## Chapter 1. Introduction

### Introduction to Project

The Task Management System is a comprehensive web application developed using ASP.NET Web

Forms with a 3-tier architecture. This system aims to streamline task management processes within organizations, allowing efficient assignment, tracking, and completion of tasks among employees.

### Project Category

This project falls under the category of application or system development, as it involves the creation of a software solution to address specific organizational needs related to task management.

### Objectives

The primary objectives of the Task Management System are as follows:

1. To provide a centralized platform for managing tasks and assignments within an organization.
2. To facilitate effective communication and collaboration among employees working on various projects.
3. To improve task visibility and accountability through real-time tracking and monitoring features.
4. To enhance productivity by streamlining task allocation, prioritization, and scheduling processes.

### ****Problem Formulation****

### 

### **In many organizations, task management is often fragmented and inefficient, leading to delays, miscommunications, and productivity issues. Manual methods such as emails, spreadsheets, and paper-based systems are prone to errors and lack scalability. The need for a comprehensive task management solution arises from the challenges associated with coordinating and tracking tasks across multiple projects and teams.**

* 1. **Identification/Reorganization of Need**

The need for the Task Management System was identified through discussions with stakeholders and a thorough analysis of existing task management practices within the organization. Feedback from employees highlighted the limitations of current methods and the potential benefits of adopting a more robust and integrated solution. The project requirements were refined through iterative discussions and feedback sessions to ensure alignment with organizational objectives and user needs.

* 1. **Existing System**

Prior to the development of the Task Management System, the organization relied on manual methods such as spreadsheets, emails, and in-person meetings to manage tasks. These methods lacked centralized control, real-time visibility, and scalability, leading to inefficiencies and communication gaps. The limitations of the existing system underscored the need for a more automated and streamlined approach to task management.

* 1. **Proposed System**

The Task Management System offers a comprehensive solution to address the shortcomings of the existing system. Key features of the proposed system include:

* Centralized task management dashboard for viewing and assigning tasks.
* Role-based access control to ensure data security and privacy.
* Real-time notifications and updates for task assignments, comments, and status changes.
* Integration with email and calendar systems for seamless communication and scheduling.
* Reporting and analytics tools to track task performance, identify bottlenecks, and make data-driven decisions.

**1.8 Unique Features of the System**

The Task Management System distinguishes itself from other solutions through its unique features, including:

* Customizable task workflows to accommodate different project requirements.
* Integration with third-party tools and services for enhanced functionality and interoperability.
* Support for mobile devices, allowing users to access and manage tasks on the go.
* Extensible architecture that allows for future enhancements and scalability.

### Core Components

App components are the essential building blocks of an Web app. Each component is an entry point through which the system or a user can enter your app. Some components depend on others. Each type serves a distinct purpose and has a distinct lifecycle that defines how the component is created and destroyed. The following sections describe the four types of app components.

### Hardware Component

* + 1. Intel Quad core 1.7 GHZ Processor or above.
    2. Minimum 100 GB HD.
    3. Standard Keyboard and Serial Mouse.
    4. Minimum 4GB Of Ram

### Software Component

* + 1. **Operating System:** Windows 10
    2. Visual Studio 2019
    3. SQL Server 2019

## Chapter 2. Requirements Analysis and System Specification

### 2.1 Feasibility Study

### ****Technical Feasibility:****

### The technical feasibility of the Task Management System was assessed by evaluating the availability of necessary technology and infrastructure to support its development and implementation. This included an analysis of hardware, software, and network requirements, as well as the feasibility of integrating with existing systems and platforms.

### ****Economic Feasibility:****

### An economic feasibility study was conducted to assess the financial viability of the project. This involved estimating the costs associated with development, deployment, maintenance, and support of the system, as well as projecting potential benefits and returns on investment. Cost-benefit analysis was performed to determine whether the benefits outweighed the costs over the project's lifecycle.

### ****Operational Feasibility:****

### The operational feasibility of the Task Management System was evaluated by considering its alignment with organizational goals, user needs, and operational processes. This included assessing the readiness of stakeholders to adopt and utilize the system, as well as identifying any potential barriers or challenges to its implementation and adoption.

### 2.2 ****Software Requirement Specification Document****

### The Software Requirement Specification (SRS) document outlines the functional and non-functional requirements of the Task Management System. It includes the following sections:

### 2.3.1 ****Data Requirements****

* The system shall store user information, including user ID, password, and role.
* Task data shall include task ID, description, assigned employee, status, priority, and deadline.
* Project data shall include project ID, name, description, start date, and end date.

### ****2.3.2 Functional Requirement****

* Users shall be able to log in to the system using their user ID and password.
* Admin users shall have privileges to create, edit, and delete projects, tasks, and employees.
* Users shall be able to view, comment on, and update tasks assigned to them.
* Admin users shall be able to assign tasks to employees and track their progress.

### ****2.3.3 Performance Requirement****

* The system shall support a minimum of 100 concurrent users.
* Response time for user interactions shall be less than 2 seconds.
* Data retrieval and processing operations shall be optimized for efficiency.

### ****2.3.4 Dependability Requirement****

* The system shall have a backup and recovery mechanism to ensure data integrity and availability.
* Error handling and logging mechanisms shall be implemented to facilitate system troubleshooting and maintenance.

### ****2.3.5 Maintainability Requirement****

* The system architecture shall be modular and extensible to accommodate future updates and enhancements.
* Code documentation and commenting standards shall be followed to facilitate maintenance by developers.

### ****2.3.6 Security Requirement****

* User authentication and authorization mechanisms shall be implemented to ensure data security and privacy.
* Data encryption shall be used to protect sensitive information during transmission and storage.

### ****2.3.7 Look and Feel Requirement****

* The user interface shall be intuitive and user-friendly, with consistent navigation and layout.
* Visual design elements shall adhere to organizational branding guidelines.

### ****2.3.8 Validation****

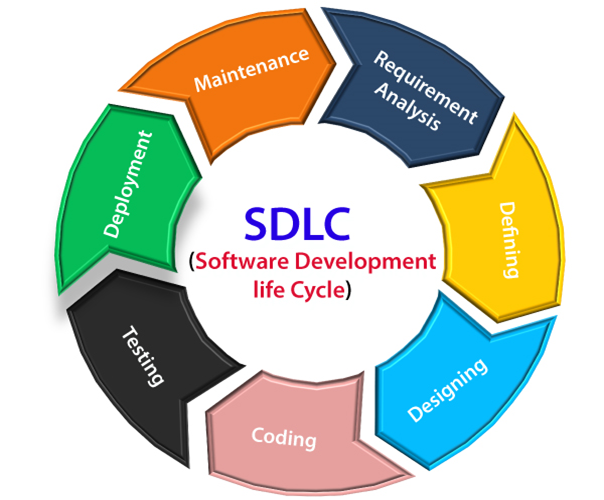
### Validation of the system requirements shall be conducted through user acceptance testing (UAT) and stakeholder reviews. Test scenarios and acceptance criteria shall be defined to verify that the system meets the specified requirements and fulfills user expectations.

### ****2.3.8 SDLC Model to be Used****

For the development of the Task Management System, the chosen Software Development Life Cycle (SDLC) model is the iterative and incremental development approach. This model emphasizes the iterative nature of software development, where the project progresses through repeated cycles of planning, development, testing, and deployment.

**Iterative Development:** In the iterative approach, the project is divided into small increments or iterations, with each iteration delivering a working subset of the final system's features. Each iteration typically lasts for a fixed time period, known as a sprint, during which the development team focuses on implementing and testing specific functionalities.

**Incremental Development:** In incremental development, the system is built incrementally, with new features and functionalities added in successive increments. Each increment builds upon the previous one, gradually enhancing the system's capabilities. This approach allows for early delivery of key features and provides stakeholders with tangible results at regular intervals.



## Chapter 3. System Design

### 3.1 ****Design Approach (Object-oriented)****

### The chosen design approach for the project is object-oriented. Object-oriented design (OOD) emphasizes the modeling of real-world entities as objects, each with its own properties, behaviors, and interactions. This approach promotes modularity, reusability, and maintainability by encapsulating related functionality within objects and defining clear interfaces between them.

### 3.2 ****System Design using various Structured Analysis and Design Tools****

### 3.2.1 DFD's (Data Flow Diagrams)

### Zero Level Data Flow Diagram (0 Level DFD) Of Project Management System:

### This is the Zero Level DFD of Project Management System, where we have elaborated the high-level process of Project Management. It’s a basic

### overview of the whole Project Management System or process being analyzed or modeled. It’s designed to be an at-a-glance view of Bug, Ticket and

### Timesheet showing the system as a single high-level process, with its relationship to external entities of Employee, Employee Salary, and Project. It

### should be easily understood by a wide audience, including Employee, Project and Bug In zero level DFD of Project Management System, we have.

### Describe the high-level flow of the Project Management system.

### High Level Entities and processes ow of Project Management System:

### Managing all the Employee

### Managing all the Employee Salary

### Managing all the Project

### Managing all the Task

### Managing all the Bug

### Managing all the Ticket

### Managing all the Timesheet

## 

### Fig 3.1 Zero level DFD

### First Level Data Flow Diagram(1st Level DFD) Of Project Management System :

### First Level DFD (1st Level) of Project Management System shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flows to or from an external agent, and which together provide all of the functionality of the Project Management System as a whole. It also identifies internal data stores of Timesheet, Ticket, Bug, Task, Project that must be present in order for the Project Management system to do its job, and shows the flow of data between the various parts of Employee, Project, Ticket, Timesheet, Bug of the system. DFD Level 1 provides a more detailed breakout of pieces of the 1st level DFD. You will highlight the main functionalities of Project Management.

