

Department of Computer Science and Engineering
G.PULLAREDDY ENGINEERING COLLEGE (AUTONOMOUS):KURNOOL
(Affiliated to JNTUA, ANANTHAPURAMU)



A Report on “SBI ATM System Management”

Developed by

S.Lakshmi Prasana (149X1A05F7)
A.Prathyusha Lakshmi (149X1A0511)
M.Sai Krishna (149X1A05A7)

Under the guidance of

Sri M.Imtiaz Ur Rahaman
Sri P.Penchala Prasad
Sri K.Bala Chowdappa

ABSTRACT

At present every bank has several branches of ATMs present at different locations. Every ATM consists of several assets and those details are maintained manually. By these they are facing problems if they miss the details so they want to store all the details in a database and they want an interface to manage the details of locations and assets and its operations. They have specified their own requirements and according to those requirements the interface has been developed. First gather the information of assets present in the ATMs and the locations where ATMs are present and store the details in the database and we can add the new assets and locations and can modify the existing details using the interface developed.

1. INTRODUCTION

What is an ATM?

ATM stands for Automated Teller Machine. It is a specialized computer that makes convenient to manage money. For example, almost all ATMs allow to withdraw money, and many allow to make deposits. At some ATMs, we can print a statement, we can check account balances transfer money between the accounts and even purchase stamps. We can usually access most of the services at an ATM that are operated in the bank. ATMs are a safe and convenient way to manage your money. There are millions of ATMs worldwide and we can use many ATMs 24 hours a day, 7 days week.

There are 4 models of ATM-

1. ATM
2. CDM
3. Recycler
4. Coin Vending Machine

1. ATM

ATM is an electronic banking outlet which allows customers to complete basic transactions without the aid of a branch representative or teller. Anyone with credit or debit card can access ATMs. Customers can withdraw cash and receive reports of their account balances.

2. CDM

CDM is known as Cash Deposit Machine. It is an ATM like machine that allows to deposit cash directly to the account using the debit card. We can use this machine to instantly credit account without visiting the branch. The transaction receipt also gives the updated account balance.

Some of the salient features of this machine are:

- Instant credit of cash deposit into the account.
- Quick and convenient way to deposit cash.
- Paperless transaction.
- Pin Change.
- Balance Enquiry

3. Recycler

A cash recycler is a complex machine that handles a couple of simple but important tasks- accepting and dispensing cash. It also stores money securely, keeps an accurate accounting of cash and automates the cash cycle. Generally, we will find them in banks, credit unions and back-office retail cash rooms. In a cash recycler banknotes are placed into a feeder and passed through a bill identifier to determine the denomination and validity of the banknotes. This cash is then stored in separate cassettes or modules for dispensing in future transactions. The cash that is deposited to a recycler is the same cash dispensed from it. Cash recycling isn't a product or a service it is an approach for cash management.

4. Coin Vending Machine

Coin vending machine enables people to get coins for their routine transactions. They ensure a continuous supply of coins and are especially useful in public places such as railway stations, bus stands etc. This machines can be installed both in-house and off-site. When installed at banks, this coin vending machines make coin distribution simple and cost effective. An easy-to-understand working mechanism eliminates long waiting queues for customers. The coin vending machines accept notes from Rs.5 to Rs.1000. These are simple, reliable and secure. Coin vending machines also prevent fake note circulation.



Fig. 1.1. Coin Vending Machine

PROBLEM STATEMENT

The bank employees especially bank manager is responsible for maintaining the ATM. ATM not only includes ATM machine but also it consists of several assets such as Air Conditioner, Surveillance Camera, Uninterrupted Power Supply(UPS), Stabilizer, Property (where the ATM is located). For each asset it has its own specifications and all the details of these assets are not maintained properly. They used to note manually and they did not maintained them properly. By these they are facing the problems. Especially during the time of Demonetisation they have faced a lot of problems. At that time Rs.500 and Rs.1000 are banned and because of that there was a shortage of cash. There was no cash in the ATM. People used to stand several hours in front of ATM for cash and due to shortage of money there was no cash in the ATM and because of frustration people has damaged the ATM machines and assets present in the ATM. Also some of the assets have been stolen. As there are no records of those assets they did not understand whether they have been stolen or not present. And also they did not know when the asset have been installed, what is its type and make. So they want all these to store in one database and they also want an Interface to maintain all these records regularly. They have their own specifications and we are going to design an Interface according to their specifications.

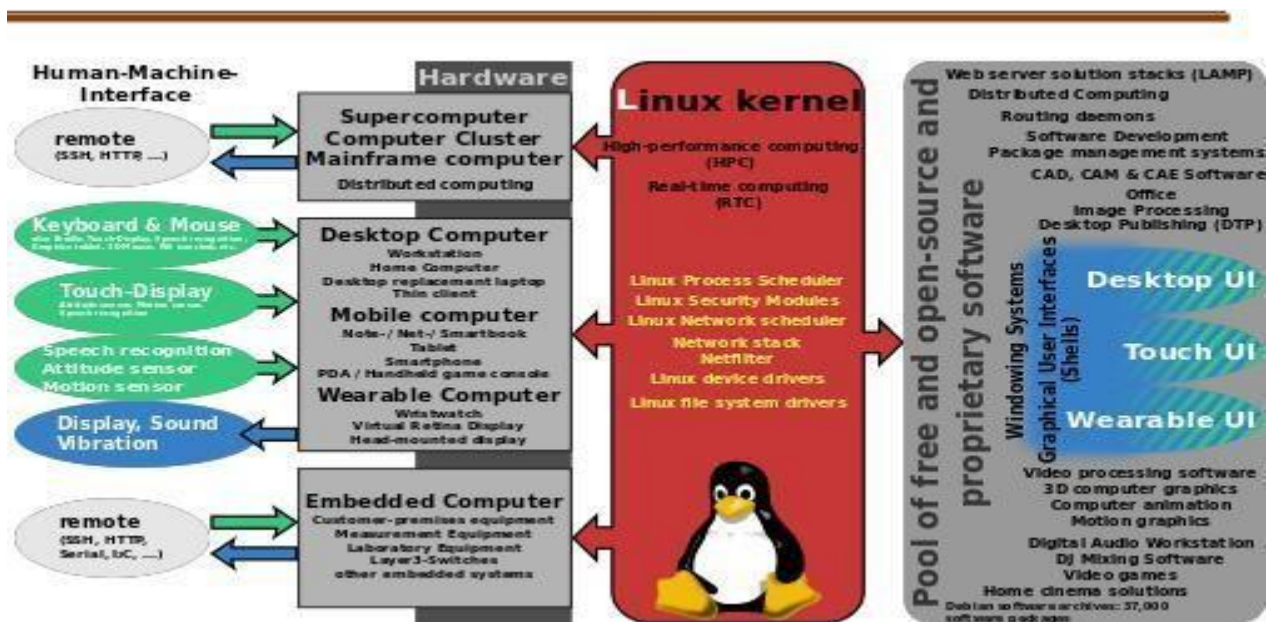
2. LITRATURE REVIEW



2.1. LINUX

Hardware support

The Linux kernel is a widely ported operating system kernel, available for devices ranging from mobile phones to supercomputers. It runs on a highly diverse range of computer architectures, including the hand-held ARM-based IPAQ and the IBM mainframes System z9 or System z10. Specialized distributions and kernel forks exist for less mainstream architectures for example, the ELKS kernel fork can run on Intel 8086 or Intel 80286 16-bit microprocessors, while the Linux kernel fork may run on systems without a memory management unit. The kernel also runs on architectures that were only ever intended to use a manufacturer-created operating system, such as Macintosh computers (with both PowerPC and Intel processors), PDAs, video game consoles, portable music players, and mobile phones. There are several industry associations and hardware conferences devoted to maintaining and improving support for diverse hardware under Linux, such as Freedom HEC. Over time, support for different hardware has improved in Linux, resulting in any off-the-shelf purchase having a "good chance" of being compatible.



Uses

Beside the Linux distributions designed for general-purpose use on desktops and servers, distributions may be specialized for different purposes including: computer architecture support, embedded systems, stability, security, localization to a specific region or language, targeting of specific user groups, support for real-time applications, or commitment to a given desktop environment. Furthermore, some distributions deliberately include only free software. As of 2015, over four hundred Linux distributions are actively developed, with about a dozen distributions being most popular for general-purpose use.

Digital security

Kali Linux is a Debian based Linux distribution designed for digital forensics and penetration testing. It comes preinstalled with several software applications for penetration testing and identifying security exploits. The Ubuntu derivative Back Box provides pre-installed security and network analysis tools for ethical hacking. There are many Linux distributions created with privacy, secrecy, network anonymity and information security in mind, including Tails, Tin Hat Linux and Tinfoil Hat Linux. Lightweight Portable Security is a distribution based on Arch Linux and developed by the United States Department of Defense. Tor-ram disk is a minimal distribution created solely to host the network anonymity software Tor.

UBUNTU

Ubuntu is an open source operating system for computers. It is a Linux distribution based on the Debian architecture. It is usually run on personal computers, and is also popular on network servers, usually running the Ubuntu Server variant, with enterprise-class features. Ubuntu runs on the most popular architectures, including Intel, AMD, and ARM-based machines. Ubuntu is also available for tablets and smart phones, with the Ubuntu Touch edition.

Ubuntu is the most popular operating system running in hosted environments, so-called clouds as it is the most popular server Linux distribution.

APACHE HTTP SERVER

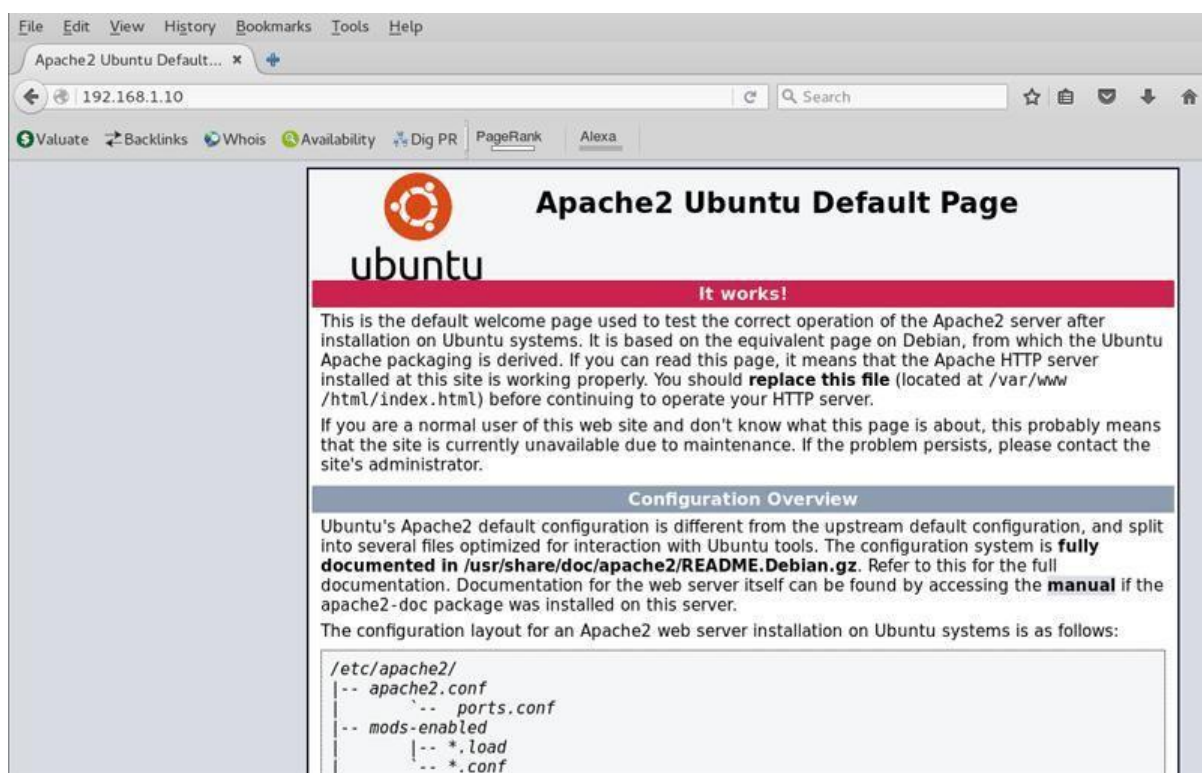
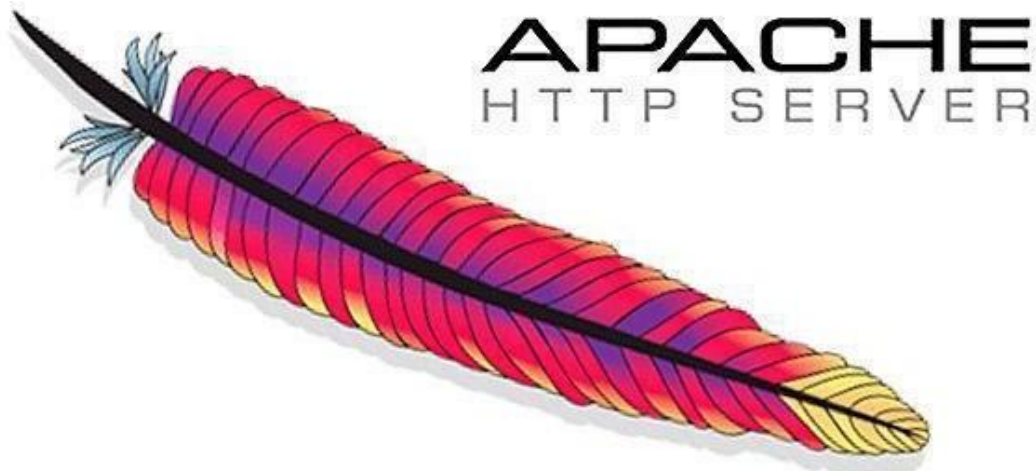


Fig. 2.2. Apache2 Ubuntu Default Page

Apache HTTP Server, colloquially called Apache, is free and open-source cross-platform web server software, released under the terms of Apache License 2.0. Apache is developed and maintained by an open community of developers under the auspices of the Apache Software Foundation. Although Apache HTTP Server is cross-platform, As of 1 June 2017, 92% of all Apache HTTPS Server copies run on Linux distributions. Version 2.0 improved support for non-Unix operating systems such as Windows and OS/2. Old versions of Apache were ported to run on OpenVMS, and NetWare.

