**CHAPTER 1**

**INTRODUCTION**

1. **AIM**

The scope of this project is to simplify the Bill Payment Process for all the Utility Services. The basic idea is to integrate all billers of the user into one single window and then perform the payment on the specified time to those billers. The aim is to design a web application where the users can create an account for them and integrate all the required services.

1. **PROBLEM STATEMENT**

“Online Bill Payments” are evolving to be the most sought option today to do all the Bill Payments. Users need an account in the respective Utility Provider’s site and they have to login once during every cycle of their bill period to check their Outstanding Bill and for the Payment process. In case if the user forgets the pending bill for any service to be paid on time, then it ends up either in late payment charges or termination of that particular service to the user. Adding more difficulty, the users are bound to remember the different login credentials associated with different biller’s site. So, the user is bound to keep track on all the billing dates, amounts and the account details of all the billers despite his work related pressure and schedule.Therefore, to overcome all these hurdles, an alternate application which performs all the above mentioned process by itself with just one single click is developed.This web application turns out handy for automated payment of all the user’s bills from one single window.

1. **DESCRIPTION**

***EASYPAY***

“Easypay” is a Web Application where the users could link all their Utility accounts to this application and make the payments from this application itself. Easypay provides for an e-Wallet facility, so that the users could add cash totheir wallet associated with their account. Users could now Link the services which they want to add to their Easypay account. The details from the respective services which the user has linked are fetched into their Easypay Account itself. Easypay provides for an Automated Payment facility which automatically pays the Bill Amount to the Biller’s Account much before the due date of the Bill, so that the possibility of late payments are avoided. So, the bill payment now comes as simple as a three step process,

* Creating an Easypay account
* Linking all the Utility accounts to Easypay
* Adding cash to Wallet

Users are notified in prior before each transaction being performed and even post payment notifications are sent to the user. For some services where the Biller also owns an “Easypay” account, it is possible to transfer the bill amount directly to the Biller’s “Easypay” account itself.

In built feature called “Insta-Pay” allows the user to pay the bill manually from the application. This comes much handy when the user needs to pay some

bills with priority at times for any specific reasons. This also could be performed as either “Easypay” transfer or Transfer to the Biller’s Account.

Adding funds to the Easypay wallet is very simple. There are two options available to add cash to the wallet,

* + **Online Payment**- User could use his/her Card to transfer funds tothe wallet. The online payment window is powered by the Payment Gateway, which processes the card and performs the transfer.
  + **Transfer through an “Easypay” representative**-The User isrequired to provide his/her details when asked in the application. Once the user submits the details, one of the Easypay representative will get in touch with the User and receives cash from them. Once the representative receives the cash the users Easypay account is updated for the addition of funds.

1. **BENEFITS** 
   * + Regular Payment of Bills within the due date.
     + Single Window Billing and Payment.
     + Avoids late payment charges.
     + Reduces the need of User to keep track on the Bills.
2. **OTHER APPLICATIONS** 
   * + Rent Payment to House Owners.
     + Wholesale-Retailers Bill Payment.
     + Electronic Fund Transfer for Utility Payment options.

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 EXISTING SYSTEM**

***2.1.1 ONLINE BILL PAYMENT***

Online bill payment systems are normal electronic payment system which are connected to the Payment Gateways so that the users could pay their bills through the gateways to the Billers. Users are supposed to login to their account in the Biller’s page. Now they are provided with all the bill details and their late payment charges and such. Now the user should manually pay the Bill amount through the payment gateway. This involves the users to keep updated on all the bills that are to be paid, their due dates and other such information.

**Drawbacks**

* + Users are supposed to keep track on all the bill amounts and bill periods regularly.
  + User needs to perform same set of tasks every time to different billers.
  + Human Tendency to miss bill payment on date therefore landingup.

1. ***SCHEDULING PAYMENT***

Banks provides the facility to pay bills from their net banking feature if the user has been registered with it. If so, Bank also offers the facility to schedule the payment on particular date to any particular Biller. That is the payment to the Biller on the specified amount is made to the biller from the user’s bank account itself. In this case user have to add sufficient funds to the account so that the bill gets paid.

**Drawbacks**

* + Users are supposed to maintain sufficient balance in order that the bill could be processed on the specified date.
  + If the User makes use of the Account Balance for some other Purpose, then the schedule gets wasted and again the User is bound to pay late
  + payment charges.

1. **PROPOSED SYSTEM**

**2.2.1 *EASYPAY***

Easypay application overcomes all those drawbacks which lasted in the above two existing models. Easypay consists of Single Window Billers Section where all the billers of the user are linked into one single window. This facilitates to the user that, with just one single login the user is now possible to get the details of all his bills that do exist. There is no possibility of late payment charges or the need for user to keep tracking of the bills every month. The only necessity every month the user is required to perform is to add the notified cash to the wallet once in a month or as he/she wishes. Thus one session of the user with EASYPAY will complete all the bills of the month to be cleared at instant.

It also maintains the features of the existing systems intact. There is an option called INSTA-PAY where the user is free to choose his billers and manually make the payment process incase if he needs to do so. The Payment gateway is linked to perform this in the most secured environment. There is also an additional feature called “Pay to Easypay Representative” which the users could make use of incase if the Online payment does not work.

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 FEASIBILITY STUDY**

The main objective of feasibility study is to test the Technical, operational and economical aspects for adding new modules and debugging old running system. All systems are feasible for adding new modules and debugging old running system.

There are aspects in the feasibility study portion of the preliminary investigation:

* + Technical feasibility
  + Economic feasibility
  + Operational feasibility

1. **TECHNICAL FEASIBILITY**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. To develop this system, we first worked with android applications and then found that running it as web application Server would be technically feasible. The requirements here are very modest because the system here supports very basic Server options which is WAMP 2.1 and above.

