VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Machhe, Belagavi, Karnataka 590018

DBMS LABORATORY WITH MINI-PROJECT REPORT

On

"ATM DATABASE MANAGEMENT SYSTEM"

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

Bachelor of Engineering
in
Information Science & Engineering
by

ANSH PANDEY	**1BG20IS008**
YASH SINHA	**1BG20IS064**
BHARGAV JK	**1BG21IS401**

Under the guidance of
Mrs. SWATHI SRIDHARAN
Assistant Professor
Department of ISE, BNMIT

Vidyayāmruthamashnuthe

B.N.M. Institute of Technology

An Autonomous Institution under VTU, Approved by

AICTE Department of Information Science and Engineering

2022 – 2023

B.N.M. Institute of Technology

An Autonomous Institution under VTU

DEPARTMENT OF INFORMATION SCIENCE & ENGINEERING

Vidyayāmruthamashnuthe

CERTIFICATE

Certified that the Mini-project entitled **ATM DATABASE MANAGEMENT SYSTEM** is carried out by **Ansh Pandey(1BG20IS008), Yash Sinha(1BG20IS064) and Bhargav JK(1BG21IS401)** the bonafide student of **B.N.M Institute of Technology** in partial fulfillment for the award of **Bachelor of Engineering** in **Information Science & Engineering** of the **Visvesvaraya Technological University**, Belagavi during the year 2022-2023. It is certified that all corrections / suggestions indicated for Internal Assessment have been incorporated in the report deposited in the department library. The mini-project report has been approved as it satisfies the academic requirements in respect of mini-project prescribed for the said Degree.

Mrs. Swathi Sridharan
Assistant Professor,
Dept. of ISE
BNMIT

Dr. S. Srividhya
Professor & Head,
Dept. of ISE
BNMIT

Name & Signature of the Examiners with date:

1.

2.

ABSTRACT

This project is basically creating a database management system for the transaction that are being taken place in the ATM of any bank. Aim of the project is to creating a backend program of the processes taking place using SQL. This model deals with the relationship between the customer, ATM machine & Bank. The other entities play a supportive weak role. So basically this helps the banks to debit or credit the amount of the customer from his respective account and also keep the record of his/her transaction made.

The ATM System is the project which is used to access their bank accounts in order to make cash withdrawals. Whenever user want to make withdrawals, they can enter their ATM card and verified PIN, then user select the withdrawal option and enter the withdrawal amounts and it will display the amount to be withdrawn in the form of50's,100's and 500's. The user also able to perform one or more transactions. Security is the foundation of good ATM system. This system will provide for secure connections between users and the bank servers. The whole process will be automated right from PIN validation to transaction completion. The card details and PIN database will be a secure module that will not be open to routine maintenance, the only possibility of access to this database will be through queries(questions) raised from an ATM in the presence of a valid bank ATM card. ATM Simulation System will enable two important features of an ATM, reduction of human error in the banking system and the possibility of24 hour personal banking.

Automated Teller Machine (ATM) is an electronic telecommunications device, which enables customers to perform banking without the need for direct interaction with bank staff. For this, every account holder must have a unique id card for the individual account having a unique pin. On the absence of this card, whatever be the adverse situation the use of this ATM service is not permitted. So, an Internet Of Things and Computer Vision based Smart ATM service is being proposed here, using Raspberrypi microcontroller based embedded system, where each person will be their own identity, where Fingerprint, Face, OTP verifications are key features for security, which in turn reduces the issue of fraud transactions, fraud ATM cards, hence security issue gets resolved.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

An Automated Teller Machine (ATM) is a safety-critical and real-time system that is highly complicated in design and implementation.

1.1 Introduction to Web Development

Web Development is the work involved in developing a web site for the internet. Web development can range from developing a simple single static page of plain text to complex web-based internet applications, electronic business, and social network services. For larger organization and businesses, web development teams can consist of hundreds of people and follow standard methods like agile methodologies while developing web sites. An ever-growing set of tools and technologies have helped developers build more dynamic and interactive web sites. Further web developers now help to deliver applications as web services which were traditionally only available as applications on desk-based computer. This has allowed for many opportunities to decentralize information and media distribution. In web development we use HTML (Hyper Text Markup Language), CSS (Cascading Style Sheets), PHP(Hypertext Preprocessor), JavaScript ..Etc.

1.2 Brief Outline of Project

The ATM/CDM System is a simulated working of an **Automated teller machine/Cash deposit machine**. The front end tool used to simulate this software database is My SQL. When a user enters his account number and his PIN he will be able to log into his Account. This is achieved by sending the login information to the database and verifying whether the credentials are matching the database records. If it matches the records then the user is able to login to his account to perform any transaction. Else he'll be required to enter his credentials properly again.

After he has access to his account he can choose between 5 options which are:

1. Withdraw cash.
2. Deposit cash.
3. Enquire balance.
4. Generate a Mini Statement.
5. Change PIN Code.

Any of the option can be chosen choose based on his requirement. If he chooses Withdraw cash option then he will be redirected to a page where he can select if he wants to Withdraw from his Savings account or a Current account. Then he can enter the amount he wishes to withdraw and if the balance is sufficient in his account the transaction will be successful, else he will be redirected back to the login page with an alert that he has insufficient funds in his account. The amount entered should be in multiples of 100s only. The amount which is withdrawn will be updated in the database of the account holder. A transaction ID will be generated and presented to the user, which he can make note of. This is the same case of operation if he chooses the 2nd option which is the Deposit operation in which case he can deposit money into his account.

When the 3rd option that is, Enquire Balance is chosen, the user is presented with a page to choose between his savings account and his current account and that account's balance is displayed on a new page. This balance will be the current balance pulled from the database.

When the 4th option that is, Mini Statement is chosen, the most recent 8 transactions of that account holder will be presented in the form of a table. The tuples of the table are Date, Time, Transaction number, Account Number, Account Type, Amount, Balance and Transaction Type. All of this tuple details will be pulled from the database and presented to the user.

The final 5th option is Change PIN option in which the user can change his account PIN Code by entering his Old PIN code, New PIN code and verifying his new PIN code. This information is sent to the database and the PIN code change is reflected in the database as well.

This a Brief outline of how the project operates.

1.3 Project Goal

The aim of this project is to speed up the transactions done by customers in a neat and easy to understand manner without too much complexities.