**Main entities and output of First Level DFD (1st Level DFD):**

### Processing Employee records and generate report of all Employee.

### Processing Employee Salary records and generate report of all Employee Salary

A diagram of a project management system

Description automatically generated

### Fig 3.2 First level DFD

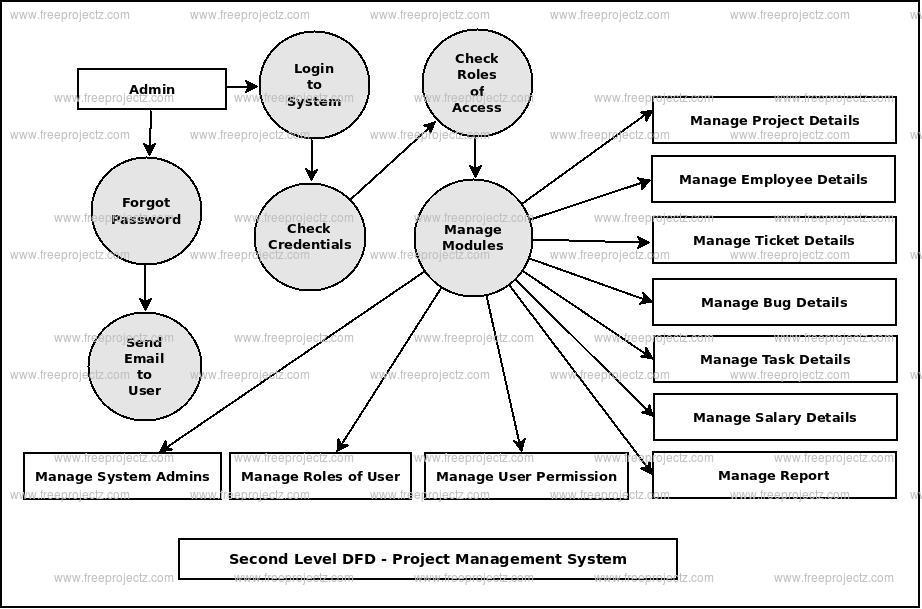
### Second Level Data Flow Diagram(2nd Level DFD) Of Project Management System :

DFD Level 2 then goes one step deeper into parts of Level 1 of Project Management. It may require more functionalities of Project Management to reach the necessary level of detail about the Project Management functioning. First Level DFD (1st Level) of Project Management System shows how the system is divided into sub-systems (processes). The 2nd Level DFD contains more details of Timesheet, Ticket, Bug, Task, Project, Employee Salary, Employee.

### Low level functionalities of Project Management System

### Admin logins to the system and manage all the functionalities of Project Management System.

### Admin can add, edit, delete and view the records of Employee, Project



### 

### Fig 3.3 Second level DFD

### 3.2.2 Data Dictionary

### Project Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Project id | int | Primary key |
| Project name | varchar (25) | Null |
| Project start date | DateTime | Null |
| Project end date | DateTime | Null |
| project\_Description | varchar(100) | Null |
| isDeleted | bit | Null |
| isActive | bit | Null |
| Employee\_id | int | Null |

### Task Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Task id | int | Primary key |
| Task name | varchar (25) | Null |
| Task start date | DateTime | Null |
| Task end date | DateTime | Null |
| task\_Description | varchar(100) | Null |
| isDeleted | bit | Null |
| isActive | bit | Null |
| Employee\_id | int | Null |

### Employee Table

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| ID | int | Primary key |
| Name | varchar (25) | Null |
| gender | varchar(10) | Null |
| EmailId | varchar(10) | Null |
| State | varchar(10) | Null |
| MobileNumber | varchar(10) | Null |
| Age | int | Null |
| Address | varchar(MAX) | Null |
| password | nvarchar(255) | Null |
| role | varchar(10) | Null |
| confirmpassword | nvarchar(255) | Null |
| comments | nvarchar(255) | Null |

### 3.2.3 UML

### Employee Task Management System Web Project using PHP and MYSQL - 1000 Projects

### Fig 3.3 UML diagram

### 3.3 ER Diagrams

### Admin and User (Chen ER Diagram) | Relationship diagram, Data modeling, Diagram

### Fig 3.4 ER Diagram showing Accessibility of Admin and User

### entity framework - ER diagram - Project, Task and Employee - Stack Overflow

### Fig 3.5 ER Diagram project, Task and Employee

### 3.3 Normalization

### Normalization is the process of organizing data in a database efficiently. This involves minimizing redundancy and dependency by dividing large tables into smaller ones and defining relationships between them. Normalization helps ensure data integrity and reduces the chances of anomalies such as insertion, update, and deletion anomalies. Common normalization forms include First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF), Boyce-Codd Normal Form (BCNF), and Fourth Normal Form (4NF).

### 3.4 Database Manipulation

Database manipulation refers to the actions performed on a database to retrieve, modify, or delete data. This includes operations such as querying data using SQL (Structured Query Language), inserting new records, updating existing records, and deleting unwanted records. Proper database manipulation techniques ensure data accuracy, consistency, and security.

### 3.5 Database Connection Controls

### Database connection controls are mechanisms used to manage connections between an application and a database. This includes establishing connections, pooling and reusing connections to improve performance, configuring connection parameters such as timeouts and maximum connections, and handling connection errors gracefully. Effective connection controls optimize resource utilization and enhance the scalability and reliability of the system.

### 3.6 Strings Methodology

Strings methodology refers to the techniques and best practices for handling strings (sequences of characters) in a programming environment. This includes operations such as string manipulation, concatenation, comparison, searching, and formatting. String manipulation functions may vary depending on the programming language or framework being used. Adhering to a consistent strings methodology helps ensure code readability, maintainability, and performance.

## Chapter 4. Implementation, Testing, and Maintenance

### Introduction to Languages, IDE’s, Tools and Technologies used for implementation

### Visual Studio, a comprehensive Integrated Development Environment (IDE) developed by Microsoft, is employed for implementing the project. Visual Studio provides a robust set of tools and features for software development, including code editing, debugging, version control, and deployment. It supports various programming languages such as C#, Visual Basic.NET, JavaScript, HTML, and CSS, making it versatile for web application development.

### Additional technologies utilized in the implementation process may include:

### ASP.NET: For building web applications and services, leveraging its extensive framework and controls.

### MS SQL Server: For database management, offering scalability, reliability, and performance.

### .NET Framework: Providing a rich set of class libraries and functionalities for application development.

### HTML, CSS, and JavaScript: For designing user interfaces and adding interactivity to web applications.

### Entity Framework: For object-relational mapping (ORM) and simplifying database interaction.

### 4.2 Coding standards of Language used:

Coding standards in Visual Studio adhere to industry best practices and guidelines specific to the chosen programming languages. These standards encompass naming conventions, code formatting, commenting practices, and code structure guidelines. Adhering to coding standards ensures consistency across the codebase, improves readability, and facilitates collaboration among developers. Visual Studio supports code analysis tools and plugins to enforce coding standards and identify potential issues during development.

### 4.3 Project Scheduling using various tools such as:

### 4.3.1 PERT

### What Is A Pert Chart And How To Use It In Project Management

### Fig 4.3.1 PERT Chart for Task Management System

### PERT (Program Evaluation and Review Technique) is a project management tool used to analyze and schedule tasks based on their estimated time for completion. Visual Studio provides integration with PERT techniques through third-party plugins or extensions. PERT charts help in visualizing project tasks, dependencies, and critical paths, aiding project managers in effective planning and resource allocation.

### 4.2 Testing Techniques and Test Plans

Testing techniques in Visual Studio encompass various methodologies to ensure the quality and reliability of the software product. These techniques include:

* Unit Testing: Testing individual components or units of code to verify their correctness.
* Integration Testing: Testing the interaction between integrated components to detect interface defects.
* System Testing: Testing the entire system as a whole to evaluate its compliance with requirements.
* Regression Testing: Testing to ensure that new changes do not adversely affect existing functionality.
* User Acceptance Testing (UAT): Testing conducted by end-users to validate that the software meets their requirements.

Test plans outline the testing approach, scope, objectives, resources, and schedules for testing activities. Visual Studio provides built-in testing frameworks such as MSTest, NUnit, and xUnit.NET for implementing and executing test cases. Additionally, Visual Studio supports test management tools for organizing test cases, executing tests, and tracking test results throughout the development lifecycle.

## Chapter 5. Results and Discussions

### 5.1 User Interface Representation

### Login Interface

### The login page offers a secure entry point to the task management system, allowing users to authenticate their credentials swiftly. Its intuitive design and streamlined interface ensure a seamless login experience. With robust encryption protocols, it prioritizes user privacy and data security.

## A screenshot of a computer Description automatically generated

### Fig 5.1 Login Interface

### Registration Interface

### The registration interface presents users with the choice to register either as a standard user or as an administrator. This dual-option approach streamlines the registration process, enabling individuals to select their desired account type based on their role and responsibilities within the system. A screenshot of a computer Description automatically generated

### Fig 5.2 Registration Interface

### Admin Interface

### The admin dashboard provides a centralized hub for managing various aspects of the task management system. Featuring a search functionality, administrators can effortlessly locate employees by entering their unique employee ID, facilitating quick access to individual profiles. Moreover, administrators can efficiently delegate tasks by forwarding them to specific employees directly from the dashboard. Admin can see Project and task list of Employees too.

### A screenshot of a computer Description automatically generated

### Fig 5.3 Admin Interface

A screenshot of a computer

Description automatically generated

### Fig 5.4 Listing all Tasks and Projects

### Forward Task Interface

### The admin dashboard streamlines task delegation with a straightforward 'Forward Task' feature, enabling quick assignment of tasks to specific employees

### A screenshot of a computer Description automatically generated

### Fig 5.5 Forward Task Interface

### Create Task Interface

### The task creation feature allows administrators to swiftly generate new tasks, streamlining project management and workflow organization.

### 

### Fig 5.6 Create Task Interface

### Create Task Interface

### The project creation feature allows administrators to swiftly generate new projects.

### A screenshot of a computer Description automatically generated

### Fig 5.7 Create Project Interface

### Delete Task and Project Interface

### The delete task feature allows administrators to swiftly delete tasks.

### 

### 

### Fig 5.8 Delete Task and Interface

### List of Tasks Interface

### The task list offers essential functionality including edit and delete options for task management. Additionally, users can provide feedback and collaborate effectively through the comment feature.

