|  |
| --- |
| Final Documentation |
|  |
|  |
| |  | | --- | | **Abhishek Singh | Aman Kumar | Abhishek Kumar Ravi** | |

|  |
| --- |
| Utility Management |



Table Layout

* Abstract ……… (2)
* System Analysis ………….. (3)
* Feasibility Study …………… (4)
* Requirement Specification ……….. (6-8)
* Directory Module ………….. (9)
* Snapshots ……………………………… (10-15)
* User Case Diagram ………………………..(16)
* Page Flow Diagram ………………….. (17)
* Data Dictionary …………………….. (18)
* Testing ………………………. (19-21)
* Maintenance …………………… (22)
* Future Scope ………………….. (23)

Abstract

It’s a payment management system. Since, we all have once paid the bill either it is a phone bill, broadband bill or electricity bill. It’s always a challenge for the corporate house or government to satisfy customer by their service. So, we have built a product on **Electricity Payment System**.

General Overview

It is much like an online payment system. But, it targets the Electricity so, what a customer needs and their security is concerned. On the other site, we have prepared a portal for the administration purpose. Since, it’s not a single-end software product. So, it can be handled easily on administrator’s side.

Site Overview

Taking on the implementation, we have assured you to put it as a web based or a desktop based. At the end, it’s a User based product which must be benefited by the common users along with the maintenance team which work hard on the record management.

System Analysis

**Identification of Need:**

Identification of the need is the process of finding out the basic requirement of the project. It is a vital stage and requires thorough analysis. In case of this project, the basic need is to develop a user friendly, feature-rich, practical Online Voting System.

Preliminary Investigation:

This activity involves interviewing the end users and customer and studying the existing documents and materials to collect all possible information. The purpose of this activity is to understand the exact requirements of the customer. We have to identify and eliminate the various problems of anomalies, inconsistencies and incompleteness.

Feasibility Study

It includes the following-

1. Technical Feasibility
2. Economical Feasibility
3. Operational Feasibility
4. **Technical Feasibility:**

Since the project is design with ASP.NET with C# as code behind and SQL server 2008 as backend, it is easy to install in the systems whenever needed. It is more efficient, easy and user friendly to understand by almost everyone. Huge amount of data can be handled efficiently using SQL server as backend. Hence this project has good technical feasibility. As per the technical aspect the project College Management System is technically suitable for the user as well as the developer.

1. **Economical Feasibility:**

Economic feasibility is mainly concern with the cost incurred during the implementation of the software. Since the project is developed using ASP.NET with C# and SQL server which is more commonly available and even the cost involved in the installation process is not high. Similarly it is to recruit persons for operating the software. Since almost all the people are aware of ASP.NET with C# and SQL server. Even if we want to train the person in these areas, the cost involved in training is also very less. Hence this project has good economic feasibility.

1. **Operational Feasibility:**

As this ASP.NET component is very easy to operate for the software engineers and easy to use. So this project has a good operational feasibility.

Hence this project is technically, economically as well as operationally feasible.

­­­­­

Requirement Specification

**Minimum Requirements for Client Side:**

**Hardware Requirement:**

* Intel i3 or above based PC
* 2 GB of RAM
* Minimum Space Required: 80GB.
* Display: 64 bit colour.

**Software Requirement:**

* Operating Environment: Windows 7 or Windows 8.
* .NET Framework 4.5

**Performance Requirement:**

* The performance of the product is dependent on the hardware.
* Security should be high such that no intruder can tamper with data.

**Minimum Requirements for Server Side:**

**Hardware Requirement:**

* Intel® i3 or above Processor
* 2 GB of RAM
* Minimum Space Required: 500GB.

**Software Requirement:**

* Operating Environment: Windows 7 or Windows 8.
* Microsoft Visual Studio 2010.
* IIS (Internet Information Service).
* My SQL Server Workbench.
* .Net frame work4.5.

**Programming Language and Developing Tools:**

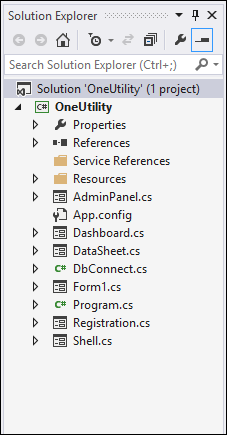
**Programming Language:**

* C#.

**Development Tools:**

* Visual Studio 2010 (embedded SQL Server 2008).
* MySQL Server Work Bench
* Operating System:- Windows 7

Directory Module



Snapshots

Fig1: Login page, which asks for the customer’s credentials.