1.4 Scope

- It can be implemented in ATM machine by owner of bank or in charge of branch
- It is easy to learn the task.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 Hardware Requirements

There are various hardware components with which the machine is required to interact. Various hardware interface requirements that need to be fulfilled for successful functioning of the software are as follows:

Processor	: intel core i3
Hard Disk	: 20 GB, 80 GB, 160 GB or above
Monitor	: 15 VGA colour, 1024*768 resolution
RAM	: 4GB or above
Input Device	: Keyboard and Mouse

2.2 Software Requirements

In order to perform various different functions, this software needs to interact with various other software. So there are certain software interface requirements that need to be fulfilled which are listed as follows:-

Operating System	: Windows 10
Tool used	: Brackets and XAMPP
Front end used	: HTML/CSS/PHP
Back end used	: SQL

CHAPTER 3

SYSTEM DESIGN

3.1 E R Diagram

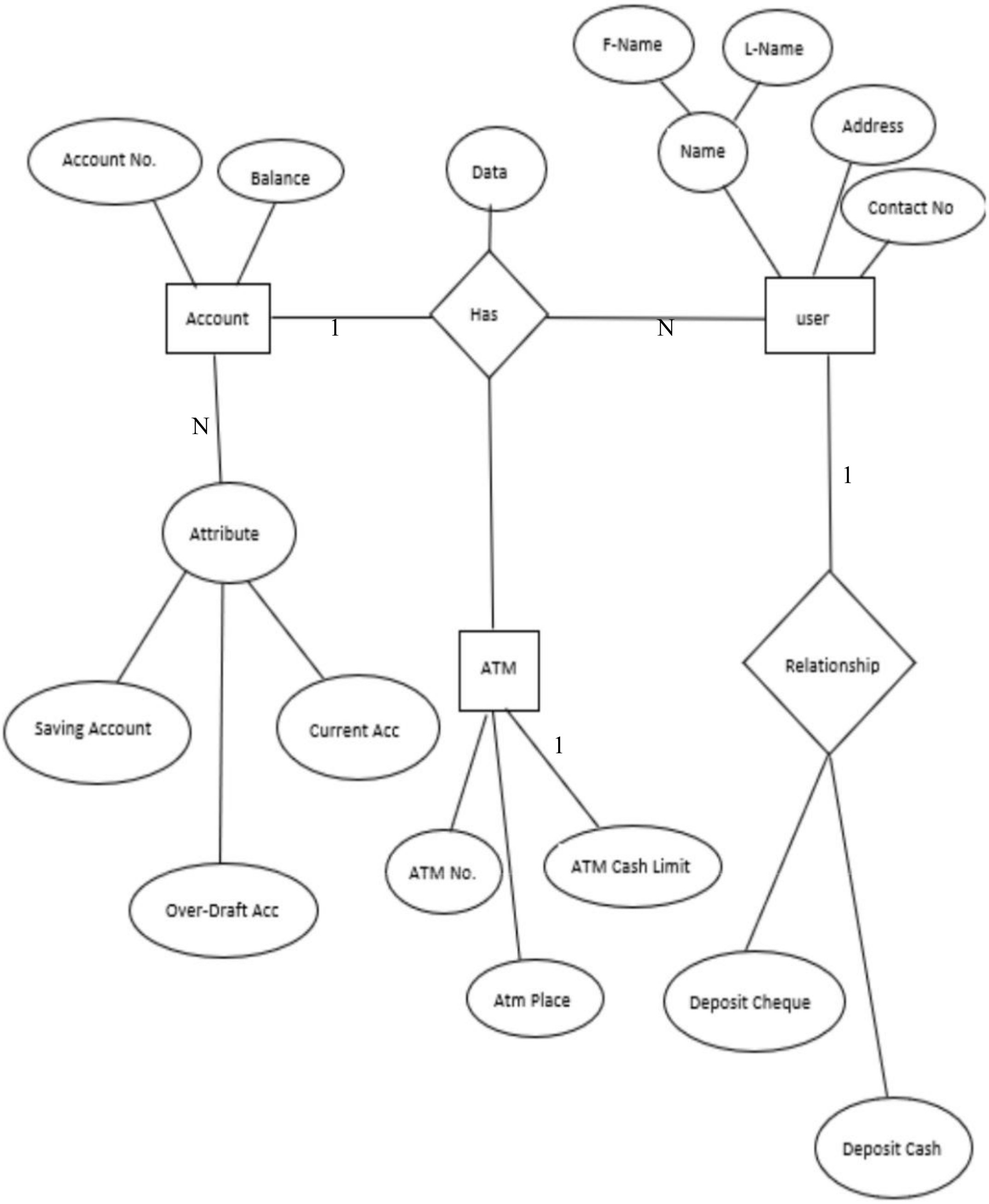


Fig:3.1 ER Diagram of ATM Management System

3.2 SCHEMA DIAGRAM:

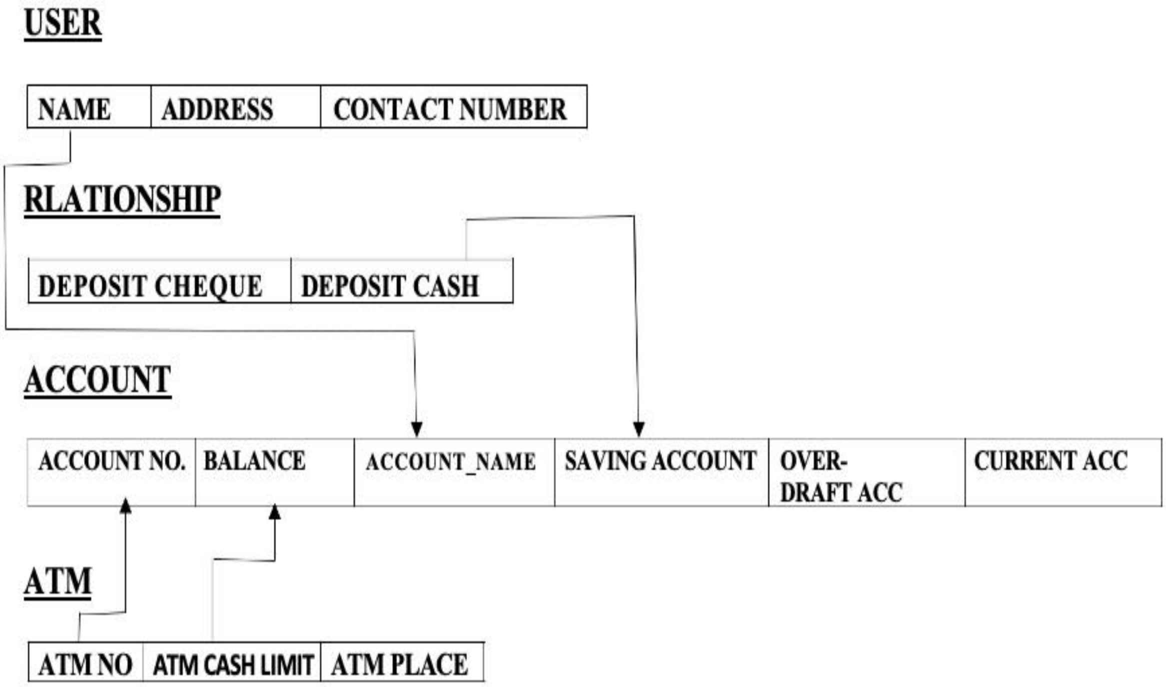


Fig: 3.2 Schema Diagram of ATM Management System

3.1 User Interface Design

The user interface used in any software must be user friendly. So, we have made PHP/HTML interface as our Front-end tool. PHP is being widely used in this era of web development. It is simple, user-friendly and HTML can be embedded within it.

To access PHP files we needed a server which is why we have used XAMPP server software which consists of Apache Server. By using this software we were able to access all our PHP pages and work on it. We used Sublime Text 3 software to code our webpages. This open-source tool can be used to code about 200 programming and scripting languages together. Finally with the help of XAMPP server software we were able to create a Database in Microsoft’s MySQL database. All the data required to run this system is stored in that database. It is highly convenient for developers and designers alike to connect MySQL database data and PHP webpages using simple SQL command queries. Any user can use our web interface and easily be able to sign-up, login to his account, make any desired transaction and exit the system with a Transaction ID (As long as its successful)

CHAPTER 4

TESTING

Testing is a dynamic technique of verification and validation. It involves executing an implementation of the software with test data and examining the outputs of the software and its operational behaviour to check that it is performing as required.