### 

### Fig 5.9 Task List Interface

### Comment on a Task Interface

### The comment feature enables users to provide feedback, exchange ideas, and collaborate seamlessly within task discussions, fostering effective communication and teamwork.

### A screenshot of a computer Description automatically generated

### Fig 6.0 Comment on Task Interface

### Edit Task Interface

### The edit task feature empowers users to modify task details and adapt to changing project requirements with ease, ensuring flexibility and agility in task management.

### A screenshot of a computer Description automatically generated

### Fig 6.1 Edit Task

### Registration Done Interface by Sending Mail

### Receiving a "Registered Successfully" email indicates that your registration was processed and confirmed by the system. This confirmation email typically contains important details such as your account information and any next steps required.

### A screenshot of a computer Description automatically generated

### Fig 6.1 Email Interface

### New Task assigned Interface

### Employees can see what task have been assigned to them.

### A screenshot of a computer Description automatically generated

### Fig 6.2 New Task Interface

### Reset Password link

### Employees can get Reset Password link via Email

### A screenshot of a computer Description automatically generated

### Fig 6.3 Reset Password Interface

Electricity Hub

### Functional Requirement User:

* In the Whole application the user performs most important role.
* User should be anyone from civilians.
* Users can perform all function related their role.
* Users have limited access on the application.

### Admin:

* Admin is the main part of our Web Application.
* Admins have access to add any needed details of all users.
* Admin will manage the users.
* Admin can update the database of the application.
* Admins have all the access in application.

### Non-Functional Requirements Security:

* + - * A separate login will be provided for the Admin, User. The real estate and inspector will not be given access to the database. All the Admin and User will have different access rights.

### Reliability:

* + - * The System must be reliable to prevent any unauthorized access.

### Availability:

* + - * Facility to keep constant track of databases, admission requests etc. will be provided.

### Compatibility:

* + - * Since the system would be built on a **.NET** platform it would be compatible with Android computable Devices.

### Interoperability:

* + - * The basic high-level design of the system will ensure interoperability as it is based on an android platform, also it will facilitate industry standards.

Page | 7

Electricity Hub

### Targeted User

The software requirements are a description of features and functionalities of the target system. Requirements convey the expectations of Targeted users from the software product. The requirements can be obvious or hidden, known or unknown, expected, or unexpected from the client’s point of view. So, these are the targeted user with their requirement.

### Admin Requirement is as follows.

* + Process Customer
  + Add Channel
    - Add Update Channel
  + Add Rate
    - Add Update Rate
  + Customer List
  + Channel List
  + Failed Customer
  + Referral Customer List
  + Add User
  + Landlord List
  + Transaction List
  + Customer Portal Account

### User Requirement is as follows.

* + Add New Connection
  + View Usage
  + View Bill
  + View Personal Details
  + Change Password

Page | 8

Electricity Hub

# System Design

Page | 9

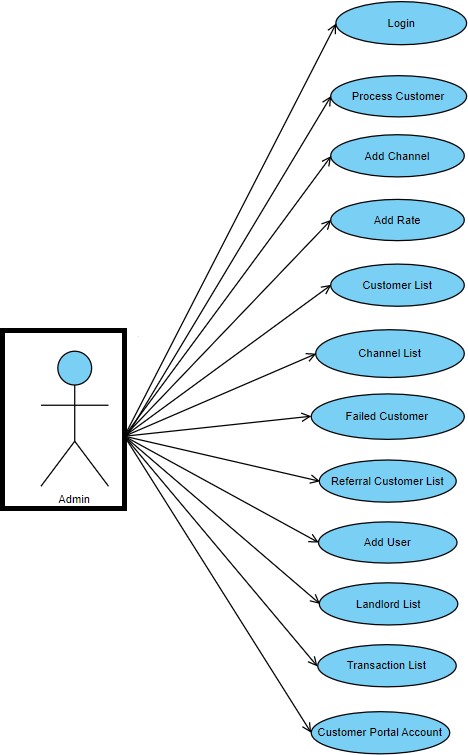
Electricity Hub

* 1. **Use Case Diagram**

1. **System Design**

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.

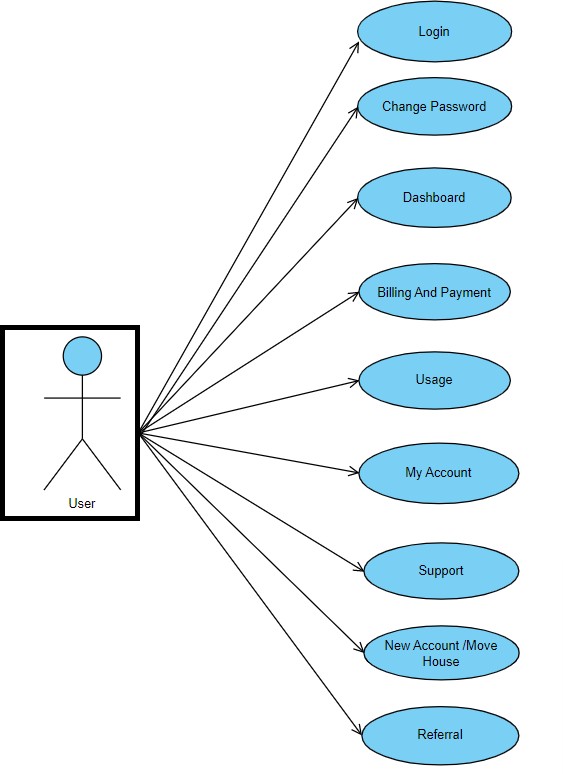
#### Use Case for Admin



Page | 10

Electricity Hub

#### 3.1.1 Use Case for User



Page | 11

Electricity Hub

* 1. **Class Diagram**

In [software engineering,](https://en.wikipedia.org/wiki/Software_engineering) a **Class Diagram** in the [Unified Modeling Language](https://en.wikipedia.org/wiki/Unified_Modeling_Language) (UML) is a type of static structure diagram that describes the structure of a system by showing the system's [classes,](https://en.wikipedia.org/wiki/Class_(computer_science)) their attributes, operations (or methods), and the relationships among objects.

The class diagram is the main building block of [object-oriented](https://en.wikipedia.org/wiki/Object-oriented_programming) modeling. It is used for general [conceptual modeling](https://en.wikipedia.org/wiki/Conceptual_model) of the systematic of the application, and for detailed modeling translating the models into [programming code](https://en.wikipedia.org/wiki/Programming_code). Class diagrams can also be used for [data](https://en.wikipedia.org/wiki/Data_modeling) [modeling.](https://en.wikipedia.org/wiki/Data_modeling) The classes in a class diagram represent both the main elements, interactions in the application, and the classes to be programmed.

|  |
| --- |
| Admin |
| -Id: Int  -User Type: String  -User Name: String  -Password: String |
| +Mange User  +Mange Electricity |

|  |
| --- |
| Electricity |
| -Id: Int  -Plan: String  -Mange Plan: String  -Add Plan: String |
| +Mange House  +Mange Rate |

|  |
| --- |
| User |
| -Id: Int  -Name: String  -Mobile: Int  -Email: String |
| +Pay Bill  +New Connection  +Power Usage |

|  |
| --- |
| Services |
| -Id: Int  -Name: String  -Payment: String  -Usage: String |
| +Supply Address  +Refer |

Page | 12

Electricity Hub

### Interaction Diagram

An interaction diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

#### Interaction Diagram for Admin

Admin

System

Database

**Log In**

**Check**

**Response**

**Verified**

**Add Chanel**

**Insert Data**

**Add Rate**

**Response**

**Response**

**Add User**

**Insert Data**

**Response**

**Response**

Page | 13

Electricity Hub

#### Interaction Diagram for User

User

System

Database

**Log In**

**Check**

**Response**

**Verified**

**Bill & Payment**

**Insert Data**

**Usage**

**Bind Data**

**Response**

**Response**

**Insert Data**

**New Connection**

**Move House**

**Insert Data**

**Response**

**Response**

Page | 14

Electricity Hub

### Activity Diagram

Activity diagrams are graphical representations of [workflows](https://en.wikipedia.org/wiki/Workflow) of stepwise activities and actions with support for choice, iteration and concurrency. In the [Unified Modeling Language,](https://en.wikipedia.org/wiki/Unified_Modeling_Language) activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

#### Activity Diagram for Admin

Login In



#### s

Condition

Process Customers

Add Channel

Add Rate Customer List

Channel List

Fail Customer

Transaction List

Name

Price Plan

Disable

Ex GST

Start Date

Inc GST

Supply

End Date

Action

Action

Log Out

Page | 15

Electricity Hub

#### Activity Diagram for User

Login In



Condition

Billing And Payment

Usage My Account Support New/Move

House

Refer

Customer Number

Customer Number

Select Plan

Supply Address

From Date

Supply Address

Add Details

To Date

Log Out

Page | 16

Electricity Hub

### Data Dictionary

#### Process Of Customer

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Customer ID | Int | Primary key |
| Customer Name | varchar (25) | Null |
| Email Address | varchar (50) | Null |
| NMI | Int | Null |
| Create Date | DateTime | Null |
| Move In Date | DateTime | Null |
| Check Credit | varchar (50) | Null |
| Concession Status | Varchar (50) | Null |
| Channel Name | Varchar (50) | Null |
| Comments | Varchar (50) | Null |
| Is Deleted | Varchar (50) | Null |

**Add Channel**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Channel ID | Int | Primary key |
| Customer Name | Varchar (50) | Null |
| Channel PartnerID | Varchar (50) | Null |
| Disabled | Varchar (50) | Null |
| Start Date | DateTime | Null |
| End Date | DateTime | Null |
| Action |  |  |

