are coded in the ASP.NET. This front end has the tools for the cache and N cache operation to get the latest updates from the source and the server machine with help of XML programming.

## **C#.net**

The middleware programming platform for the business logic coding are coded in the C#.net. this business logic of RASTAX is related to the tax calculation and the updating the SQL with the related tax with respect to the client business nature. The programming logic are coded in the class modules and they are inherited in the client system (raspberry pi) for the further tax calculations.

## **SQL**

The SQL is storage unit for the larger data , the small size data from the client machine and also to store the updated latest version control. The connection string is coded to connect with various SQL server from the different location with raspberrry pi and the main SQL server of chartered accountant

## **Team viewer**

The team viewer is the software used for the remote desktop access. If the client office or branch is located in the remote area for the application setting and installation of RASTAX will be done through the team viewer software. There by the developers can save the time for the travelling and manual effort for the installation. Also the team viewer can be used to fix error or network issues occurred during the tax calculation and mining.

## **Team build**

The software tool used for the building the application by integrating the raspberrry pi, the SQL server and visual studio. That is if the developers are needed to generate the value in RASTAX unit which is running in the Linux and the values are collected from the SQL server and the rule or tax percentage are gathered from the CA windows system the building will be complex and needed third party intermediate software. Team build will integrate all three system and build the application in the common platform.



## VM ware

The VM ware is the software used for the creating the virtual system architecture. The VM ware is mainly used by the software testing users. That is the module of RASAX which is programmed to run Linux module will be successfully executed in the Linux platform. But there can be technical issues when Linux and visual studio work mutually for the tax calculations. So developers will create the virtual setup of Linux in the windows OS and execute the module and make sure it is working properly before implemented in the final server.

### 2.4 HARDWARE AND SOFTWARE REQUIREMENTS.

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

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

#### Software requirements

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

## CHAPTER 3

### SOFTWARE REQUIREMENTS SPECIFICATION

#### 3.1 USERS TYPES

The user in the RASTAX application is are divided as users in the main associate organisation and the users in the remote client users. The admin user must be among the main associate users and the technical support users from the developers unit is also included in this application.

- Admin users
- Clerk associate
- Client employee
- Technical support engineer

##### Admin users

The user from the associates who have the supreme power to control the activities and decision taking in the application. This users as the director of the organisation, the accounts head, the chartered accountant are categories in this user level. The decision over creating client, client organisation controls, the tax calculation the bank details (source bank -client, the own bank – associate ) are created in the user controls.

<b>Module</b>	<b>Permission for View content</b>	<b>Permission for Edit content</b>	<b>Permission for Delete content</b>
Chartered associate module	Yes	Yes	Yes
Client machine module	Yes	Yes	No
Network server/ data management	Yes	No	No
Tax calculation	Yes	Yes	Yes

##### Clerk associate

The employee of associates who works in the tax calculation and client interactions. Even though the application auto generate the tax and report, there could be exceptional

possibilities like tax refund, the prepaid tax information, export duty refund etc which needed client employee communication.

<b>Module</b>	<b>Permission for View content</b>	<b>Permission for Edit content</b>	<b>Permission for Delete content</b>
Chartered associate module	Yes	No	No
Client machine module	Yes	Yes	Yes
Network server/ data management	No	No	No
Tax calculation	Yes	Yes	Yes

#### Client employee

The employees of the third party organisation have given the access to certain modules access. The modules for view the tax updates, status of the pending tax payment, important update form the associates regarding the pending payment ,the bank reconciliation statement etc are included in the client employee module.

<b>Module</b>	<b>Permission for View content</b>	<b>Permission for Edit content</b>	<b>Permission for Delete content</b>
Chartered associate module	Yes	No	No
Client machine module	Yes	Yes	Yes
Network server/ data management	No	No	No
Tax calculation	Yes	Yes	No

#### Technical support engineer

The user who are assigned to install he application in the clients machine, to manage the technical error occurs during the application running and also to give customer care support. These users do not have access in the client data or server or in the tax calculation modules.

These users will be working to ensure the application performance assurance and keep the data integration among the varies machine block less for the fast tax calculation.

### 3.2 FUNCTIONAL REQUIREMENTS

Functional requirement for create organisation

Requirement number	Text box name	Input	Description	Output
1	TXT_ORG_NAME	Name of organization	Name of the clients organization	Check required filed validate and save in SQL
2	LBL_ORD_ID	NA (read only)	Auto generate ID for organisation to avoid confusion if there is two company with same name	Save the ID in SQL server
3	TXT_BUSINESS_MANAGER	Name of user in charge of organisation	Check the validation and no special characters allowed , maximum 50 letter	Display and save name in SQL
4	DDL_REGISTRATION	Auto fill ID in combo box,	data are fetched the SQL server loaded with registered company under ministry	Select and short list the ID in the combo box
5	CAL_CONTRACT_DATE	Select calendar with auto selected current date	To save the date of registration is been made.	Store the date in the SQL
6	CAL_CONTRACT_END_DATE	Auto select contract end date	Auto generate one year as expiry. The contract will remain for one year and	Auto select the date.

			needed to renew	
7	TXT_ trade allowance	Manual enter if allowance is eligible	Only integers, no characters or special characters allowed	Display the rate
8	RD_NON_TAXABLE	Radio button if non taxable (0% GST) is permitted	Manual select the radio button	Save the 0 percentage in SQL
9	TXT_REG_FEE	Enter the amount	First time payment with associates	Display the rate in the SQL server

### Functional requirement for general tax

Requirement number	Text box name	Input	Description	Output
1	LBL_SALES_ID	Auto generated sales bill number	Sales invoice ID for the tax calculation reference	Generate the sales ID
2	TXT_DATE	Text box fill with current system date	Date of sales billing	Save date
3	CAL_DATE	Link button with calendar pop up	If the user wants to select new date other than current date click link button and select	Display date
4	DDL_TAX_TYPE	Select tax type	Type included VAT, or GST or output tax etc	Display the selected tax
5	Radio permit classification	Radio button with YES or NO	For bigger organisation more tax feature like carried forward,	SAVE YES or NO

			prepaid tax etc will be included. User need to click YES for special cases	
6	REMARK	Enter remark	Multi –text box with maximum 500 words permitted	Save the remark

### 3.3 NON FUNCTIONAL REQUIREMENTS

The non functional requirement is prepared by the developers to ensure the RASTAX software’s final product property, the future upgrading and business expansion possibilities. This non functional requirement is carried in the areas of

- Integrity
- Reliability
- Security
- Flexibility

#### Integrity

The integrity of the software will ensure the raspberry pi and the ASP.NET application communication. The steps needed to be take for the ASP and non ASP application interaction , the communication between the Linux and windows will show the application integrity

#### Reliability

This application is based on the tax report generation, so application is coded to ensure that all the report generated will be clear and without mistake. Particularly in the area of tax carried forward or VAT time tax pending etc are given more importance. Because a small mistake in the area of calculation will generate huge loss for the government or for the client who wants to pay taxes.

#### Security

The detail which are loaded in the SQL server has banking information, the client details, the mode of revenue generated, the source of income generated etc. These details are confidential

and the clients do not want to share with third party. So RASTAX application will give high importance in the data security and data access.

### Flexibility

Application flexibility in the tax calculation and the report generations, the single RASTAX application can be used for VAT based tax generation, the GST calculation, the employee income tax, tax from other source, the export duty calculation etc. Single RASTAX unit will connect multiple server to generate the required tax report.

# CHAPTER 4

## SYSTEM DESIGN

### 4.1 SYSTEM PERSPECTIVE

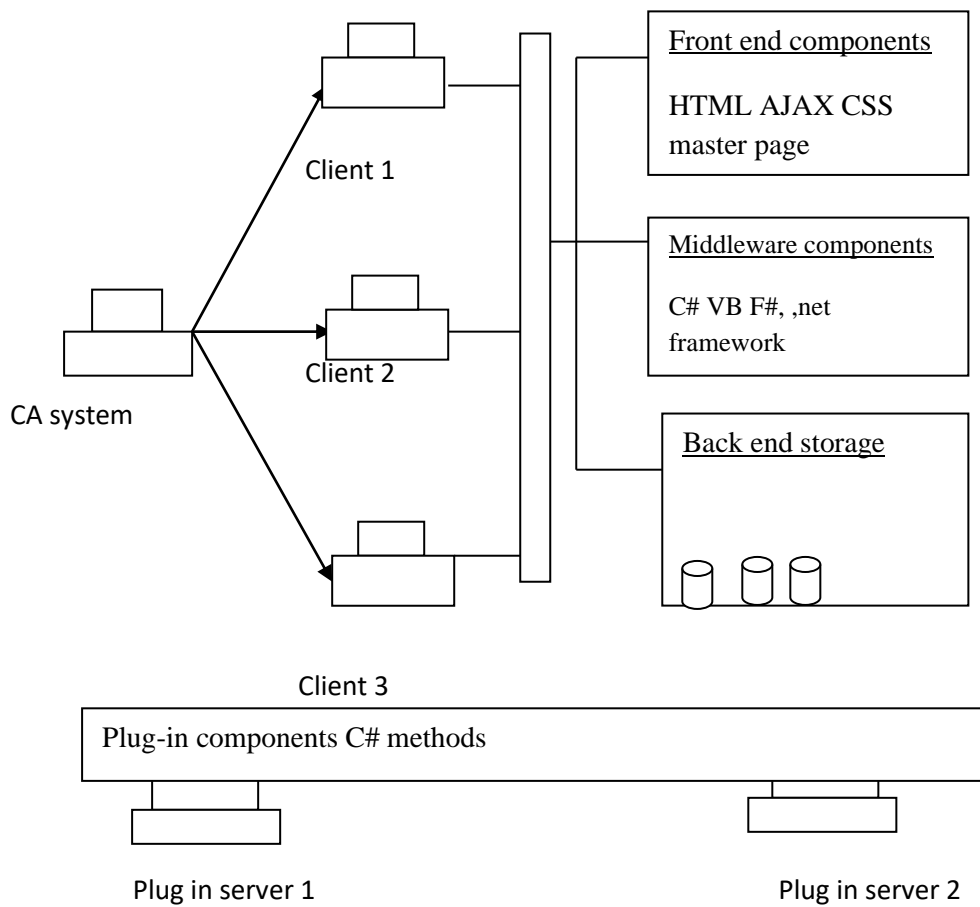


Fig 4.1: the architecture diagram for CA system the client and plug in server.

The developing platform for the CA and the client are created in three tier architecture. The plug n is connected network layer with C# components of network communication. The program for the tax and other business logic is developed in the main three architecture



## 4.2 CONTEXT DIAGRAM

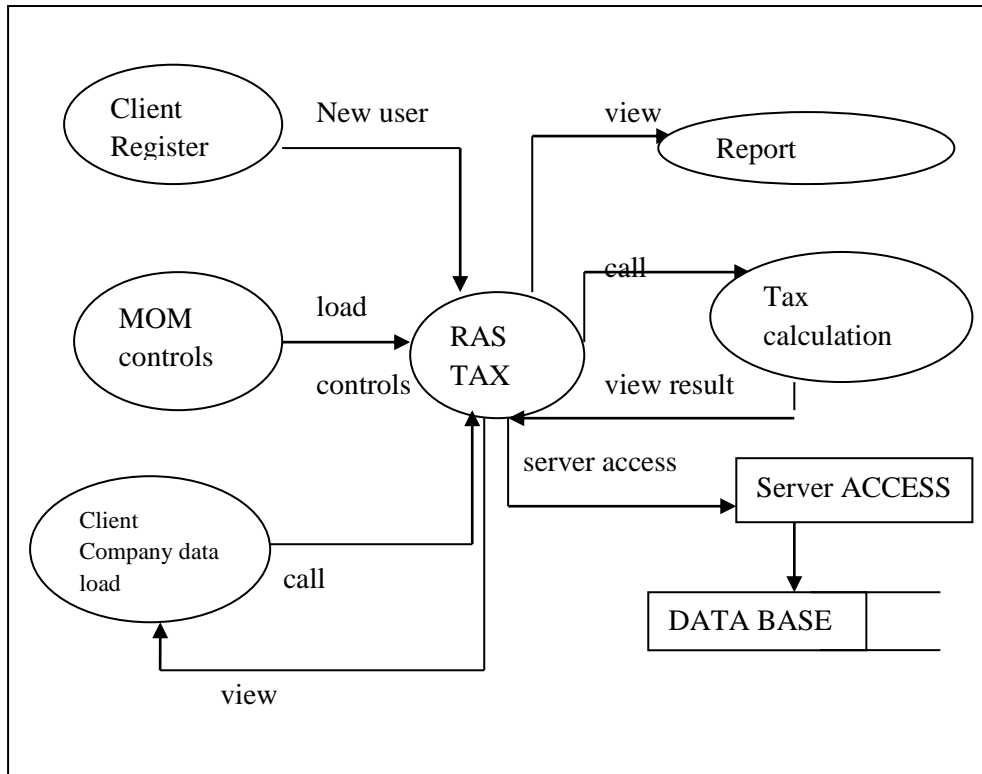


Fig 4.2: context diagram for the RAS tax integrations

The context diagram for the accounting plug in operations with other application objects are explained in the figure 4.2. the MOM is the network based architecture created for the loading control in the IDE, this MOM will function in the security, the storage and encryption of data integration with all the tax related module.

# CHAPTER 5

## DETAILED DESIGN

### 5.1 USE CASE DIAGRAM

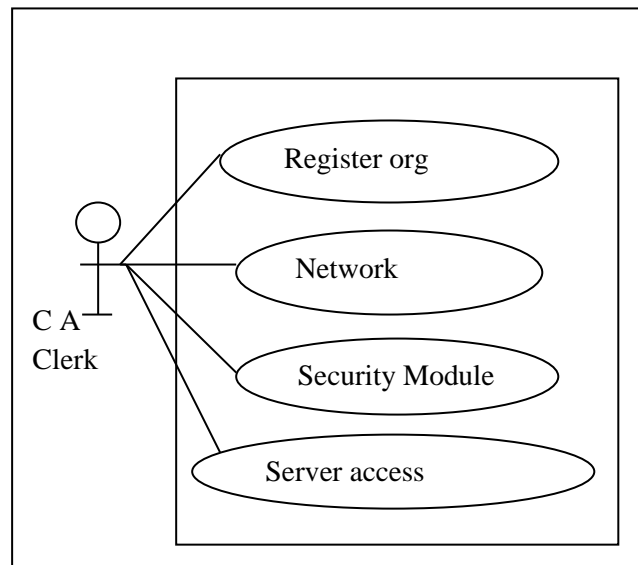


Fig 5.1.1: use case of CA clerk

The CA employee or clerk has the activities over the shop registration and the server access data needed for the create origination to communicate with RASTAX plug in software. This access of the clerk to the server is made with application authorization module and the data integration is in the server access and the organisation is permitted.