The following statements serve as the objectives for testing:

1. Testing is a process of executing a program with the intent of finding error
2. A good test case is one that has a high probability of finding an as-yet undiscovered error.
3. A successful test is one that uncovers as-yet undiscovered error.

Verification and validation is intended to show that a system confirms to its specification and that the system meets the expectations of the customer. Verification involves checking that the software confirms to its specification. We should check that the system meets its specified functional and non-functional requirements. Validation ensures that the software meets the expectations of the customer. It goes beyond checking conformance of the system to its specification to showing that the software does what the customer expects as distinct from what has been specified.

The testing process should proceed in stages where testing is carried out incrementally in conjunction with system implementation. Figure 4.1 shows a five stage testing process where system components are tested, the integrated system is tested and, finally, the system is tested with the customer's data.

The stages in the testing process are:

4.1 Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. Thus the project has been successfully tested.

4.2 Integration Testing

Main function is design to call many sub functions, where different options are given in the sub functions. Now, different functions are included in the main separately and tested for error. Compile and tested the project without any error’s get the output as we expected. Thus the project has been successfully tested.

4.3 Sub-system Testing

This phase involves testing collections of modules which have been integrated into sub-systems. The sub-system test process should concentrate on the detection of module interface errors by rigorously exercising these interfaces. Thus the project has been successfully tested.

4.4 System Testing

The sub-systems are integrated to make up the system. This process is concerned with finding errors that result from unanticipated interactions between sub-systems and sub-system interface problems. It is also concerned with validating that the system meets its functional and non-functional requirements and testing the emergent system properties. Thus the project has been successfully tested.

4.5 Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system customer rather than simulated test data. Thus the project has been successfully tested.

CHAPTER 5

IMPLEMENTATION

```
SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";

SET AUTOCOMMIT = 0;

START TRANSACTION;

SET time_zone = "+00:00";



CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_current_dep` (IN `tran_id` INT(7), IN `acc_no`
VARCHAR(20), IN `t_amt` INT(7), IN `balance` INT(7)) NO SQL

insert INTO transaction values(CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,tran_id
,acc_no,'current',t_amt,balance,'credit')$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_current_with` (IN `tran_id` INT(7), IN
`acc_no` VARCHAR(20), IN `t_amt` INT(7), IN `balance` INT(7)) NO SQL

insert INTO transaction
values(CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,tran_id,acc_no,'current',t_amt,balance,'debit')
$$
CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_saving_dep` (IN `tran_id` INT(7), IN `acc_no`
VARCHAR(20), IN `t_amt` INT(7), IN `balance` INT(7)) NO SQL

insert INTO transaction values(CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,tran_id
,acc_no,'saving',t_amt,balance,'credit')$$

CREATE DEFINER=`root`@`localhost` PROCEDURE `trans_saving_with` (IN `tran_id` INT(7), IN `acc_no`
VARCHAR(20), IN `t_amt` INT(7), IN `balance` INT(7)) NO SQL

insert INTO transaction values(CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,tran_id
,acc_no,'saving',t_amt,balance,'debit')$$

,acc_no,'current',t_amt,balance,'debit')$$
```

```
-- Table structure for table `account_details`


CREATE TABLE `account_details` (`c_id` int(11) NOT NULL,`acc_no` varchar(20) NOT
NULL,`atm_pin` int(11) DEFAULT NULL) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `account_details`

INSERT INTO    `account_details`     (`c_id`, `acc_no`,      `atm_pin`)

VALUES

(1, '1234567890', 2222),

(2, '2345678901', 2018),

(3, '3456789012', 2016),

(4, '4567890123', 2014);

-- Table structure for table `atm`

CREATE TABLE `atm` (`a_id` int(11) NOT NULL,`a_location` varchar(20) DEFAULT NULL)
ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `atm`


INSERT INTO `atm` (`a_id`, `a_location`) VALUES (1, 'Kammanahalli');

-- Table structure for table `banks`

CREATE TABLE `banks` (`b_id` int(11) NOT NULL,`b_name` varchar(20) DEFAULT
NULL, `b_location` varchar(20) DEFAULT NULL) ENGINE=InnoDB DEFAULT
CHARSET=latin1;




-- Dumping data for table `banks`
```

```
INSERT INTO `banks` (`b_id`, `b_name`, `b_location`) VALUES (1, 'State Bank of India',
'Kammanahalli'), (2, 'State Bank of India', 'Hebbal'),

(3, 'State Bank of India', 'New Bel'),

(4, 'State Bank of India', 'R.T Nagar'),

(5, 'State Bank of India', 'Malleshwaram'),

(6, 'State Bank of India', 'Indiranagar'),

(7, 'State Bank of India', 'M.G Road'),

(8, 'State Bank of India', 'Cunningham Road'),

(9, 'State Bank of India', 'Sahakarnagar'),

(10, 'State Bank of India', 'J.P Nagar');


-- Table structure for table `current_account`

CREATE TABLE `current_account` (

`c_id` int(11) NOT NULL,

`current_balance` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;

-- Dumping data for table `current_account`

INSERT INTO   `current_account`    (`c_id`, `current_balance`)

VALUES

(1, 245900),

(2, 2000),

(3, 3000),

(4, 4000);
```

```
-- Table structure for table `customer`

CREATE TABLE `customer` (`c_id` int(11) NOT NULL,`b_id` int(11) NOT NULL,`c_fname`
varchar(20) DEFAULT NULL, `c_lname` varchar(20) DEFAULT NULL, `c_address` varchar(30)
DEFAULT NULL, `c_mob` varchar(20) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;


-- Dumping data for table `customer`

INSERT INTO `customer` (`c_id`, `b_id`, `c_fname`, `c_lname`, `c_address`, `c_mob`) VALUES

(1, 1, 'Rishil', ' Rajan', 'Kammanahalli', '7019209690'),

(2, 1, 'Sayanth', 'V', 'M G Road', '8660591081'),

(3, 1, 'Shramith', 'Kumar', 'Murgesh Palya', '8073878189'),

(4, 2, 'Nitya', 'Prasad', 'Bidarahalli', '8861776693');


-- Table structure for table `saving_account`


CREATE TABLE `saving_account` (

`c_id` int(11) NOT NULL,

`saving_balance` int(11) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;



-- Dumping data for table `saving_account`
```

```
INSERT INTO `saving_account` (`c_id`, `saving_balance`) VALUES

(1, 36000),

(2, 1000),

(3, 3000),

(4, 4000);



-- Table structure for table `transaction`



CREATE TABLE `transaction` (

`date` date NOT NULL,

`time` time NOT NULL,

`t_id` int(11) NOT NULL,

`acc_no` varchar(20) NOT NULL,

`account_type` varchar(20) NOT NULL,

`t_amount` int(11) DEFAULT NULL,

`t_balance` int(11) NOT NULL,

`t_type` varchar(10) DEFAULT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;


-- Dumping data for table `transaction`
```

```
INSERT INTO `transaction` (`date`, `time`, `t_id`, `acc_no`, `account_type`, `t_amount`,
`t_balance`, `t_type`) VALUES ('2018-11-27', '18:50:39', 1053, '1234567890', 'current', 123000,
247600, 'debit'),

('2018-11-28', '08:25:33', 1054, '1234567890', 'current', 12300, 259900, 'debit'),

('2018-11-28', '08:25:43', 1055, '1234567890', 'current', 10000, 249900, 'debit'),

('2018-12-07', '16:13:05', 1056, '1234567890', 'current', 1000, 250900, 'debit'),

('2018-12-07', '16:22:47', 1057, '1234567890', 'current', 5000, 245900, 'debit'),

('2018-12-11', '16:30:26', 1058, '1234567890', 'saving', 100000, 37000, 'debit'),

('2019-08-07', '14:25:08', 1059, '1234567890', 'saving', 1000, 36000, 'debit');


-- Table structure for table `transaction_id`


CREATE TABLE `transaction_id` (

  `t_id` int(11) NOT NULL

) ENGINE=InnoDB DEFAULT CHARSET=latin1;
-- Dumping data for table `transaction_id`


INSERT INTO `transaction_id` (`t_id`)
VALUES (1059);
--        Indexes for dumped tables

--        Indexes for table `account_details`

ALTER TABLE `account_details`

ADD PRIMARY KEY (`acc_no`,`c_id`)

-- Indexes for table `atm`
```

```
ALTER TABLE `atm`

ADD PRIMARY KEY (`a_id`)

-- Indexes for table `banks`

ALTER TABLE `banks`

ADD PRIMARY KEY (`b_id`);

 -- Indexes for table `current_account`

ALTER TABLE `current_account`
ADD PRIMARY KEY (`c_id`);

-- Indexes for table `customer`

ALTER TABLE `customer`
ADD PRIMARY KEY (`c_id`,`b_id`);
-- Indexes for table `saving_account`

ALTER TABLE `saving_account`
ADD PRIMARY KEY (`c_id`);


-- Indexes for table `transaction`

ALTER TABLE `transaction`
ADD PRIMARY KEY (`t_id`);

-- Indexes for table `transaction_id`
ALTER TABLE `transaction_id`
ADD PRIMARY KEY (`t_id`);
COMMIT;
```