#### Add Rate

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Network Id | Int | Primary key |
| Customer\_Type | Varchar (50) | Null |
| Price Plan Name | Varchar (50) | Null |
| Tariff Code1 | Int | Null |
| Tariff Code2 | Int | Null |

Page | 17

Electricity Hub

#### Customer List

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Customer ID | Int | Primary key |
| Customer Number | Varchar (10) | Null |
| Email Address | Varchar (50) | Null |
| Customer Name | Varchar (50) | Null |
| Customer Category | Varchar (50) | Null |
| NMI | Int | Null |
| Plan Name | Varchar (50) | Null |
| Create Date | DateTime | Null |
| Move In Date | DateTime | Null |
| Channel Name | Varchar (50) | Null |
| Credit Check | Varchar (50) | Null |
| Concession Check | Varchar (50) | Null |
| Process Notes | Varchar (50) | Null |
| Onboard Status | Varchar (50) | Null |
| Provision Status | Varchar (50) | Null |
| Start Date | DateTime | Null |
| End Date | DateTime | Null |

**Transaction List**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Transaction ID | Int | Primary key |
| Payment Reference No | Int | Null |
| Statement Number | Int (50) | Null |
| Billed Amount | Int (50) | Null |
| Paid Amount | Int | Null |
| Email | Varchar (50) | Null |
| Created Date | DateTime | Null |
| Start Date | DateTime | Null |
| End Date | DateTime | Null |

#### Customer Registration

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Customer Number | Int | Primary key |
| NMI | Int | Null |
| First Name | Varchar (50) | Null |
| Last Name | Varchar (50) | Null |
| Email Address | Varchar (50) | Null |
| Mobile numbers | Varchar (50) | Null |

Page | 18

Electricity Hub

#### Billing And Payment

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Supply Address | Varchar (50) | Null |
| Select Start Date | DateTime | Null |
| Select End Date | DateTime | Null |

**Usage**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Customer Number | Varchar (50) | Null |
| Supply Address | Varchar (50) | Null |

#### My Account Personal Details

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Customer Number | Int | Primary key |
| Contact Type | Int | Null |
| First Name | Varchar (50) | Null |
| Last Name | Varchar (50) | Null |
| Email Address | Varchar (50) | Null |
| Mobile numbers | Varchar (50) | Null |

**Authorized Personal Details**

|  |  |  |
| --- | --- | --- |
| **Column Name** | **Data Type** | **Constraints** |
| Title | Varchar (50) | Primary key |
| First Name | Varchar (50) | Null |
| Last Name | Varchar (50) | Null |
| Phone Number | Int (10) | Null |
| Date Of Birth | DateTime | Null |
| Email | Varchar (50) | Null |
| Contact Adress | Varchar (50) | Null |
| Contact Adress 2 | Varchar (50) | Null |
| City | Varchar (50) | Null |
| Contact State | Varchar (50) | Null |
| Contact Postal Date | Varchar (50) | Null |

Page | 19

Electricity Hub

# Development

Page | 20

Electricity Hub

## Development

Software development is the process of conceiving, specifying, designing, programming, documenting, testing, and bug fixing involved in creating and maintaining applications, frameworks, or other software components. Software development is a process of writing and keeping the source code, but in a broader sense, it includes all that is involved between the conception of the desired software through to the final manifestation of the software, sometimes in a planned and structured process. Therefore, software development may include research, new development, prototyping, modification, reuse, re- engineering, maintenance, or any other activities that result in software products.

## Coding Standards

Writing efficient software code requires a thorough knowledge of programming. This knowledge can be implemented by following a coding style which includes several guidelines that help in writing the software code efficiently and with minimum errors. These guidelines, known as coding guidelines, are used to implement individual programming language constructs, comments, formatting, and so on. These guidelines, if followed, help in preventing errors, controlling the complexity of the program, and increasing the readability and understandability of the program.

A set of comprehensive coding guidelines encompasses all aspects of code development. To ensure that all developers work in a harmonized manner (the source code should reflect a harmonized style as a single developer had written the entire code in one session), the developers should be aware of the coding guidelines before starting a software project. Moreover, coding guidelines should state how to deal with the existing code when the software incorporates it or when maintenance is performed.

Since there are numerous programming languages for writing software codes, each having different features and capabilities, coding style guidelines differ from one language to another. However, there are some basic guidelines which are followed in all programming Languages. These include naming conventions, commenting conventions, and formatting conventions.

### ASP.NET

ASP.NET is a web development platform, which provides a programming model, a comprehensive software infrastructure and various services required to build robust web applications for PC, as well as mobile devices.

ASP.NET works on top of the HTTP protocol and uses the HTTP commands and policies to set a browser-to-server bilateral communication and cooperation.

ASP.NET is a part of the Microsoft .Net platform. ASP.NET applications are compiled codes, written using the extensible and reusable components or objects present in .Net framework. These codes can use the entire hierarchy of classes in .Net framework.

Page | 21

Electricity Hub

* + - * The ASP.NET application codes can be written in any of the following languages:
        + C#
        + Visual Basic.Net
        + HTML (Hyper Text Markup Language)
        + Java script
        + CSS (Cascading Style Sheets)

ASP.NET is used to produce interactive, data-driven web applications over the internet. It consists of a large number of controls such as text boxes, buttons, and labels for assembling, configuring, and manipulating code to create HTML pages.

### ASP.NET Overview

Here are some points that give a quick overview of ASP.NET.

* + ASP.NET provides services to allow the creation, deployment, and execution of Web Applications and Web Services
  + Like ASP, ASP.NET is a server-side technology.
  + Web Applications are built using Web Forms. ASP.NET comes with built-in Web Forms controls, which are responsible for generating the user interface. They mirror typical HTML widgets like text boxes or buttons. If these controls do not fit your needs, you are free to create your own user controls.
  + Web Forms are designed to make building web-based applications as easy as building Visual Basic applications.

### Advantages of ASP.NET

1. **Separation of Code from HTML:**

To make a clean sweep, with ASP.NET you can completely separate layout and business logic. This makes it much easier for teams of programmers and designers to collaborate efficiently.

### Support for compiled languages:

Developers can use VB.NET and access features such as strong typing and object-oriented programming. Using compiled languages also means that ASP.NET pages do not suffer the performance penalties associated with interpreted code. ASP.NET pages are precompiled to **bytecode** and **Just In Time (JIT)** compiled when first requested. Subsequent requests are directed to the fully compiled code, which is cached until the source changes.

Page | 22

Electricity Hub

### Use services provided by the .NET Framework:

The .NET Framework provides class libraries that can be used by your application. Some of the key classes help you with input/output, access to operating system services, data access, or even debugging. We will go into more detail on some of them in this module.

### Graphical Development Environment:

Visual Studio .NET provides a rich development environment for web developers. You can drag and drop controls and set properties the way you do in Visual Basic 6. And you have full IntelliSense support, not only for your code, but also for HTML and XML.

### State management:

To refer to the problems mentioned before, ASP.NET provides solutions for session and application state management. State information can, for example, be kept in memory or stored in a database. It can be shared across web farms, and state information can be recovered, even if the server fails or the connection breaks down.

### Update files while the server is running:

Components of your application can be updated while the server is online and clients are connected. The Framework will use the new files as soon as they are copied to the application. Removed or old files that are still in use are kept in memory until the clients have finished.

### XML-Based Configuration Files:

Configuration settings in ASP.NET are stored in XML files that you can easily read and edit. You can also easily copy these to another server, along with the other files that comprise your application.

Page | 23

Electricity Hub

### Telerik Rad Controls

Telerik Rad Controls for ASP.NET is the preferred toolset for professional web development. From the most advanced HTML editor and fastest AJAX data grid to SEO- optimized navigation controls, the Telerik products allow you to build highly rich and responsive applications. Customers often recommend Rad Controls for their superior performance, optimized HTML output, comprehensive design-time experience, and native ASP.NET AJAX support. Always hand-in-hand with Microsoft releases, Telerik is traditionally the first component vendor to provide compatibility with emerging browsers, standards and updates of Visual Studio and the .NET Framework.

Telerik controls are supported through a wide range of resources: demos, online documentation, very active forums, Knowledge Base articles, code library, sample applications. Version Q1 2007 include a Manager for Microsoft ASP.NET AJAX. Inspired by the design-time capabilities of Telerik Ajax, the new control will offer completely codeless development experience, but this time with Microsoft ASP.NET AJAX framework. Introduced are also a new set of widgets built on top of ASP.NET AJAX (color-picker, screen tip/tooltip, slider/scroller, spinner, split button, etc.). In addition, new major versions of Rad Input, Rad Chart and Rad Dock are presented.

### Advantages:

1. By using this control our lots of coding is reduced. For ex: - Telerik Grid automatically handled Paging, Filter, Sorting...etc. So, we can have reduced development time.
2. Using Telerik style builder, we can easily create our custom theme to match our site theme.
3. Any .NET developer gets good hand on this control in very short time.
4. We can get easily support from Telerik team, Telerik MVP and Telerik users.
5. In trail demo all the features is available, so we can implement this controls in our page and check the how it looks and match our requirement.
6. Demos and documents available live. If the internet is not available in our system, then we can also install this demo in our system and we can check it offline.
7. The number of controls is very high so after buying these controls we do not need to buy any other controls.
8. In-addition its controls also provide client-side events and API.

Page | 24

Electricity Hub

### MS SQL

* MS SQL is used for web application because it is designed to scalable and to perform well in multi-user environment.
* It is reliable, platform compatible and easy to use. It provides Excellent Database platform for
  + Large-scale online transaction processing.
  + Data warehousing, and e-commerce Application.
* MS SQL is a database system used on the web.
* MS SQL is a database system that runs on a server.
* MS SQL is ideal for both small and large applications.
* MS SQL is very fast, reliable, and easy to use.
* MS SQL uses standard SQL.
* MS SQL compiles on a number of platforms.
* MS SQL is free to download and use.
* MS SQL is developed, distributed, and supported by Microsoft Corporation.