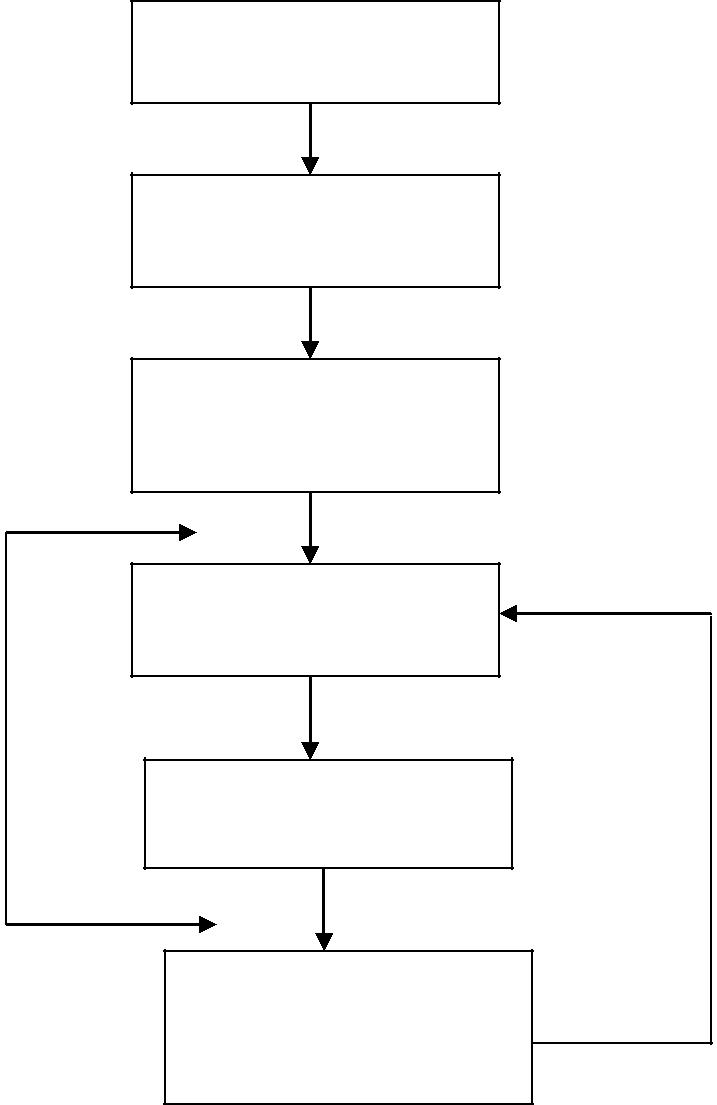
So, this system is technically feasible because it needs only WAMP with a basic version.

**3.1.2 ECONOMIC FEASIBILITY**

Economic feasibility is the most frequently used method for evaluating the effectiveness of the proposed system. More commonly known as cost-benefit

analysis, the procedure that costs for a proposed system and weights them against the tangible and intangible benefits of the system.

The system is cost effective because it is available as a web application and can be accessed from the internet. The system is economically feasible because the users can normally load the application over the computers and mobiles and use them.



Alternative Project

Analysis

Needs Analysis

Initial Screening

Market and

Demand Analysis

Technical Analysis

Impact Assessment

Financial and

Economic Analysis

Project Change

Required

**Fig. 3.1: Feasibility Analysis**

**3.1.3 OPERATIONAL FEASIBILITY**

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. This system will not threaten the user instead it is friendly in its operation. All the user needs to have is an Internet connection provided to his mobile. The system is operationally feasible because the user can run the application just by clicking on the mic icon present there and then continue talking. The whole process is done only through the speech recognition.

**3.2 HARDWARE USED**

RAM :1GB and above

Processor : Intel Processor (Core i3)

**3.3 SOFTWARE USED**

Operating System : Windows 7 Home Premium

Version : 2.4 and above

Server : WAMP

Version : 2.2 and above

Front End : HTML

Back End : PHP, My SQL

**CHAPTER 4**

**DETAILED DESIGN**

**4.1 SYSTEM ARCHITECTURE**

The following is the architecture of Integrated Billing and Automated

Payment System.



**Fig 4.1: System Architecture**

**MODULES**

1. **Login Authentication Module:** It authorizes the login of the Valid User.Unauthorized users are prevented from viewing the content of the protected page.
2. **Add Services/Module:** After the User is successfully logged into hisaccount, he is allowed to link any services to his account. This module performs all those functions.
3. **View Bill Details:** This module displays all the associated bill details of theuser that he has registered with. Non Registered bills are not active to the user.
4. **Change Password:** It is strictly mandatory for the user to change thepassword periodically for the security reasons. So, there is a separate module used to change the Easypay passwords alone. It should be noted that only Easypay passwords could be changed from this module and not the passwords of linked services.
5. **Instapay:** This module takes care of the instant bill payment process whichis initiated by the user. Here, the user could make payments to the Service Provider manually.
6. **Adding Cash to Wallet:** This module is the most important module and isconnected to the Payment gateway page for accessing the account details of the user. Funds are transferred through this module from the gateway.
7. **Logout:** This module terminates all the user activity in one particularsession and enables the other user to take hold of the system.

**FLOW DIAGRAM**

The following diagram describes about the flow that takes place in the

process of ticket booking.

Add Services

Add Cash to Wallet

Yes

No

If billpayment&&Wallet balance >Bill Amount

Pay Bill

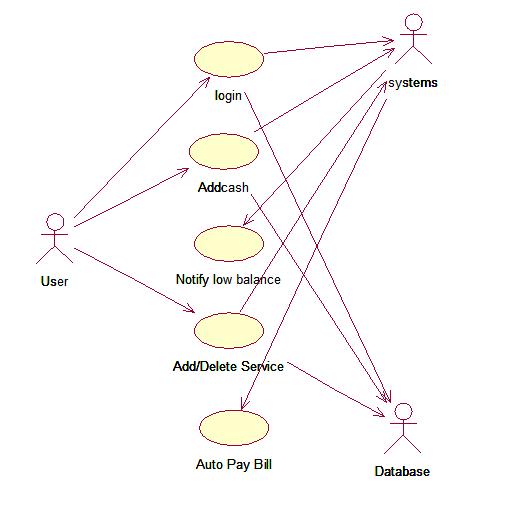
**Fig 4.2: Flow Diagram for ticket booking**

**4.3 UML DIAGRAMS**

The Unified Modelling Language (UML) is a general purpose modelling language in the field of software engineering. The basic level provides a set of graphic notation techniques to create visual methods of object-oriented software-intensive systems. Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects.

**4.3.1 USE CASE DIAGRAM**

Use case describes the interaction between one or more actors and the system itself, represented as a sequence of simple steps that take part in a sequence of activities in a dialog with the system to achieve goal.

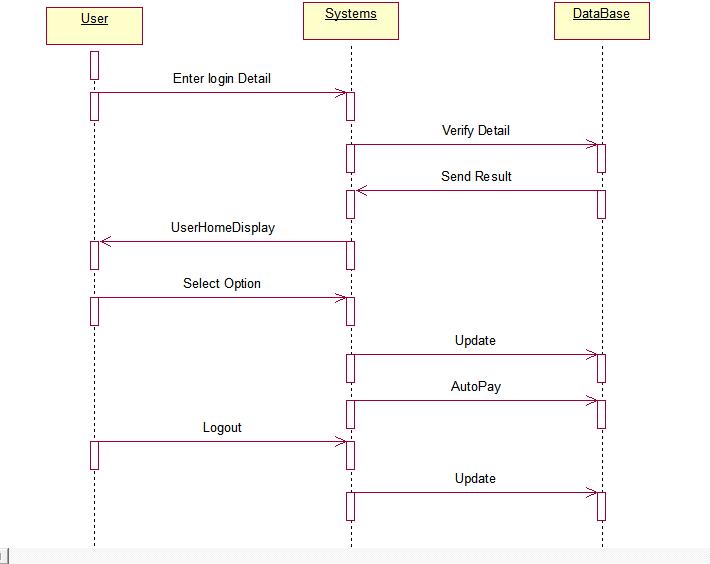


**Fig 4.3: Use Case Diagram**

There are three actors who are user, system and database necessary to run the application. The various functions of these actors like storing, entering updating and retrieving the details are represented through use cases.