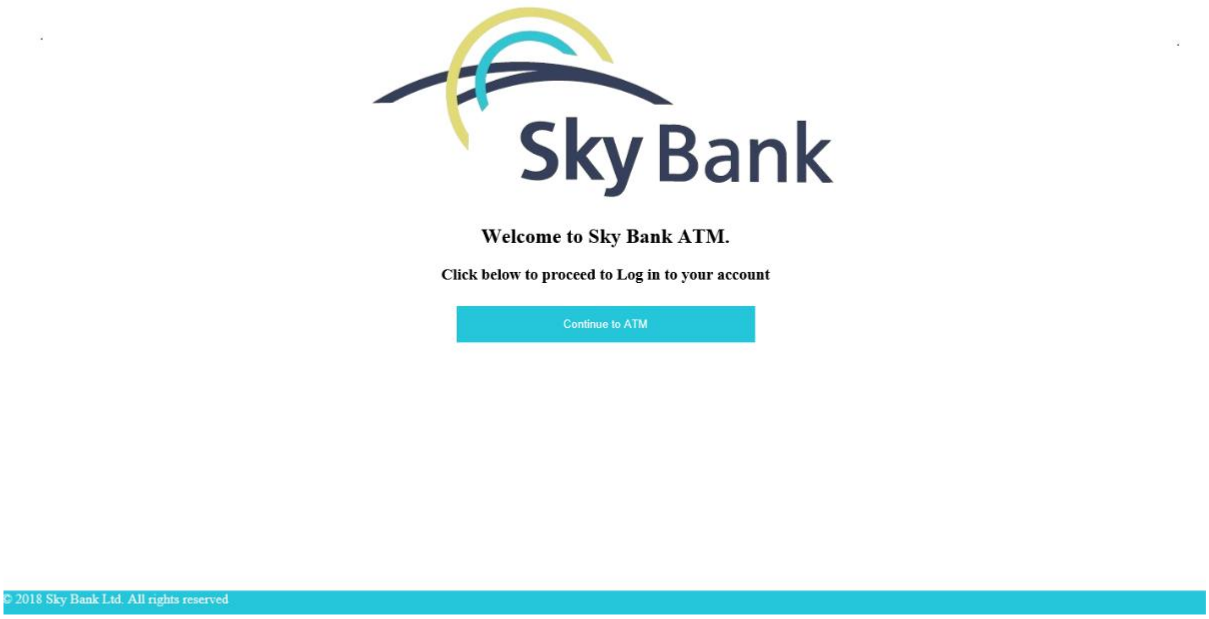
TRIGGERS:

```
CREATE TRIGGER transaction
AFTER
UPDATE
ON saving_account
FOR EACH ROW
DECLARE
prev_bal integer;
new_bal integer;
diff integer;
cid integer;
acc_no varchar(20);
t_id integer;
```

```
BEGIN

prev_bal:= OLD.saving_balance;

new_bal:= NEW.saving_balance;
cid:=NEW.c_id;

acc_no:=SELECT acc_no from account_details WHERE c_id=cid;

IF new_bal>prev_bal THEN

diff:=new_bal-prev_bal;

insert into transaction values(CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,t_id,"acc_no","Savings",diff,new_bal,"Credit");

t_id=t_id+1;

ELSE

diff:=prev_bal-new_bal;

insert into transaction values(CURRENT_TIMESTAMP,CURRENT_TIMESTAMP,t_id,"acc_no","Savings",diff,new_bal,"Debit");

t_id=t_id+1;

END;

BEGIN

DECLARE SET @name:="Deep"

DECLARE SET @sex:="male"

DECLARE SET @address:="nowhere"

INSERT into a VALUES(@name,@sex,@address);

END
```

CHAPTER 6

RESULTS

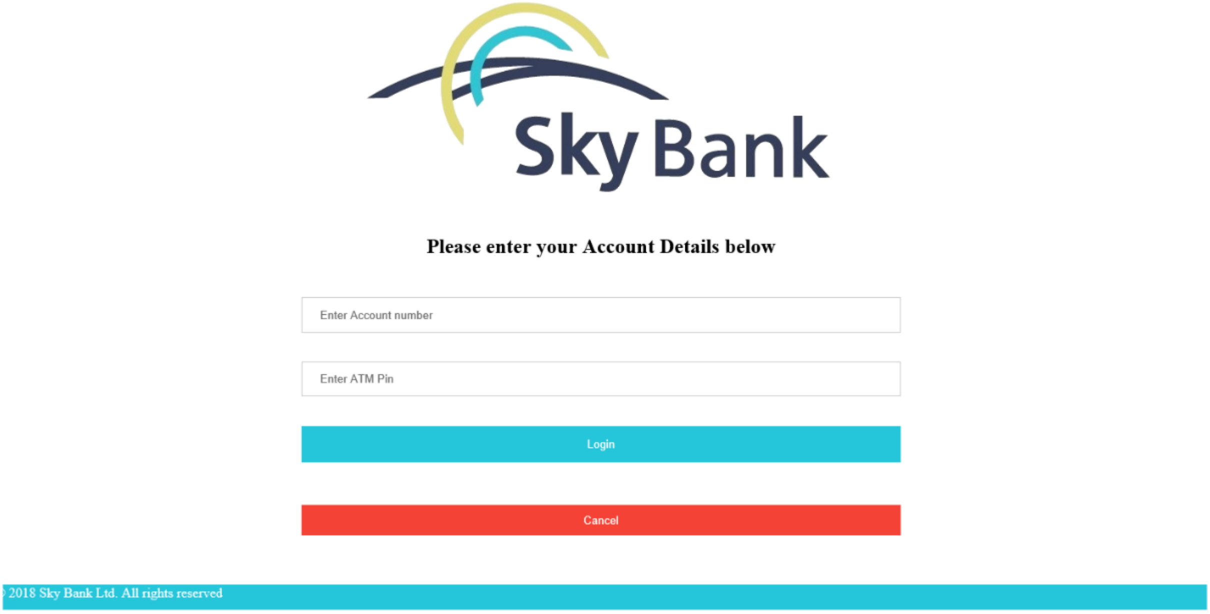


Welcome to Sky Bank ATM.