Installation



```
$ sudo apt-get update  
$ sudo apt-get install apache2
```

Open terminal in Ubuntu and type the above stated lines. First you need to check for updates And then update if required. Later install apache version.2 (apache2).

Give the required permissions during installation.

Performance

Instead of implementing a single architecture, Apache provides a variety of Multi Processing Modules (MPMs), which allow Apache to run in a process-based, hybrid (process and thread) or event-hybrid mode, to better match the demands of each particular infrastructure. This implies that the choice of correct MPM and the correct configuration is important. Where compromises in performance need to be made, the design of Apache is to reduce latency and increase throughput, relative to simply handling more requests, thus ensuring consistent and reliable processing of requests within reasonable time-frames.

MYSQL



Fig.2.3. MySQL

My SQL is an open-source relational database management system (RDBMS). Its name is a combination of "My", the name of co-founder Michael Widenius daughter, and "SQL" the abbreviation for Structured Query Language. The My SQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. My SQL was owned and sponsored by a single for-profit firm, the Swedish company My SQL AB, now owned by Oracle Corporation. For proprietary use, several paid editions are available, and offer additional functionality.

My SQL is a central component of the LAMP open-source web application software stack. LAMP is an acronym for "Linux, Apache, My SQL, Perl/PHP/Python". Applications that use the My SQL database include: TYPO3, MODx, Joomla, WordPress, phpBB, MyBB, and Drupal. MySQL is also used in many high-profile, large-scale websites, including Google, Facebook, Twitter, Flickr and YouTube.

Limitations

When using some storage engines other than the default of InnoDB, MySQL does not comply with the full SQL standard for some of the implemented functionality, including foreign key references and check constraints.

Up until MySQL 5.7, triggers are limited to one per action / timing, meaning that at most one trigger can be defined to be executed after an INSERT operation, and one before INSERT on the same table. No triggers can be defined on views.

MySQL database's inbuilt functions like UNIX_TIMESTAMP() will return 0 after 03:14:07 UTC on 19 January 2038. Recently, there had been an attempt to solve the problem which had been assigned to the internal queue.

Installation

```
$ sudo apt-get update
```

```
$ sudo apt-get install mysql-server
```

```
$ mysql_secure_installation
```

Install the SQL-server in Ubuntu by going to terminal updating the system and then typing the second Line of code give related permissions .create a password for it. Secure the installation.



After installation check whether it is working good.

```
matt@ubuntu: ~  
matt@ubuntu:~$ mysql -u mynewuser -p  
Enter password:  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 7  
Server version: 5.7.12-0ubuntu1 (Ubuntu)  
  
Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> █
```

```
jeremy@CrashBox: ~  
Welcome to the MySQL monitor.  Commands end with ; or \g.  
Your MySQL connection id is 45  
Server version: 5.5.27-0ubuntu2 (Ubuntu)  
  
Copyright (c) 2000, 2011, Oracle and/or its affiliates. All rights reserved.  
  
Oracle is a registered trademark of Oracle Corporation and/or its  
affiliates. Other names may be trademarks of their respective  
owners.  
  
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.  
mysql> show databases;  
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
| test |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql>
```

PHP

PHP is a server-side scripting language designed primarily for web development but also used as a general-purpose programming language. Originally created by Rasmus Lerdorf in 1994, the PHP reference implementation is now produced by The PHP Development Team. PHP originally stood for Personal Home Page, but it now stands for the recursive acronym PHP: Hypertext Preprocessor. PHP code may be embedded into HTML or HTML5 markup, or it can be used in combination with various web template systems, web content management systems and web frameworks. PHP code is usually processed by a PHP interpreter implemented as a module in the web server or as a Common Gateway Interface (CGI) executable. The web server software combines the results of the interpreted and executed PHP code, which may be any type of data, including images, with the generated web page. PHP code may also be executed with a command-line interface (CLI) and can be used to implement standalone graphical applications. The standard PHP interpreter, powered by the Zend Engine, is free software released under the PHP License. PHP has been widely ported and can be deployed on most web servers on almost every operating system and platform, free of charge. The PHP language evolved without a written formal specification or standard until 2014, leaving the canonical PHP interpreter as a de facto standard. Since 2014 work has gone on to create a formal PHP specification.

PHP 5

On July 13, 2004, PHP 5 was released, powered by the new Zend Engine II. PHP 5 included new features such as improved support for object-oriented programming, the PHP Data Objects (PDO) extension (which defines a lightweight and consistent interface for accessing databases), and numerous performance enhancements. In 2008 PHP 5 became the only stable version under development. Late static binding had been missing from PHP and was added in version 5.3. Many high-profile open-source projects ceased to support PHP 4 in new code as of February 5, 2008, because of the GoPHP5 initiative, provided by a consortium of PHP developers promoting the transition from PHP 4 to PHP 5. Over time, PHP interpreters became available on most existing 32-bit and 64-bit operating systems, either by building them from the PHP source code, or by using pre-built binaries. For the PHP versions 5.3 and 5.4, the only available Microsoft Windows binary distributions were 32-bit x86 builds, requiring Windows 32-bit compatibility mode while using Internet Information Services (IIS) on a 64-bit Windows platform. PHP version 5.5 made the 64-bit x86-64 builds available for Microsoft Windows.

REFERENCE TEXTBOOKS FOR PHP

1. PHP: The Complete Reference – by Steven Holzner
2. Practical PHP and MySQL Website Database: A Simplified Approach-by Adrian.w.west

PHP My Admin



Fig. 2.5. PHP MyAdmin

PHP My Admin is a free and open source administration tool for My SQL and Maria DB. As a portable web application written primarily in PHP, it has become one of the most popular My SQL administration tools, especially for web hosting services

Features

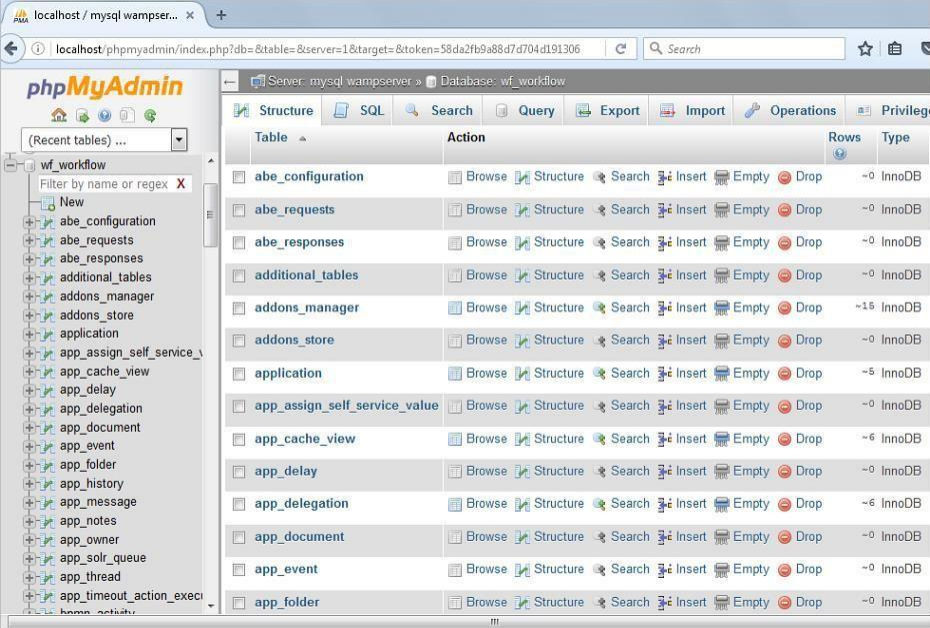
Features provided by the program include:

1. Web interface
2. MySQL and MariaDB database management
3. Import data from CSV and SQL
4. Export data to various formats: CSV, SQL, XML, PDF (via the TCPDF library), ISO/IEC 26300 - OpenDocument Text and Spreadsheet, Word, Excel, LaTeX and others
5. Administering multiple servers
6. Creating PDF graphics of the database layout
7. Creating complex queries using query-by-example (QBE)
8. Searching globally in a database or a subset of it
9. Transforming stored data into any format using a set of predefined functions, like displaying BLOB-data as image or download-link
10. Live charts to monitor My SQL server activity like connections, processes, CPU/memory usage, etc.

References

1. "PHP My Admin 4.7.5 is released '. PHP MY ADMIN.net. Retrieved 2017-10-24
2. "Translations". PHP My Admin. Retrieved 2014-12-23
3. http://www.pcworld.com/article/233948/PHP_MYADMIN.html
4. "PHP MY ADMIN-About". PHP MY ADMIN. Retrieved 2013-03-03.
5. Delisle, Marc (2010). Mastering PHP MY ADMIN 3.3.x for Effective MySQL Management Packt Publishing.p.359.ISBN 978-1-84951-354-8.Operating systems.

Maintain Databases

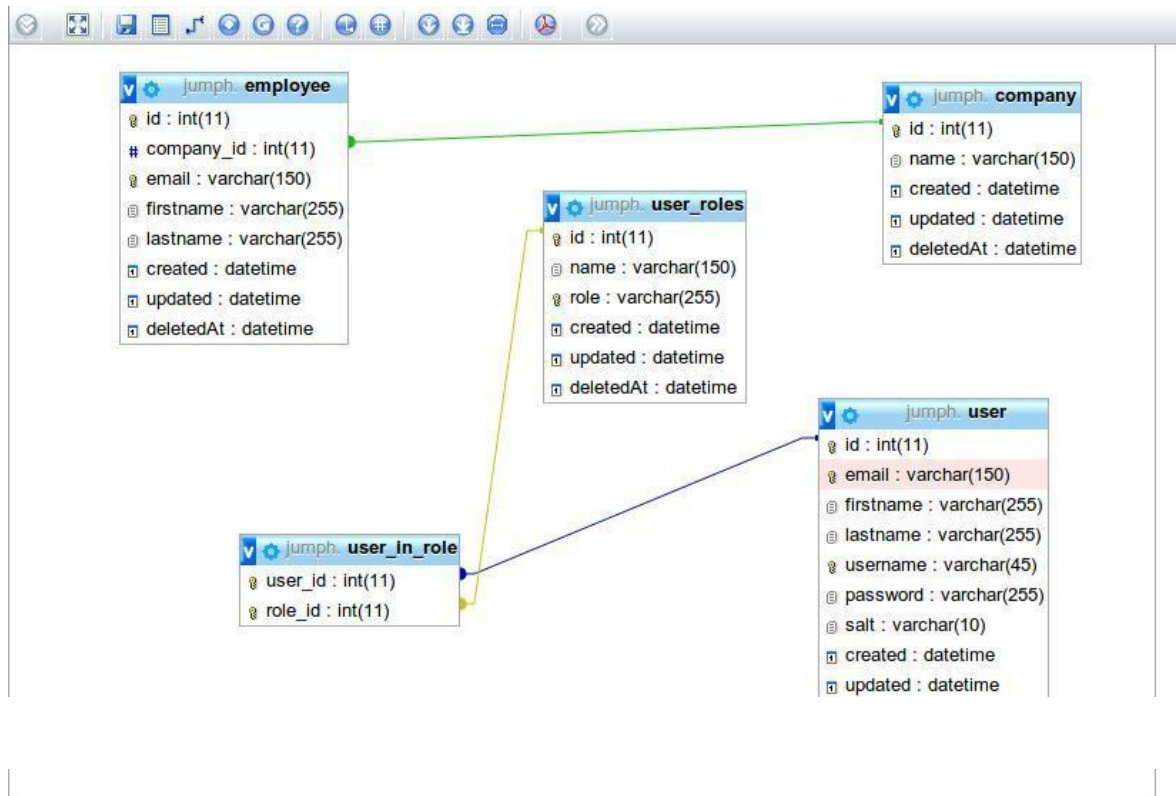


The screenshot shows the phpMyAdmin interface for a database named 'wf_workflow'. The left sidebar lists various tables, and the main area displays the structure of the 'wf_workflow' database. The table structure is as follows:

Table	Action	Rows	Type
abe_configuration	Browse Structure Search Insert Empty Drop	~0	InnoDB
abe_requests	Browse Structure Search Insert Empty Drop	~0	InnoDB
abe_responses	Browse Structure Search Insert Empty Drop	~0	InnoDB
additional_tables	Browse Structure Search Insert Empty Drop	~0	InnoDB
addons_manager	Browse Structure Search Insert Empty Drop	~15	InnoDB
addons_store	Browse Structure Search Insert Empty Drop	~0	InnoDB
application	Browse Structure Search Insert Empty Drop	~5	InnoDB
app_assign_self_service_value	Browse Structure Search Insert Empty Drop	~0	InnoDB
app_cache_view	Browse Structure Search Insert Empty Drop	~6	InnoDB
app_delay	Browse Structure Search Insert Empty Drop	~0	InnoDB
app_delegation	Browse Structure Search Insert Empty Drop	~6	InnoDB
app_document	Browse Structure Search Insert Empty Drop	~0	InnoDB
app_event	Browse Structure Search Insert Empty Drop	~0	InnoDB
app_folder	Browse Structure Search Insert Empty Drop	~0	InnoDB

Database table

Structures and relations between tables of a database



2.6. SPECIFICATIONS

1. MACHINE ID

Each ATM will be having a unique 13 digit alpha numeric field

2. MACHINE TYPE

There are 4 models. ATM, CDM, Recycler, Coin Vending Machine

3. MACHINE MAKE

Wincore, Diebold, Hyosung

4. MACHINE LOCATION

Geographical address of the ATM/site. 4 lines space required for DNo. Plot No. Building Name, Street Name, Colony/Area, City. It should also indicate whether the location is in “Metro/Urban, Semi Urban, Rural’ areas.

5. INSTALLATION DATE

Date of installation of machine or opening of the site

6. NOTE CAPACITY

It is applicable for each individual machine of ATM type only. The machine contains 4 bins to load cash viz Rs.2000, Rs.500, Rs.100, Rs.50/-. The maximum loading capacity in each bin in terms of number of notes and relative amount will be keyed in.

7. MACHINE REPLACEMENT DETAILS

4 lines to record authority for replacement, letter number and date, date of cash evacuation, date of disposal, name of the officer to whom the machine was handed over.

8. UPS ID

Each UPS will be having a unique 13 digit alpha numeric field

9. UPS TYPE

There are two types – battery charged and solar power charged.

10. UPS MAKE

Different suppliers as approved by Head Office for a period of 1 year normally.

11. UPS CAPACITY

The UPS capacity is measured in KV. A two digit space will be in order.

12. INSTALLATION DATE

Date of installation of machine or opening of the site

13. AMC FOR UPS

Normally, the supplier of UPS will undertake AMC [Annual Maintenance Contract] at certain cost for a period of 12 months. Name & address of the AMC provider - 5 lines, mobile phone 10 digits separately [not mixed with address], email ID of the AMC provider - 1 line of 50 characters.

14. MACHINE REPLACEMENT DETAILS

4 lines to record authority for replacement, letter number and date, date of disconnection, date of disposal, replacement value in Rs -----, name of the officer to whom the machine was handed over.

15. BATTERY ID

Each battery will be having a unique 13 digit alpha numeric field.

16. BATTERY TYPE

There are two types – battery sealed maintenance proof [SMPF] or distilled water maintenance. Normally it is SMPF.

17. BATTERY MAKE

Different suppliers as approved by Head Office for a period of 1 year normally.

18. BATTERY CAPACITY

The battery capacity is measured in **Amh**. A three digit space will be in order.

19. INSTALLATION DATE

Date of installation of machine or opening of the site.

20. AMC FOR BATTERIES

Normally, the supplier of batteries will undertake AMC [Annual Maintenance Contract] at certain cost for a period of 12 months. Name & address of the AMC provider - 5 lines, mobile phone 10 digits separately [not mixed with address], email ID of the AMC provider - 1 line of 50 characters.

21. BATTERY REPLACEMENT DETAILS

4 lines to record authority for replacement, letter number and date, date of disconnection, date of disposal, Buy Back value in Rs -----, name of the officer to whom the batteries were handed over.

22. AC ID

Each AC will be having a unique 13 digit alpha numeric field.

23. AC TYPE

There are three types – window, split and cassette ACs.

24. AC MAKE

Different suppliers as approved by Head Office for a period of 1 year normally.

25. AC CAPACITY

The AC capacity is measured in **tonnage**. A three digit with decimal value space will be in order.

26. INSTALLATION DATE

Date of installation of AC or opening of the site.

27. AMC FOR AC

Normally, the supplier of AC or a local technician team will undertake AMC [Annual Maintenance Contract] at certain cost for a period of 12 months. Name & address of the AMC provider - 5 lines, mobile phone 10 digits separately [not mixed with address], email ID of the AMC provider - 1 line of 50 characters.

28. AC REPLACEMENT DETAILS

4 lines to record authority for replacement, letter number and date, date of disconnection, date of disposal, Buy Back value in Rs -----, name of the officer to whom the batteries were handed over.

29. STABILIZER ID

Each AC will be having a unique 13 digit alpha numeric field.

30. STABILIZER MAKE

Different suppliers as approved by Head Office for a period of 1 year normally.

31. INSTALLATION DATE

Date of installation of stabilizer or opening of the site.

32. AMC FOR STABILIZER

Normally, the supplier of AC or a local technician team will undertake AMC [Annual Maintenance Contract] at certain cost for a period of 12 months. Name & address of the AMC provider - 5 lines, mobile phone 10 digits separately [not mixed with address], email ID of the AMC provider - 1 line of 50 characters.

33. AMC REPLACEMENT DETAILS

4 lines to record authority for replacement, letter number and date, date of disconnection, date of disposal, Buy Back value in Rs -----, name of the officer to whom the batteries were handed over.

34. Camera ID

Each AC will be having a unique 13 digit alpha numeric field.

35. Camera TYPE

There are four types – Normal, Infra Red, Dome, IP.

36. Camera MAKE

Different suppliers as approved by Head Office for a period of 1 year normally.

37. Camera CAPACITY

No capacity

38. INSTALLATION DATE

Date of installation of camera or opening of the site.

39. AMC FOR CAMERAS

Normally, the supplier of camera or a local technician team will undertake AMC [Annual Maintenance Contract] at certain cost for a period of 12 months. Name & address of the AMC provider - 5 lines, mobile phone 10 digits separately [not mixed with address], email ID of the AMC provider - 1 line of 50 characters.

40. AMC REPLACEMENT DETAILS

4 lines to record authority for replacement, letter number and date, date of disconnection, date of disposal, Buy Back value in Rs -----, name of the officer to whom the batteries were handed over.

41. NAME OF THE OWNER [FULL NAME]

42. PERMANENT ADDRESS

4 lines to be used for Door No., building name, street name, colony, area, city, pin code

43. ADDRESS FOR COMMUNICATION

4 lines to be used for Door No., building name, street name, colony, area, city, pin code. 1 additional line to record email ID.

44. MOBILE NUMBER- 10 DIGIT SPACE

45. ALTERNATE CONTACT NUMBER -12 digit space

46. LEASE AGREEMENT FOR PERIOD – TWO DIGITS to be recorded in months

47. LEASE AGREEMENT DATES

FROM [DATE 10 CHARS] TO DATE [10 CHARS]

With a provision to add rows

48. LEASE RENT

5 digits

With a provision to add rows

49. OWNER PAN CARD

50. OWNER AADHAR CARD

51. OWNER SB ACCOUNT NUMBER

52. IF OWNERS ACCOUNT WITH OTHER BANK:

Bank name, Branch Name, IFSC Code, Net amount, Bank charges, TDS, gross amount
[this equals ATM monthly rent]

53. Electricity Service Connection No:

ERO Office name

ERO Office address

ERO Office contact number

54. SECURITY PERSONNEL

Each ATM may be guarded by security personnel, who are engaged from outside agencies.
The details are to be recorded are

A] Name of the Agency providing security staff

B] Constitution [Pvt/Public Ltd Company]

C] KYC of the entity

D] Authority who has permitted such hiring

E] Letter reference number, date etc

F] Names of the key persons to whom cash is to be delivered

55. CASH REPLENISHMENT AGENCY

A] Name of the Agency providing security staff

B] Constitution [Pvt/Public Ltd Company]

C] KYC of the entity

D] Authority who has permitted such hiring

A] b] c] Full name, KYC details, permanent address, temporary address, communication details

56. DETAILS OF VEHICLE ENGAGED

Name of the owner of the vehicle – add row facility

Vehicle details – add row facility

Vehicle Registration Number – add row facility

Name of the Driver

KYC of the driver

Contact details

Name of the Stand by driver

KYC of stand by driver

Contact details

2.6.1. CASH DELIVERY ACCOUNTING & LOADING

ATM Number

ATM Location

BGL A/c No

Date of Loading

Cash Indent

	Rs.50/-	Rs.100/-	Rs.500/-	Rs.1000/-	Rs.2000/-	Total
No. of pieces						
Amount						

Cash Delivered

	Rs.50/-	Rs.100/-	Rs.500/-	Rs.1000/-	Rs.2000/-	Total
No. of pieces						
Amount						

SBI ATM SYSTEM MANAGEMENT

Pre Loading – as per Admin Slip

	Rs.50/-	Rs.100/-	Rs.500/-	Rs.1000/-	Rs.2000/-	Total
No. of pieces						
Amount						

Cash Loaded

	Rs.50/-	Rs.100/-	Rs.500/-	Rs.1000/-	Rs.2000/-	Total
No. of pieces						
Amount						

Post Loading – as per Admin Slip

	Rs.50/-	Rs.100/-	Rs.500/-	Rs.1000/-	Rs.2000/-	Total
No. of pieces						
Amount						

Difference, if any

	Rs.50/-	Rs.100/-	Rs.500/-	Rs.1000/-	Rs.2000/-	Total
No. of pieces						
Amount						

3. ANALYSIS

3.1 FEASIBILITY STUDY

A feasibility study is a sophisticated case description of the complete Design Process and system analysis. This study starts by the problem definition classification. Feasibility is to come to a decision if it's significant doing or not. One time as the acceptance problem definition has been created; a logical model of the system is developed by the analyst. A look for alternatives is analyzed suspiciously.

The key considerations involved in the feasibility study are.

Operational Feasibility

Questions that are going to be posed are as follows:

- If the system is developed and implemented will it be used?
- If there was adequate support from the management and from the users for the project.
- Have the users been participated in planning and expansion of the project.
- Will the system makes of poorer quality result in any respect or region?

This type of system can be implemented in the association as there is sufficient hold up from organization and users. Being implemented in Java so that the essential operations are accepted mechanically.

Technical Feasibility

Questions that are going to be posed are as follows:

- Does the required technology be present to do what is has been suggested.
- Does the projected equipment have the technical ability for using the latest system?
- Are there technical assures of accurateness, consistency and information security?
- Whether the project is implemented with 1GB RAM on Pentium III.
- The environment necessary is windows XP platform in the expansion of system.
- Observer and Factory patterns will bring up to date the results finally
- The language used is JAVA 1.6.0 & Windows Environment in the development.

Economic Feasibility

It is also called as the cost or budget estimated analysis. It determines if the project development was possible with given resource constraints? Are the benefits that will accrue from the new system worth the costs? What are the savings that will result from the system,

including tangible and intangible ones? What are the development and operational costs? These issues are taken care by the Economic feasibility.

Social Feasibility

In this we study the acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as necessity.