**4.3.2 SEQUENCE DIAGRAM**

A Sequence diagram shows, as parallel vertical lines different processes or objects that live simultaneously and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

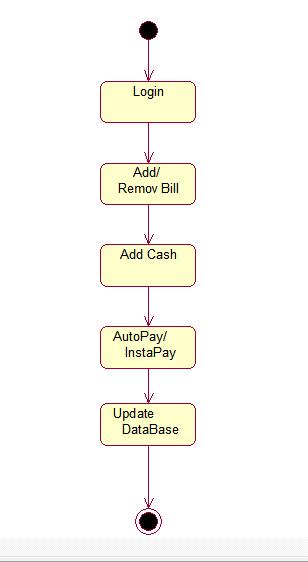


**Fig 4.4: Sequence Diagram**

The actions such as entering the details, storing and so on are expressed in a sequential order through vertical lines. The exchange of data between the actors(user, system and database) are represented by a set of horizontal lines.

**4.3.3 ACTIVITY DIAGRAM**

Activity diagram are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.

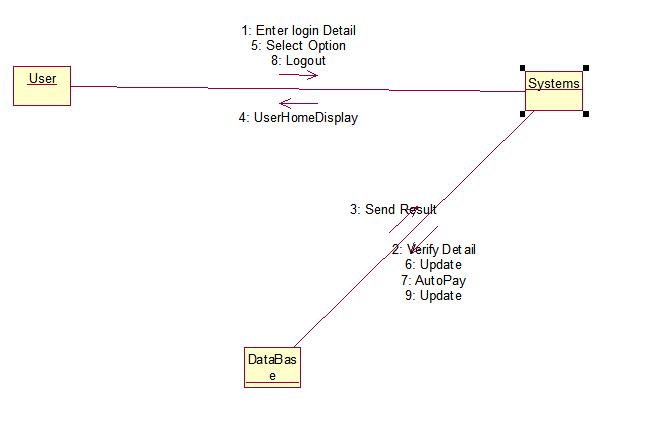


**Fig 4.5: Activity Diagram**

The set of activities that takes place among the actors is represented through activity diagram. It has a start and stop state which indicates the user to start and stop the activity respectively. The sequence of activities are represented through arrow marks.

**4.3.4 COLLABORATION DIAGRAM**

A Collaboration diagram is easily represented by modeling objects in a system and representing the association between the objects as links. The interaction between the objects is denoted by arrows. To identify the sequence of invocation of these objects, a number is placed next to each of arrows.

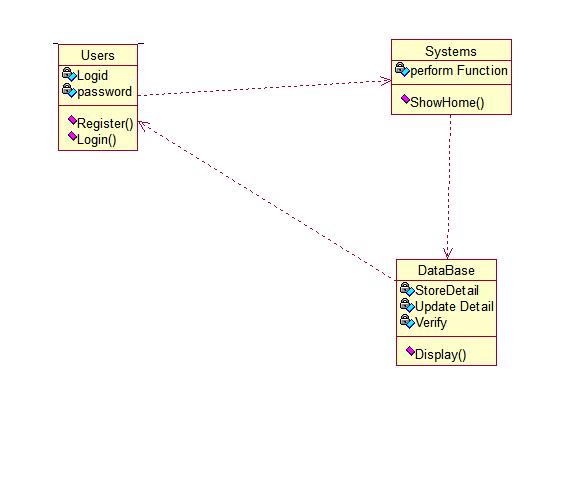


**Fig 4.6: Collaboration Diagram**

According to the sequence diagram drawn, an automatic collaboration diagram would be generated which again indicates the relationship among the actors.

**4.3.5 CLASS DIAGRAM**

Class Diagram provides an overview of the target system by describing the objects and classes inside the system and the relationships between them. It provides a wide variety of usages; from modeling the domain-specific data structure to detailed design of the target system. With the share model facilities,you can reuse your class model in the interaction diagram for modeling the detailed design of the dynamic behavior. The Form Diagram allows you to generate diagram automatically with user-defined scope.



**Fig 4.7: Class Diagram**

Class diagram describes the attributes and operations of different actors. Eg: For actor user the attributes are name, gender, source anddestination and the operations are entering the details and payment.

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

1. **IMPLEMENTATION**

This Project deals with the Bill Payment of the Utility Services the user has

registered with. To begin with, the user needs to create an account with Easypay. Once the account creation is done, the password is mailed to the registered email-id to verify the account. After verification the user is now allowed to change password of his choice in the first login attempt to his account. The User Screen provides list of options to the user to select with. The user can link the necessary services that he needs into his Easypay Window and later on view their bill details. The user has to add cash to the wallet in order to initiate the Payment process. There are two options available to add cash to the wallet and the user required option is selected. Once the cash is available in the wallet, then the payment process happens in an automated environment.

There is an option for Instant Payment available so that the user if needed could pay the bill amount manually. Insta-Pay also facilitates to be paid wither to an Easypay account or the Billers Account. Once any transaction is made the summary of the transaction is displayed and then the page is redirected to the user’s homepage.

The logout feature disables the current session used by any particular user and takes the user to the login screen where he is supposed to begin from the beginning. Also there is a guide scrolling to instruct the Users on how to use the application in case of any doubts on using.

**MODULES**

There are different set of codes performing different functions. Those codes could be reused whenever and wherever it could be linked with any other blocks in the Codes. These Reusable blocks are termed as modules and Easypay consists of the following modules,

**Login Authentication Module:** It authorizes the login of the Valid User.Unauthorized users are prevented from viewing the content of the protected page.

**Add Services/Module:** After the User is successfully logged into his account, he isallowed to link any services to his account. This module performs all those functions.

**View Bill Details:** This module displays all the associated bill details of the userthat he has registered with. Non Registered bills are not active to the user.

**Change Password:** It is strictly mandatory for the user to change the passwordperiodically for the security reasons. So, there is a separate module used to change the Easypay passwords alone. It should be noted that only Easypay passwords could be changed from this module and not the passwords of linked services.

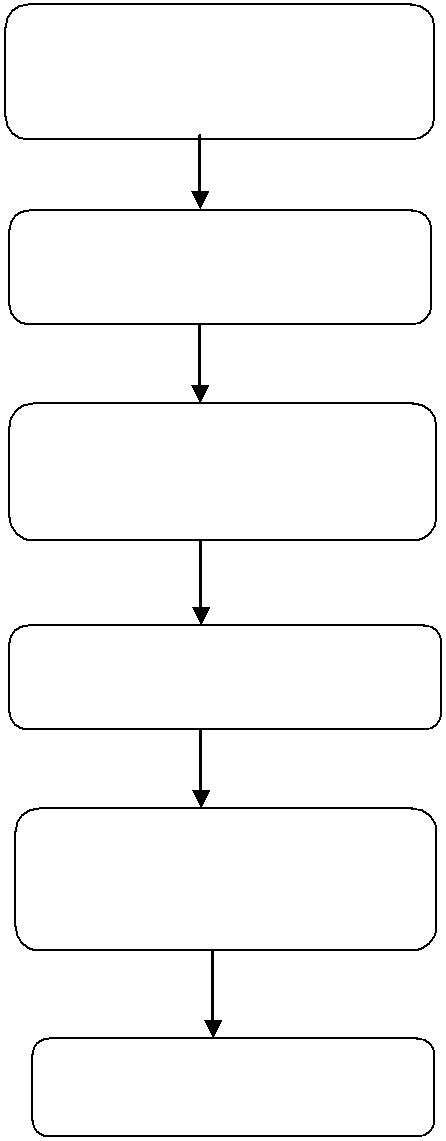
**Instapay:** This module takes care of the instant bill payment process which isinitiated by the user. Here, the user could make payments to the Service Provider manually.