Click below to proceed to Log in to your account

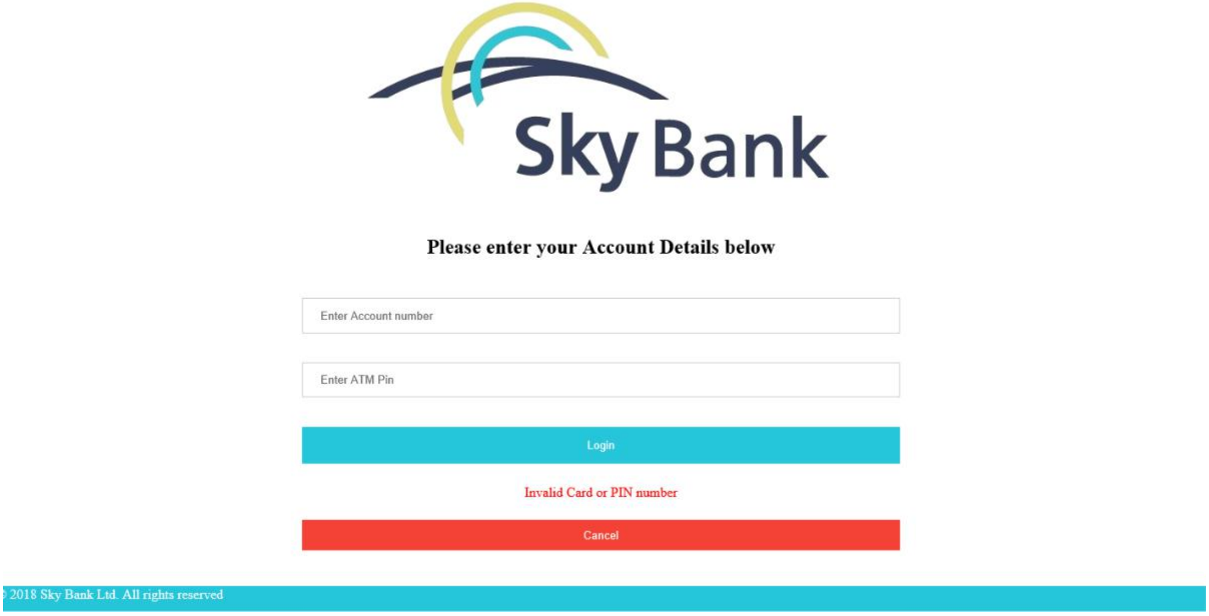
© 2018 Sky Bank Ltd. All rights reserved

Fig. 6.1: Welcome Screen



Sky Bank
Please enter your Account Details below
Enter Account number
Enter ATM Pin
Login
Cancel
© 2018 Sky Bank Ltd. All rights reserved

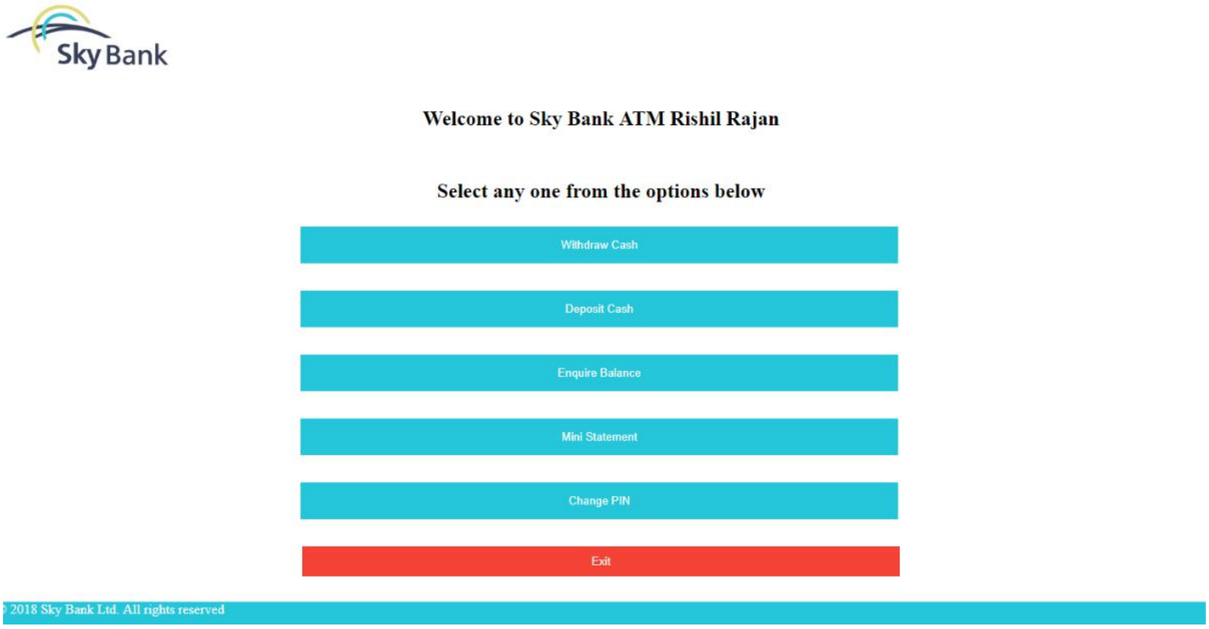
Fig. 6.2: Login Screen



Sky Bank
Please enter your Account Details below
Enter Account number
Enter ATM Pin
Login
Cancel
© 2018 Sky Bank Ltd. All rights reserved