3.2. TESTING

INTRODUCTION

The purpose of testing is to find out errors. Testing is the procedure of trying to find out every possible fault or fault in a work product. It provides a way to test the functionality of the following i) components, ii) sub-assemblies, iii) assemblies and/or an iv) final product. It is the procedure of implementing software with plan of ensuring that the software system attains its requirements and user chance does not fail in an appropriate way.

We have different types of testing. Every test type addresses a detailed testing requirement. Software testing is a main feature of software quality assurance and represents the essential reuse of requirement. Design and code testing represents attractive differently for the software throughout earlier description and development phase, it was attempted to construct software from an conceptual concept to real execution.

The testing phase contains, testing of the improvement of the system which using various testing techniques such as i) white box testing, ii) control structure Testing.

TESTING TECHNIQUES

It contains two types of techniques. One is white box testing and second one is Black box testing.

White Box Testing

This white box testing is one among the test case blue print techniques that uses the control arrangement of the practical design to attain test cases. After applying white box testing it was recognized that.

Leave Recording System (LRS) software assures that all autonomous paths with in the modules have been exercised at least one time.

- a) It has exercised all logical decisions on either 0(true) or 1 (false) sides.

- b) It was tested to perform all loops at their margins and within their operational limits.
- c) It was tested for the inner data structures to make sure their authority.

Control structure training

The following tests were performed and it was made famous that the BCBS is testing them fine.

- Basic path testing
- Condition testing
- Data flow testing
- Loop testing

Black Box Testing

Black- box testing concentrates on functional requirement of the software. It enables to obtain sets of input conditions that will totally work out all functional necessities of a program.

Black box testing tries to locate error in the following group:

- a) Wrong or missing function
- b) Interface errors
- c) Errors in data structures or outside database access and presentation errors

TESTING STRATEGIES

An approach for software testing must contain low-level tests that are essential to prove that a tiny source code segment has been properly implemented and also high level over customer necessities.

VERIFICATION

Verification is a basic idea in the software plan. Verification refers to the set of actions that make sure that software properly implements exact role. Validation refers to a dissimilar set of actions that make sure that software that has been constructed is a noticeable to customers. Validation is not anything but the procedure of using software in live surroundings in order to discover errors.

Verification is usually finished in two steps:

- Verification of the requirements: Verification that the software requirement definition satisfies the consumer requirements.
- Verification of design: Verification that the design satisfies the necessities description.

Verification and Validation include a broad collection of software quality assurance (SQA) actions that contain official technical use again quality and design audits, presentation monitoring, replication, possible study. One of the most important SQA activities is Software reviews.

Reviews provide as a filter for the software process, deleting errors whereas they are moderately low-cost to discover and correct appropriately validate a system data regarding software engineering procedure should be collected, evaluated and dispersed. SQA helps to look up the quality of the product and software procedure itself.

Following are the few of the testing methods applied to this successful project.

Source Code Testing

This source code testing is used to examine the logic of the system. If we are receiving the output that is necessary by the user, at that time we can declare that the logic is correct.

Specification Testing

We can set with, what program should do and how it should execute under various circumstances. This testing is a relative study of growth of system performance and system necessities.

Module Level Testing

In this testing the error will be initiated at each and every module, it encourages the programmer to discover and set right the errors without disturbing the remaining modules.

Unit Testing

This Unit testing concentrates on validating the attempt on the least unit of software-module. In this the confined data structure is verified to make sure that the date stored for the moment maintains its honesty during all steps in the algorithm's implementation. Boundary surroundings are tested to make sure that the module operates correctly at boundaries recognized to limit or confine processing.

Integration Testing

In this integration testing data is tested over an interface. One module can contain an unintended, unfavourable result on the other. Integration testing is a methodical method for constructing a program structure at the same time conducts tests to discover errors linked with interring.

Validation Testing

This validation testing continues later the integration testing is effectively assembled. Validation succeeds when the software functions in a approach that can be logically established by the customer. In this the best part of the validation is finished all through the data entry operation wherever there is at most possibility of ingoing wrong data. Other validation will be performed in all procedure wherever accurate particulars and data should be entered to obtain the necessary outcome.

Performance Testing

This type of testing is used to test runtime routine of software inside the perspective of an integrated structure. Performance test are frequently joined with stress testing and involve equally software instrumentation.

System Testing

System Testing is used to test the complete system as a whole and inspect for its accuracy is system testing. The system is planned for dispensaries among the system and its unique objectives. This project was successful and well-organized.

Output Testing

After the validation testing is performed, the subsequent step is output testing of the planned system because no system would be termed as helpful until it does create the necessary output in the particular format. Output format is measured in two traditions, the screen arrangement and the printer set-up.

User Acceptance Testing

User Acceptance Testing is the key feature for the accomplishment of any system. The system beneath thought is tested for user acceptance by continuously maintaining in touch with potential system users at the time of increasing and creating changes whenever necessary. The user acceptance testing points are as follows.

- Input Screen design
- Output Screen design
- Menu-driven System

4. DESIGN AND IMPLEMENTATION**SCHEMA-1****1. LOCATION**

LOCATION	ID	Address	Location_type
datatype	int	varchar(200)	varchar(400)
literals	123	Address	Metro/urban,semiurban,rural
PK	X		

2. AMC

AMC	ID	Name	Address	Mobile	Email
datatype	int	varchar(40)	varchar(100)	varchar(40)	varchar(40)
literals	1,2,3				
PK	x				
FK					

3. REPLACEMENT

REPLACEMENT	ID	Authority	LetterNum	Letterdate
datatype	int	varchar (40)	varchar (40)	date
literals	1,2 ,3			
PK	x			
FK				

DisconnectionDate	DisposalDate	Replacementvalue	Officername
date	date	int	varchar (40)

SBI ATM SYSTEM MANAGEMENT

4. ATM

ATM	ID	Asset-ID	Location ID	Type	Make
datatype	int	varchar(40)	int	varchar(40)	varchar(40)
literals	1,2,3	alpha-numeric		ATM, CDM, Recycler,CVM	Wincore,Deibold ,Hyoung
PK	x	unique			
FK			x(location-ID)		

Installation Date	Note Capacity -2000	Note Capacity -1000	Note Capacity -500	Note Capacity -100	Note Capacity -50	Replacement Details
date	int	int	int	int	int	int
date	number	Number	number	number	Number	int
						x(replacement-ID)

5. UPS

UPS	ID	Asset-ID	Location ID	Type
datatype	int	varchar(40)	int	varchar(40)
literals	1,2,3	alpha-numeric		Batterycharged/solar
PK	x	unique		
FK			x(location-ID)	

Make	Capacity	Installation_Date	AMC_Details	Replacement_Details
varchar(40)	int	date	int	int
Supplier name with 1year Warranty	KV(2digit)	date	int	int
			x(AMC:ID)	x (Replacement:ID)

SBI ATM SYSTEM MANAGEMENT

6. BATTERY

BATTERY	ID	Asset-ID	Location-ID	Type
datatype	int	varchar(40)	int	varchar(40)
literals	1,2,3	alpha-numeric		Sealed/water
PK	x	Unique		
FK			x(location-ID)	

Make	Capacity	Installation_Date	AMC_Details	Replacement_Details
varchar(40)	int	date	int	int
Supplier name with 1year Warrenty	Amh	date	int	int
			x(AMC:ID)	x (Replacement:ID)

7. AC

AC	ID	Asset-ID	Location-ID	Type
datatype	varchar(40)	varchar(40)	int	varchar(40)
literals	alpha-numeric	alpha-numeric		Window/split/casatteAC
PK	x	unique		
FK			x(location-ID)	

Make	Capacity	Installation_Date	AMC_Details	Replacement_Details
varchar(40)	float	date	int	int
Supplier name with 1year Warranty	tonnage	date	int	int
			x(AMC:ID)	x(Replacement:ID)

SBI ATM SYSTEM MANAGEMENT

8. STABILIZER

AC	ID	Asset-ID	Location-ID	Make
datatype	varchar(40)	varchar(40)	int	varchar(40)
literals	alpha-numeric	alpha-numeric		Supplier name with 1year Warranty
PK	x	Unique		
FK			x(location-ID)	

Installation-Date	AMC-Details	Replacement Details
date	int	int
date	int	int
	x(AMC:ID)	x (Replacement:ID)

9. CAMERA

CAMERA	ID	Asset-ID	Location-ID	Type
datatype	varchar(40)	varchar(40)	int	varchar(40)
literals	alpha-numeric	alpha-numeric		Normal,InfraRed,Domo,IP
PK	x	unique		
FK			x(location-ID)	

Make	Installation_Date	AMC_Details	Replacement_Details
varchar(40)	date	int	int
Supplier name with 1year Warranty	date	int	int
		x(AMC:ID)	x (Replacement:ID)

SBI ATM SYSTEM MANAGEMENT

10. OWNER

OWNER	ID	Name	Location-ID	Permanent Address
datatype	int	varchar(40)	int	varchar(100)
literals	1,2,3	Name		
PK	x			
FK			x(location-ID)	

Communication Address	Mobile	Alternate Contact	SB Account Number	Other Account Details
varchar(100)	int	int		varchar(2000)
				json

11. LEASE

LEASE	ID	Location-ID	Lease Months
datatype	int	int	int
literals	1,2,3		
PK	X		
FK		x(location-ID)	

Lease From	Lease To	Lease Rent
date	date	Number(5)

12. ESC

ESC	ID	ESC-NO	Location-ID
datatype	int	int	int
literals	1,2,3		
PK	X	unique	
FK			x(location:ID)

Office-Name	Office_Address	Office-Phone
varchar(40)	varchar(40)	varchar(40)

SBI ATM SYSTEM MANAGEMENT

13. AGENCY

AGENCY	ID	Name
datatype	int	varchar(40)
literals	1,2,3	
PK	X	
FK		

14. SECURITY

SECURITY	ID	Agency	Location	Constitution
datatype	int	int	int	varchar(40)
literals	1,2,3			private/public
PK	x			
FK		x(Agency : ID)	x(location : ID)	

KYC	Authority	Letter Reference Number	Cash-Handled by
varchar(40)	varchar(40)	varchar	varchar(200)
KYC proof	Authority name		Fullname,KYCdetails,permanent address,temporary address

15. CRA

CRA	ID	Agency	Name
datatype	int	int	varchar(40)
literals	1,2,3		
PK	X		
FK		X(Agency : ID)	

KYC	Permanent Address	Communication Address	Mobile
varchar(40)	varchar(100)	varchar(100)	int
KYC proof			

SBI ATM SYSTEM MANAGEMENT

16. DRIVER

DRIVER	ID	Name	KYC	Permanent Address	Communication Address	Mobile
datatype	int	varchar(40)	varchar(40)	varchar(100)	varchar(100)	int
literals	1,2,3		KYC proof			
PK	x					
FK						

17. VEHICLE

VEHICLE	ID	Owner	Details	Registration	Driver	Standby_driver
datatype	int	varchar(40)	varchar(40)	varchar(40)		
literals	1,2,3	name	alphanumeric	alphanumeric		
PK	x					
FK					x(Driver:ID)	x(Driver : ID)

SCHEMA-2

1. LOCATION

LOCATION	ID	Address	Type
datatype	int	varchar(200)	varchar(40)
literals	1,2,3	Address	urban,semiurban,rural
PK	x		

2. ASSET

ASSET	ID	Location-ID	Type
datatype	int	int	varchar(40)
literals	1,2,3		ATM,UPS,Battery,AC,Stabilizer,Camera
PK	x		
FK		x(location-ID)	

Make	INS_Date
Varchar(40)	date
Wincore, Diebold, Hyosung	date

SBI ATM SYSTEM MANAGEMENT

3. ASSET_PROP

ASSET_PRO P	ID	Asset-ID	PROP_NAME	PROP_VALU E
datatype	int	int	Varchar(40)	Varchar(100)
literals	1,2, 3		note_capacity_2000= note_capacity_1000= note_capacity_500= note_capacity_100= note_capacity_50= type=ATM, CDM, Recycler, Coin Vending Machine, Battery Charged, Solar, Sealed, Water, Window/split/casatteAC, Normal, Infra Red, Dome, IP capacity= AMC_NAME= AMC_ADDRESS= AMC_MOBILE= AMC_EMAIL= replacement_authority= replacement_letter_no= replacement_letter_date= replacement_disconnection_date = replacement_disposal_date= replacement_value= replacement_office_name=	
PK	x			
FK		x(asset:ID)		

4. OWNER

OWNER	ID	Name	Location-ID	Permanent Address
datatype	int	varchar(40)	int	varchar(100)
literals	1,2,3	Name		
PK	x			
FK			x(location-ID)	