The data in a MS SQL database are stored in tables. A table is a collection of related data, and it consists of columns and rows.

Page | 25

Electricity Hub

## Screen Shots

**Admin Side**

#### Admin Login

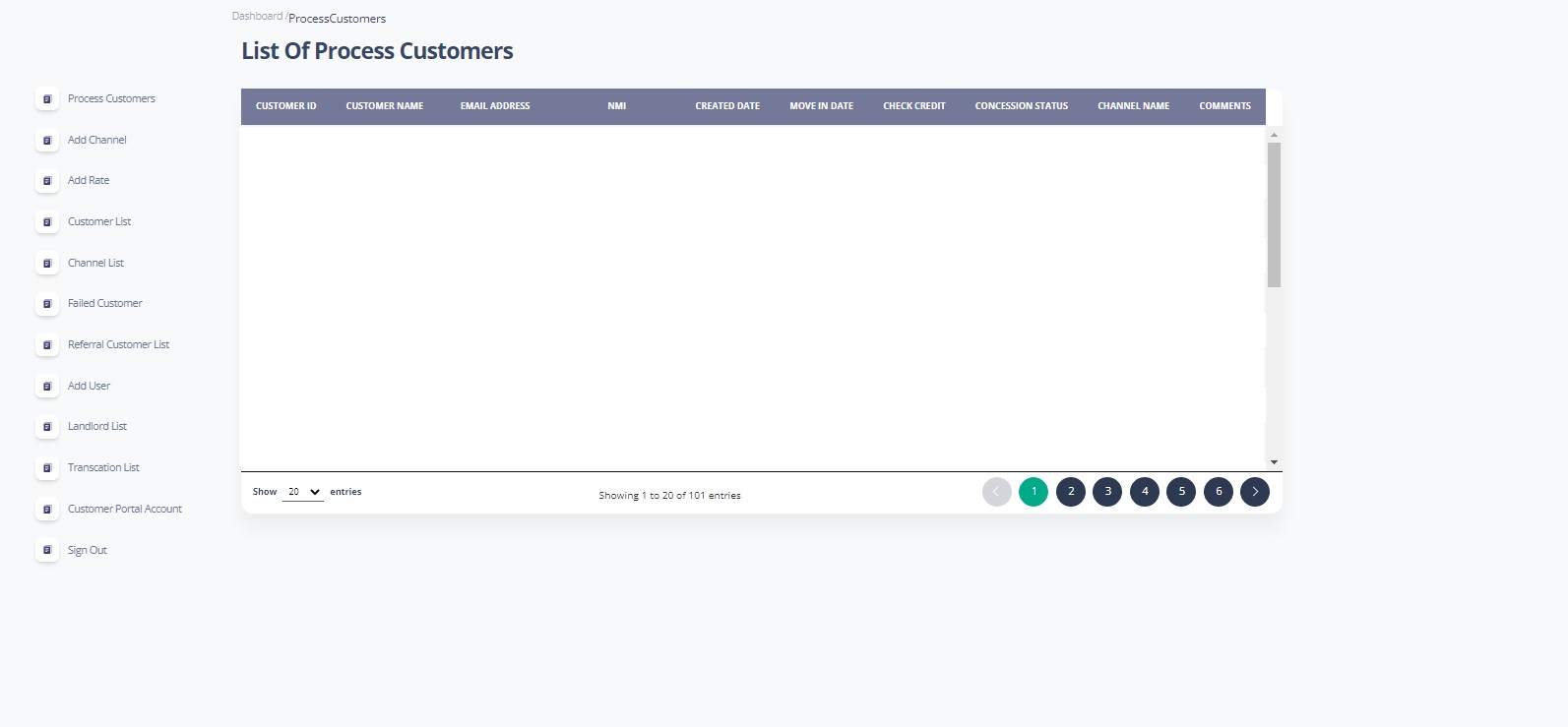


* **Admin can login by using this.**

Page | 26

Electricity Hub

#### List Of Process Customer

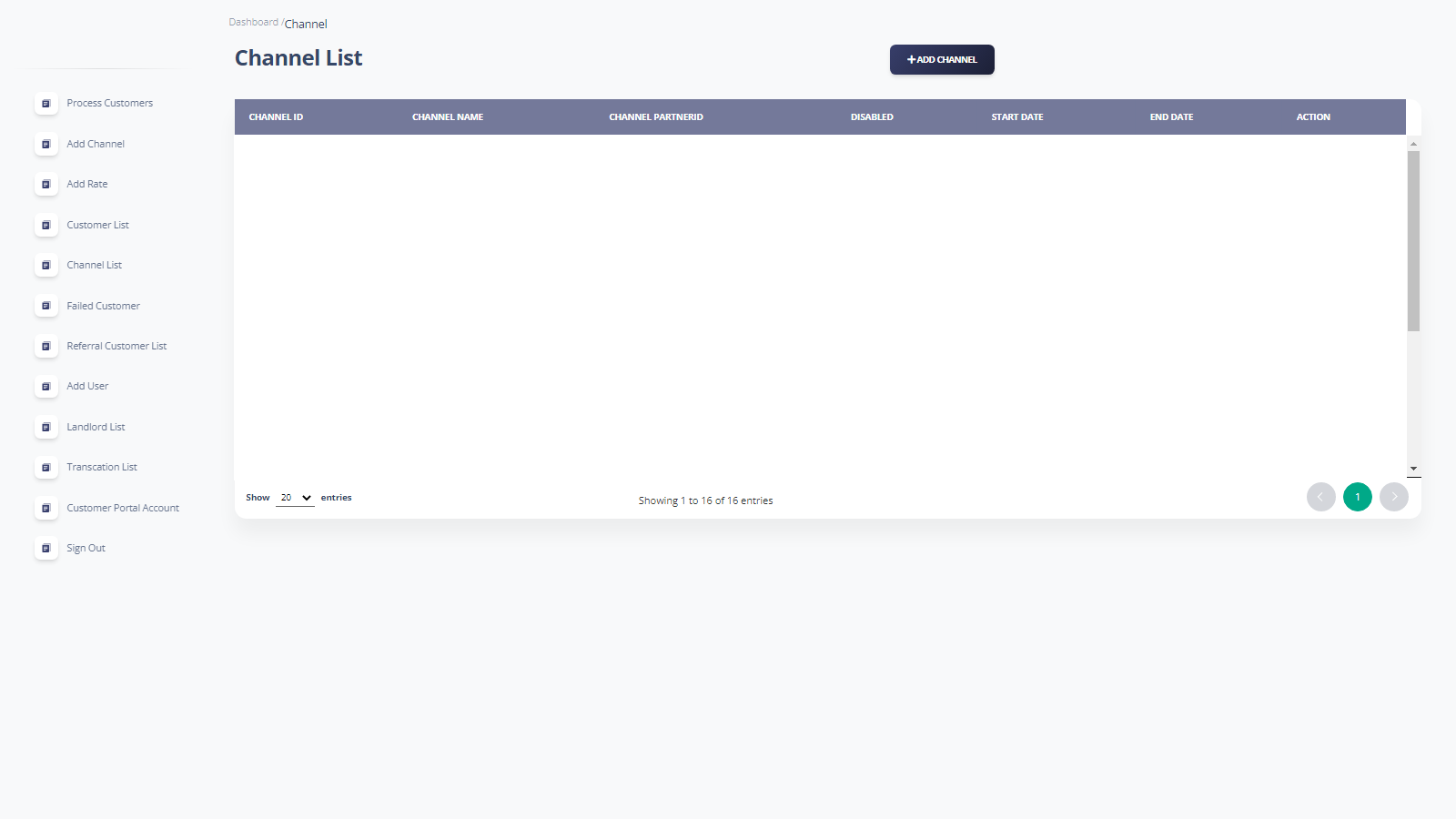


* **This Module shows all the information of Customers to the Admin.**
* **Admin can Add and Update all data in this page.**