**Adding Cash to Wallet:** This module is the most important module and isconnected to the Payment gateway page for accessing the account details of the user. Funds are transferred through this module from the gateway.

**Logout:** This module terminates all the user activity in one particular session andenables the other user to take hold of the system.

**5.2 TESTING**

Testing is an important phase that focuses on an empirical investigation in which the results describe the quality of the system. It cannot confirm system functions properly under all conditions but can establish that it fails under specific conditions. The prime purpose of testing is to guarantee that system successfully built and tested in the development phase meets all the requirements and design parameters.



TEST PLANNING

TEST ANALYSIS

TEST CASE

IDENTIFICATION

TEST EXECUTION

FINAL TESTING

IMPLEMENTATION

**Fig 5.1 Process of Testing**

1. **UNIT TESTING**

***USER LOGIN***

Table 5.1: Login Authentication

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Case** |  | **Expected** |  | **Observed** |  |  | **Result** |
|  |  |  | **Output** |  | **Output** |  |  |  |
|  |  |  |  |  |  |  |  |  |
| 1 | Valid |  | Display | the | User’s |  |  | Pass |
|  | Username | and | User |  | Homepage | | is |  |
|  | Password |  | Homepage |  | displayed |  |  |  |
|  |  |  |  |  |  |  | |  |
| 2 | Invalid |  | Display | an | An | error | | Pass |
|  | Username | and | error message | | message |  | is |  |
|  | password |  |  |  | displayed |  |  |  |
|  |  |  |  |  |  |  |  |  |

***CHANGE PASSWORD***

Table 5.2:Changing Password

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected** | **Observed** | **Result** |
|  |  | **Output** | **Output** |  |
|  |  |  |  |  |
| 1 | Enter new | Old Password | Old password | Pass |
|  | password | Should not | doesn’t works |  |
|  |  | work but new | but the new |  |
|  |  | password | password |  |
|  |  | should work | works |  |
|  |  |  |  |  |

***GRAPHICAL INTERFACE***

Table 5.3: Graphical interface

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected** | **Observed** | **Result** |
|  |  | **Output** | **Output** |  |
|  |  |  |  |  |
| 1 | Add any | Button | Button | Pass |
|  | service | renamed as | Renamed |  |
|  |  | Delete Service |  |  |
|  |  |  |  |  |
| 2 | Add Cash to | Pop Up | Pop Up | Pass |
|  | Wallet | Options in | Options comes |  |
|  |  | right |  |  |
|  |  |  |  |  |

***NAVIGATION MANAGEMENT***

Table 5.4:Navigation Management

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected** |  | **Observed** | **Result** |
|  |  | **Output** |  | **Output** |  |
|  |  |  | |  |  |
| 1 | Logout | Takes to login | | Login screen | Pass |
|  |  | screen | after |  |  |
|  |  | ending session | |  |  |
|  |  |  |  |  |  |
| 2 | Back during | Ask | for | Warning | Pass |
|  | Payment | Warning | to | message pops |  |
|  |  | conform |  | up |  |
|  |  |  |  |  |  |

**5.2.2 INTEGRATION TESTING**

Table 5.5: Integration testing

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **No** | **Test Case** |  | **Expected** | **Observed** | **Result** |
|  |  |  | **Output** | **Output** |  |
|  |  |  |  |  |  |
| 1 | View Bill |  | Display added | Display added | Pass |
|  |  |  | biller’s details | biller’s details |  |
|  |  |  |  |  |  |
| 2 | Insta | Pay | Rent Already | Rent Already | Pass |
|  | second time | | paid message | paid message |  |
|  |  |  |  |  |  |

**5.2.3 FUNCTIONAL TESTING**

Table 5.6: Functional Testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **No** | **Test Case** |  | **Expected** |  | **Observed** |  | **Result** |
|  |  |  | **Output** |  | **Output** |  |  |
|  |  |  |  |  |  |  |  |
| 1 | Insta Pay | is | Debited | from | Debited | from | Pass |
|  | done |  | tenant | and | tenant | and |  |
|  |  |  | credited | to | credited | to |  |
|  |  |  | Owner |  | Owner |  |  |
|  |  |  |  |  |  |  |  |

**5.2.4 ACCEPTANCE TESTING**

Table 5.7: Acceptance Testing

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Test Case** | **Expected** | **Observed** | **Result** |
|  |  | **Output** | **Output** |  |
|  |  |  |  |  |
| 1 | Bills are added | Payment | Payments are | Pass |
|  |  | should be done | done |  |
|  |  |  |  |  |
| 2 | Incorrect Card | Display Error | Error Message | Pass |
|  | Details Given | message | displayed |  |
|  |  |  |  |  |

**5.3 TEST PLAN**

The project is tested to verify its correctness and identify the bugs. The test plan includes the various test cases that acts as the set of conditions or variables that determine whether the corresponding feature in the system is working as it originally established to do so. When this test plan is executed, the errors spotted are rectified and the final testing yields following result.

**5.4 TEST ANALYSIS**

In this phase of testing, the requirements for software testing are analyzed and later its feasibility is determined. In the feasibility study the possibility of project development is found through suitable test cases.

**5.5 RESULT**

The application is tested and found to function as expected with no errors. This application provides an interface for the users to pay the bills in an efficient way. Thus the payment is done with user’s less effort thereby saving much time and work.

**CHAPTER 6**

**CONCLUSION AND FUTURE ENHANCEMENT**

**6.1 CONCLUSION**

Easypay provides the easiest platform to Pay Utility Bills over the internet thereby eliminating the human need to remember different bill details and his carelessness leading to the late payment charges.

**6.2 FUTURE ENHANCEMENT**

This application can be developed in a way users can pay all their financial transactions through this . The same idea can be applied in different fields such as Retailer-Whole-Seller Billing, Money Exchange Platformsetc. This application can be adapted to various other platforms. The application can also be enhanced to process the alternative to Banking.

**APPENDIX-A**

**SAMPLE SOURCE CODE**

**LOGIN:**

The following code tells us how the Login Authentication is done to the Easypay account.