Communication Address	Mobile	Alternate Contact	SB Account Number	Other Account Details
varchar(100)	int	int		Varchar(2000)
				json

SBI ATM SYSTEM MANAGEMENT

5. LEASE

LEASE	ID	Location-ID	Lease Months
datatype	int	int	int
literals	1,2,3		
PK	X		
FK		x(location-ID)	

Lease From	Lease To	Lease Rent
date	date	Number(5)

6. ESC

ESC	ID	ESC-NO	Location-ID
datatype	int	int	int
literals	1,2,3		
PK	X	unique	
FK			x(location:ID)

Office-Name	Office_Address	Office-Phone
varchar(40)	varchar(100)	varchar(40)

7. AGENCY

AGENCY	ID	Name
datatype	int	varchar(40)
literals	1,2,3	
PK	X	
FK		

SBI ATM SYSTEM MANAGEMENT

8. SECURITY

SECURITY	ID	Agency	Location	Constitution
datatype	int	int	int	varchar(40)
literals	1,2,3			private/public
PK	x			
FK		x(Agency:ID)	x(location-ID)	

KYC	Authority	Letter Reference Number	
varchar(40)	varchar(40)	varchar	
KYC proof	Authority name		

9. CRA

CRA	ID	Agency	Name
datatype	int	int	Varchar(40)
literals	1,2,3		
PK	X		
FK		X(Agency:ID)	

KYC	Permanent Address	Communication Address	Mobile
varchar(40)	varchar(100)	varchar(100)	int
KYC proof			

10. DRIVER

DRIVER	ID	Name	KYC	Permanent Address	Communication Address	Mobile
datatype	int	varchar(40)	varchar(40)	varchar(100)	varchar(100)	int
literals	1,2,3		KYC proof			
PK	x					
FK						

11. VEHICLE

VEHICLE	ID	Owner	Details	Registration	Driver	Stand by driver
datatype	int	varchar(40)	varchar(40)	varchar(40)	int	int
literals	1,2,3	name	alphanumeric	alphanumeric		
PK	x					
FK					x(Driver:ID)	x(Driver:ID)

MODULES

- Location
- Assets
- Asset Properties
- Owner Information
- Agency Details
- Vehicle Details
- Security Personnel Details
- Cash Replenishment Details

Module Description**Location**

In this module location of ATM is specified. It includes the location of ATM.

Assets

In this module assets present in the ATM are specified. At which location which type of assets are present are specified.

Asset properties

In this module properties of assets are specified. Every asset has its own properties and those are specified.

Owner Information

In this module details of owner that is from whom the place has been taken for lease for ATM is present.

Agency Details

In this module details of Agencies are present. These agencies supply Security guards, Vehicles and Drivers for vehicles for ATM purpose.

Vehicle Details

In this module Vehicle details that is Registration number, driver details and stand by driver details are included. These vehicles are used to take money to the ATM locations and to place in the ATM.

Security personnel Details

In this module Details of Security personnel are present. Each ATM may be guarded by security personnel, who are engaged from outside agencies and the details of security personnel are recorded.

Cash Replenishment Details

In this module details of the person who is responsible for keeping money in ATM at different locations are maintained.

HARDWARE SPECIFICATION

System	:	Intel core i5 processor
Hard Disk	:	1 TB
Ram	:	1 GB.

SOFTWARE SPECIFICATION Operating

system	:	UBUNTU 14.04 Coding
language	:	PHP, HTML, CSS
Database	:	MYSQL Oracle
Tool	:	Apache Version-2

IMPLEMENTATION

Login Page

```
<!DOCTYPE HTML>

<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
    <title>Login</title>

    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <?php
        /*
        echo "<pre>";
        print_r($_SERVER);
        echo "</pre>";
        */
    ?>

    <div class="heading">
        <center><h1>Welcome to SBI dashboard!</h1></center>
    </div>

    <form method="post" action="validate.php">
    <div class="tableswrapper">
        <table align="center" class="mytable">
            <tr>
                <th>Username</th>
                <td><input type="text" name="username"></td>
            </tr>
            <tr>
                <th>Password</th>
                <td><input type="password" name="password"></td>
            </tr>
        </table>
    </div>
    <div class="buttonswrapper">
        <button class="btn" type="submit" value="Login"
name="LoginButton">Login</button>
    </div>
</form>
</body>
</html>
```

Home Page

```
<?php
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?><!DOCTYPE HTML>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
        <title>SBI Home</title>

        <link rel="stylesheet" href="style.css" />
    </head>
    <body>
        <?php require_once('header.php'); ?>
        <div class="heading">
            <center><h1>Welcome <?php echo $_SESSION['username'];
?>!</h1></center>
            <center><h2>This dashboard allows you to manage all SBI locations, their
assets and operations.</h2></center>
        </div>
        <div class="tableswrapper">
            <table align="center" class="mytable">
                <tr>
                    <td>
                        <center><h3><u>Locations</u></h3></center>
                        <table align="center" class="mytable">
                            <tr><td><a href="addlocations.php">Add locations</a></td></tr>
                            <tr><td><a href="viewlocations.php">View locations</a></td></tr>
                            <tr><td><a href="modifylocations.php">Modifylocations</a></td></tr>
                        </table>
                    </td>
                    <td>
                        <center><h3><u>Assets</u></h3></center>
                        <table align="center" class="mytable">
                            <tr><td><a href="addassets.php">Add assets</a></td></tr>
                            <tr><td><a href="viewassets.php">View assets</a></td></tr>
                            <tr><td><a href="modifyassets.php">Modify assets</a></td></tr>
                        </table>
                    </td>
                    <td>
                        <center><h3><u>Asset Property</u></h3></center>
                        <table align="center" class="mytable">
                            <tr><td><a href="addap.php">Add assetsProp</a></td></tr>
```

```
<tr><td><a href="viewap.php">View assetsProp</a></td></tr>
<tr>
<td><a href="modifyap.php">Modify assetsProp</a></td></tr>
</table>
</td>
<td>
<center><h9><u>Owner Information</u></h9></center>
<table align="center" class="mytable">
<tr>
<td><a href="addoi.php">AddOwnerInfo</a></td></tr>
<tr><td><a href="viewoi.php">View OwnerInfo</a></td></tr>
<tr><td><a href="modifyoi.php">Modify OwnerInfo</a></td></tr>
</table>
</td>
</tr>
<tr>
<td>
<center><h9><u>Vehicle Details</u></h9></center>
<table align="center" class="mytable">
<tr><td><a href="addvehicle.php">Add vehicle details</a></td></tr>
<tr><td><a href="viewvehicle.php">View vehicle details</a></td></tr>
<tr><td><a href="modifyvehicle.php">Modify vehicle details</a></td></tr>
</table>
</td>
<td>
<center><h9><u>Agencies</u></h9></center>
<table align="center" class="mytable">
<tr><td><a href="addAgency.php">Add Agency</a></td></tr>
<tr><td><a href="viewAgency.php">View Agency</a></td></tr>
<tr><td><a href="modifyAgency.php">Modify Agency</a></td></tr>
</table>
</td>
<td>
<center><h9><u>Security Details</u></h9></center>
<table align="center" class="mytable">
<tr><td><a href="addsecurity.php">Add security details</a></td></tr>
<tr><td><a href="viewsecurity.php">View security details</a></td></tr>
<tr><td><a href="modifysecurity.php">Modify security details</a></td></tr>
</table>
</td>
<td>
<center><h9><u>Cash Replenishment Details</u></h9></center>
<table align="center" class="mytable">
<tr><td><a href="addcra.php">Add Cash replenishment
details</a></td></tr>
<tr><td><a href="viewcra.php">View Cash replenishment
details</a></td></tr>
<tr><td><a href="modifycra.php">Modify Cash replenishment
details</a></td></tr>
</tr>
```

```
        </table>
      </td>
    </tr>
  </table>
</div>
</body>
</html>
```

Locations

```
<?php
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?>
<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
        <title>Add, View or Modify SBI Locations</title>

        <link rel="stylesheet" href="style.css" />
    </head>
    <body>
        <?php require_once('header.php'); ?>

        <div class="heading">
            <center><h1>Add, View or Modify SBI Locations!</h1></center>
        </div>

        <div class="tablewrapper">
            <table align="center" class="mytable">
                <tr>
                    <td><a href="addlocations.php">Add
locations</a></td>
                </tr>
                <tr>
                    <td><a href="viewlocations.php">View
locations</a></td>
                </tr>
                <tr>
                    <td><a href="modifylocations.php">Modify
locations</a></td>
                </tr>
            </table>
        </div>
```

```
</body>
</html>
```

Add Locations

```
<?php
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?>
<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
    <title>Add Locations</title>

    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <?php
        require_once('header.php');
        require_once('db.php');
        require_once('utils.php');

        $location = $type = "";
        $locationErr = $errorMsg = "";

        if (isPost()) {
            // clear the errorMsg to begin with
            $errorMsg = "";

            $location = test_input($_POST['location']);
            $type = test_input($_POST['type']);

            if (empty($location)) {
                $locationErr = "Enter a valid location.";
                $errorMsg = "Fix highlighted errors.";
            }

            if ($errorMsg == "" && insertLocation($location, $type, $errorMsg)) {
                resetFields();
            }
        }
        function resetFields()
        {
            global $location, $type;
```

```
        $location = "";
        $type = "";
    }
?>
<div class="heading">
    <center><h1>Add a new SBI ATM location!</h1></center>
</div>
<form method="post" action="<?php echo htmlspecialchars($_SERVER['PHP_SELF']);
?>">
    <div class="tableswrapper">
        <table align="center" class="mytable">
            <tr>
                <th align="left">Location</th>
                <td>
                    <textarea name="location" rows="10" cols="40"
maxlength="100"><?php echo $location;?></textarea>
                    <span class="error">*<?php echo
$locationErr;?></span>
                </td>
            </tr>
            <tr>
                <th align="left">Type</th>
                <td align="left"><select name="type">
                    <?php
                        foreach($locationTypeArr as $key) {
                            ?>
                            <option value="<?= $key ?>" title="<?=
htmlspecialchars($key) ?>" <?php if ($key==$type) echo "selected=\"selected\"";?> >
                                <?= htmlspecialchars($key) ?>
                            </option>
                        <?php
                            }
                        ?>
                    </select>
                </td>
            </tr>
        </table>
    </div>

    <div class="buttonswrapper">
        <button class="btn" type="submit" value="Add"
name="AddButton">ADD</button>
    </div>
</form>
<?php
    if (isPost()) {
        if ($errorMsg == "")
            echo_success("Location added successfully");
        else
```

```
        echo_error($errorMsg);
    }
    ?>
</body>
</html>

View Locations

<?php
    // http://www.webslesson.info/2016/09/angularjs-tutorial-with-php-fetch-select-data-
    from-mysql-database.html
    // https://stackoverflow.com/questions/19122942/angularjs-sorting-rows-by-table-
    header
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
    $_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
    ?>

<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
        <title>View Locations</title>

        <link rel="stylesheet" href="style.css" />
        <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
    </head>
    <body>
        <?php
            require_once('header.php');
            ?>

            <div class="heading">
                <center><h1>All locations of SBI ATMs!</h1></center>
            </div>

            <div class="tableswrapper" ng-app="myapp" ng-controller="assetcontroller"
ng-init="displayData()">
                <table align="center" class="mytable">
                    <tr bgcolor="yellow">
                        <th>
                            <a href="#" ng-click="orderByField='id';
reverseSort = !reverseSort"> ID
                                <span ng-show="orderByField == 'id'">
                                    <span ng-show="!reverseSort">
```



```

                                ^
                                </span>
                                <span ng-show="reverseSort">
                                v
                                </span>
                                </span>
                                </a>
                                </th>
                                <th>
                                <a href="#" ng-click="orderByField='location';
reverseSort = !reverseSort"> Location
                                <span ng-show="orderByField ==
'location'">
                                <span ng-show="!reverseSort">
                                ^
                                </span>
                                <span ng-show="reverseSort">
                                v
                                </span>
                                </span>
                                </a>
                                </th>
                                <th>
                                <a href="#" ng-click="orderByField='type';
reverseSort = !reverseSort"> Type
                                <span ng-show="orderByField ==
'type'">
                                <span ng-show="!reverseSort">
                                ^
                                </span>
                                <span ng-show="reverseSort">
                                v
                                </span>
                                </span>
                                </a>
                                </th>
                                </tr>
                                <tr ng-repeat="x in locations |
orderBy:orderByField:reverseSort">
                                <td>{{ x.id }}</td>
                                <td>{{ x.location }}</td>
                                <td>{{ x.type }}</td>
                                </tr>
                                </table>
                                </div>
                                </body>
                                </html>
                                <script>
                                var app = angular.module("myapp",[]);
                                app.controller("assetcontroller", function($scope, $http)