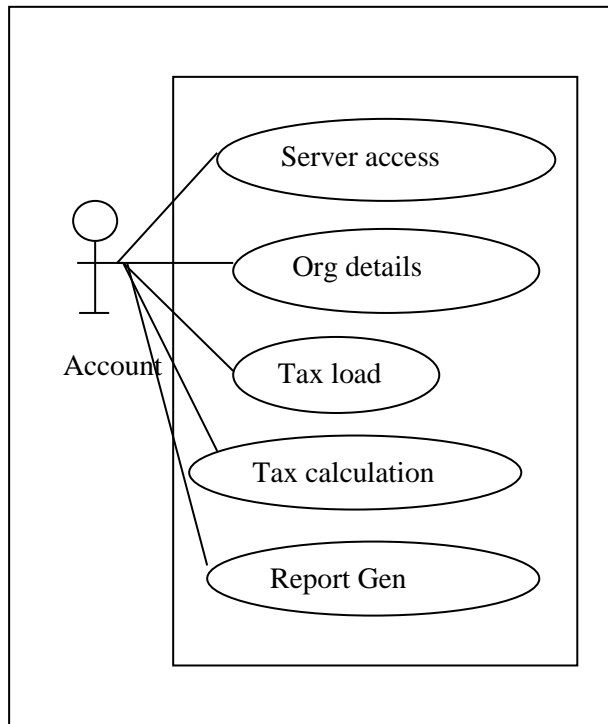


Fig 5.1.2: use case for accountant .

The tax related operations are created by the accountant users. These users can access the server of the client where the business transactions which are needed for the tax calculations are loaded. The tax based details will be loaded in the page and used for the tax report generations.

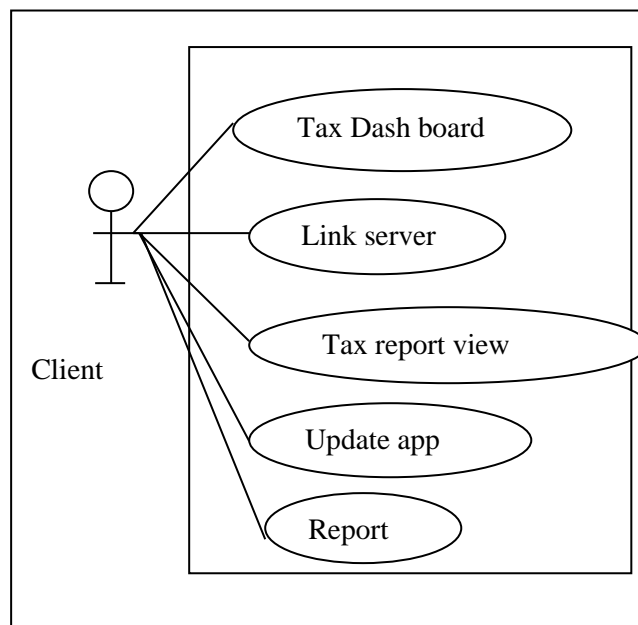


fig 5.1.3: use case for the client operations

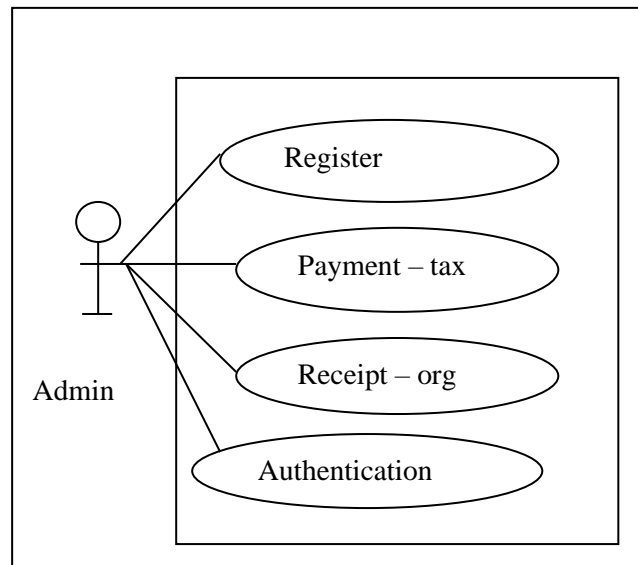


Fig 5.1.4: use case for the admin

The client is the user for who the RASAX plug in is used for the tax generation. These users have the tax dash board where the updates regarding the tax generation and the rules will be loaded. The updates form the government regarding the latest tax rules and the other tax updates are given for the users.

## 5.2: SEQUENCE DIAGRAM

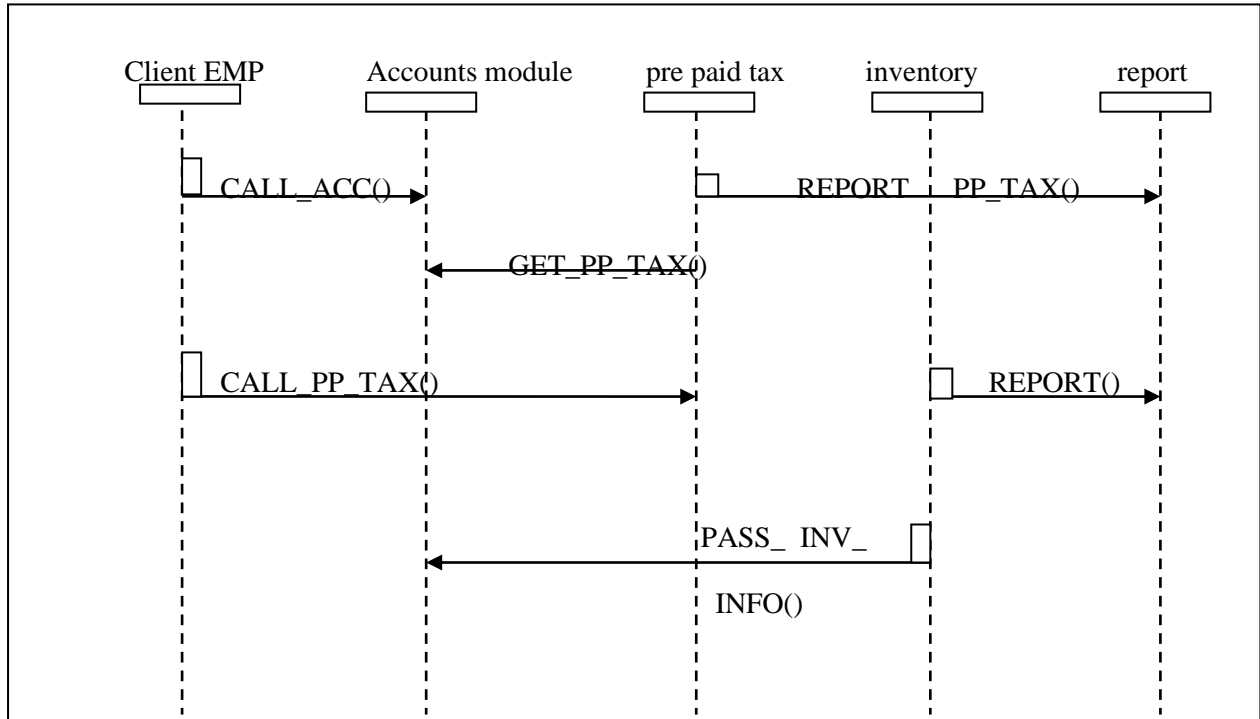


Fig 5.2.1: Sequence diagram for accounting head

The accountant head works in the pre paid tax related objects. The prepaid tax work is the first state of work to be carried out and the detail to be loaded in the server before making the current year tax transactions. The work over create client employee, fetch the pre paid and the inventory based data to the created client employee are added.

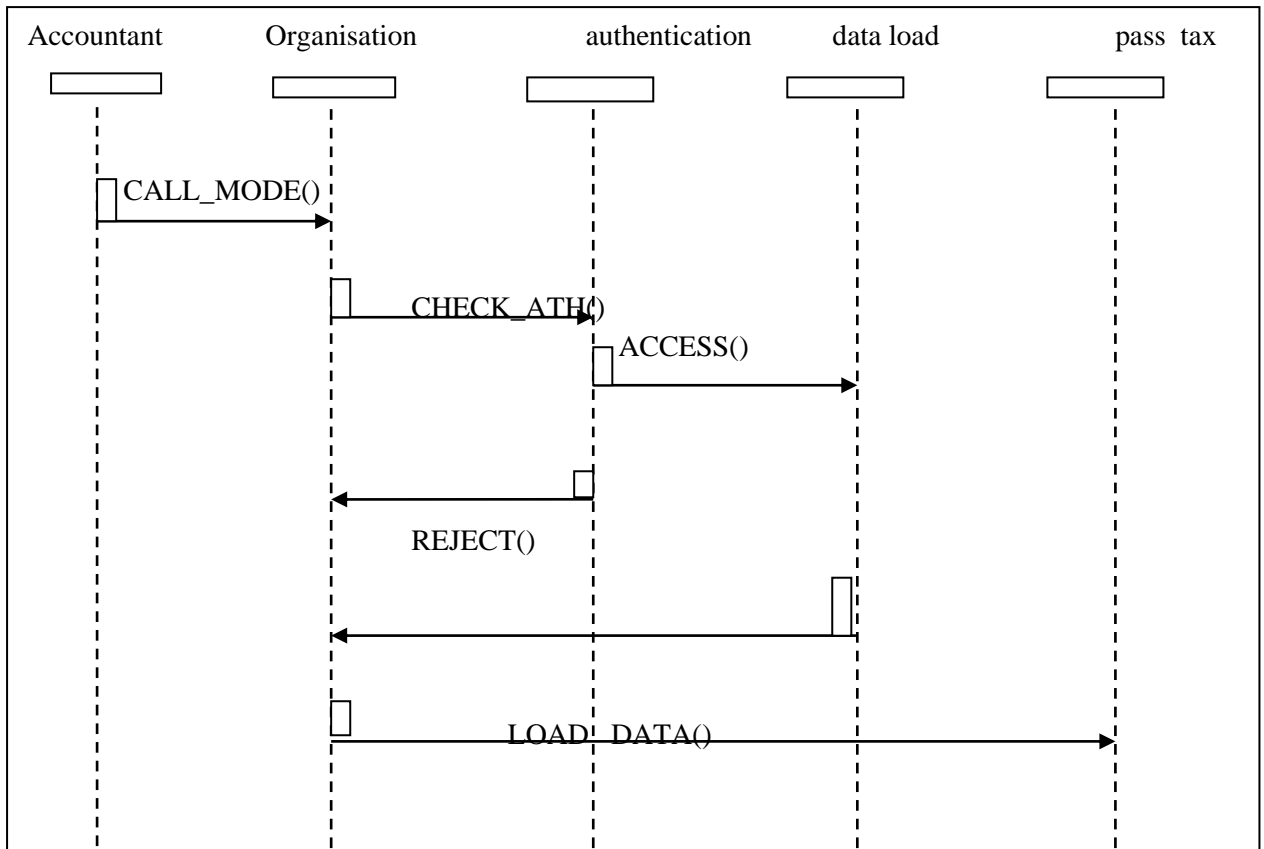


Fig 5.2.2: the sequence for the data loading by authentication check

The accountant will call the organisation module access and based on the accountant authorisation the data will be loaded. For example if the accountant is for tax value calculation then data loaded will be related to the tax. And if the accountant is for non tax like billing and inventory process the data loaded will be related to the billing

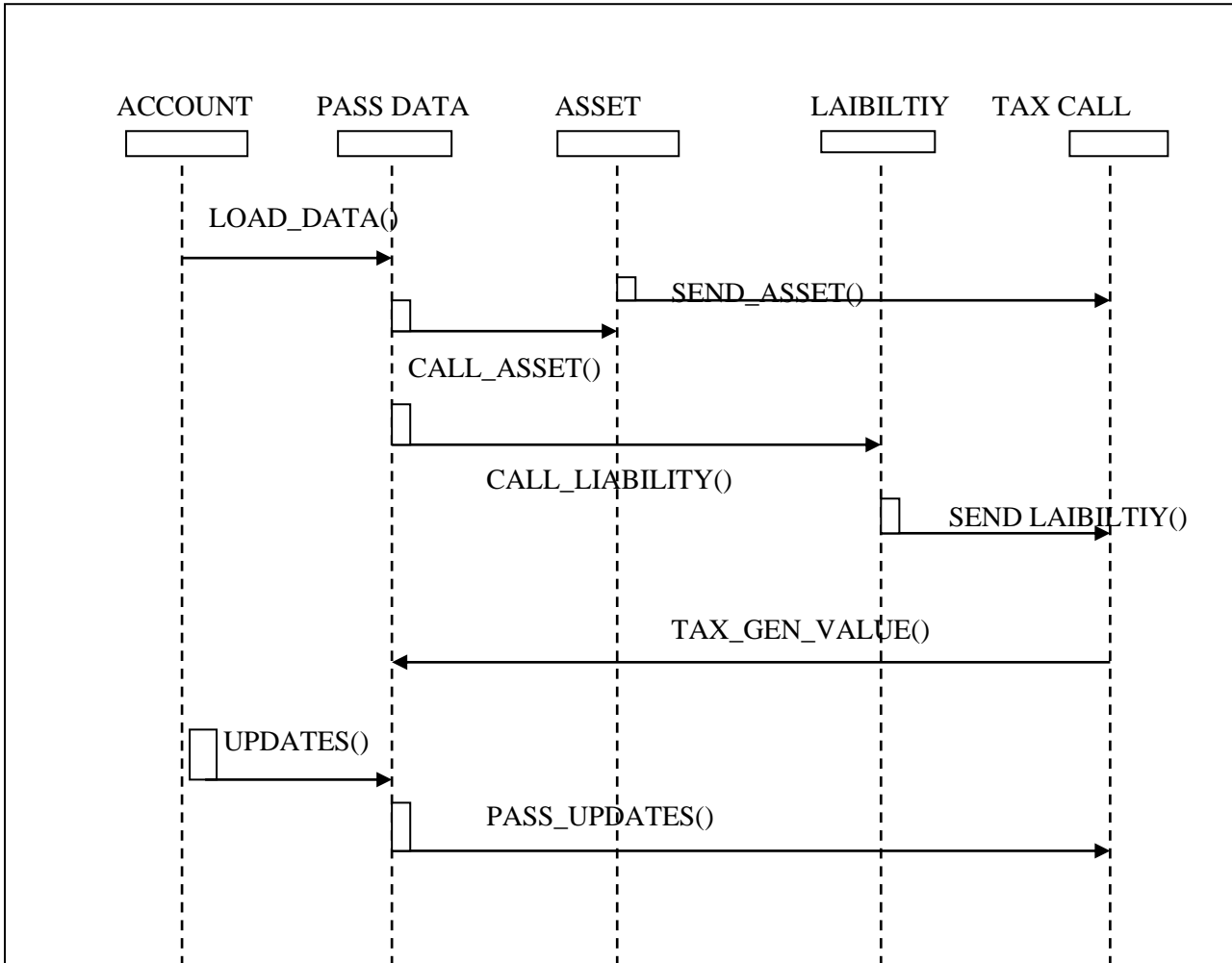


Fig 5.2.3: sequence for tax generation calling.