**Login.php**

<?phpsession\_start();

$servername = "localhost"; $username = "root"; $password = "";

$dbname = "test\_database";

$conn = mysqli\_connect($servername, $username, $password,$dbname);.

if ($\_SERVER['REQUEST\_METHOD'] == 'POST')

{

$user = htmlspecialchars(($\_POST['user'])); $password = htmlspecialchars(($\_POST['password']));

$sql="SELECT \* FROM tbl\_login WHERE loginid='".$user."' and password='".$password."'";

$result = mysqli\_query($conn, $sql) or die(mysqli\_error($conn));

if($count==1)

{

$\_SESSION['user']=$user; $\_SESSION['password']=$password;

header("Location: userhome.php");

}

else

{

?><!DOCTYPE html><html>

<body>

<p>"Wrong UserName or Password"</p><a href="index.html">Login Again</a></body>

</html>

<?php

}

}

?>

**REGISTRATION**

The below Code is used to perform registration when a new user registers for the Easypay Account.

<!DOCTYPE html><html>

<head>

<title>EASYPAY</title>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/><link href="css/style.css" rel="stylesheet" type="text/css">

<script type="text/javascript">functionconfrmemail()

{

var email = document.getElementById("cde").value varconfirmemail = document.getElementById("abc").value if(email != confirmemail) {

alert('Email Not Matching!');

}

}

functionconfrmpass()

{

var pass1 = document.getElementById("pass").value var pass2 = document.getElementById("conpass").value if(pass1 != pass2) {

alert('Passwords Not Matching!');

} }

functiongenerateActivationString() { $randomSalt = '\*&(\*(JHjhkjnkjn9898'; $uniqId = uniqid(mt\_rand(), true);

return md5($randomSalt.$uniqId);

}

window.onload = function()

{

varabc = document.getElementById('abc'); abc.onpaste = function(e) { e.preventDefault();

}

}

</script>

</head>

<body>

<body style="margin:0;"><div id="container"><div id="header">

<center><imgsrc="logo.png" alt="logo" style="width:200px;height:70px; "></center>

</div>

<div id="content">

<form action="register.php" class="register" method="POST">

<p style="text-align:center;margin-top:-5px;padding-top:15px;padding-bottom:-15px;margin-bottom:5px; font-weight:900;color:#0170cd;font-size:x-large;">Register for Your Easy Pay Account</p>

<fieldset class="row1">

<legend class="mainheading">Account Details </legend>

<p><label>Email \* </label>

<input type="email" required name="email" id="cde"/><label>Repeat email \*

</label>

<input type="email" name="confirmemail" id="abc" onblur="confrmemail()"/></p>

<p>

<label>Password\*

</label>

<input type="password" name="password" id="pass"/><label>Repeat Password\*

</label>

<input type="password" name="confirmpassword" id="conpass" onblur="confrmpass()"/>

</p>

</fieldset>

<fieldset class="row2">

<legend class="mainheading">Personal Details </legend>

<p><label>Name \* </label>

<input type="text" class="long" name="name"/></p>

<p><label>Phone \* </label>

<input type="text" maxlength="10" name="phone"/></p>

<p>

<label class="optional">Street </label>

<input type="text" class="long" name="Streetname"/></p>

<p><label>City \* </label>

<input type="text" class="long" name="city"/></p>

<p><label>Country \* </label>

<select name="Country" >

<option value="" disabled selected >Select Country</option><option value="United States" >United States</option><option value="India" >India</option>

</select>

</p>

</fieldset>

<fieldset class="row3">

<legend class="mainheading">Further Information </legend>

<p>

<label>Gender \*</label>

<input name="case" id="case1" type="radio" value="male" /><label class="gender">Male</label>

<input name="case" id="case1" type="radio" value="female" /><label class="gender">Female</label>

</p>

<p><label>Birthdate \* </label>

<select class="date" name="datenum">

<option value="" disabled selected >Select Date</option><option value="1">01</option>

<option value="2">02 </option>

<option value="3">03</option><option value="4">04</option><option value="5">05</option><option value="6">06</option><option value="7">07</option><option value="8">08</option><option value="9">09</option><option value="10">10</option><option value="11">11</option><option value="12">12</option><option value="13">13</option><option value="14">14</option><option value="15">15</option><option value="16">16</option><option value="17">17</option><option value="18">18</option><option value="19">19</option><option value="20">20</option><option value="21">21</option><option value="22">22</option>

<option value="23">23</option><option value="24">24</option>

<option value="25">25</option><option value="26">26</option>

<option value="27">27</option>

<option value="28">28</option>

<option value="29">29</option>

<option value="30">30</option>

<option value="31">31</option></select>

<select name="month" >

<option value="" disabled selected >Select Month</option><option value="1">January</option>

<option value="2">February</option><option value="3">March</option><option value="4">April</option><option value="5">May</option>

<option value="6">June</option>

<option value="7">July</option>

<option value="8">August</option>

<option value="9">September</option>

<option value="10">October</option>

<option value="11">November</option>

<option value="12">December</option></select>

<input class="year" name="yearofbirth" type="text" size="3" maxlength="4"/>e.g 1976

</p>

</fieldset>

<fieldset class="row4">

<legend class="mainheading">Terms and Conditions </legend>

<p class="agreement">

<input type="checkbox" required value=""/>

<label>\* I accept the <a href="rules.html" target="\_blank">Terms and Conditions</a></label>

</p>

<p class="agreement">

<input type="checkbox" value=""/>

<label>I want to receive personalized offers by your site</label></p>

<p class="agreement">

<input type="checkbox" value=""/>

<label>Allow partners to send me personalized offers and related services</label></p>