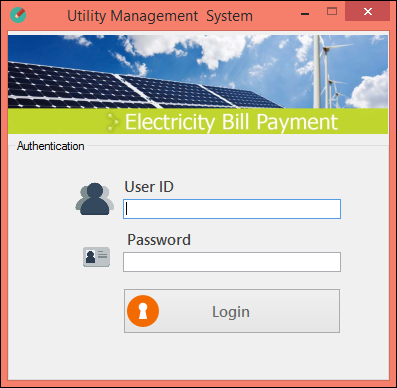


Fig2: Dashboard page, where user can pay their bills.

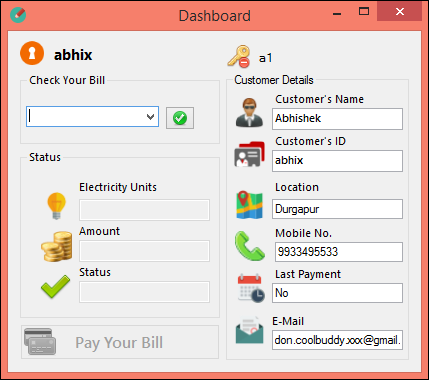


Fig3: After Selecting the payment month,

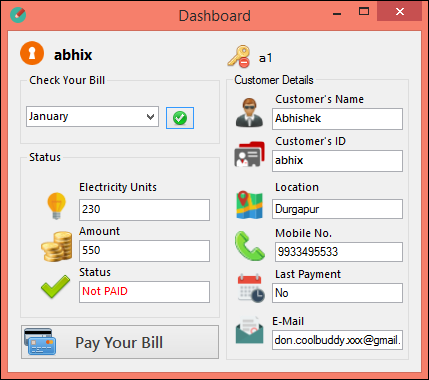


Fig4: And, when you click on the PAY button,

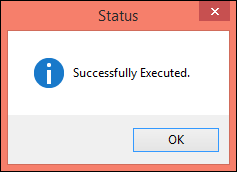
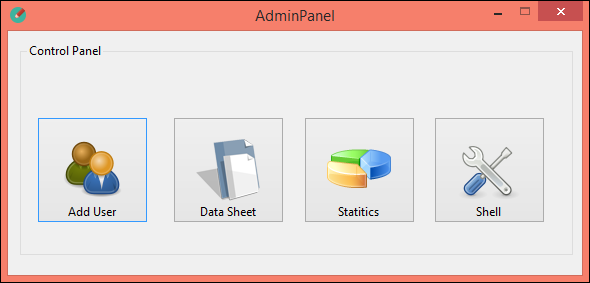


Fig5: And, what if I login as an ADMINISTRATOR



And, when we click on SHELL it would be like.

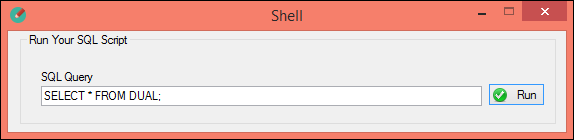


Fig 6: When you click on Registration Page,

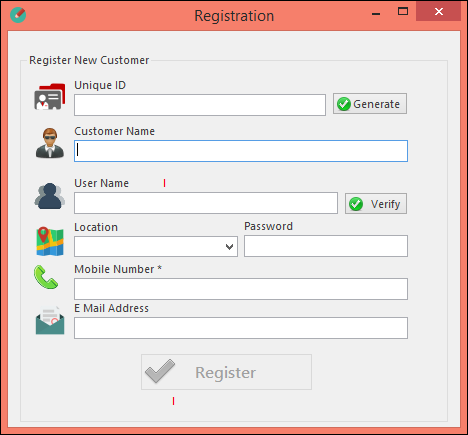
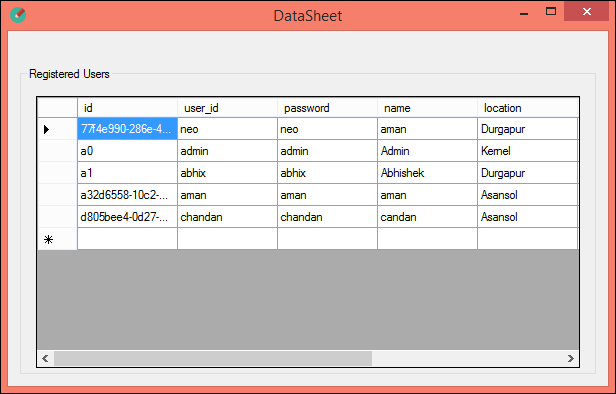
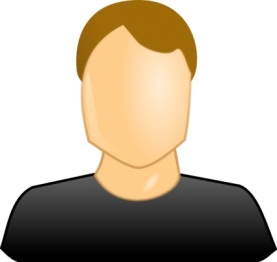


Fig 8: When you click on Data Sheet Button, then it will show you the grid of all the record in the database.