After the success full data load for the accountant these loaded data will be rearranged to the asset and liability part group by RASTAX software. This will help to identify where the client' organization is under loss or profit. After analysis the profit and loss the tax call method will be link with loaded data for the report.

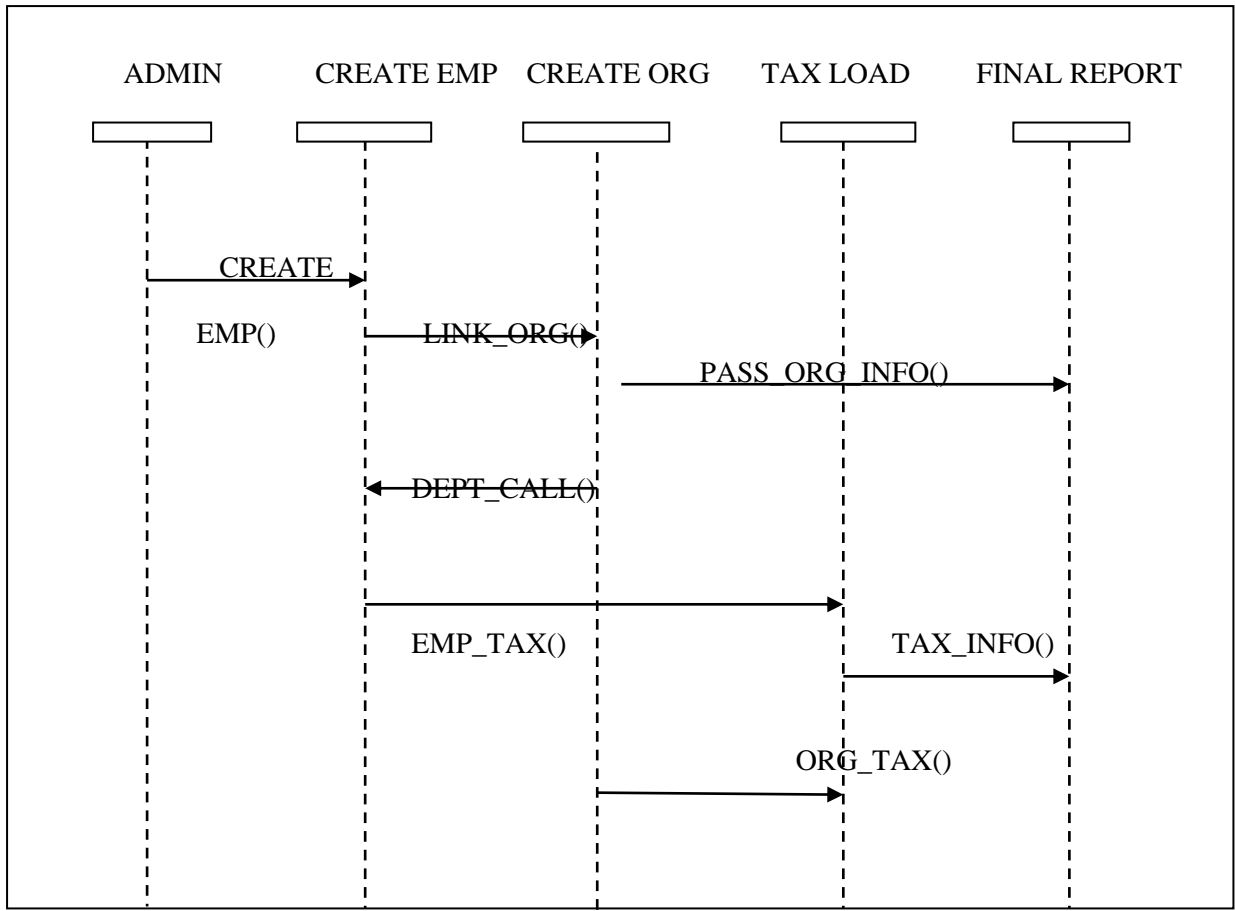


Fig 5.2.4: EMP and Organisation based tax generation

The RASTAX is coded to generate tax for organisation also for the each employee too. The data of organization , the tax the asset , the liability and the other employee related data are saved in different system which is connected together by admin with RASTAX application. the RASTAX application interface is carried to generate the tax value report for each origination and the employee separately.



### 5.3 COLLABORATION DIAGRAM

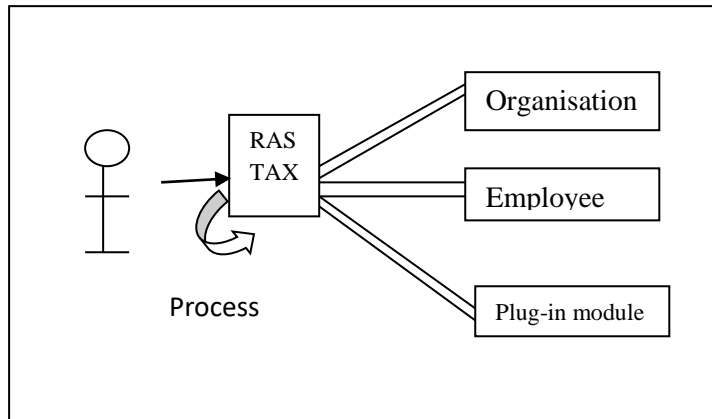


Fig 5.3.1: the collaboration diagram for RAS TAX integration

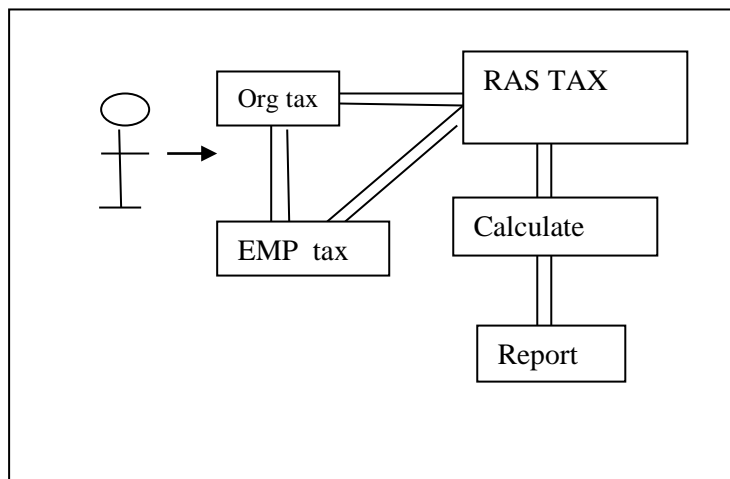


Fig 5.3.2: the collaboration diagram for RAS TAX integration each employee to the organisation where he belongs

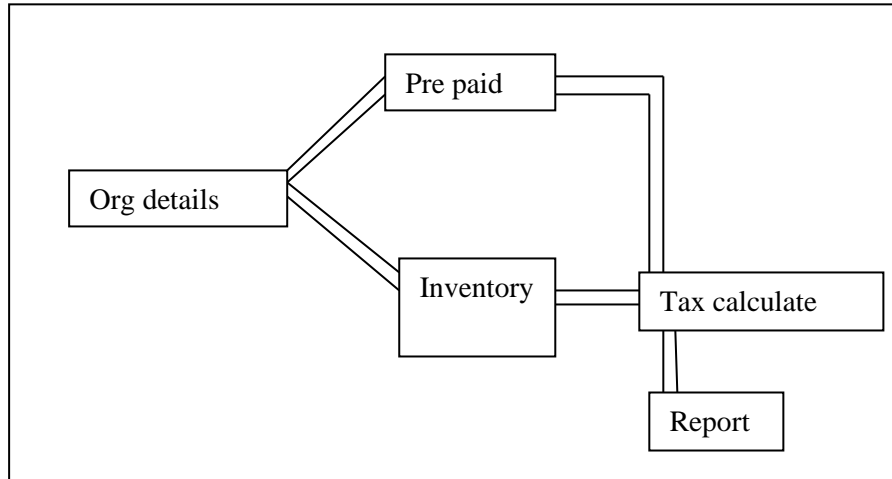


Fig 5.3.3: the collaboration diagram for RAS TAX integration in organisation

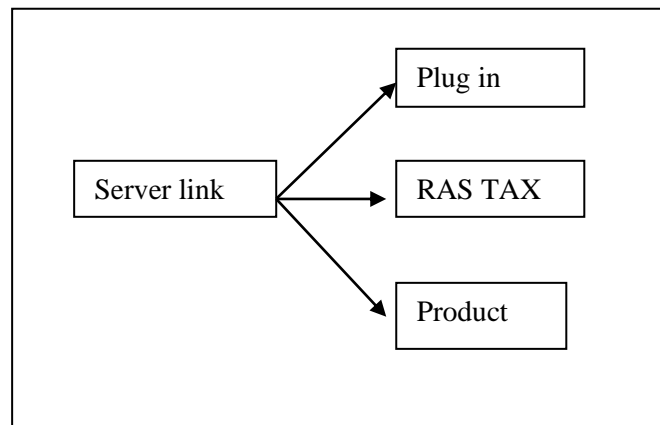


Fig 5.3.4: the collaboration diagram for product link

## 5.4 ACTIVITY DIAGRAM

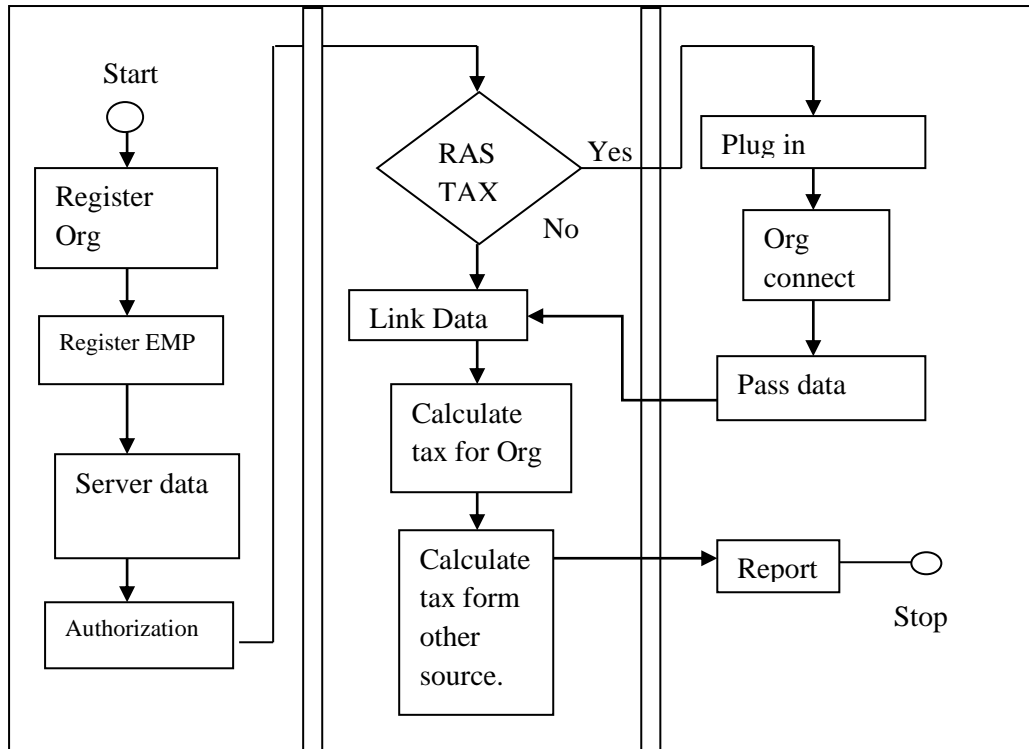


Fig 5.4.1: the activity diagram for employee tax generation with RASTAX plug in

The tax generation of the employee is depends in the income generated per year, the tax percentage will be varied form 2.5 percentage to t 30 % depends of the employee. The application is programmed to check and other source of income for the employee. The final report generated will be reconciliation of all the income and the tax payable.