Invalid Card or PIN number

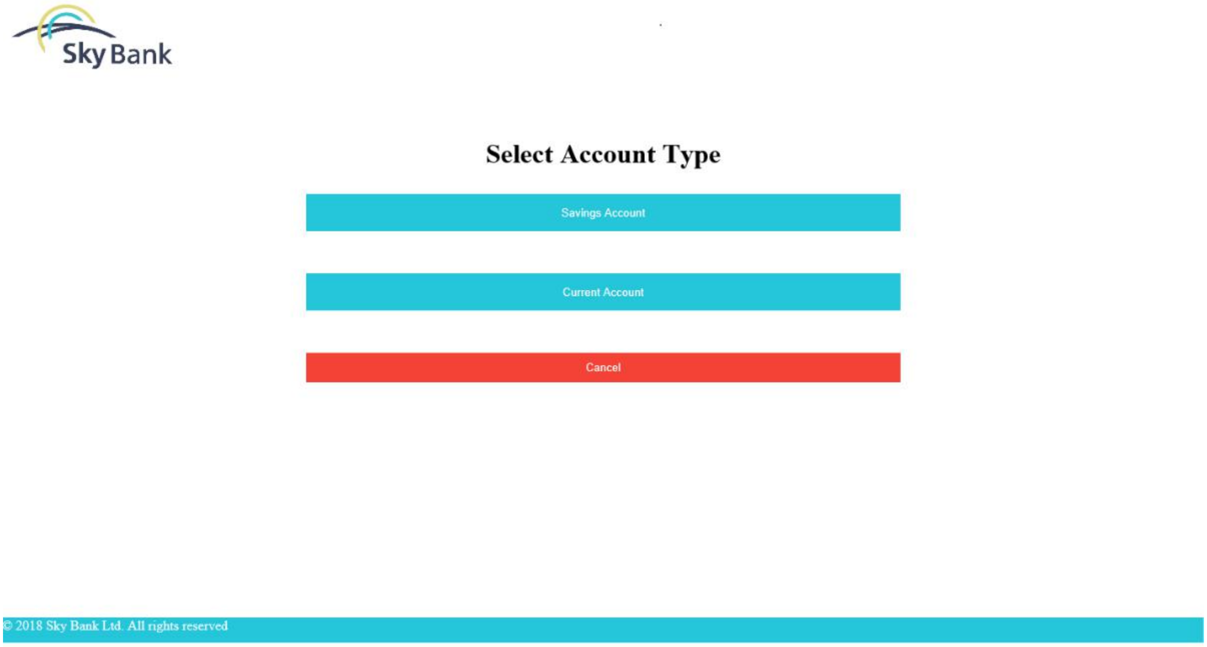
Fig .6.3: Login Failed Screen



Welcome to Sky Bank ATM Rishil Rajan

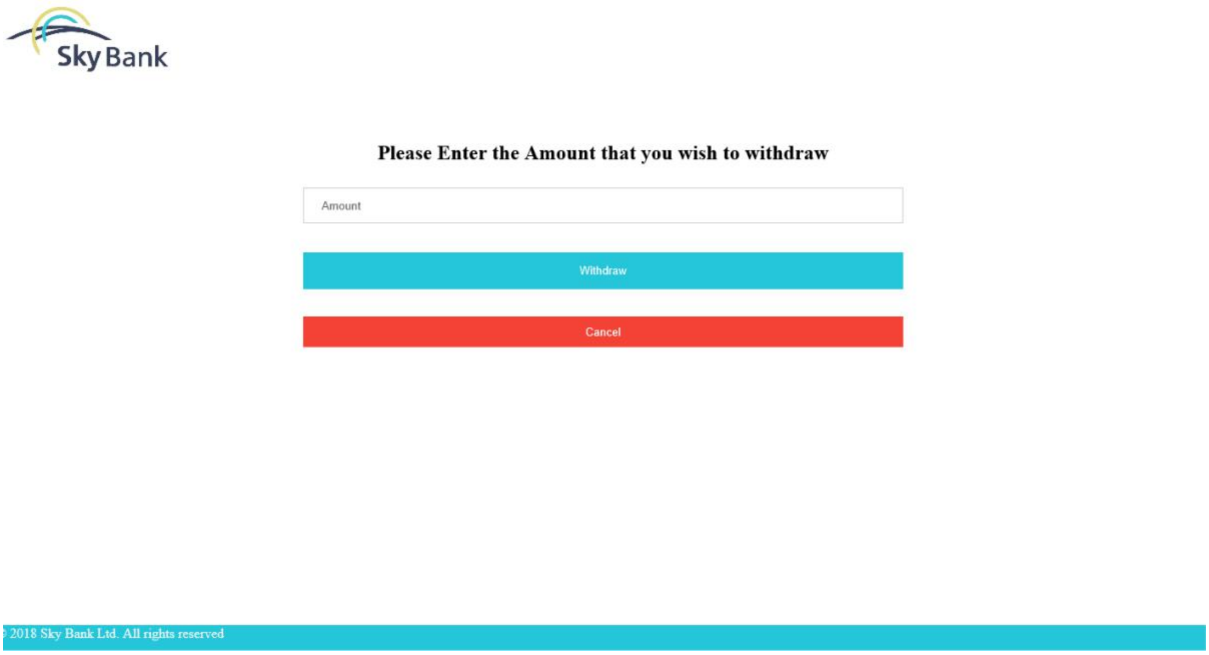
Select any one from the options below

Fig .6.4: ATM Menu Screen



Select Account Type

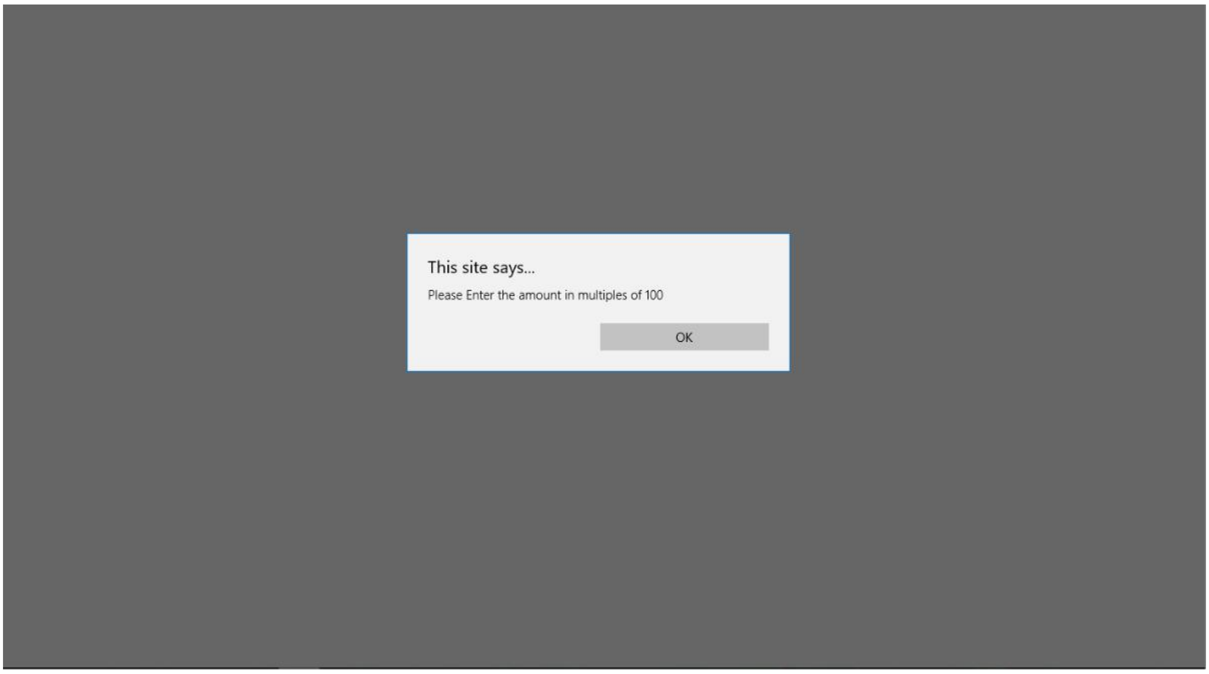
Fig .6.5: Select Account Type Screen