User Case Diagram

Payment Utility



**Pay Bill**

User

**Add Users**

**Change Data**



**Handle Database**

Administrator

Page Flow Diagram

**Shell**

**Data Sheet**

**Registration**

**Admin Control Panel**

**Payment Status**

**Bill Data**

**Login Page**

**User Page**

**Admin Page**

Data Dictionary

**gx\_utility**

* **Id**
* **Month**
* **Amount**
* **isPaid**
* **unit**
* **Id**
* **User\_id**
* **Password**
* **Name**
* **Location**
* **Mobile**
* **Lay\_pay**
* **Email**

**Tbl\_bill**

**Tbl\_users**

Testing

* **White Box Testing:**

**White-box testing** (also known as **clear box testing**, **glass box testing,transparentbox testing** and **structural testing**) is a method of testing [software](http://en.wikipedia.org/wiki/Software) that tests internal structures or workings of an application.

Using **white**-**box** **testing** methods, we derived test cases that:

* Guarantee that all independent paths within a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds, and
* Exercise internal data structures to ensure their validity.

The goal of white box testing is to cover testing as many of the statements, decision point, and branches in the code base as possible.

# **Black Box Testing:**

Black-box testing, also called **behavioral**testing, focuses on the method of [software testing](http://en.wikipedia.org/wiki/Software_testing) that examines the functionality of an application (e.g. what the software does) without peering into its internal structures or workings (see [white-box testing](http://en.wikipedia.org/wiki/White-box_testing)).

**Black**-**box** **testing** attempts to find errors in the following categories:

* incorrect or missing functions,
* interface errors,
* errors in data structures or external database access,
* behavior or performance errors, and
* Initialization and termination errors.
* **Unit Testing:**

**Unit testing** refers to tests that verify the functionality of a specific section of code, usually at the function level.

These types of tests are usually written by developers as they work on code (white-box style), to ensure that the specific function is working as expected. One function might have multiple tests, to catch corner cases or other branches in the code. Unit testing alone cannot verify the functionality of a piece of software, but rather is used to assure that the building blocks the software uses work independently of each other.

The goal of unit testing is to isolate each part of the program and show that the individual parts are correct.

* **Integration Testing:**

**Integration testing** (sometimes called Integration and Testing, abbreviated "I&T") is the phase in software in which individual software modules are combined and tested as a group. It occurs after [unit testing](http://en.wikipedia.org/wiki/Unit_testing). Integration testing works to expose defects in the interfaces and interaction between integrated components (modules). Progressively larger groups of tested software components corresponding to elements of the architectural design are integrated and tested until the software works as a system.

The purpose of integration testing is to verify functional, performance and reliability requirements placed on major design items.

* **System Testing:**

**System testing** of software is testing conducted on a complete, integrated system to evaluate the system's compliance with its specified [requirements](http://en.wikipedia.org/wiki/Requirements). System testing takes, as its input, all of the "integrated" software components that have successfully passed [integration testing](http://en.wikipedia.org/wiki/Integration_testing) and also the software system itself integrated with any applicable hardware system(s).

Maintenance

Maintenance is the enigma of system development. Maintenance can be classified as corrective, adaptive and perfective.

Adaptive maintenance means changing the program function.

Perfective maintenance means enhancing the performance or modifying the program to respond to the user additional or changing needs.

Maintenance covers a wide range of activities, including correcting coding and design errors, updating documentation and test data and upgrading user support. Several MIS organization has done to attack the ever growing problem of software maintenance through a maintenance reduction-plan that consists of two phases.

i).Maintenance management audit.

ii).Software system audit

Maintenance management audit is done through interviews and questionnaires, evaluates the quality of maintenance effort.

Software system audit entails an overall view of the system documentation and an assessment of the quality of data files and database and system reliability and efficiency.

Software implementation which consists of program writes system level update and re audit of manual system to make sure that the errors have been corrected.

Future Scope

* Implement the Graphical Representation of the Data/collection.
* Re-design the Database architecture for better and cross-application support.
* Make it available for the Web as Web Application which supports responsive design.
* Add Payment Gateway System, so we could support Internet Baking and accept Debit/Credit card.
* Create more admin rights and privileges so that, they can customize the application according to themselves.