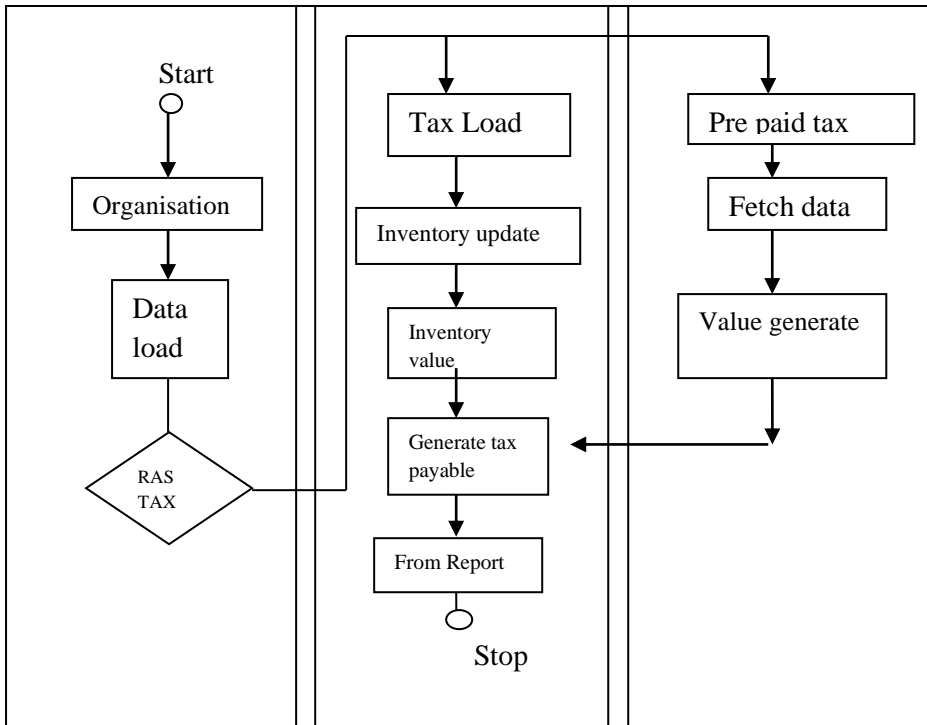


Fig 5.4.1: the activity diagram for organisation tax report generation.

The tax calculation for the an organisation needed the data form the pre paid tax, the organisation inventory related data, the server connection to the organisation and the government based server and access the data for the final report generation.

## 5.5 DATABASE DESIGN

Table name : DBO . CM\_SRN\_RS\_

Data type	Parameters	Constraints
CM_SRN_RS_TAX_ID	INT	PRIMARY KEY
CM_SRN_RS_ORG	VARCHAR (21) ,	NA
CM_SRN_RS_REG_TYPE	VARCHAR (25) ,	NA
CM_SRN_RS_BANK_LINK	VARCHAR (11)	NA

Column Name	Data Type	Allow Nulls
CM_SRN_RS_TAX_ID	int	<input type="checkbox"/>
CM_SRN_RS_ORG	varchar(21)	<input checked="" type="checkbox"/>
CM_SRN_RS_REG_TYPE	varchar(25)	<input checked="" type="checkbox"/>
CM_SRN_RS_BANK_LI...	varchar(11)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Table name : DBO . CM\_SRN\_RS\_ADV\_INC\_TAX

Data type	Parameters	Constraints
CM_SRN_RS_TAX_ID	INT	PRIMARY KEY
CM_SRN_RS_TDS_38	FLOAT,	NA
CM_SRN_RS_TDS_3C,	FLOAT	NA
CM_SRN_RS_TOTAL_BENI	FLOAT,	NA
CM_SRN_RS_REMARK	VARCHAR (251)	NA

Column Name	Data Type	Allow Nulls
CM_SRN_RS_TAX_ID	int	<input type="checkbox"/>
CM_SRN_RS_TDS_38	float	<input checked="" type="checkbox"/>
CM_SRN_RS_TDS_3C	float	<input checked="" type="checkbox"/>
CM_SRN_RS_TOTAL_...	float	<input checked="" type="checkbox"/>
CM_SRN_RS_REMARK	varchar(251)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Table name : DBO . CM\_SRN\_RS\_BENIFITS

Data type	Parameters	Constraints
CM_SRN_RS_TAX_ID	INT	PRIMARY KEY
CM_SRN_RS_TOTAL_BEN,	FLOAT	NA
CM_SRN_RS_INT_BEN	FLOAT	NA
CM_SRN_RS_BEN_PAYABLE	FLOAT	NA
CM_SRN_RS_BEN_REMARK	VARCHAR (125)	NA

Column Name	Data Type	Allow Nulls
CM_SRN_RS_TAX_ID	int	<input type="checkbox"/>
CM_SRN_RS_TOTAL_BEN	float	<input checked="" type="checkbox"/>
CM_SRN_RS_INT_BEN	float	<input checked="" type="checkbox"/>
CM_SRN_RS_BEN_PAYABLE	float	<input checked="" type="checkbox"/>
CM_SRN_RS_BEN_REMARK	varchar(125)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Table name : DBO. CM\_SRN\_RS\_ADV\_INC\_TAX

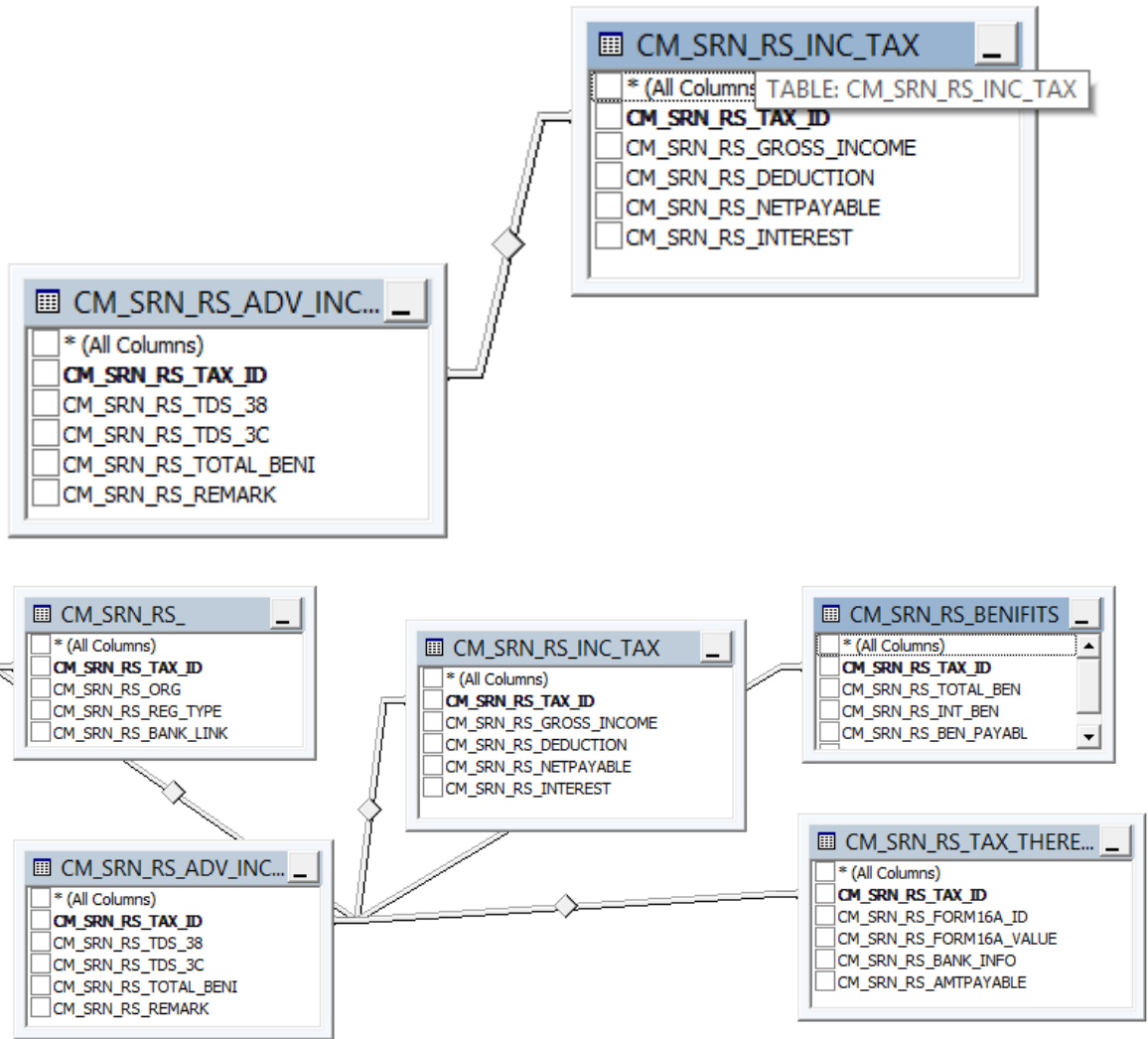
Data type	Parameters	Constraints
CM_SRN_RS_TAX_ID	INT	PRIMARY KEY
CM_SRN_RS_TDS_38	FLOAT	NA
CM_SRN_RS_TDS_3C	FLOAT	NA
CM_SRN_RS_TOTAL_BENI	FLOAT,	NA
CM_SRN_RS_REMARK	VARCHAR (251)	NA

Column Name	Data Type	Allow Nulls
CM_SRN_RS_TAX_ID	int	<input type="checkbox"/>
CM_SRN_RS_GROSS_INCOME	float	<input checked="" type="checkbox"/>
CM_SRN_RS_DEDUCTION	float	<input checked="" type="checkbox"/>
CM_SRN_RS_NETPAYABLE	float	<input checked="" type="checkbox"/>
CM_SRN_RS_INTEREST	float	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

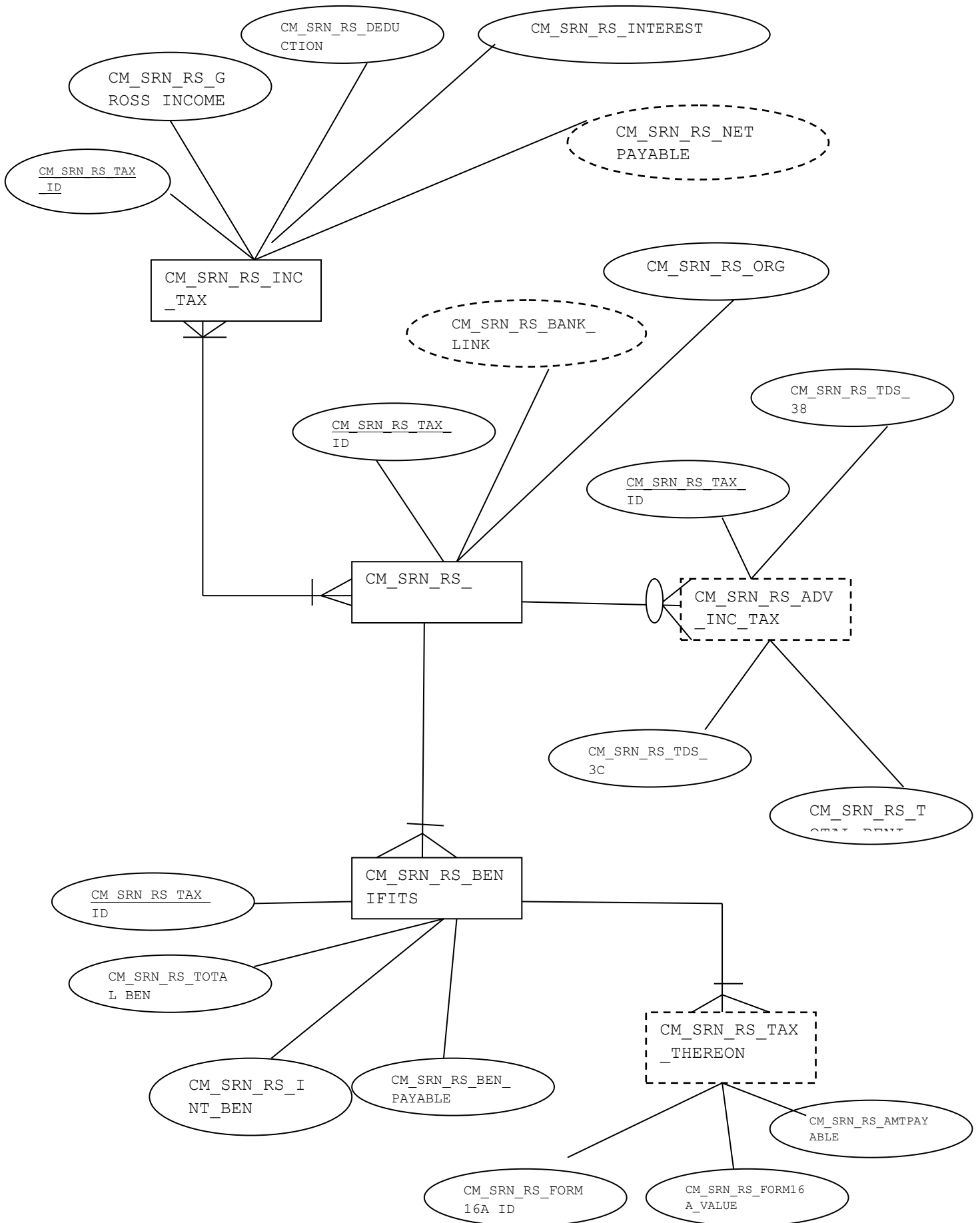
Table name : DBO. CM\_SRN\_RS\_TAX\_THEREON

Data type	Parameters	Constraints
CM_SRN_RS_TAX_ID	INT PRIMARY KEY,	PRIMARY KEY
CM_SRN_RS_FORM16A_ID	CHAR (7) ,	NA
CM_SRN_RS_FORM16A_VALUE	FLOAT,	NA
CM_SRN_RS_BANK_INFO	VARCHAR (63) ,	NA
CM_SRN_RS_AMTPAYABLE	FLOAT	NA

Column Name	Data Type	Allow Nulls
CM_SRN_RS_TAX_ID	int	<input type="checkbox"/>
CM_SRN_RS_FORM16A_ID	char(7)	<input checked="" type="checkbox"/>
CM_SRN_RS_FORM16A_VALUE	float	<input checked="" type="checkbox"/>
CM_SRN_RS_BANK_INFO	varchar(63)	<input checked="" type="checkbox"/>
CM_SRN_RS_AMTPAYABLE	float	<input checked="" type="checkbox"/>
		<input type="checkbox"/>