```

```
{
    $scope.displayData = function()
    {
        $http.get("api/getlocations.php").success(function(data)
        {
            $scope.orderByField = 'id';
            $scope.reverseSort = false;
            $scope.locations = data;
        });
    }
});
</script>
```

Modify Locations

```
<?php
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?>

<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
    <title>Modify Locations</title>

    <link rel="stylesheet" href="style.css" />
</head>
<body>
    <?php
        require_once('header.php');
        require_once('db.php');
        require_once('utils.php');
        $id_deleted = "0";
        $idName = $location = $type = "";
        $idErr = $locationErr = $errorMsg = "";
        reloadLocationIdNames($errorMsg);
        resetFields();
        if ($errorMsg == "" && isPost()) {
            $idName = $_POST['idNames'];
            if ($idName == "0") {
                $idErr = "Select valid location id.";
                $errorMsg = "Fix highlighted errors.";
            }
        }
    }
?>
```

```
        else if (!isset($_POST['ChangeButton']) &&
!isset($_POST['DeleteButton'])) {
            // this was triggered by a change in the id drop down!
            $data = NULL;
            foreach($records as $r) {
                if ($r['id'] == getIdFromIdName($idName)) {
                    $data = $r;
                    $location = $data['location'];
                    $type = $data['type'];
                    break;
                }
            }
        }
        else {
            $location = $_POST['location'];
            $type = $_POST['type'];
            if (isset($_POST['ChangeButton'])) {
                updateLocation(getIdFromIdName($idName),
$location, $type, $errorMsg);
            }
            else if (isset($_POST['DeleteButton']) &&
deleteLocation(getIdFromIdName($idName), $errorMsg)) {
                $id_deleted = $idName;
                resetFields();
            }
            reloadLocationIdNames($errorMsg);
        }
    }
    function reloadLocationIdNames(&$errorMsg) {
        global $records, $idNames;
        $records = fetchAllLocations($errorMsg);
        $idNames = array();
        array_push($idNames, "0");

        foreach($records as $r)
            array_push($idNames, $r['id']. '-' . $r['location']);
    }
    function resetFields() {
        global $idName, $location, $type;

        $idName = "0";
        $location = "";
        $type = "";
    }
?>
<div class="heading">
    <center><h1>Modify any SBI ATM location!</h1></center>
</div>
```

SBI ATM SYSTEM MANAGEMENT

```
<form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>">
    <div class="tablewrapper">
        <table align="center" class="mytable">
            <tr>
                <th align="left">ID</th>
                <td align="left">
                    <select id = "idNames" name="idNames"
onchange="this.form.submit()">
                        <?php
                            foreach($idNames as $key) {
                                ?>
                                <option value="<?= $key ?>" title="<?=
htmlspecialchars($key) ?>" <?php if
(getIdFromIdName($key)==getIdFromIdName($idName)) echo "selected=\"selected\"";?> >
                                    <?= htmlspecialchars($key) ?>
                                </option>
                                <?php
                                    }
                                ?>
                            </select>
                            <span class="error">* <?php echo $idErr;?> </span>
                        </td>
                    </tr>
                    <tr>
                        <th align="left">Location</th>
                        <td align="left"><textarea name="location" rows="10"
cols="40" maxlength="100"><?php echo $location;?></textarea>
                        </td>
                    </tr>
                    <tr>
                        <th align="left">Type</th>
                        <td align="left"><select name="type">
                            <?php
                                foreach($locationTypeArr as $key) {
                                    ?>
                                    <option value="<?= $key ?>" title="<?=
htmlspecialchars($key) ?>" <?php if ($key==$type) echo "selected=\"selected\"";?> >
                                        <?= htmlspecialchars($key) ?>
                                    </option>
                                    <?php
                                        }
                                    ?>
                                </select>
                            </td>
                        </tr>
                    </table>
                </div>
```

```
<div class="buttonswrapper">
    <button class="btn" type="submit" value="Change"
name="ChangeButton">CHANGE</button>
    <button class="btn" type="submit" value="Delete" name="DeleteButton"
onclick="return confirm('Are you sure you want to DELETE?');">DELETE</button>
</div>
</form>

<?php
    if (isPost()) {
        if ($errorMsg == "" && isset($_POST['ChangeButton']))
            echo_success("Location id " . getIdFromIdName($idName) . "
changed successfully!");
        else if ($errorMsg == "" && isset($_POST['DeleteButton']))
            echo_success("Location " . $id_deleted . " deleted
successfully");
        else
            echo_error($errorMsg);
    }
?>
</body>
</html>
```

Add Owner Information

```
<?php
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?>

<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
    <title>Modify Owner Information</title>

    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

    <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    <script>
        $(
            function()    {
                                $( "#leasedate" ).datepicker();
                            }
        );
    </script>
</head>
<body>
    <?php
        require_once('header.php');
        require_once('db.php');
        require_once('utils.php');

        $location_id_name_deleted = "0";
        $locationErr = $nameErr = $errorMsg = "";

        reloadLocationIdNames($errorMsg);
        resetFields();

        if ($errorMsg == "" && isPost()) {
            $name = $_POST['name'];
            if ($name == "select") {
                $location_id_nameErr = "Select valid owner.";
                $errorMsg = "Fix highlighted errors.";
            }
        }
    </?php
        </body>
</html>
```

```
        else if (!isset($_POST['ChangeButton']) &&
!isset($_POST['DeleteButton'])) {
            // this was triggered by a change in the id drop down!
            $data = NULL;
            foreach($records as $r) {
                if ($r['o_id'] == getIdFromname($name)) {
                    $data = $r;
                    $location_id_name = $data['location'];
                    $address = $data['address'];
                    $leasedate = $data['leasedate'];
                    $leaseperiod = $data['leaseperiod'];
                    $leaserent = $data['leaserent'];
                    $mobilenos = $data['mobilenos'];
                    $pancard = $data['pancard'];
                    $Aadhaarcid = $data['Aadhaarcid'];
                    $acno = $data['acno'];
                    break;
                }
            }
        }
        else {
            $location_id_name = $_POST['location'];
            $address= $_POST['address'];
            $leasedate = $_POST['leasedate'];
            $leaseperiod = $_POST['leaseperiod'];
            $leaserent = $_POST['leaserent'];
            $mobilenos = $_POST['mobilenos'];
            $pancard = $_POST['pancard'];
            $Aadhaarcid = $_POST['Aadhaarcid'];
            $acno = $_POST['acno'];
            if (isset($_POST['ChangeButton'])) {
                updateownerinfo(getIdFromname($name), $address,
sql_date($leasedate), $leaseperiod, $leaserent, $mobilenos, $pancard, $Aadhaarcid, $acno,
$errorMsg);
            }
            else if (isset($_POST['DeleteButton']) &&
deleteownerinfo(getIdFromname($name), $errorMsg)) {
                $location_id_name_deleted = $name;
                resetFields();
            }
        }
    }
    function reloadLocationIdNames(&$errorMsg) {
        global $records, $locationIdNameArr;
        $records = fetchAllownerinfo($errorMsg);
        $locationIdNameArr = array();
        array_push($locationIdNameArr, "select");
        foreach($records as $r)
            array_push($locationIdNameArr,$r['o_id']. '-' . $r['name']);
    }
}
```

```
    }

    function resetFields() {
        global $location_id_name, $name, $address, $leasedate, $leaseperiod,
        $leaserent, $mobileneno, $pancard, $Aadhaarcad, $acno ;

        $location_id_name = $name = $address = $leasedate = $leaseperiod =
        $leaserent = $mobileneno = $pancard = $Aadhaarcad = $acno = "";

    }
?>
<div class="heading">
    <center><h1>Modify SBI ATM Location Owner Information!</h1></center>
</div>

<form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>">
    <div class="tablewrapper">
        <table align="center" class="mytable">
            <tr>
                <th align="left">Name</th>
                <td align="left">
                    <select type = "name" name="name"
onchange="this.form.submit()">
                        <?php
                            foreach($locationIdNameArr as $key) {
                                ?>
                                <option value="<?= $key ?>" title="<?=
htmlspecialchars($key) ?>" <?php if ($key==$name) echo "selected=\"selected\"";?> >
                                    <?= htmlspecialchars($key) ?>
                                </option>
                            <?php
                                }
                            ?>
                        </select>
                        <span class="error">* <?php echo
$location_id_nameErr;?> </span>
                    </td>
                </tr>

                <tr>
                    <th align="left">Location</th>
                    <td align="left"><input type="text" id="location"
name="location" value="<?php echo $location_id_name;?>">
                    </td>
                </tr>

                <tr>
                    <th align="left">Address</th>
                    <td>
```


SBI ATM SYSTEM MANAGEMENT

```
<textarea name="address" rows="5" cols="25"
maxlength="100"><?php echo $address;?></textarea>
</td>
</tr>
<tr>
<th align="left">Lease Start Date</th>
<td align="left"><input type="text" id="leasedate"
name="leasedate" value="<?php echo $leasedate;?>">
</td>
</tr>
<tr>
<th align="left">Lease Period</th>
<td align="left"><input type="text" id="leaseperiod"
name="leaseperiod" value="<?php echo $leaseperiod;?>">
</td>
</tr>
<tr>
<th align="left">Lease Rent</th>
<td align="left"><input type="text" id="leaserent"
name="leaserent" value="<?php echo $leaserent;?>">
</td>
</tr>
<tr>
<th align="left">Mobile Number</th>
<td align="left"><input type="text" id="mobileneno"
name="mobileneno" value="<?php echo $mobileneno;?>">
</td>
</tr>
<tr>
<th align="left">Pan Card</th>
<td align="left"><input type="text" id="pancard"
name="pancard" value="<?php echo $pancard;?>">
</td>
</tr>
<tr>
<th align="left">Aadhaar Card</th>
<td align="left"><input type="text" id="Aadhaarcad"
name="Aadhaarcad" value="<?php echo $Aadhaarcad;?>">
</td>
</tr>
<tr>
<th align="left">Account Number</th>
<td align="left"><input type="text" id="acno" name="acno"
value="<?php echo $acno;?>">
</td>
</tr>
</table>
</div>
```

```
<div class="buttonswrapper">
    <button class="btn" type="submit" value="Change"
name="ChangeButton">CHANGE</button>
    <button class="btn" type="submit" value="Delete" name="DeleteButton"
onclick="return confirm('Are you sure you want to
DELETE?');">DELETE</button>
</div>
</form>

<?php
    if (isPost()) {
        if ($errorMsg == "" && isset($_POST['ChangeButton']))
            echo_success("Owner details " .
getIdFromIdName($location_id_name) . " changed successfully!");
        else if ($errorMsg == "" && isset($_POST['DeleteButton']))
            echo_success("Owner " . $location_id_name_deleted . " deleted
successfully");
        else
            echo_error($errorMsg);
    }
?>
</body>
</html>
```

View Owner Information

```
<?php
    // http://www.webslesson.info/2016/09/angularjs-tutorial-with-php-fetch-select-data-
from-mysql-database.html
    // https://stackoverflow.com/questions/19122942/angularjs-sorting-rows-by-table-
header
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?>

<!DOCTYPE HTML>
<html>
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
        <title>View Owners Information</title>

        <link rel="stylesheet" href="style.css" />
        <script
src="http://ajax.googleapis.com/ajax/libs/angularjs/1.4.8/angular.min.js"></script>
```