</fieldset>

<center><div><button class="btn" >Register </button></div></center></form>

</div>

</div>

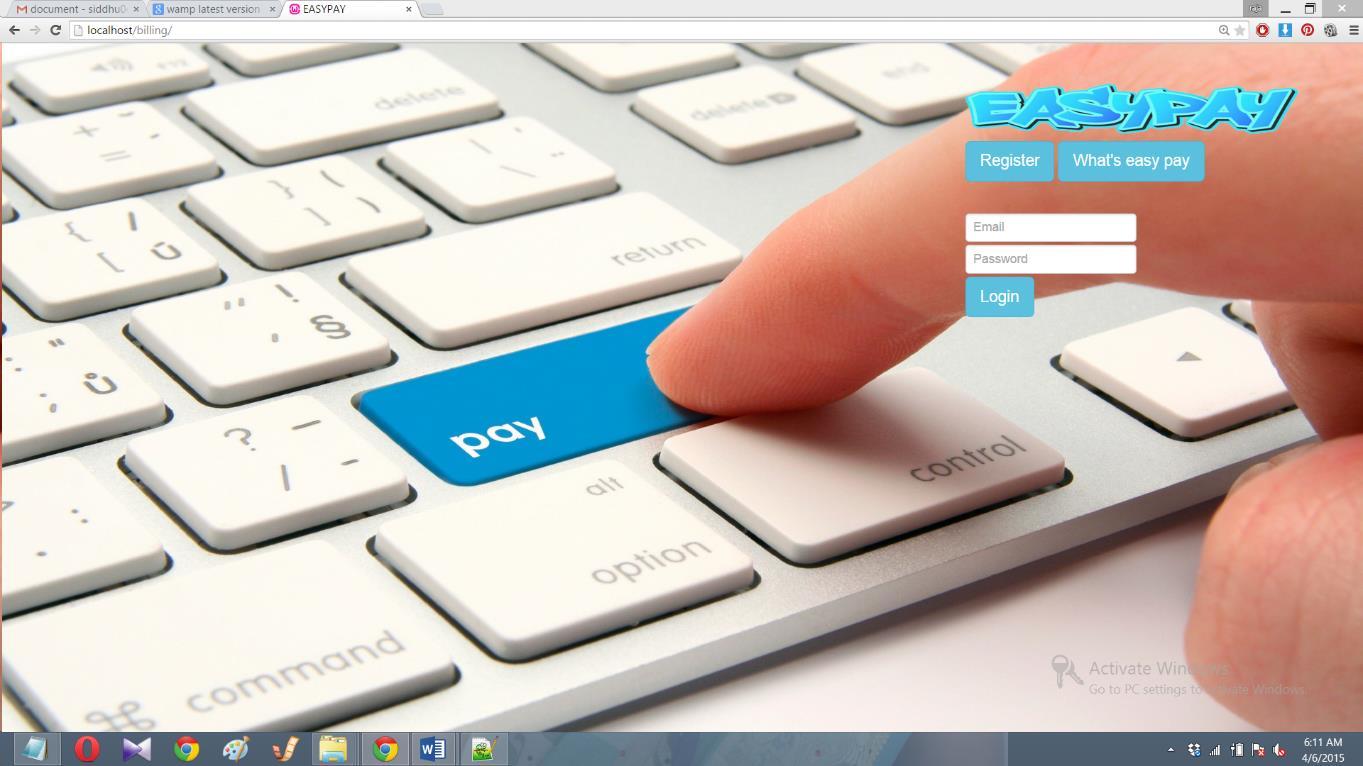
</body>

</html>

**APPENDIX-B**

**SCREEN SHOTS**

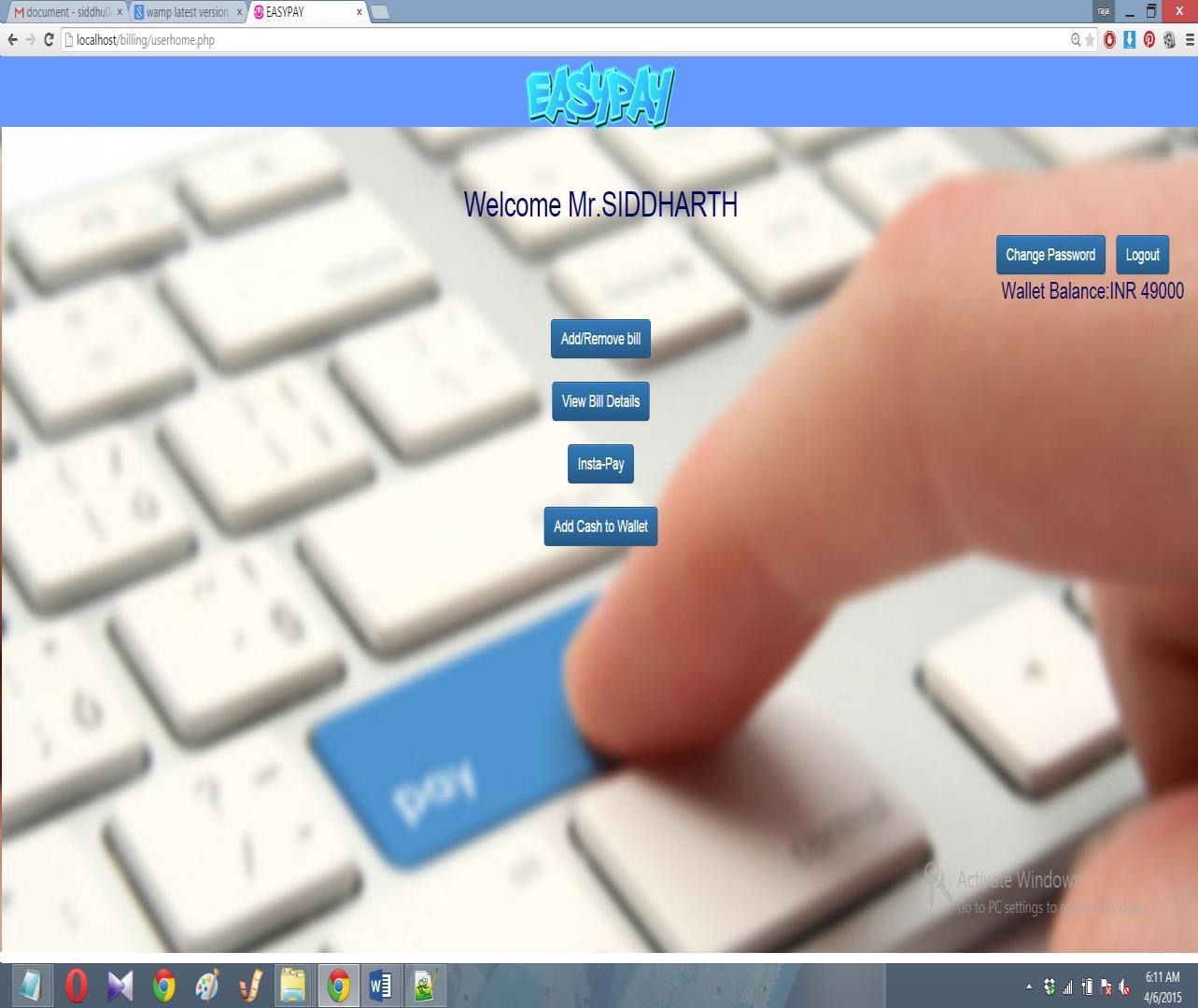
**LOGIN PAGE:**



**Fig b.1:Login Screen**

This this is the index page where the users are supposed to login to their respective accounts.There is also a brief description given about Easypay.