## 5.6 ER diagram





# CHAPTER 6

## IMPLEMENTATION

### 6.1 SCREENSHOT

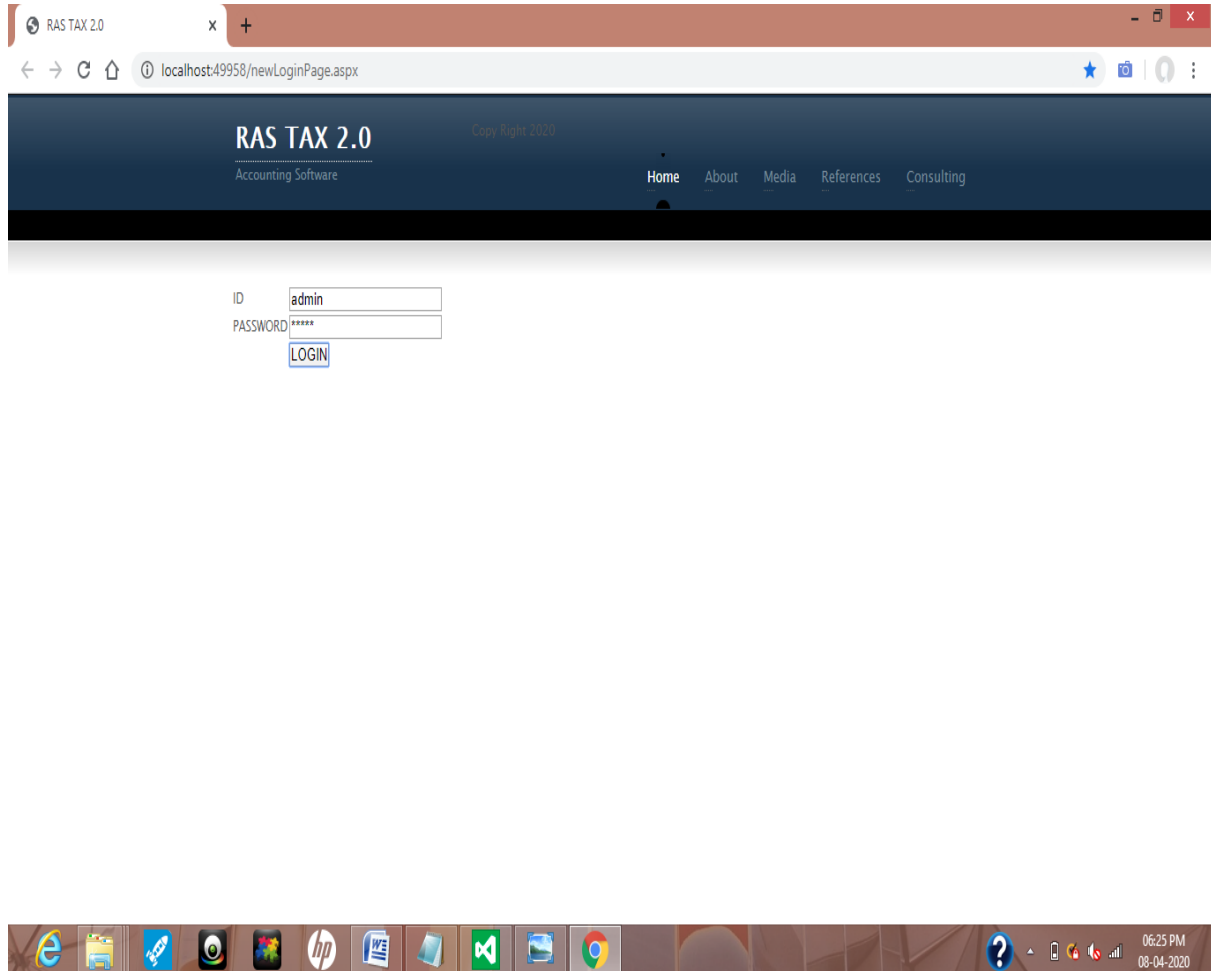


Fig 6.2.1: the login page for the RASTAX plug in

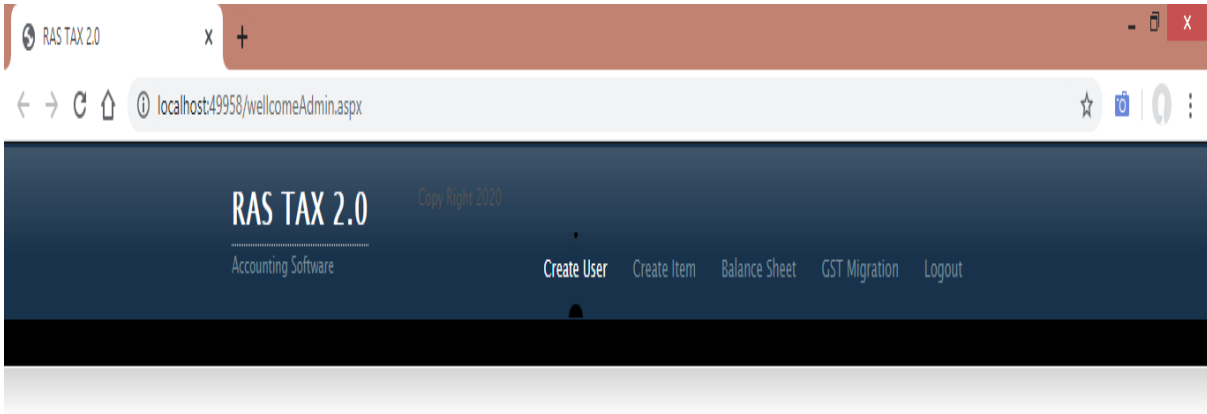
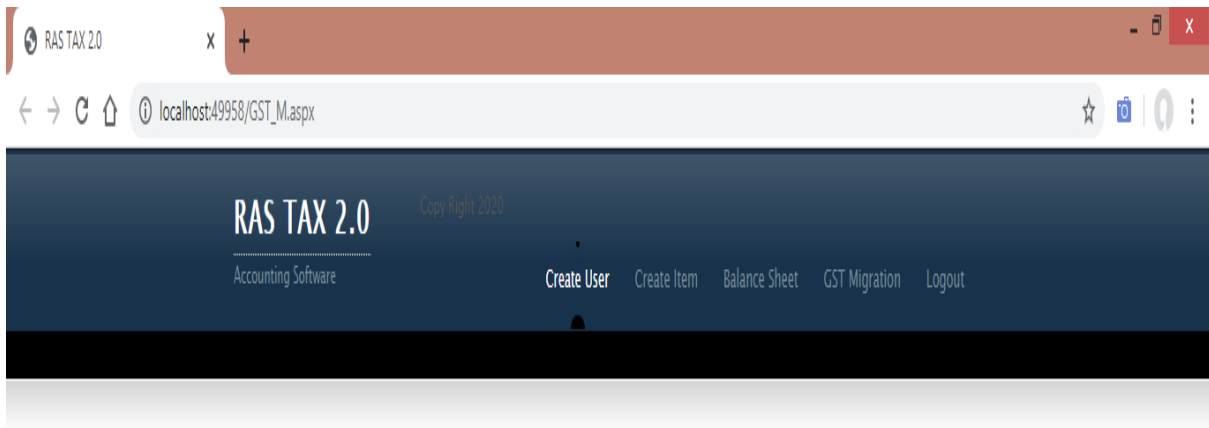


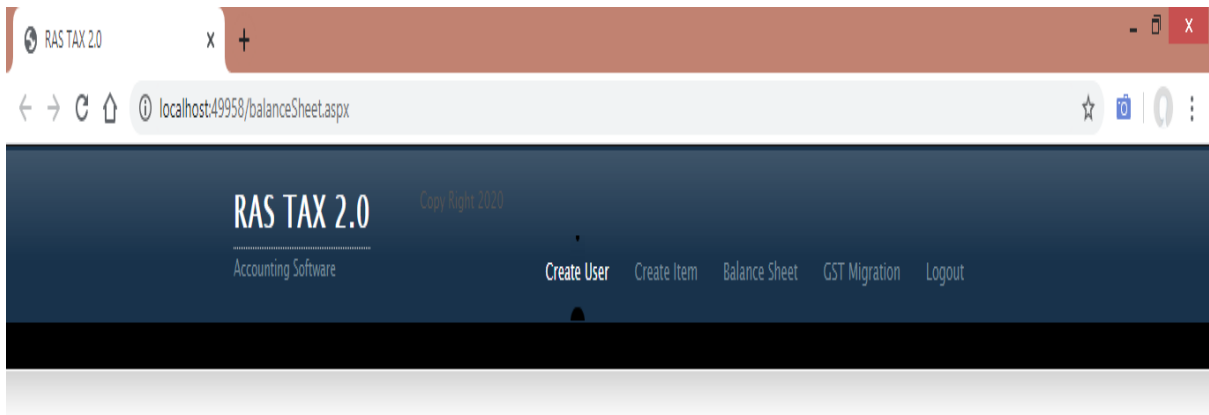
Fig 6.2.2: the welcome page for RASTAX.



enter IP



Fig 6.2.3: user enter the IP address of the remote system (client machine) to import clients organisation details for tax calculations.



MIGRATE			
ASSET		LIABILITY	
CASH	0	CAPITAL	0
CURRENT ASSET		CURRENT LIABILITY	0
SALES		PURCHASE	
STOCK INVESTMENT	0	TAX PAYABLE	0
FIXED ASSET	0	EQUITY	0
MACHINE	0	CAPITAL STOCK	0
RECEIVABLE AMOUNT		PAYABLE AMOUNT	
TOTAL	0	TOTAL	0



Fig 6.2.4: by default the asset side and liability side of the RASTAX will be filled with zero, when the user clicks the migrate button the application will calculate the total asset and liability and if the both end matches then data will be fetched form the server and loaded in the users portal.

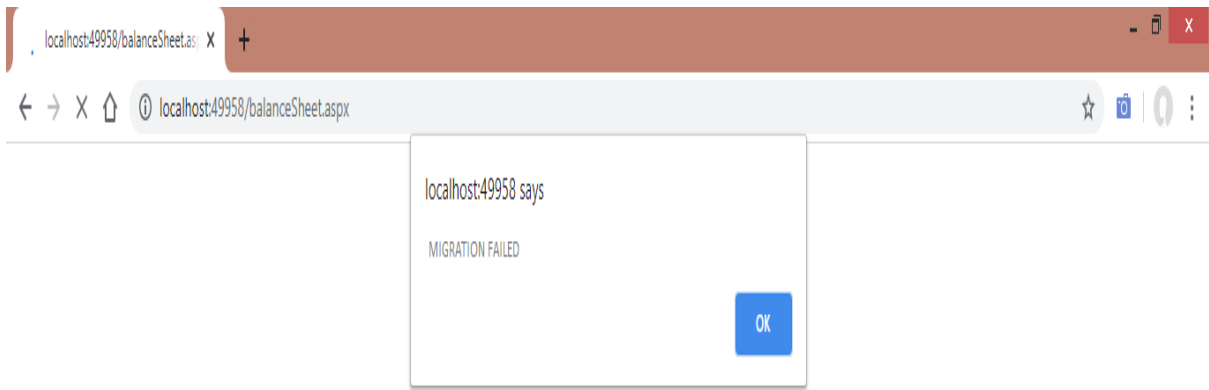
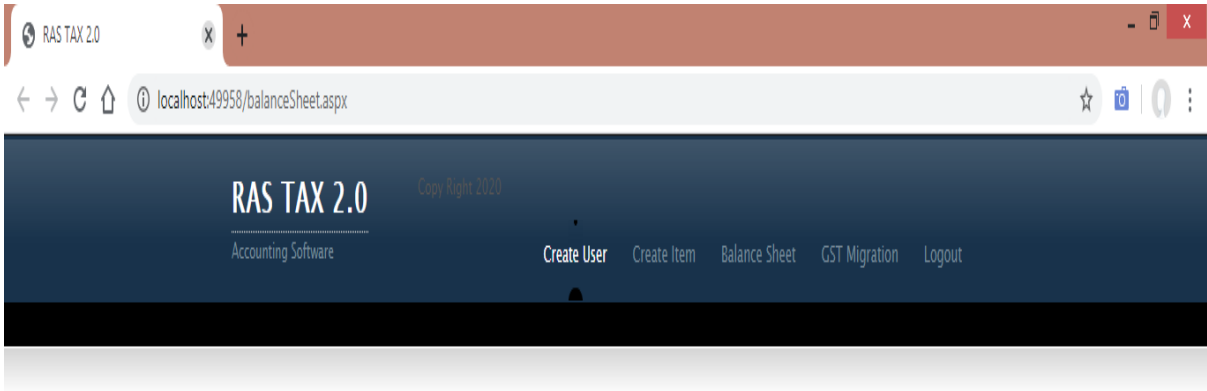


Fig 6.2.5: migration failed message showed. When the user clicks the OK button the page will navigate to previous page.



MIGRATE			
ASSET		LIABILITY	
CASH	1000	CAPITAL	950
CURRENT ASSET		CURRENT LIABILITY	0
SALES		PURCHASE	
STOCK INVESTMENT	0	TAX PAYABLE	0
FIXED ASSET	0	EQUITY	0
MACHINE	0	CAPITAL STOCK	0
RECEIVABLE AMOUNT		PAYABLE AMOUNT	
<b>TOTAL</b>	<b>1000</b>	<b>TOTAL</b>	<b>934</b>



Fig 6.2.6: RASPLX plug in application show the total asset and liability of selected client organisation do not match. Now the chartered accountant needed to open the client machine accessing through team view or by directly accessing the data.