Please Enter the Amount that you wish to withdraw

© 2018 Sky Bank Ltd. All rights reserved

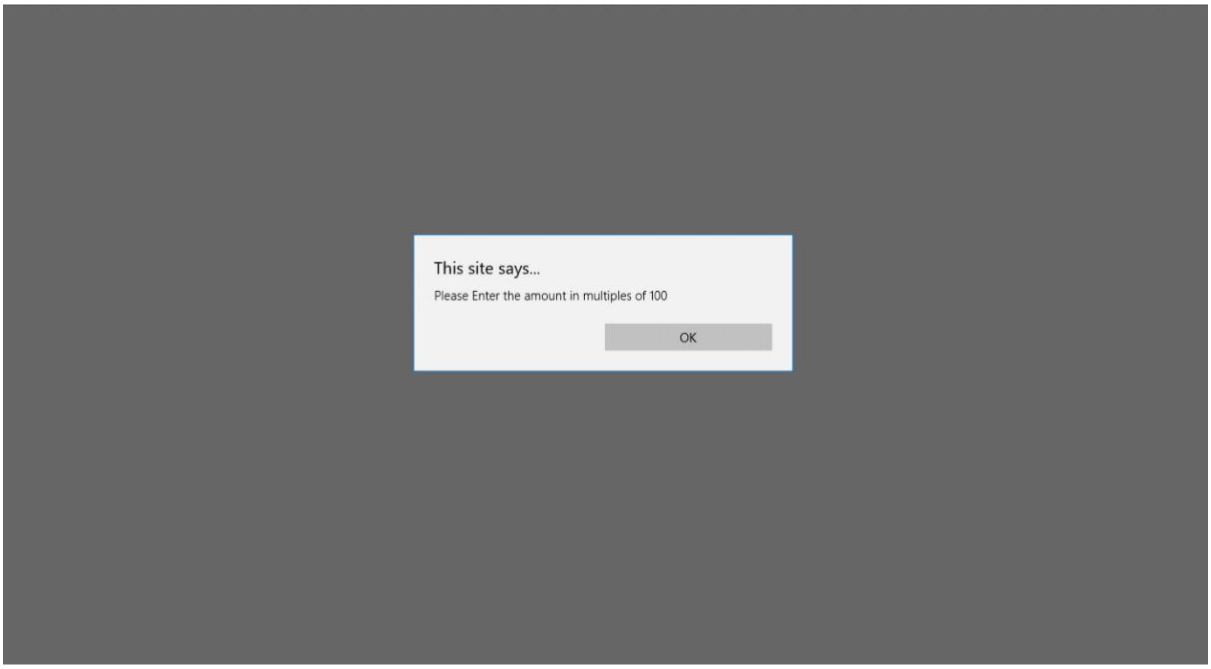
Fig. 6.6: Withdraw Screen



This site says...

Please Enter the amount in multiples of 100

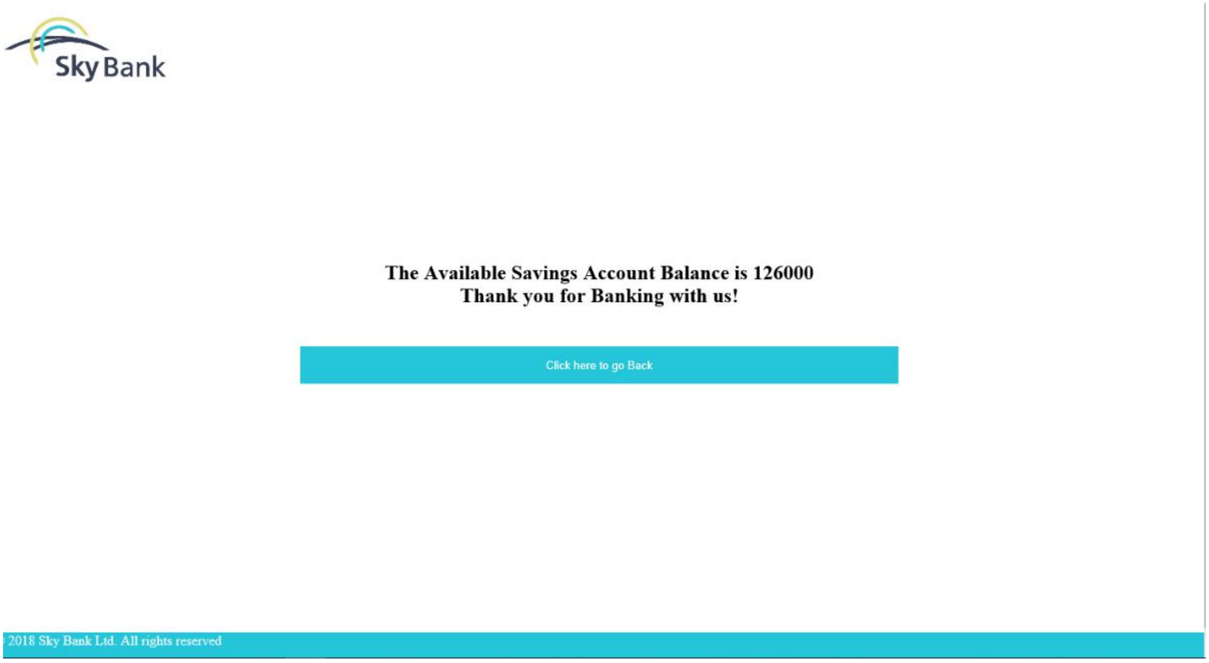
Fig.6.7: Withdraw Multiples Error Alert Screen



This site says...