Page | 27

Electricity Hub

#### Channel List



* **This Module shows list of all channels.**

#### Admin can view all the channel list which have user selected.

Page | 28

Electricity Hub

#### Add Update Channel



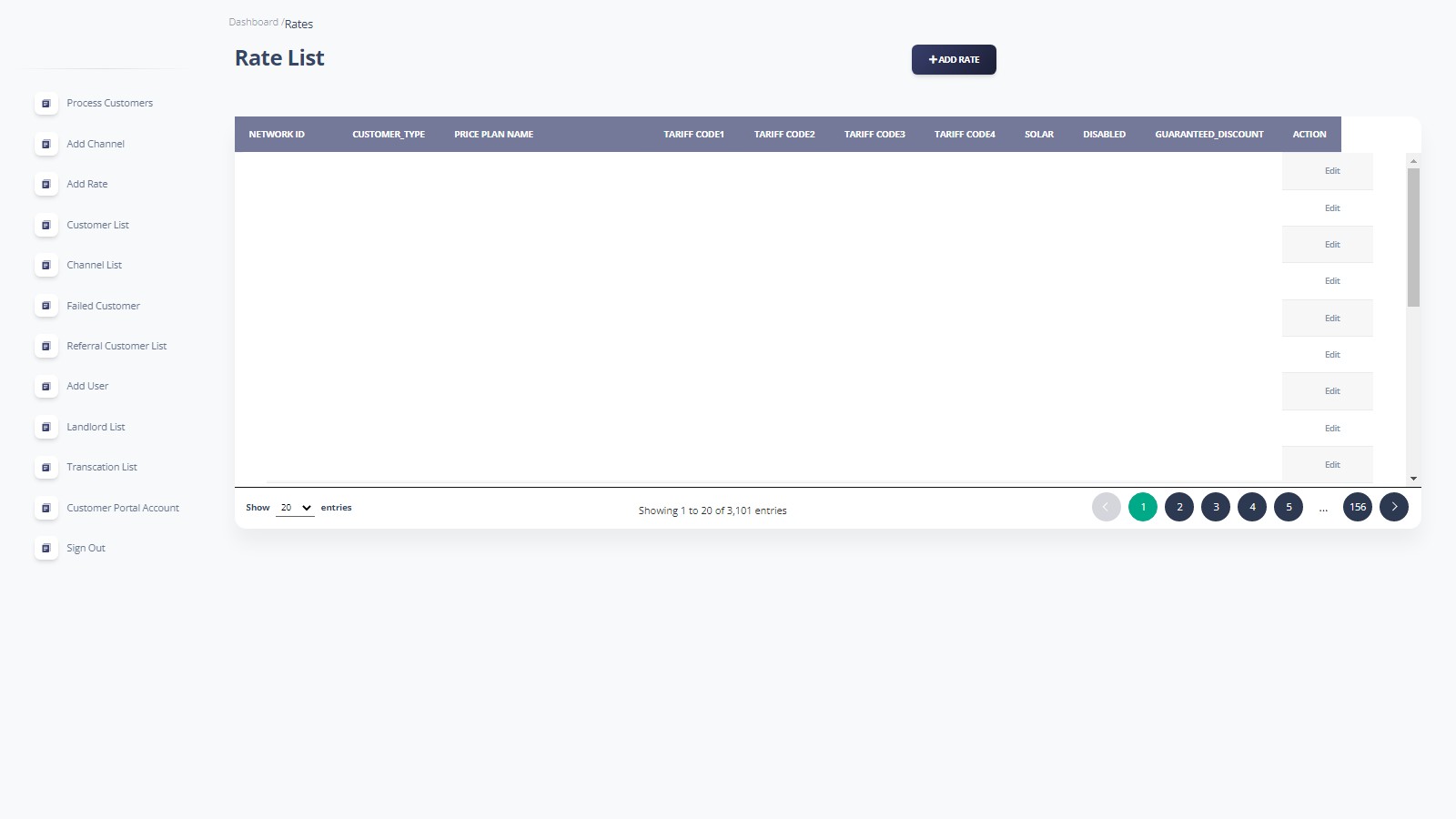
* **Admin can Add and Update any channel by using this Module.**