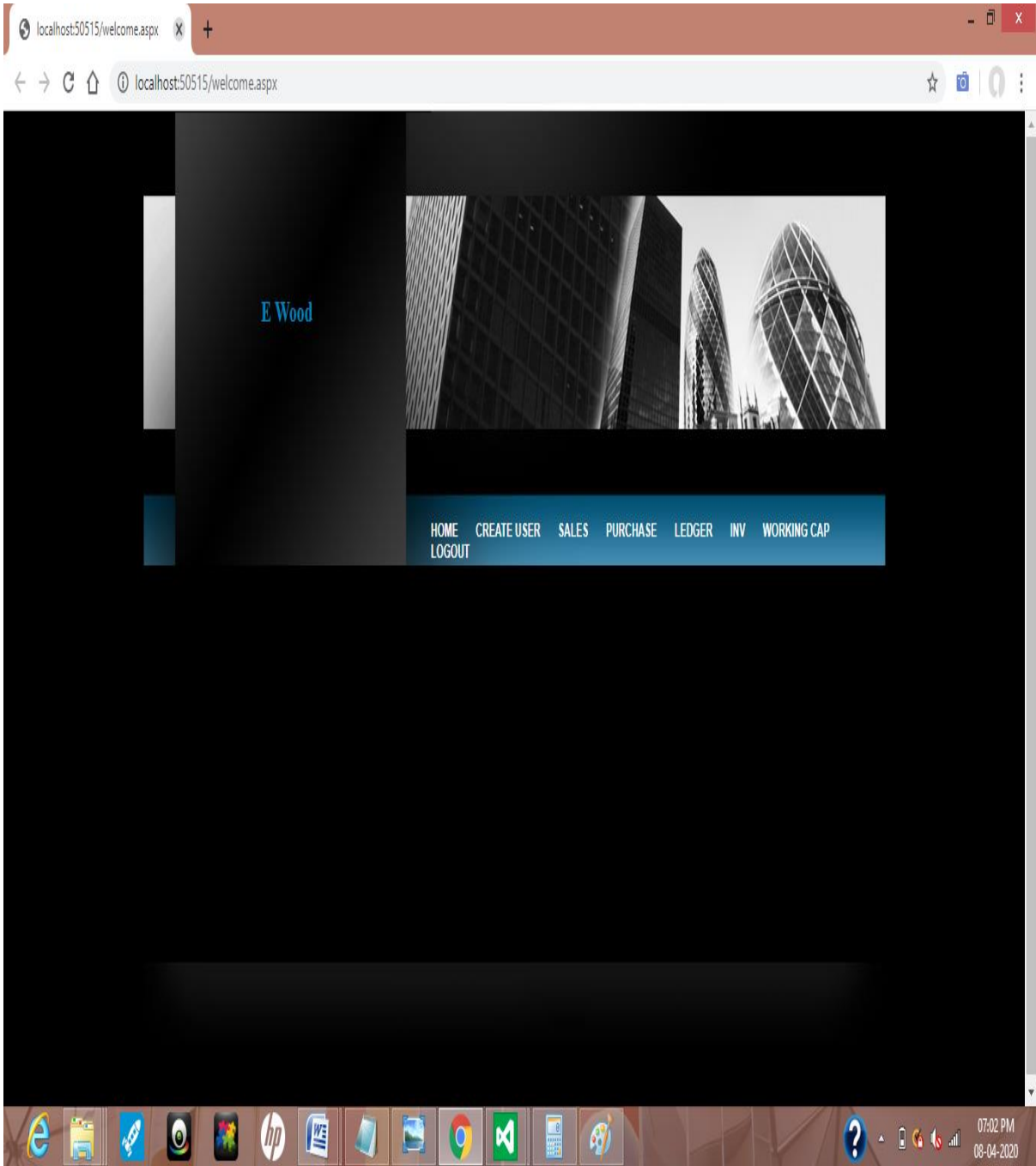


Fig 6.2.7: the real page of client with different master page design form RASTAX.

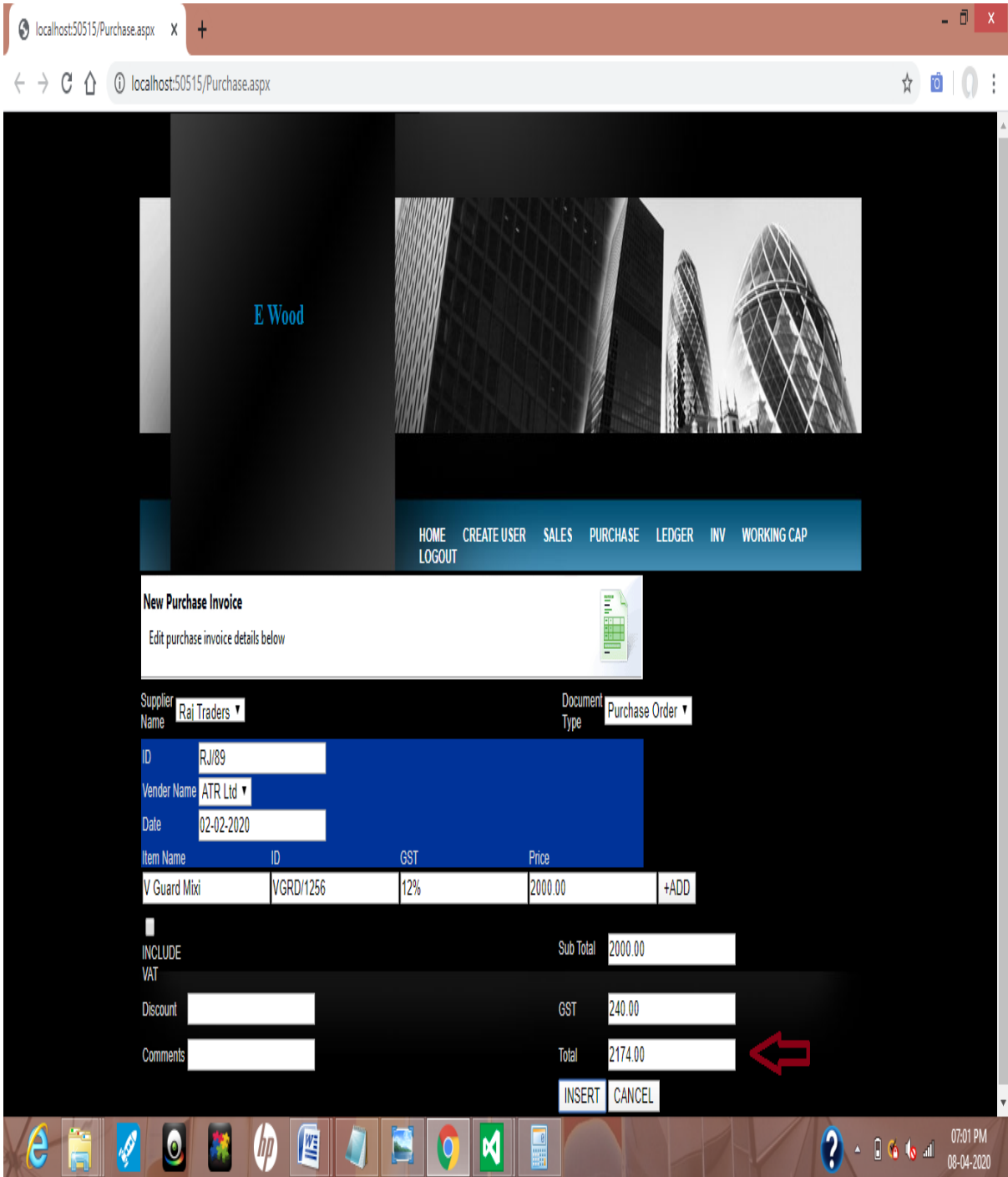


Fig 6.2.8: When the accountant explore the application the user found the error in the total calculation where the discount give was marked as advanced paid. And the balance amount is carried forward for the next month which never occur made.



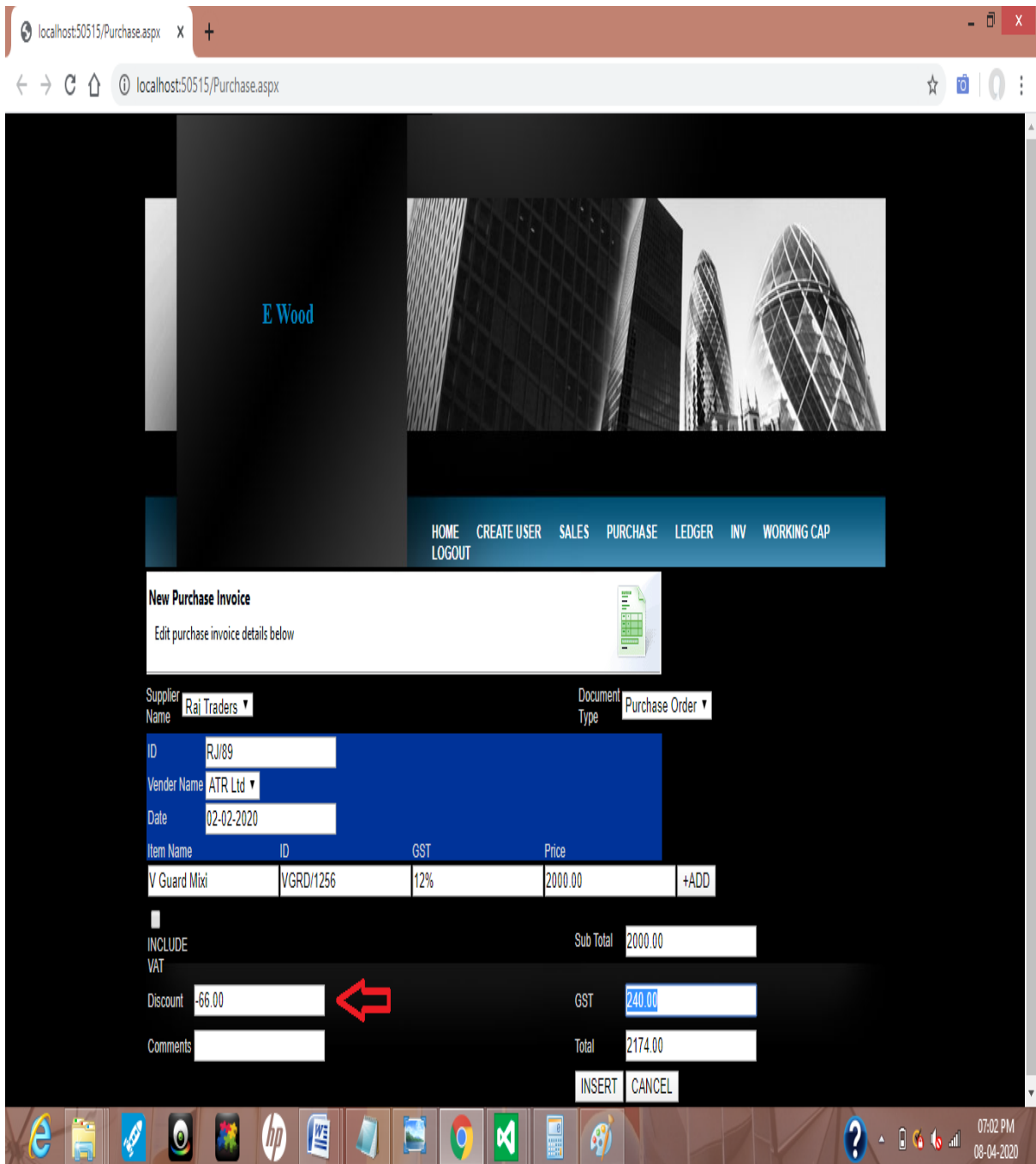


Fig 6.2.9: now the user updated 66 rupees has discount given and the carried forward of 66.00 is saved in the account of discount.

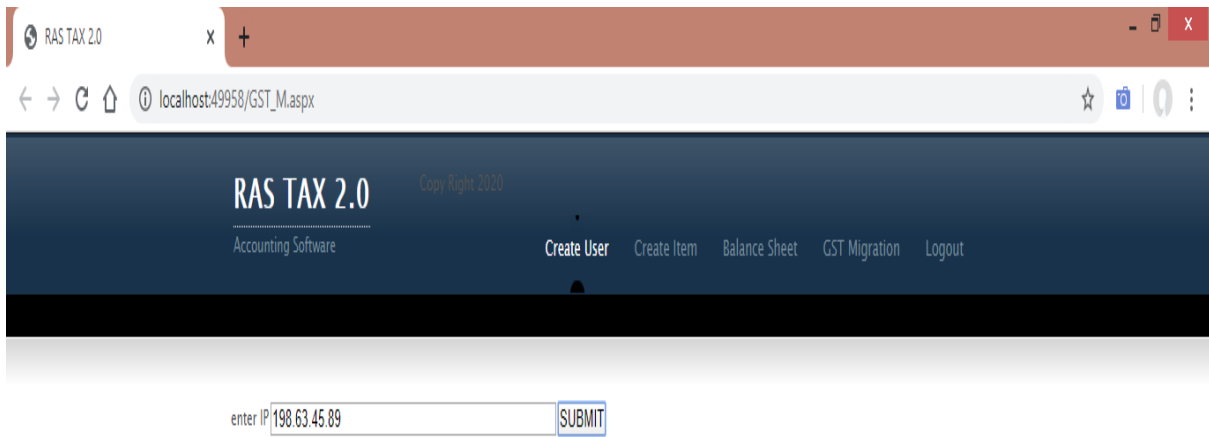


Fig 6.2.10: after making the update the users again enter the IP address and data are called for the tax calculations.