Please Enter the amount in multiples of 100

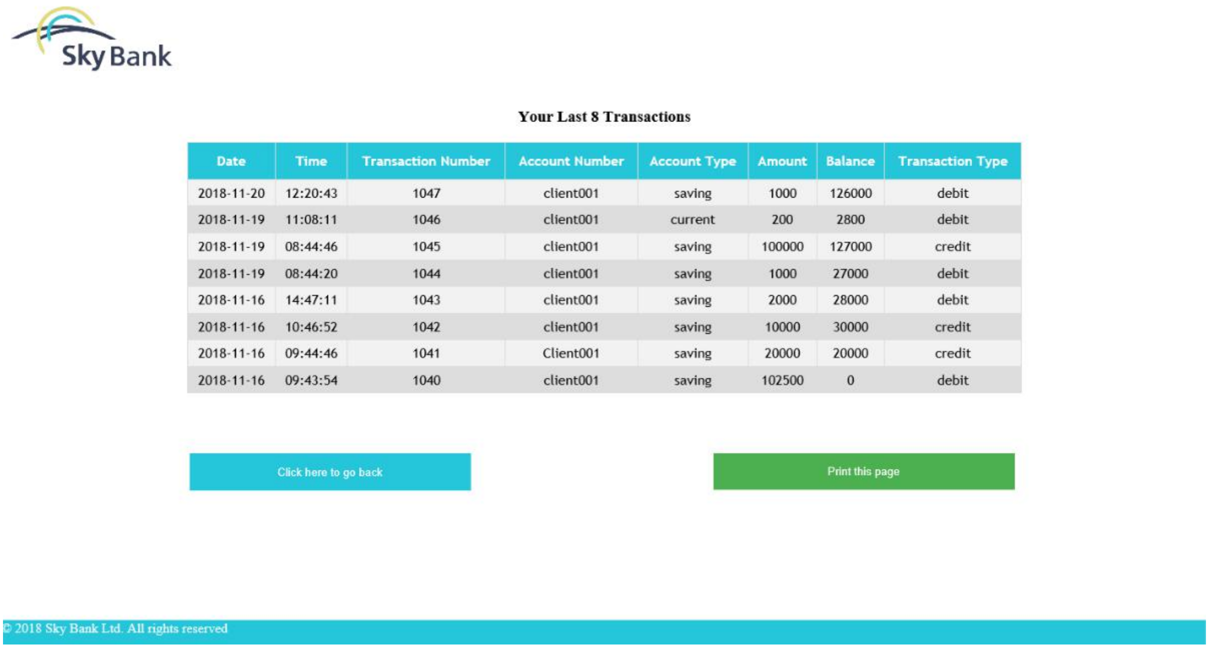
Fig. 6.8: Transaction Successful Screen



The Available Savings Account Balance is 126000
Thank you for Banking with us!

2018 Sky Bank Ltd. All rights reserved

Fig. 6.9: Balance Amount Screen

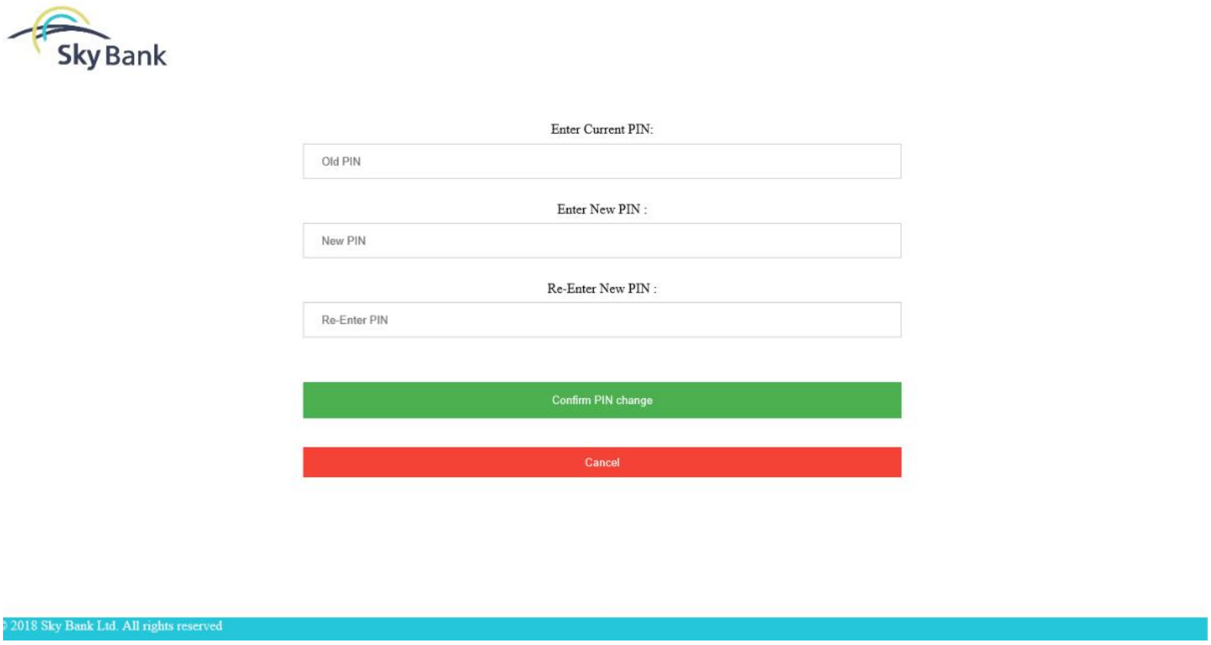


Your Last 8 Transactions

Date	Time	Transaction Number	Account Number	Account Type	Amount	Balance	Transaction Type
2018-11-20	12:20:43	1047	client001	saving	1000	126000	debit
2018-11-19	11:08:11	1046	client001	current	200	2800	debit
2018-11-19	08:44:46	1045	client001	saving	100000	127000	credit
2018-11-19	08:44:20	1044	client001	saving	1000	27000	debit
2018-11-16	14:47:11	1043	client001	saving	2000	28000	debit
2018-11-16	10:46:52	1042	client001	saving	10000	30000	credit
2018-11-16	09:44:46	1041	Client001	saving	20000	20000	credit
2018-11-16	09:43:54	1040	client001	saving	102500	0	debit

© 2018 Sky Bank Ltd. All rights reserved

Fig. 6.10: Mini Statement Screen



Enter Current PIN:

Enter New PIN :

Re-Enter New PIN :

© 2018 Sky Bank Ltd. All rights reserved

Fig. 6.11: Change PIN Screen

CHAPTER 7

CONCLUSION

The development of the mini project has given us a good exposure to HTML, CSS, PHP , JAVASCRIPT and SQL by which we have learnt some of the techniques which help in development of websites and manipulation of databases. Hence it is helpful for us even to take up this field as our career too and develop new websites and databases and improve existing ones and help create a more secure database.

The future developments that can be done to this project is, use of Bio-metric authentication can improve security as card and PIN is not secure enough.

REFERENCES

[1] Ramez Elmasri and Shamkant B.Navathe, ‘Database system model, Language, Design and application programming’, 7th Edition, 2017.

[2] Steven Holzner, ‘The complete Reference PHP’.

[3] Luke Welling, Laura Thomson, “PHP and MYSQL web Development”, 5th Edition Pearson Education, 2016

[4] Robin Nixon, ‘Learning PHP, MYSQL & Javascript with Jquery, CSS and HTML5’, 4th Edition, O’Reilly Publications, 2015

[5] Paul, Ryan (24 October 2005). "A guided tour of the Microsoft Command Shell". Ars Technica. Retrieved 10 April 2011.

[6] "Media Type registration for application/sql". Internet Assigned Numbers Authority. 10 April 2013. Retrieved 10 April 2013.

[7] Shafranovich, Y. (April 2013). "The application/sql Media Type, RFC 6922". Internet Engineering Task Force: 3. doi:10.17487/RFC6922. Retrieved 10 April 2013.

[8] Beaulieu, Alan (April 2009). Mary E Treseler (ed.). Learning SQL (2nd ed.). Sebastopol, CA, USA: O'Reilly. ISBN 978-0-596-52083-0.

[9] Chamberlin, Donald D. (2001-10-03). "Oral history interview with Donald D. Chamberlin". Retrieved 2020-01-14. We changed the original name "SEQUEL" to SQL because we got a letter from somebody's lawyer that said the name "SEQUEL" belonged to them. We shortened it to SQL, for Structured Query Language, and the product was known as SQL/DS.

[10] SQL-92, 4.22 SQL-statements, 4.22.1 Classes of SQL-statements "There are at least five ways of classifying SQL-statements:", 4.22.2, SQL statements classified by function "The following are the main classes of SQL-statements:"; SQL:2003 4.11 SQL-statements, and later revisions.

[11] Chatham, Mark (2012). Structured Query Language By Example - Volume I: Data Query Language. p. 8. ISBN 978-1-29119951-2.