SBI ATM SYSTEM MANAGEMENT

```
</head>
<body>
  <?php
    require_once('header.php');
  ?>

  <div class="heading">
    <center><h1>SBI ATM Location Owner Details</h1></center>
  </div>

  <div class="tablewrapper" ng-app="myapp" ng-controller="assetcontroller"
ng-init="displayData()">
    <table align="center" class="mytable">
      <tr bgcolor="yellow">
        <th>
          <a href="#" ng-click="orderByField='o_id';
reverseSort = !reverseSort"> Owner Id
          <span ng-show="orderByField ==
'o_id'">
            <span ng-show="!reverseSort">
              ^
            </span>
            <span ng-show="reverseSort">
              v
            </span>
          </span>
        </a>
      </th>
      <th>
          <a href="#" ng-click="orderByField='name';
reverseSort = !reverseSort"> Name
          <span ng-show="orderByField ==
'name'">
            <span ng-show="!reverseSort">
              ^
            </span>
            <span ng-show="reverseSort">
              v
            </span>
          </span>
        </a>
      </th>
      <th>
          <a href="#" ng-click="orderByField='address';
reverseSort = !reverseSort"> Address
          <span ng-show="orderByField ==
'address'">
            <span ng-show="!reverseSort">
              ^
            </span>
          </span>
        </a>
      </th>
    </tr>
  </table>
</div>
```

```

                                <span ng-show="reverseSort">
                                    v
                                </span>
                            </span>
                        </a>
                    </th>
                    <th>
                        <a href="#" ng-click="orderByField='leasedate';
reverseSort = !reverseSort"> Lease Start Date
                                <span ng-show="orderByField ==
'leasedate'">
                                    <span ng-show="!reverseSort">
                                        ^
                                    </span>
                                    <span ng-show="reverseSort">
                                        v
                                    </span>
                                </span>
                            </a>
                        </th>
                        <th>
                            <a href="#" ng-
click="orderByField='leaseperiod'; reverseSort = !reverseSort"> Lease period
                                <span ng-show="orderByField ==
'leaseperiod'">
                                    <span ng-show="!reverseSort">
                                        ^
                                    </span>
                                    <span ng-show="reverseSort">
                                        v
                                    </span>
                                </span>
                            </a>
                        </th>
                        <th>
                            <a href="#" ng-click="orderByField='leaserent';
reverseSort = !reverseSort"> Lease rent
                                <span ng-show="orderByField ==
'leaserent'">
                                    <span ng-show="!reverseSort">
                                        ^
                                    </span>
                                    <span ng-show="reverseSort">
                                        v
                                    </span>
                                </span>
                            </a>
                        </th>
                        <th>
```

SBI ATM SYSTEM MANAGEMENT

```
reverseSort = !reverseSort"> Mobile Number
reverseSort = !reverseSort"> Pan card
reverseSort = !reverseSort"> Aadhaar card
reverseSort = !reverseSort"> Account Number

<a href="#" ng-click="orderByField='mobilen';
  <span ng-show="orderByField ==
    <span ng-show="!reverseSort">
      ^
    </span>
    <span ng-show="reverseSort">
      v
    </span>
  </span>
</a>
</th>
<th>
  <a href="#" ng-click="orderByField='pancard';
    <span ng-show="orderByField ==
      <span ng-show="!reverseSort">
        ^
      </span>
      <span ng-show="reverseSort">
        v
      </span>
    </span>
  </a>
  </th>
  <th>
    <a href="#" ng-
click="orderByField='Aadhaarc'; reverseSort = !reverseSort"> Aadhaar card
    <span ng-show="orderByField ==
      <span ng-show="!reverseSort">
        ^
      </span>
      <span ng-show="reverseSort">
        v
      </span>
    </span>
  </a>
  </th>
  <th>
    <a href="#" ng-click="orderByField='acno';
reverseSort = !reverseSort"> Account Number
    <span ng-show="orderByField ==
      <span ng-show="!reverseSort">
        ^
      </span>
      <span ng-show="reverseSort">
```

```

                                v
                                </span>
                            </span>
                        </a>
                    </th>
                    <th>
                        <a href="#" ng-click="orderByField='l_id';
reverseSort = !reverseSort"> Location Id
                                <span ng-show="orderByField ==
'l_id'">
                                    <span ng-show="!reverseSort">
                                        ^
                                    </span>
                                    <span ng-show="reverseSort">
                                        v
                                    </span>
                                </span>
                            </a>
                    </th>
                    <th>
                        <a href="#" ng-click="orderByField='location';
reverseSort = !reverseSort"> Location
                                <span ng-show="orderByField ==
'location'">
                                    <span ng-show="!reverseSort">
                                        ^
                                    </span>
                                    <span ng-show="reverseSort">
                                        v
                                    </span>
                                </span>
                            </a>
                    </th>
                </tr>
                <tr ng-repeat="x in ownerinfo |
orderBy:orderByField:reverseSort">
                    <td>{{ x.o_id }}</td>
                    <td>{{ x.name }}</td>
                    <td>{{ x.address }}</td>
                    <td>{{ x.leasedate }}</td>
                    <td>{{ x.leaseperiod }}</td>
                    <td>{{ x.leaserent }}</td>
                    <td>{{ x.mobileno }}</td>
                    <td>{{ x.pancard }}</td>
                    <td>{{ x.Aadhaarcard }}</td>
                    <td>{{ x.acno }}</td>
                    <td>{{ x.l_id }}</td>
                    <td>{{ x.location }}</td>
                </tr>
            </table>

```

```
        </div>
    </body>
</html>
<script>
    var app = angular.module("myapp",[]);
    app.controller("assetcontroller", function($scope, $http)
    {
        $scope.displayData = function()
        {
            $http.get("api/getowner.php").success(function(data)
            {
                $scope.orderByField = 'o_id';
                $scope.reverseSort = false;
                $scope.ownerinfo = data;
            });
        }
    });
</script>
```

Modify Owner Information

```
<?php
    session_start();
    if (!isset($_SESSION['username']) || ($_SESSION['username'] != "sbiadmin" &&
$_SESSION['username'] != "sbiguest")) {
        header('location:login.php');
        exit;
    }
?>

<!DOCTYPE HTML>
<html>
<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,initial-scale=0.1,user-
scalable=yes">
    <title>Modify Owner Information</title>

    <link rel="stylesheet" href="style.css" />
    <link rel="stylesheet" href="//code.jquery.com/ui/1.12.1/themes/base/jquery-ui.css">

    <script src="https://code.jquery.com/jquery-1.12.4.js"></script>
    <script src="https://code.jquery.com/ui/1.12.1/jquery-ui.js"></script>
    <script>
        $(
            function()      {
                                $( "#leasedate" ).datepicker();
                            }
        );
    </script>
```

```
</head>
<body>
    <?php
        require_once('header.php');
        require_once('db.php');
        require_once('utils.php');

        $location_id_name_deleted = "0";
        $locationErr = $nameErr = $errorMsg = "";

        reloadLocationIdNames($errorMsg);
        resetFields();

        if ($errorMsg == "" && isPost()) {
            $name = $_POST['name'];
            if ($name == "select") {
                $location_id_nameErr = "Select valid owner.";
                $errorMsg = "Fix highlighted errors.";
            }
            else if (!isset($_POST['ChangeButton']) &&
!isset($_POST['DeleteButton'])) {
                // this was triggered by a change in the id drop down!
                $data = NULL;
                foreach($records as $r) {
                    if ($r['o_id'] == getIdFromname($name)) {
                        $data = $r;
                        $location_id_name = $data['location'];
                        $address = $data['address'];
                        $leasedate = $data['leasedate'];
                        $leaseperiod = $data['leaseperiod'];
                        $leaserent = $data['leaserent'];
                        $mobilenos = $data['mobilenos'];
                        $pancard = $data['pancard'];
                        $Aadhaarcid = $data['Aadhaarcid'];
                        $acno = $data['acno'];
                        break;
                    }
                }
            }
            else {
                $location_id_name = $_POST['location'];
                $address= $_POST['address'];
                $leasedate = $_POST['leasedate'];
                $leaseperiod = $_POST['leaseperiod'];
                $leaserent = $_POST['leaserent'];
                $mobilenos = $_POST['mobilenos'];
                $pancard = $_POST['pancard'];
                $Aadhaarcid = $_POST['Aadhaarcid'];
                $acno = $_POST['acno'];
                if (isset($_POST['ChangeButton'])) {
```

SBI ATM SYSTEM MANAGEMENT

```
        updateownerinfo(getIdFromname($name), $address,
sql_date($leasedate), $leaseperiod, $leaserent, $mobilenos, $pancard, $Aadhaarcad, $acno,
$errorMsg);
    }
    else if (isset($_POST['DeleteButton']) &&
deleteownerinfo(getIdFromname($name), $errorMsg)) {
        $location_id_name_deleted = $name;
        resetFields();
    }
}
}
function reloadLocationIdNames(&$errorMsg) {
    global $records, $locationIdNameArr;
    $records = fetchAllownerinfo($errorMsg);
    $locationIdNameArr = array();
    array_push($locationIdNameArr, "select");
    foreach($records as $r)
        array_push($locationIdNameArr,$r['o_id']. '-' . $r['name']);
}

function resetFields() {
    global $location_id_name, $name, $address, $leasedate, $leaseperiod,
$leaserent, $mobilenos, $pancard, $Aadhaarcad, $acno ;

    $location_id_name = $name = $address = $leasedate = $leaseperiod =
$leaserent = $mobilenos = $pancard = $Aadhaarcad = $acno = "";
}
?>
<div class="heading">
    <center><h1>Modify SBI ATM Location Owner Information!</h1></center>
</div>

    <form method="post" action="<?php echo
htmlspecialchars($_SERVER['PHP_SELF']); ?>">
    <div class="tableswrapper">
        <table align="center" class="mytable">
            <tr>
                <th align="left">Name</th>
                <td align="left">
                    <select type = "name" name="name"
onchange="this.form.submit()">
                        <?php
                            foreach($locationIdNameArr as $key) {
                                ?>
                                <option value="<?=$key ?>" title="<?=$
htmlspecialchars($key) ?>" <?php if ($key==$name) echo "selected=\"selected\"";?> >
                                    <?=$ htmlspecialchars($key) ?>
                                </option>
```

```
<?php
    }
    ?>
</select>
<span class="error">*<?php echo
$location_id_nameErr;?> </span>
</td>
</tr>

<tr>
    <th align="left">Location</th>
    <td align="left"><input type="text" id="location"
name="location" value="<?php echo $location_id_name;?>">
    </td>
</tr></td>

<tr>
    <th align="left">Address</th>
    <td>
        <textarea name="address" rows="5" cols="25"
maxlength="100"><?php echo $address;?></textarea>
    </td>
</tr>
<tr>
    <th align="left">Lease Start Date</th>
    <td align="left"><input type="text" id="leasedate"
name="leasedate" value="<?php echo $leasedate;?>">
    </td>
</tr>

<tr>
    <th align="left">Lease Period</th>
    <td align="left"><input type="text" id="leaseperiod"
name="leaseperiod" value="<?php echo $leaseperiod;?>">
    </td>
</tr>

<tr>
    <th align="left">Lease Rent</th>
    <td align="left"><input type="text" id="leaserent"
name="leaserent" value="<?php echo $leaserent;?>">
    </td>
</tr>

<tr>
    <th align="left">Mobile Number</th>
    <td align="left"><input type="text" id="mobilenno"
name="mobilenno" value="<?php echo $mobilenno;?>">
    </td>
</tr>

<tr>
    <th align="left">Pan Card</th>
    <td align="left"><input type="text" id="pancard"
name="pancard" value="<?php echo $pancard;?>">
```

SBI ATM SYSTEM MANAGEMENT

```

        </td>
    </tr>
<tr>
    <th align="left">Aadhaar Card</th>
    <td align="left"><input type="text" id="Aadhaarcad"
name="Aadhaarcad" value="<?php echo $Aadhaarcad;?>">
    </td>
</tr>
<tr>
    <th align="left">Account Number</th>
    <td align="left"><input type="text" id="acno" name="acno"
value="<?php echo $acno;?>">
    </td>
</tr>
</table>
</div>

<div class="buttonswrapper">
    <button class="btn" type="submit" value="Change"
name="ChangeButton">CHANGE</button>
    <button class="btn" type="submit" value="Delete" name="DeleteButton"
onclick="return confirm('Are you sure you want to
DELETE?');">DELETE</button>
</div>
</form>

<?php
    if (isPost()) {
        if ($errorMsg == "" && isset($_POST['ChangeButton']))
            echo_success("Owner details " .
getIdFromIdName($location_id_name) . " changed successfully!");
        else if ($errorMsg == "" && isset($_POST['DeleteButton']))
            echo_success("Owner " . $location_id_name_deleted . " deleted
successfully");
        else
            echo_error($errorMsg);
    }
?>
</body>
</html>
```