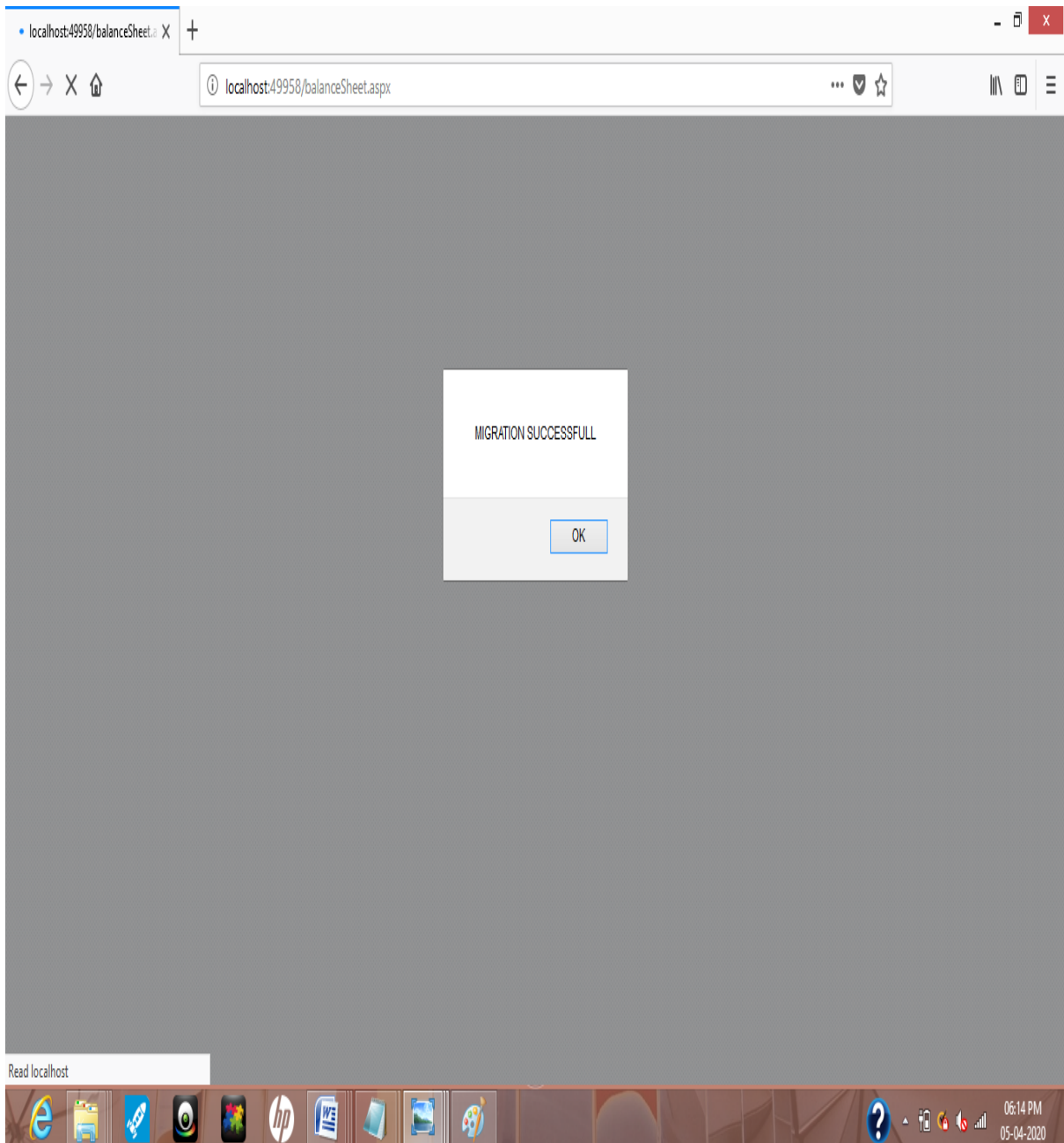
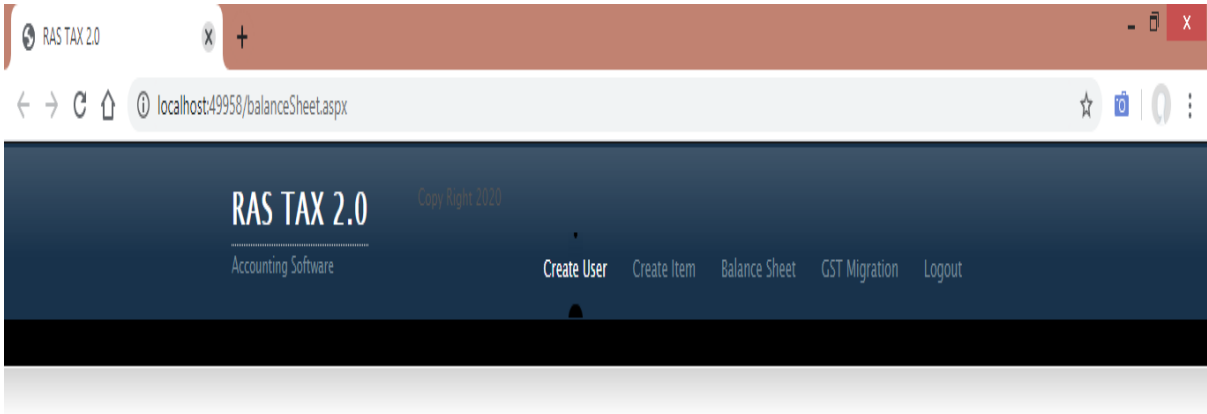


Fig 6.2.11: this time the migration is successful and the data from the connected SQL server will be migrated to CA server or the centralised server there all accountant system is connected



MIGRATE			
ASSET		LIABILITY	
CASH	1000	CAPITAL	1000
CURRENT ASSET		CURRENT LIABILITY	0
SALES		PURCHASE	
STOCK INVESTMENT	0	TAX PAYABLE	0
FIXED ASSET	0	EQUITY	0
MACHINE	0	CAPITAL STOCK	0
RECEIVABLE AMOUNT		PAYABLE AMOUNT	
<b>TOTAL</b>	<b>1000</b>	<b>TOTAL</b>	<b>1000</b>



Fig 6.2.12: this time total liability and asset shows balanced.

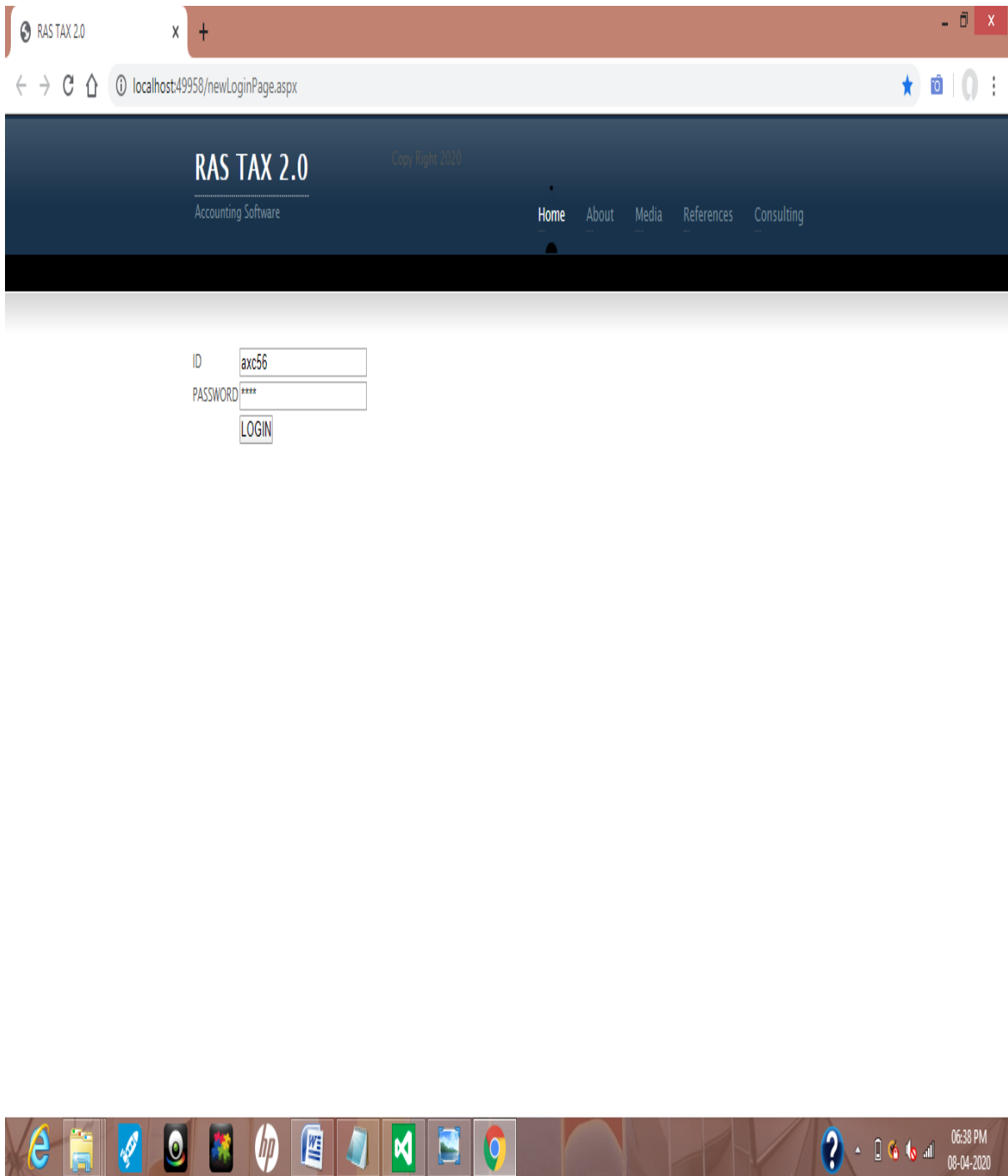


Fig 6.2.13: After the successful migration of the data the duty of the accountant will start to generate the tax generation calculation. Accountant login page

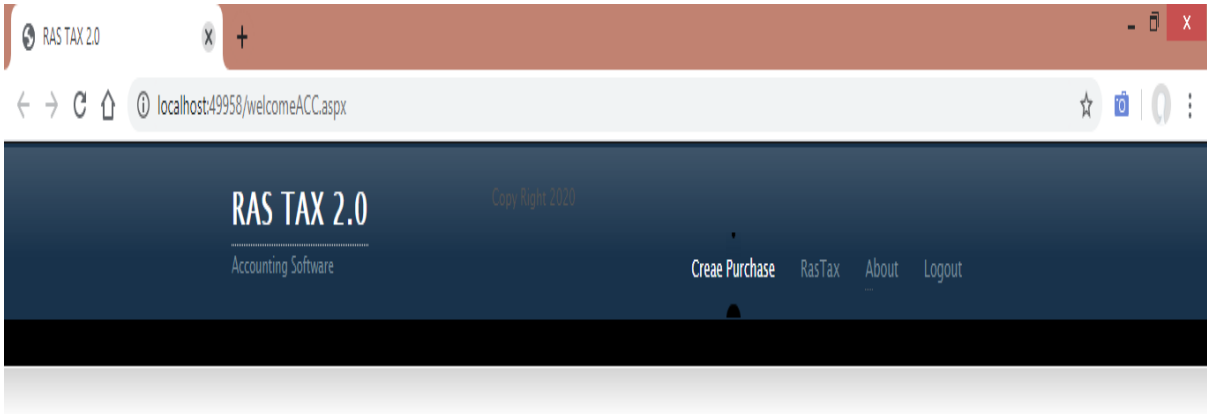


Fig 6.2.14: the welcome page .

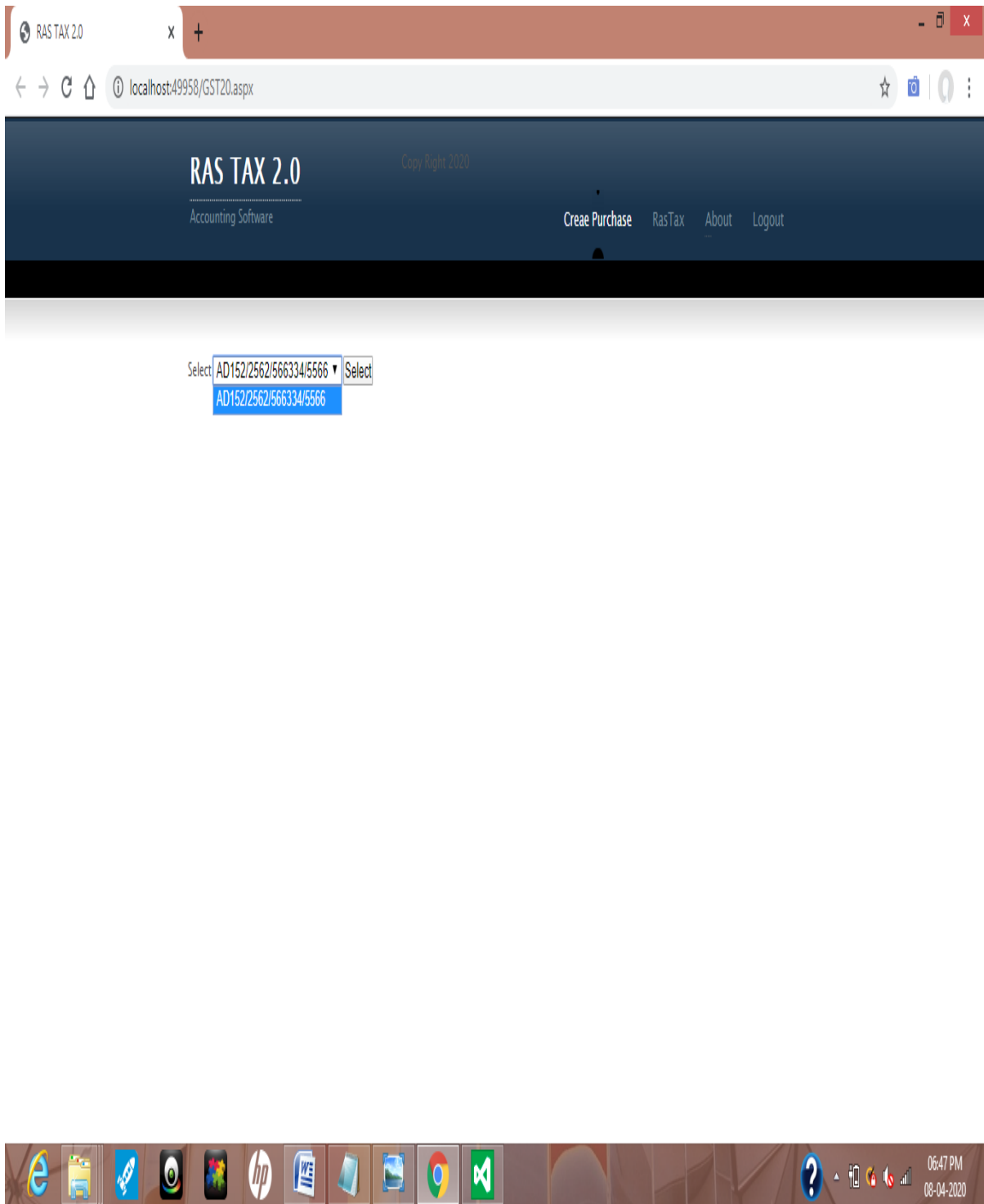
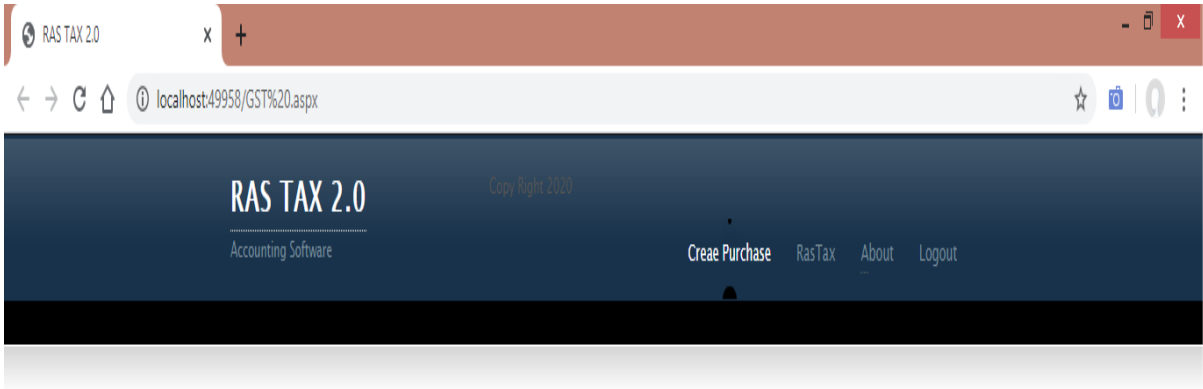


Fig 6.2.15: the drop down list loaded with reference number of each migrated data. The user can select the reference number and call the process of tax calculation.