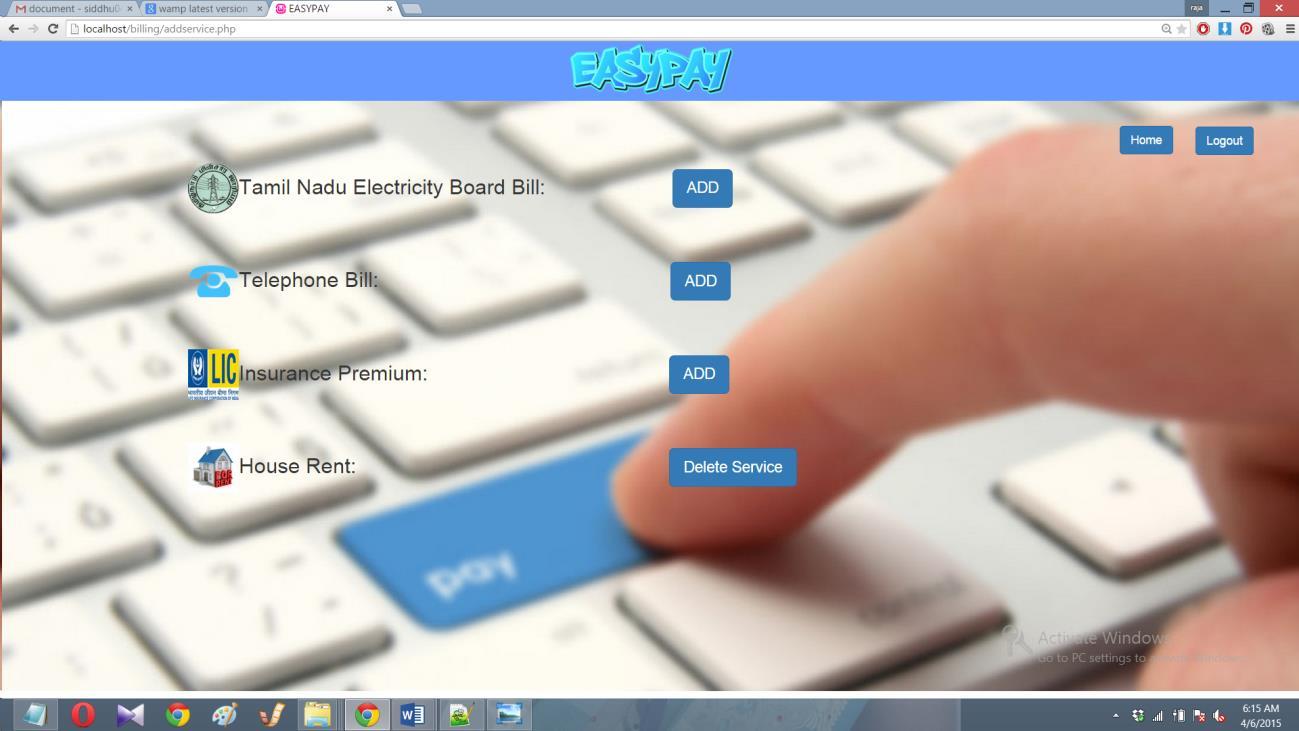
**USERHOME PAGE:**



**Fig b.2:Userhome After Successful Login**

User, after successfully logging in is directed to his own page where he can perform any option of his choice.

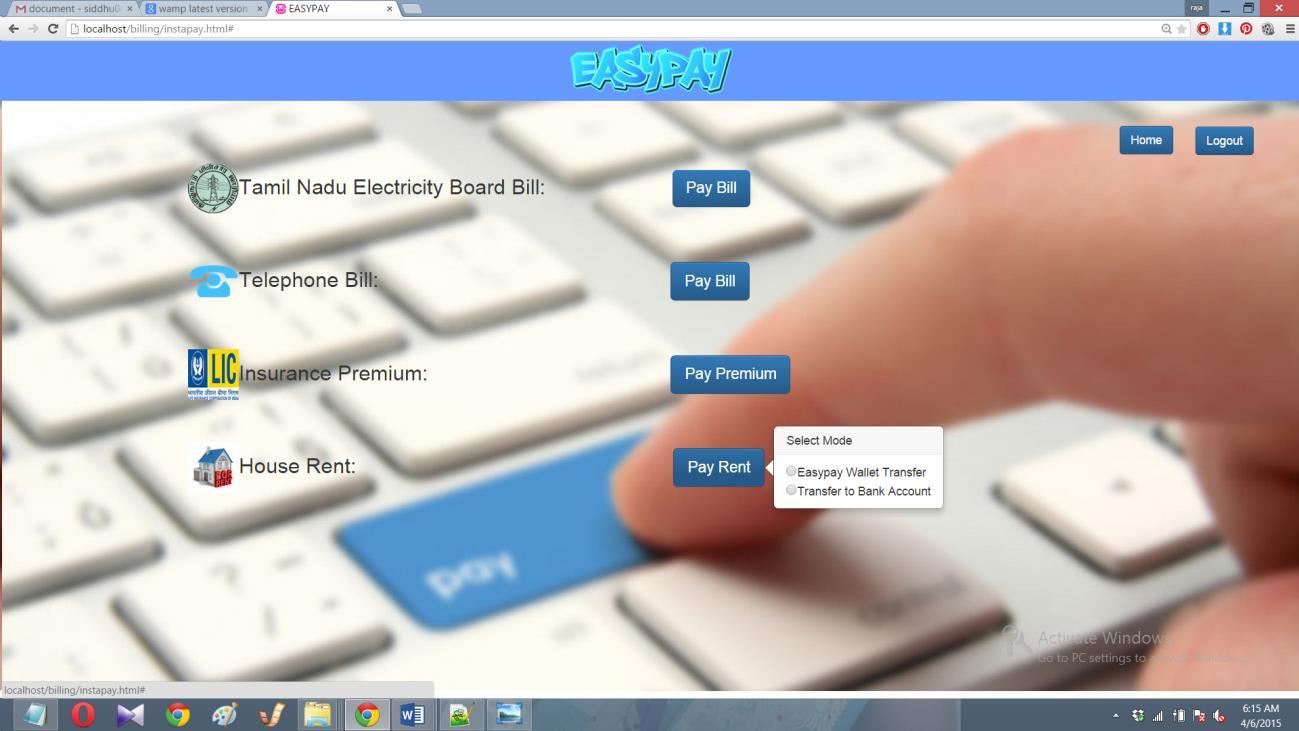
**ADDING SERVICES:**



**Fig b.3: Adding New Services**

User can add any new utility service he needs by clicking on the “ADD” in the above page and the same service could also be deleted.