#### Admin also can remove any channel.

Page | 29

Electricity Hub

#### Rate List

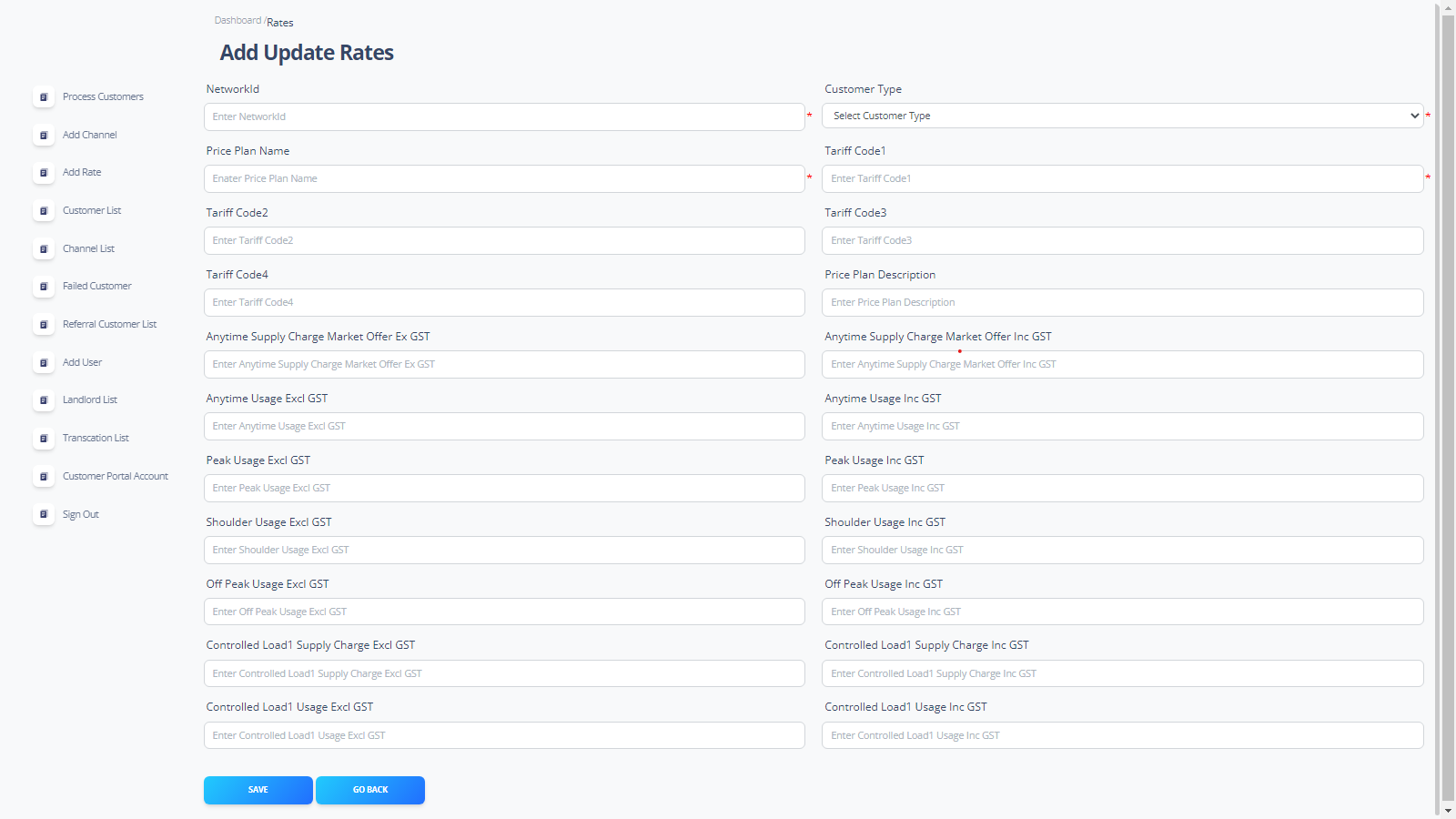


* **This Module describes information/list about rates of electricity plans.**

Page | 30

Electricity Hub

#### Add Update Rates

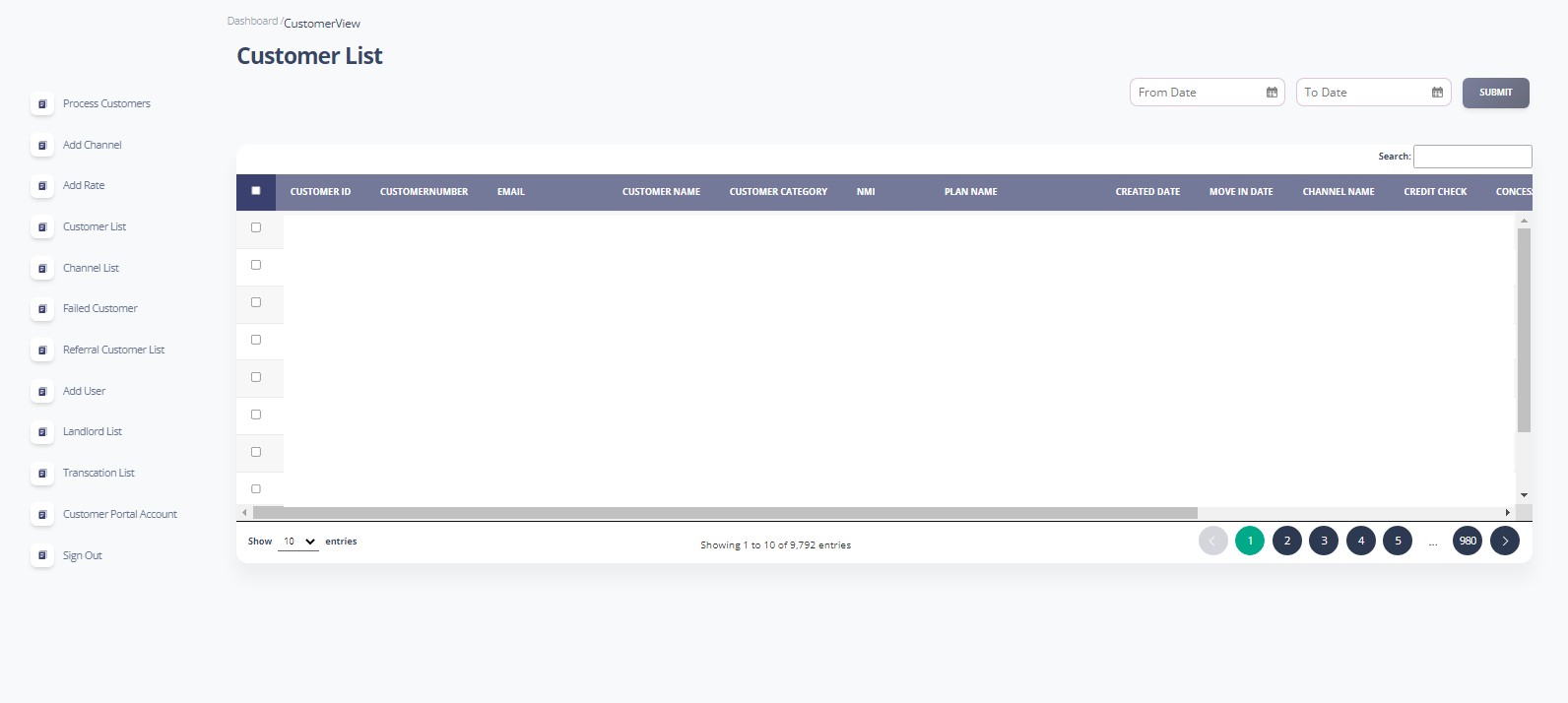


* **Admin Add and Update list of rates in this Module.**

Page | 31

Electricity Hub

#### Customer List



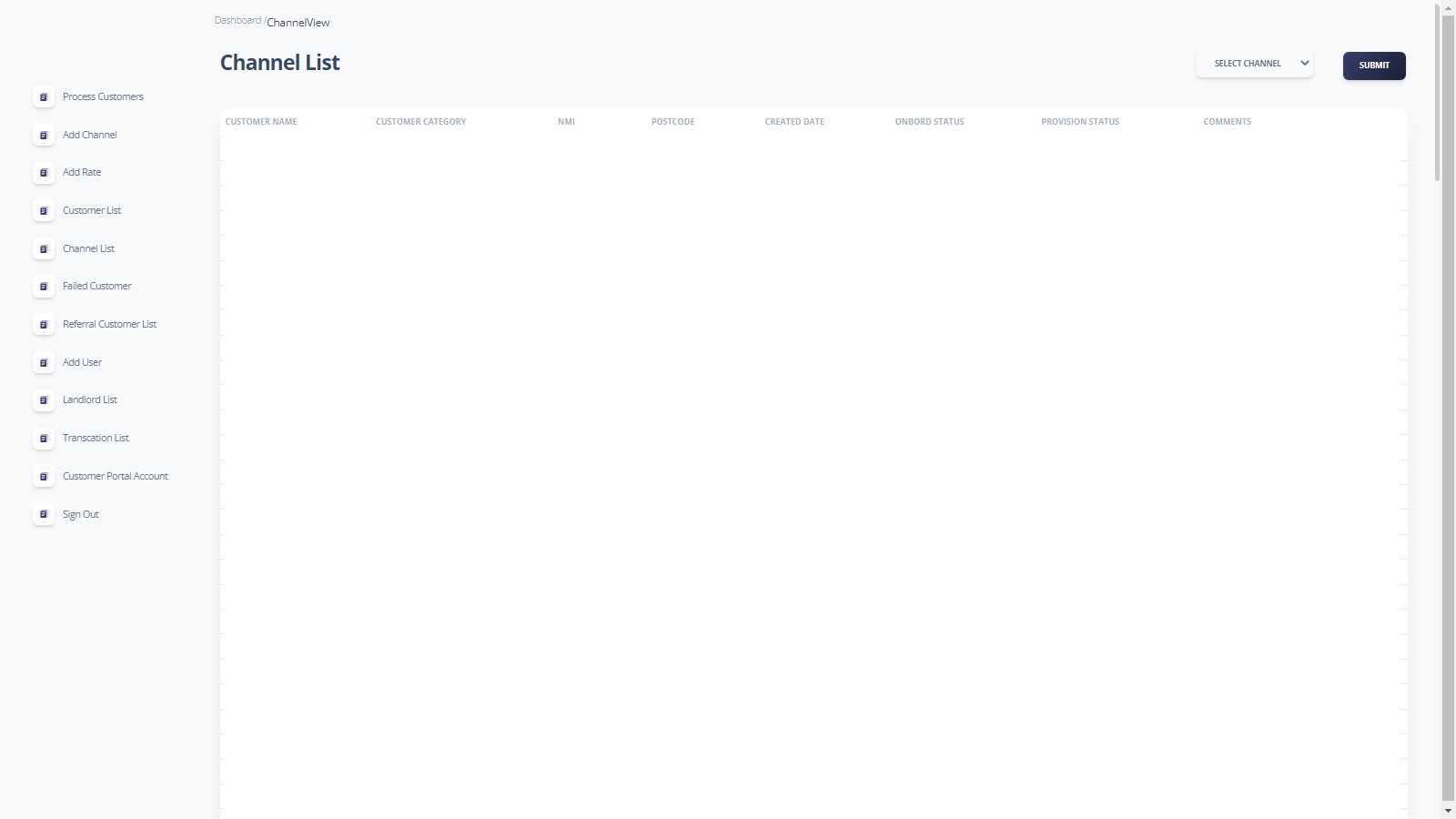
* **This Module shows the list of all customers / Users.**

#### Admin can Add and Update customer list.

Page | 32

Electricity Hub

#### Channel List



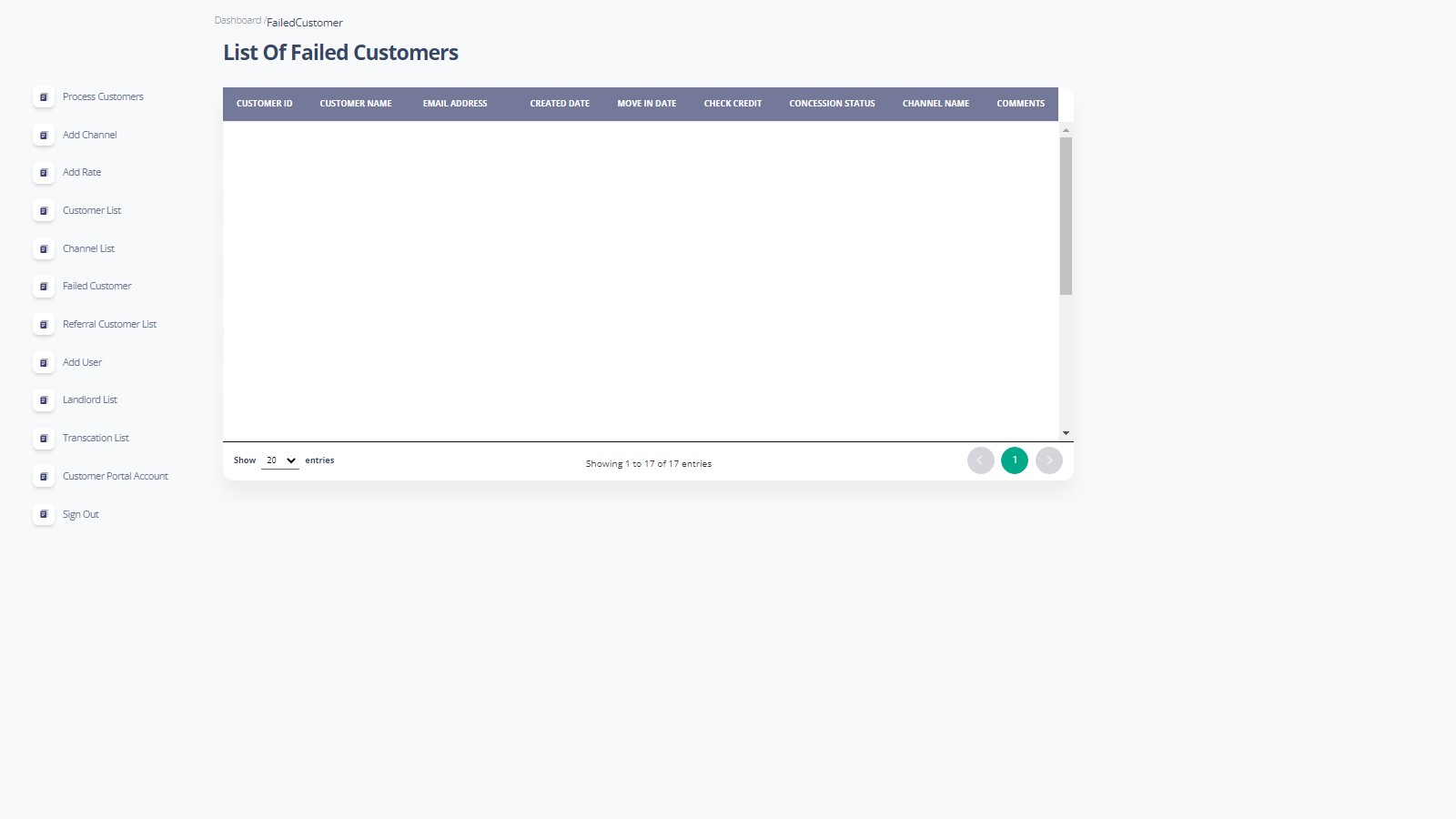
* **Admin can select any channel by using this channel Module.**