5. RESULT ANALYSIS

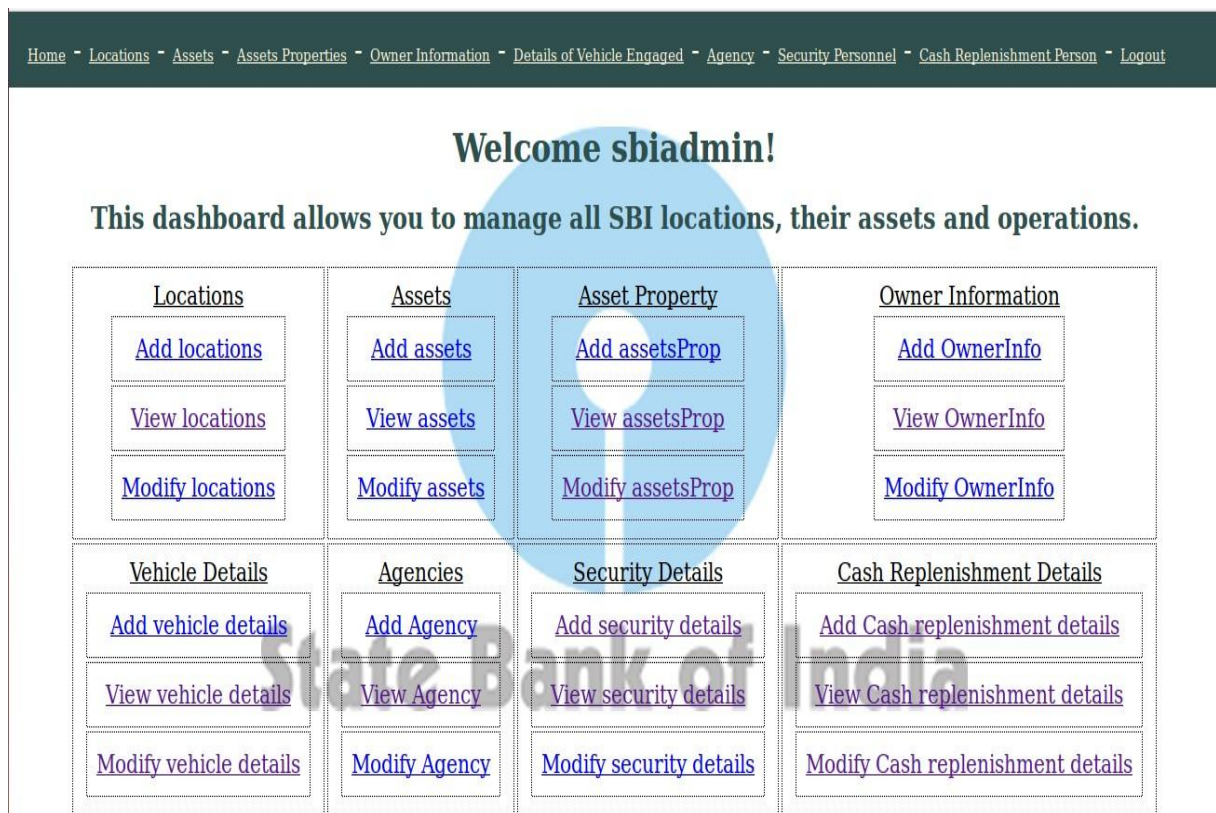
OUTPUT SCREENS

Login Page



Screenshot 5.1. Login Page

Home Page



Screenshot 5.2. Home Page

SBI ATM SYSTEM MANAGEMENT

Add Locations

Add a new SBI ATM location!

Location	<input type="text"/>
Type	<input type="text" value="Urban"/>

ADD

Screenshot 5.3. Add Locations

View Locations

All locations of SBI ATMs!

ID ^	Location	Type
1	Bellary road, Kurnool 518006 A.P	Urban
10	Main branch, Kurnool 518012, A.P	Urban
2	N.R.R colony, kurnool 518002 A.P	Semi-urban
3	Old bus stand, Kurnool 518004 A.P	Semi-urban
4	New bus stand, Kurnool 518006 A.P	Rural
5	Flower market, Kurnool 518003 A.P	Rural
6	Raja vihar junction, Kurnool 518002 A.P	Urban
7	Tandrapadu, Kurnool 518007 A.P	Rural
8	Pasupala, Kurnool 518007, A.P	Urban
9	Alampur check post, Kurnool 518010, A.P	Semi-urban

5.4. View Locations

Modify Locations

Modify any SBI ATM location!

ID	<input type="text" value="0"/> *
Location	<div></div>
Type	<input type="text" value="Urban"/>

Screenshot 6.5. Modify Locations

Add Assets

Add a new asset!

Location	<input type="text" value="1-Bellary road, Kurnool 518006 A.P"/>
Type	<input type="text" value="Select"/>
Make	<input type="text"/>
Installation date	<input type="text"/> *

Screenshot 5.6. Add Assets

SBI ATM SYSTEM MANAGEMENT

View Assets

All assets of SBI!

Asset ID ^	Asset Type	Make	Installation date	Location ID	Location
10	ATM	Wincore	2017-07-01	2	N.R.R colony, kurnool 518002 A.P
11	ATM		2017-06-01	3	Old bus stand, Kurnool 518004 A.P
12	ATM		2017-06-02	5	Flower market, Kurnool 518003 A.P
13	Battery	Exide	2017-06-09	2	N.R.R colony, kurnool 518002 A.P
14	AC	Huawei	2017-04-01	4	New bus stand, Kurnool 518006 A.P
15	ATM	Hitech	2017-07-07	9	Alampur check post, Kurnool 518010, A.P
16	ATM	Wincore	2017-07-12	10	Main branch, Kurnool 518012, A.P
17	Camera	Logitech	2017-07-07	10	Main branch, Kurnool 518012, A.P
18	AC	LG	2017-09-01	10	Main branch, Kurnool 518012, A.P
19	UPS	APC	2018-01-10	9	Alampur check post, Kurnool 518010, A.P

Screenshot 5.7. View Assets

Modify Assets

Modify any SBI Asset!

ID	<input type="text" value="0"/> *
Location	<input type="text" value="0"/>
Type	<input type="text" value="Select"/>
Make	<input type="text"/>
Installation date	<input type="text"/> *

CHANGE

DELETE

Screenshot 5.8. Modify Assets

SBI ATM SYSTEM MANAGEMENT

Add Asset Properties

Add a new asset property!

Asset select

Property

Property details

ADD

Screenshot 5.9. Add Asset Properties

View Asset Properties

SBI ATM Asset Property Details

Property Id ^	Asset Id	Type	Property	Property details
1	6	UPS	Capacity	20000mah
10	18	AC	AMC	maintained by voltas service m
11	5	Stabilizer	AMC	12/05/1996
3	14	AC	AMC	28/03/2017
4	20	Battery	Capacity	10000mah
5	8	Camera	Type	cc
7	3	Battery	Capacity	2000mah
8	14	AC	Type	split
9	12	ATM	MachineReplacementDe	Replaced on 25th jan 2015

Screenshot 5.10. View Asset Properties

Modify Asset Properties

Modify SBI ATM Asset Properties!

Asset Property Id	<input type="text" value="select"/> *
Asset	<input type="text"/>
Property_details	<input type="text"/>
Property	<input type="text"/>

Screenshot 5.11. Modify Asset Properties

Add Owner Information

Add a new owner information

Location	<input type="text" value="1-Bellary road, Kurnool 518006 A.P"/>
Name	<input type="text"/> *
Address	<input type="text"/>
Lease Start Date	<input type="text"/>
Lease Period	<input type="text"/>
Lease Rent	<input type="text"/>
Mobile Number	<input type="text"/>
Pan Card	<input type="text"/>

Screenshot 5.12. Add Owner Information

SBI ATM SYSTEM MANAGEMENT

View Owner Information

SBI ATM Location Owner Details									
Owner Id ^	Name	Address	Lease Start Date	Lease period	Lease rent	Mobile Number	Pan card	Aadhaar card	Account
13	Rajeev	87/2, kurnool	2018-03-11	5	4	8309724634	FJB262J	849029798625	8910
14	Ramu	90/4 Yammiganuru	2016-10-10	3	42000	9342444484	BVHJ91B	890123416271	11071
15	Vikram	9/3, Kurnool	2017-07-17	3	54000	8912087192	KIUIJ123H	712910234152	6710
16	Ravi	90/3, Pasupula	2016-04-12	6	20000	7810283612	JKLA12G	789120123415	1839

Screenshot 5.13. View Owner Information

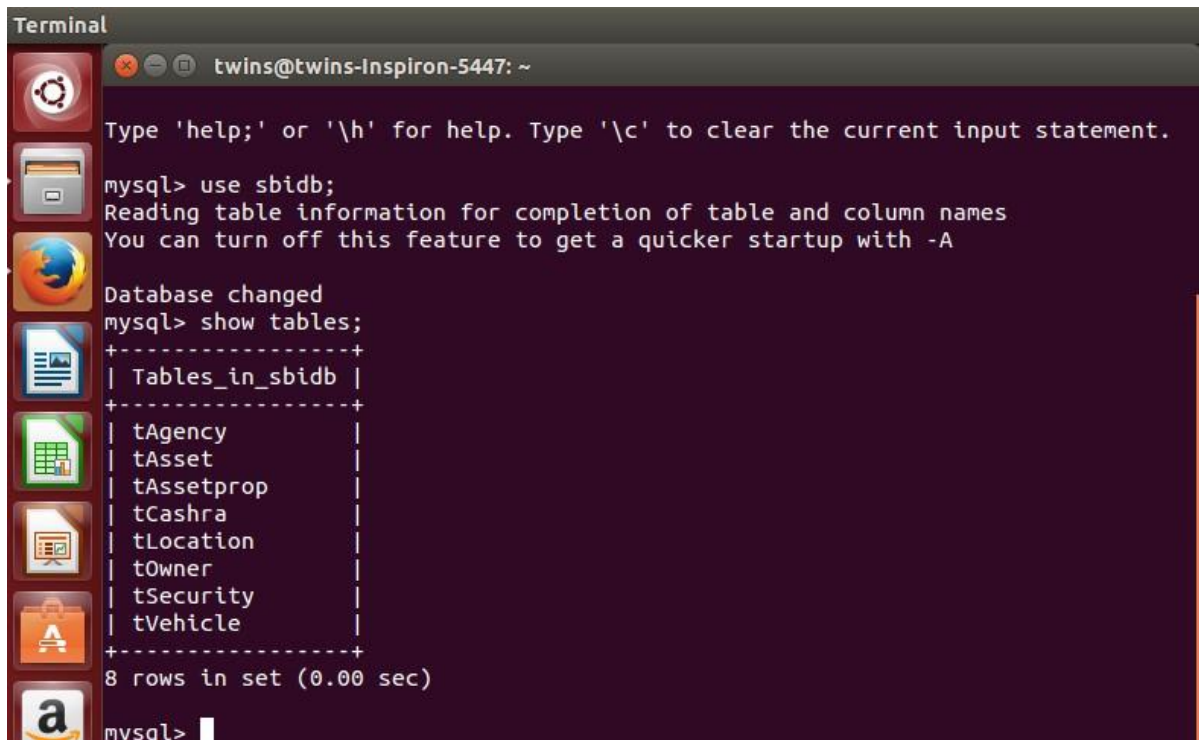
Modify Owner Information

Modify SBI ATM Location Owner Information!

Name	<input type="text" value="select"/> *
Location	<input type="text"/>
Address	<input type="text"/>
Lease Start Date	<input type="text"/>
Lease Period	<input type="text"/>
Lease Rent	<input type="text"/>
Mobile Number	<input type="text"/>
Pan Card	<input type="text"/>

Screenshot 5.14. Modify Owner Information

Tables



The screenshot shows a terminal window titled "Terminal" with the user "twins@twins-Inspiron-5447: ~". The terminal displays the following text:

```
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
mysql> use sbidb;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_sbidb |
+-----+
| tAgency        |
| tAsset          |
| tAssetprop      |
| tCashra         |
| tLocation       |
| tOwner          |
| tSecurity       |
| tVehicle        |
+-----+
8 rows in set (0.00 sec)

mysql>
```

Screenshot 5.15. Tables

6. CONCLUSION

We used Linux as Operating System. PHP, Java Scripts and some CSS styles in the frontend. Apache2 Web Server as backend and MySQL to store databases. Using all these the project have been developed according to the specifications given. By these the details of ATM is available that is the location where ATM is present, details of assets present in the ATM and its properties. Also Owner details from whom the ATM land has been taken for lease is available. Agency details is also present that is which agency is supplying security personnel, vehicles and drivers. Security personnel details, Vehicle details and Driver details is also available. By these all the details of ATM is available and it is easy for bank to maintain all the details.

REFERENCES

1. "PHP My Admin 4.7.5 is released". PHP MY ADMIN.net. Retrieved 2017-10-24.
2. "Translations". PHP My Admin. Retrieved 2014-12-23.
3. http://www.pcworld.com/article/233948/PHP_MYADMIN.html.
4. "PHP MYADMIN-About". PHP MY ADMIN. Retrieved 2013-03-03.
5. Delisle, Marc (2010). Mastering PHP MY ADMIN 3.3.x for Effective MySQL Management Packt Publishing.p.359.ISBN 978-1-84951-354-8. Operating systems.
6. C.J.Date.An Introduction to Database Systems (Addison-Wesley).
7. Codd. E.F. The Relational Model for Database Management Version 2.