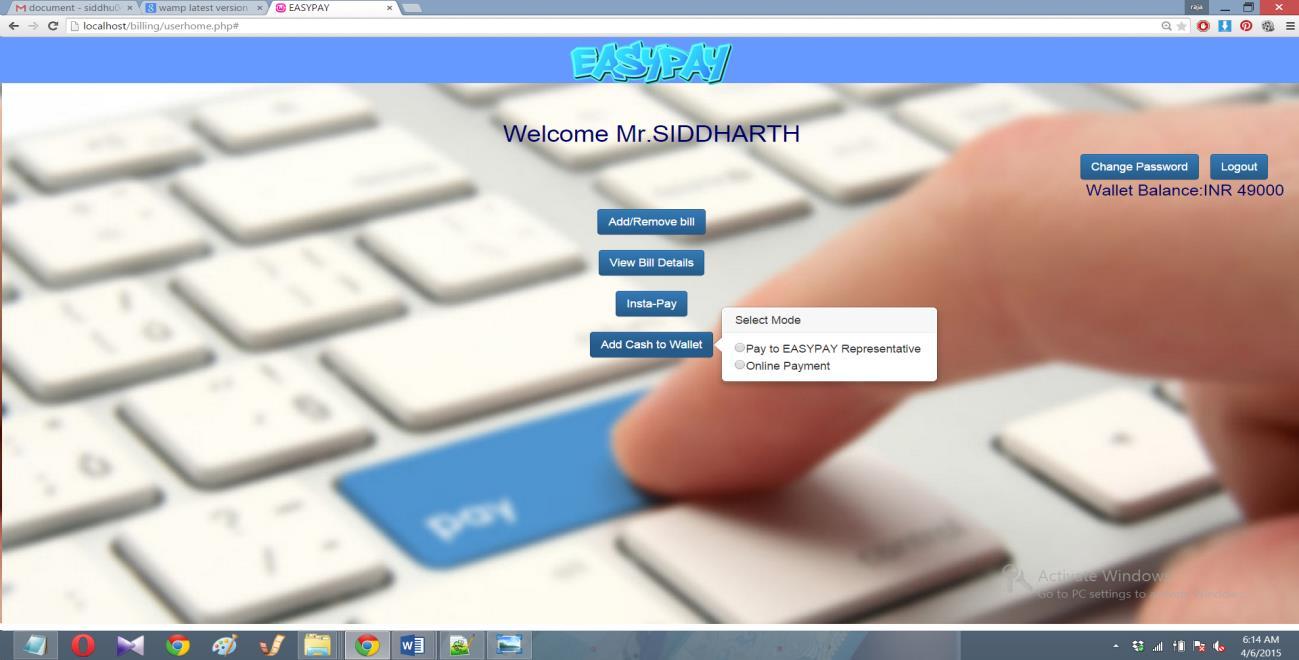
**INSTANT PAYMENT:**



**Fig b.4: Instant Payment**

The above page allows the users to pay the bill manually for any of the associated accounts without pre registering for the service.

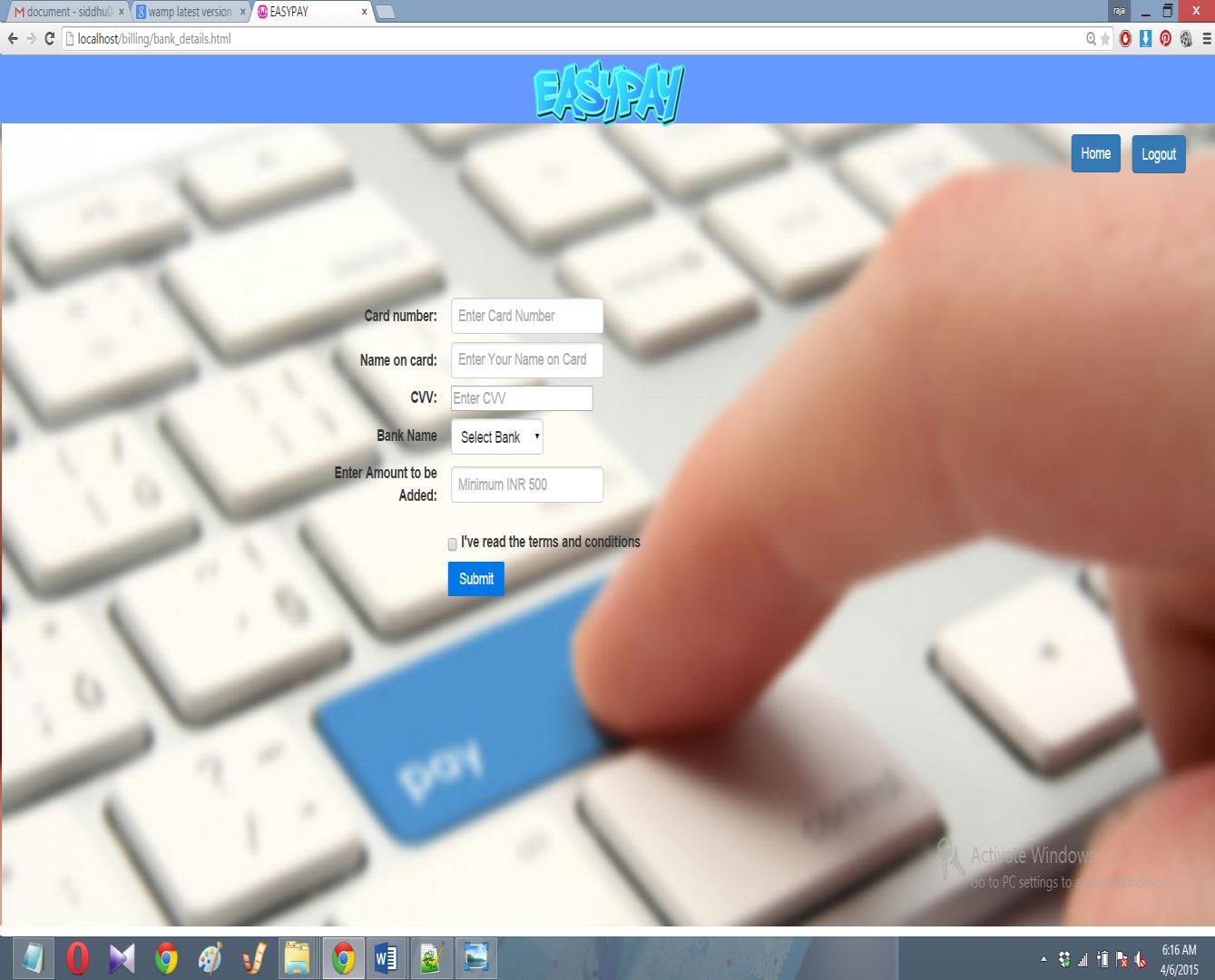
**ADDING CASH TO WALLET:**



**Fig b.5: Adding Cash to Wallet**

User when needed to add cash to his own wallet with Easypay,he can make use of the two options available in the above screen.

**ONLINE PAYMENT:**



**Fig b.6: Adding Cash to Wallet-Online**

User can add funds to Easypay Wallet using the payment Gateway option

**REFERENCES**

1. "Notice of Retraction Research on Unified Payment Gateway Online Payment Mode," by Qifeng Yang; Zhengwei Cheng; Ping Song in *Wireless Communications,Networking and Mobile Computing, 2008. WiCOM '08. 4th International Conference on*,vol.,no.,pp.1,5,2-14Oct.2008doi: 10.1109/WiCom.2008.2118
2. "Online payment system using steganography and visual cryptography," by Roy,

S.; Venkateswaran, P. in *Electrical, Electronics and Computer Science (SCEECS),2014IEEE-Students'-Conference,on*,vol.,no.,pp.1,5,1-2-March-2014doi: 10.1109/SCEECS.2014.6804449

[3] "Research on Online Payment Mode Based On Internet Banking Payment Gateway," by Qifeng Yang; Zhengwei Cheng; Ping Song in *Convergence InformationTechnology, 2007. International Conference on* , vol.,no.,pp.2043,2048,21-23Nov.-2007doi: 10.1109/ICCIT.2007.122