AD152/2562/566334/5566 ▾ SELECT

**Part II**      **Computation of Tax**      [ATC](#)

Taxable Transaction / Industry Classification	ATC	Taxable Amount	Tax Rate	Tax Due
19 Total Tax Due			19	15000
20 Less: Tax Credits/Payments				
20A Creditable Percentage Tax Withheld Per BIR Form No. 2307 (See Schedule 1)			20A	2450
20B Tax Paid in Return Previously Filed, if this is an amended return			20B	
21 Total Tax Credit/Payments (Sum of Items 20A & 20B)			21	4785
22 Tax Payable (Overpayment) (Item 19 less Item 21)			22	7896
23 Add Penalties				
Surcharge	23A	0.00		
Interest	23B	0.00		
Compromise	23C	0.00		
23D			23D	485
24 Total Amount Payable (Overpayment) (Sum of Items 22 and 23D)			24	76

If Overpayment, mark one box only     To be Refunded     To be issued a Tax Credit Certificate

NEXT



Fig 6.2.16: the application shows tax calculated value and the respective form number is added to the generated value.



RAS TAX 2.0 x +

localhost:49958/gst3.aspx

### computation of income tax and there on

Gross Income	125.00
Deduction under chapter 4	560.00
Tota Income	96.00
Net tax Payable	5.00
Interest Payable	96.00
Tax Paid	0
Advanced Paid	0
TDS 3B	96.00
TDS 3C	3.00
Payable	981.00

### computation of benefits/payable/tax thereon

Total benefits	56.00
Benefit Liability	78.00
Interest on Benefits	63.00
Total payable	2.00
Liability Payable	6.00
Advance tax	.00
Form 16 A	1.00
Gen Account B	0
Gen Account C	56
Sub Total	271.00

**Total 1252.00**

06:48 PM  
08-04-2020

Fig 6.2.17: tax value generated for the employee.

# CHAPTER 7

## SOFTWARE TESTING

### 7.1 TESTING

<b>Test case name</b>	<b>Tax form test</b>
<b>Page tested</b>	<b>FORM.ASPX</b>
<b>Attempt number</b>	<b>1</b>

Field to test	Parameter	Input	Test status
TXT_CM_SRN_RS_TAX_ID	@CM_SRN_RS_TAX_ID	1	Pending
TXT_CM_SRN_RS_FORM16A_ID	@CM_SRN_RS_FORM16A_ID	A16	Pending
TXT_CM_SRN_RS_FORM16A_VAL UE	@CM_SRN_RS_FORM16A_V ALUE	A16V	Pending
TXT_CM_SRN_RS_BANK_INFO	@CM_SRN_RS_BANK_INFO	SBI	Pending
TXT_CM_SRN_RS_AMTPAYABLE	@CM_SRN_RS_AMTPAYABL E	8975.63	Pending

priority 4

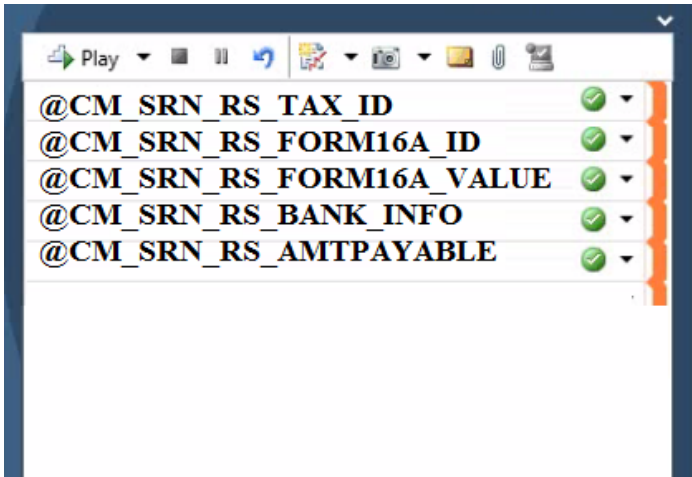
STEPS SUMMARY TESTED BACKLOG ITEMS LINKS ATTACHMENTS ASSOCIATED AUTOMATION

Change steps Insert step Insert shared steps Insert parameter

B / U A

Action	Expected Result
1. @CM_SRN_RS_TAX_ID	= 1
2. @CM_SRN_RS_FORM16A_ID	= A16
3. @CM_SRN_RS_FORM16A_VALUE	= A16V
4. @CM_SRN_RS_BANK_INFO	= SBI
5. @CM_SRN_RS_AMTPAYABLE	= 8975.63
6.	
7.	

Delete iteration Rename parameter Delete parameter



Parameter	Result	Test status
@CM_SRN_RS_TAX_ID	TXT_CM_SRN_RS_TAX_ID shows the value 1	Success
@CM_SRN_RS_FORM16A_ID	TXT_CM_SRN_RS_FORM16A_ID shows the value A16	Success
@CM_SRN_RS_FORM16A_VALUE	TXT_CM_SRN_RS_FORM16A_VALUE shows the value A16V	Success
@CM_SRN_RS_BANK_INFO	TXT_CM_SRN_RS_BANK_INFO shows the value SBI	Success
@CM_SRN_RS_AMTPAYABLE	TXT_CM_SRN_RS_AMTPAYABLE shows the value 8975.63	Success

<b>Test case name</b>	<b>Tax form test</b>
<b>Page tested</b>	<b>FORM.ASPX</b>
<b>Attempt number</b>	<b>2</b>

Field to test	Parameter	Input	Test status
TXT_CM_SRN_RS_TAX_ID	@CM_SRN_RS_TAX_ID	NULL	Pending
TXT_CM_SRN_RS_FORM16A_ID	@CM_SRN_RS_FORM16A_ID	NULL	Pending
TXT_CM_SRN_RS_FORM16A_VALUE	@CM_SRN_RS_FORM16A_VALUE	NULL	Pending
TXT_CM_SRN_RS_BANK_INFO	@CM_SRN_RS_BANK_INFO	NULL	Pending
TXT_CM_SRN_RS_AMTPAYABLE	@CM_SRN_RS_AMTPAYABLE	NULL	Pending

Assigned To Adam Barr Automation status Not Automated  
 State Design Area FabrikamFiber  
 Priority 2

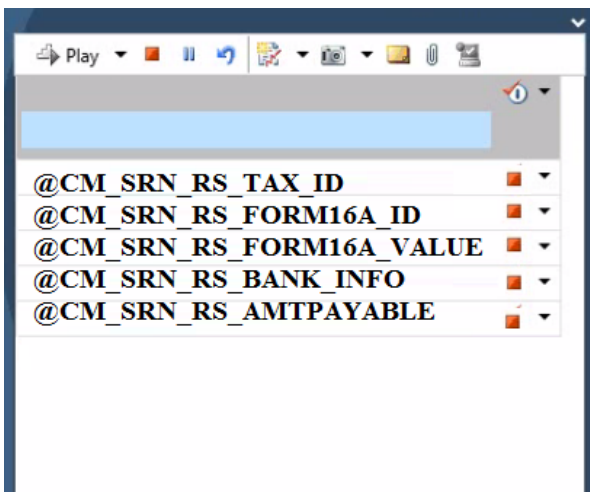
STEPS SUMMARY TESTED BACKLOG ITEMS LINKS ATTACHMENTS ASSOCIATED AUTOMATION

Change steps Insert step Insert shared steps Insert parameter

B / U A

Action	Expected Result
1. @CM_SRN_RS_TAX_ID	= NULL
2. @CM_SRN_RS_FORM16A_ID	= NULL
3. @CM_SRN_RS_FORM16A_VALUE	= NULL
4. @CM_SRN_RS_BANK_INFO	= NULL
5. @CM_SRN_RS_AMTPAYABLE	= NULL
6.	
7.	

Delete iteration Rename parameter Delete parameter



<b>Parameter</b>	<b>Result</b>	<b>Test status</b>
@CM_SRN_RS_TAX_ID	TXT_CM_SRN_RS_TAX_ID cannot <b>shows the value NULL</b>	<b>Success</b>
@CM_SRN_RS_FORM16A_ID	TXT_CM_SRN_RS_FORM16A_ID <b>shows the value NULL</b>	<b>Success</b>
@CM_SRN_RS_FORM16A_VALUE	TXT_CM_SRN_RS_FORM16A_VALUE <b>shows the value NULL</b>	<b>Success</b>
@CM_SRN_RS_BANK_INFO	TXT_CM_SRN_RS_BANK_INFO <b>shows the value NULL</b>	<b>Success</b>
@CM_SRN_RS_AMTPAYABLE	TXT_CM_SRN_RS_AMTPAYABLE <b>shows the value NULL</b>	<b>Success</b>

## **CHAPTER 8**

### **CONCLUSION**

This RASTAX application is accounting based plug in software used for the network based software integrations. The main focus of this software of RASTAX is its flexibility of tax calculation. The single application of RASTAX can be used for the any business organisation and calculate the tax over their domain used. So the users can spend less amount in the software purchasing too many software which was used in the early systems. The ACE 7.2 is the software which has control for the manufacturing units, the GTAX is the software application used for managing the export and import based applications and all. With the RASTAX the user can use a single product which can communicate with centralised server and update the latest tax released by the government and other rules.

## **CHAPTER 9**

### **FUTURE ENHANCEMENT**

The developers seeking the possibility of use the RASTAX as plug in product which can be used in other accounting software to indentifying and fixing the error for moderator software . The plug in without GUI and the GUI of application where the RASTAX is implements will be act as application GUI. For example if a accounting software name A cannot process the tax generation due to errors caused in violation of accounting rule or clerical mistake the RATAAX will highlight the area of where error has occurred in the software page of A. If the organisation do working with a software for a long period of time the administration of the company will not ready to change to a new accounting software since the user employee will be familiarise and also the data will be stored in the SQL . So by an RASTAX plug in a company the users can use their own accounting software and also make use of extra features like error fixing and tax generation features.

## CHAPTER 10

### BIBLIOGRAPHY

ADO.NET Entity Framework on MSDN®:

<http://msdn.microsoft.com/library/bb399572.aspx>

The Repository Pattern on MSDN:

<http://msdn.microsoft.com/en-us/library/ff649690.aspx>

Unit of Work Pattern:

<http://www.martinfowler.com/eaCatalog/unitOfWork.html>

Catalog of Patterns of Enterprise Application Architecture:

<http://martinfowler.com/eaCatalog/>

Understanding Models, Views, and Controllers on ASP.NET:

<http://www.asp.net/mvc/tutorials/understanding-models-views-and-controllers-cs>

Best Practices for ASP.NET MVC:

[http://blogs.msdn.com/b/aspnetue/archive/2010/09/17/second\\_2d00\\_post.aspx](http://blogs.msdn.com/b/aspnetue/archive/2010/09/17/second_2d00_post.aspx)

Dependency Injection in *MSDN Magazine*:

<http://msdn.microsoft.com/en-us/magazine/cc163739.aspx>

Unity Application Block on MSDN:

<http://www.msdn.com/unity>

Post/Redirect/Get Pattern:

<http://en.wikipedia.org/wiki/Post/Redirect/Get>

For more information on how clients can consume and use **data-** attributes, see Chapter 6, "[Client Data Management and Caching](#)" and "[Using Data- Attributes](#)" in Chapter 7, "[Manipulating Client-Side HTML](#)."

For more information on unit testing see Chapter 13, "[Unit Testing Web Applications](#)"



# CHAPTER 11

## USER MANUAL

Install all the softwares which is necessary to run web based the project.

The IDE visual studio intergrations :

1. Visual studio 2013 professional edtion
2. SQL server management studio 2008 R2
3. IIS server
4. .NET frame work 2.0 or above

### **Path Setting for the we appliciton**

**Set the following path after the installation of Visual studio software:**

MyComputer>Properties>Advanced>SystemSettings>EnvironmentVariables>

SystemVariables>

Login page :

User id: admin

Password: admin

### **Next go to the MySql developer and provide the connection**

Connection>Right Click>New Database Connection

Username:SQL\_VJ123

Password:SQL\_VJ\_123

Hostname: localhost

SID: XE

Now press Test>If Success>Connect

### **Execution**

**The user should follow the steps below to run the project in the visual studio IDE:**

1. Set the visual studio path in in window>preferences>IIS>Version 6.0>Home:

Give the complete path and click enter

2. Start IIS Server

3. Type the web address as <http://localhost:8080/Obs/login.jsp> in

any web browser.