#### Admin can apply channel to customer’s account.

Page | 33

Electricity Hub

#### List Of Failed Customer

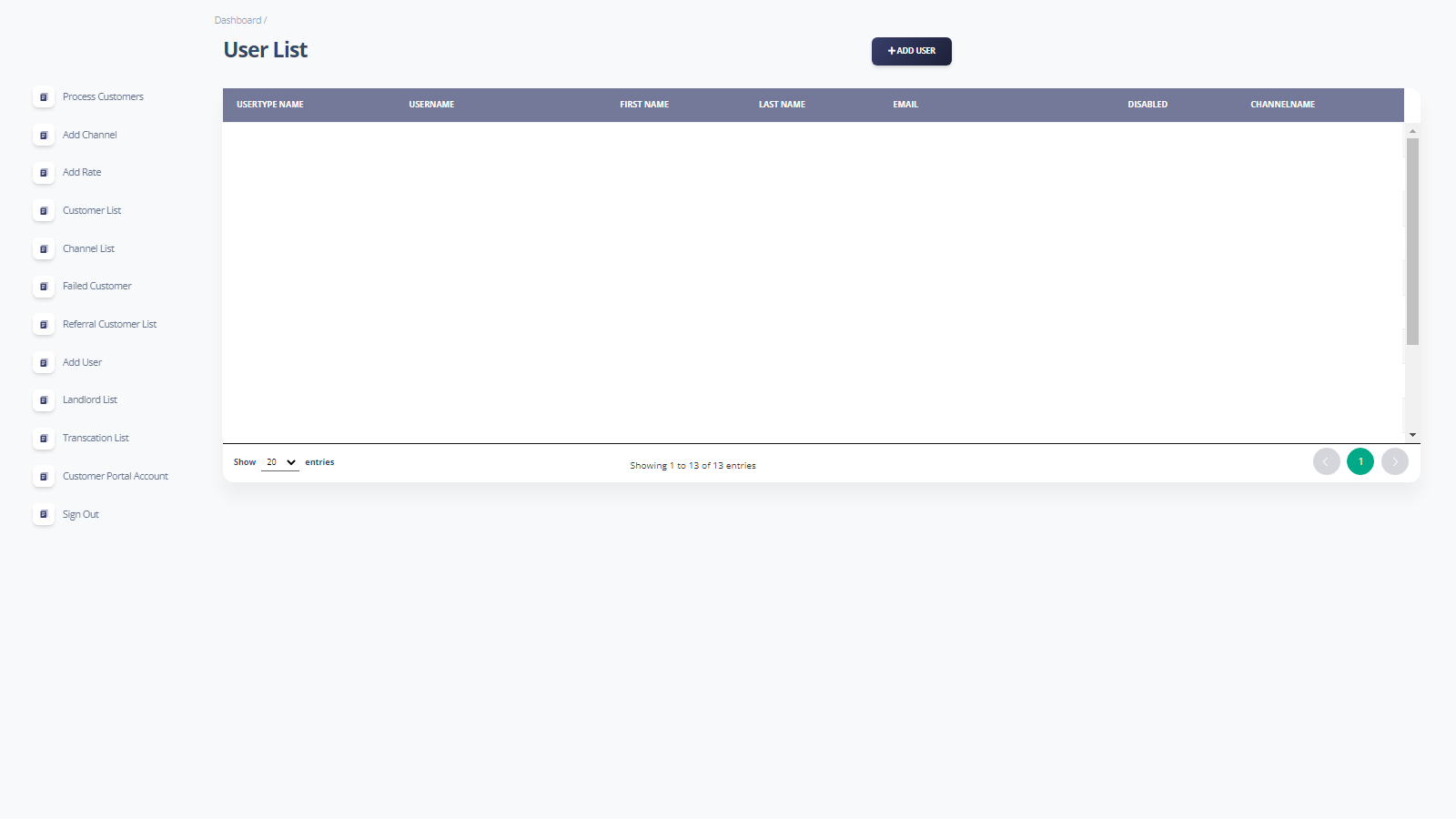


* **This Module is created to show the list of customers which have been failed in changing connection and which have applied for new connection.**

Page | 34

Electricity Hub

#### User List



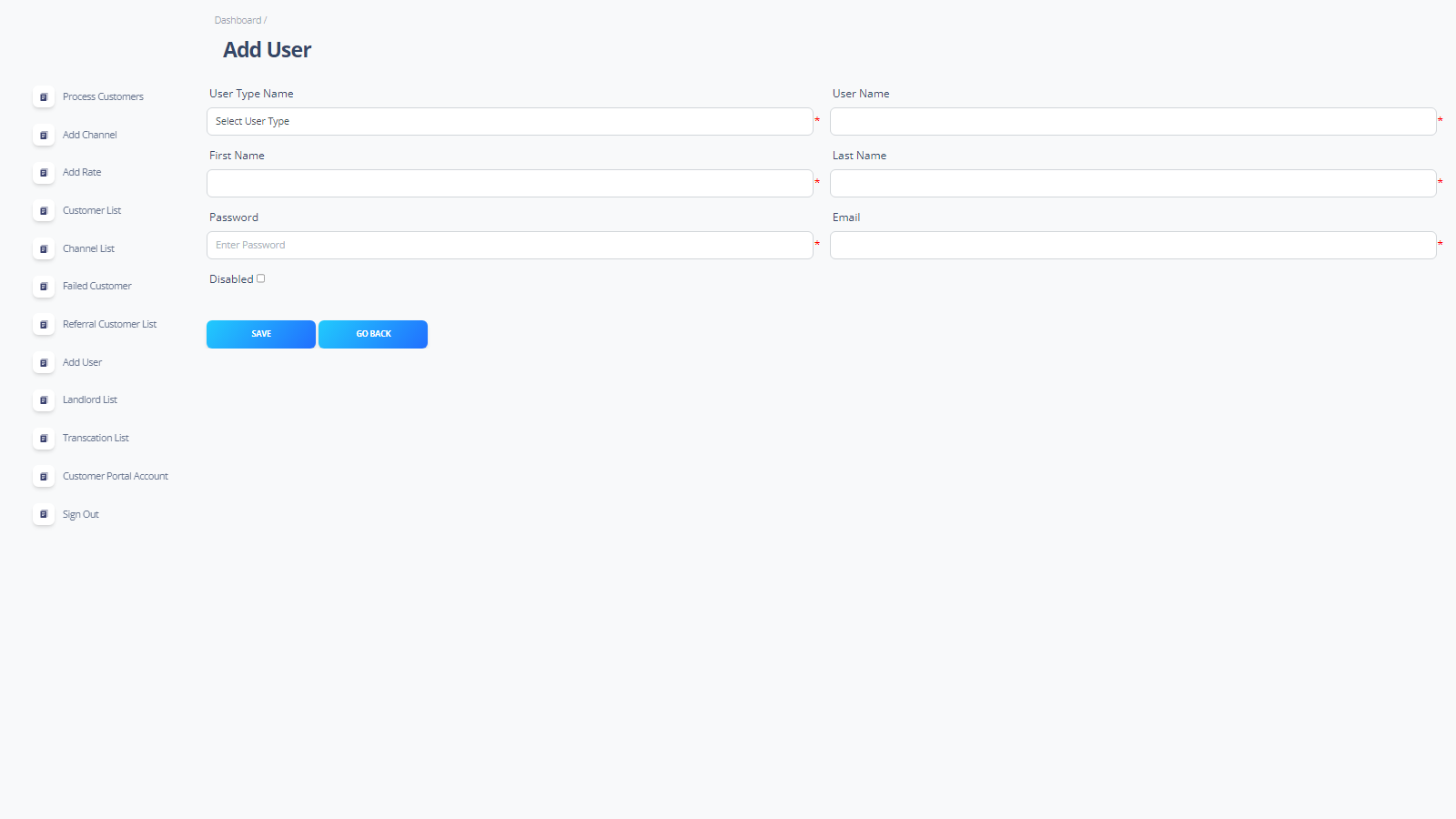
* **This Module shows the information of users like their plans and their channel.**

#### Admin can add user by this using Add user button.

Page | 35

Electricity Hub

#### Add User

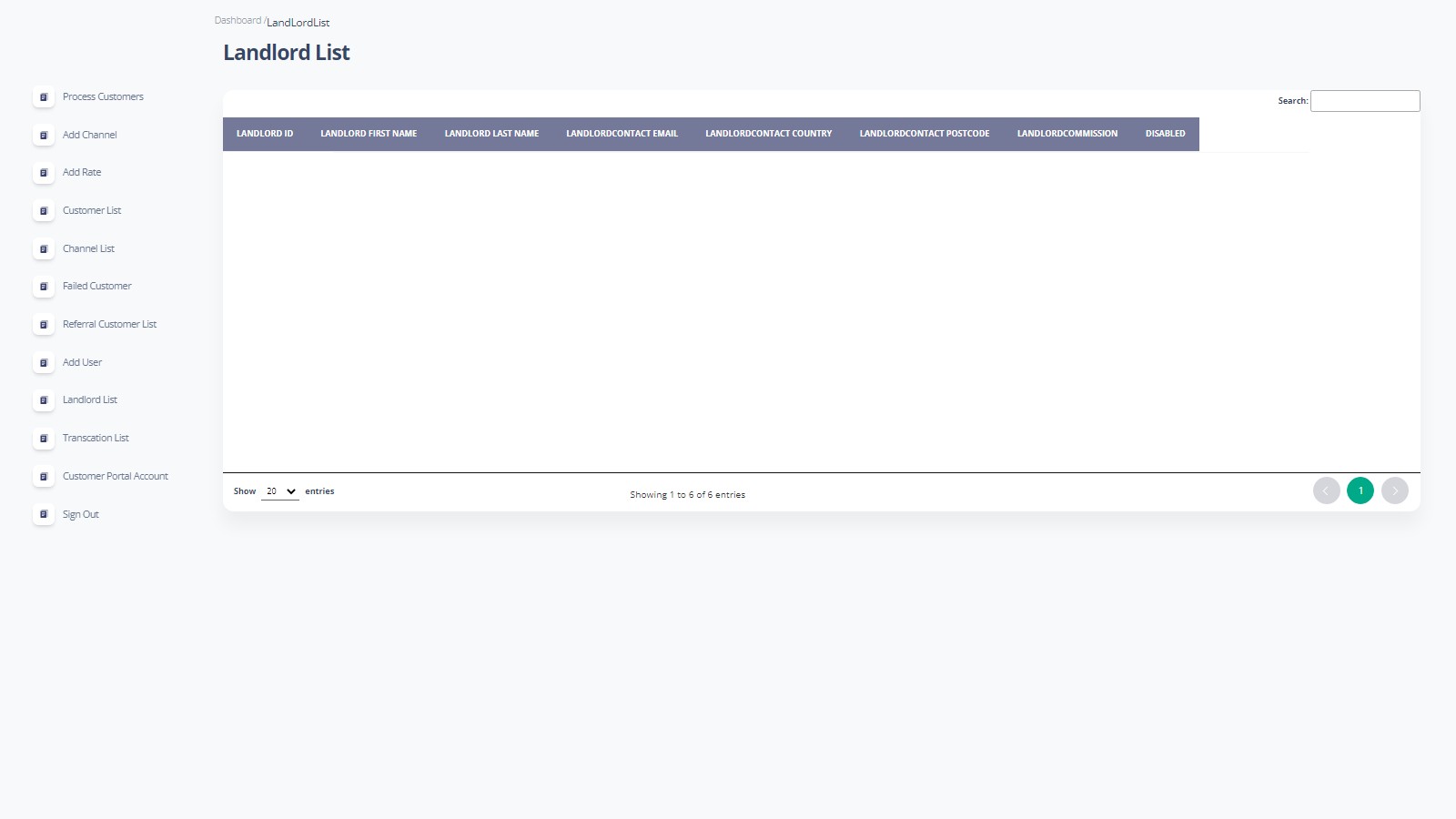


* **This Module is created to Add User in User List.**

Page | 36

Electricity Hub

#### Landlord List

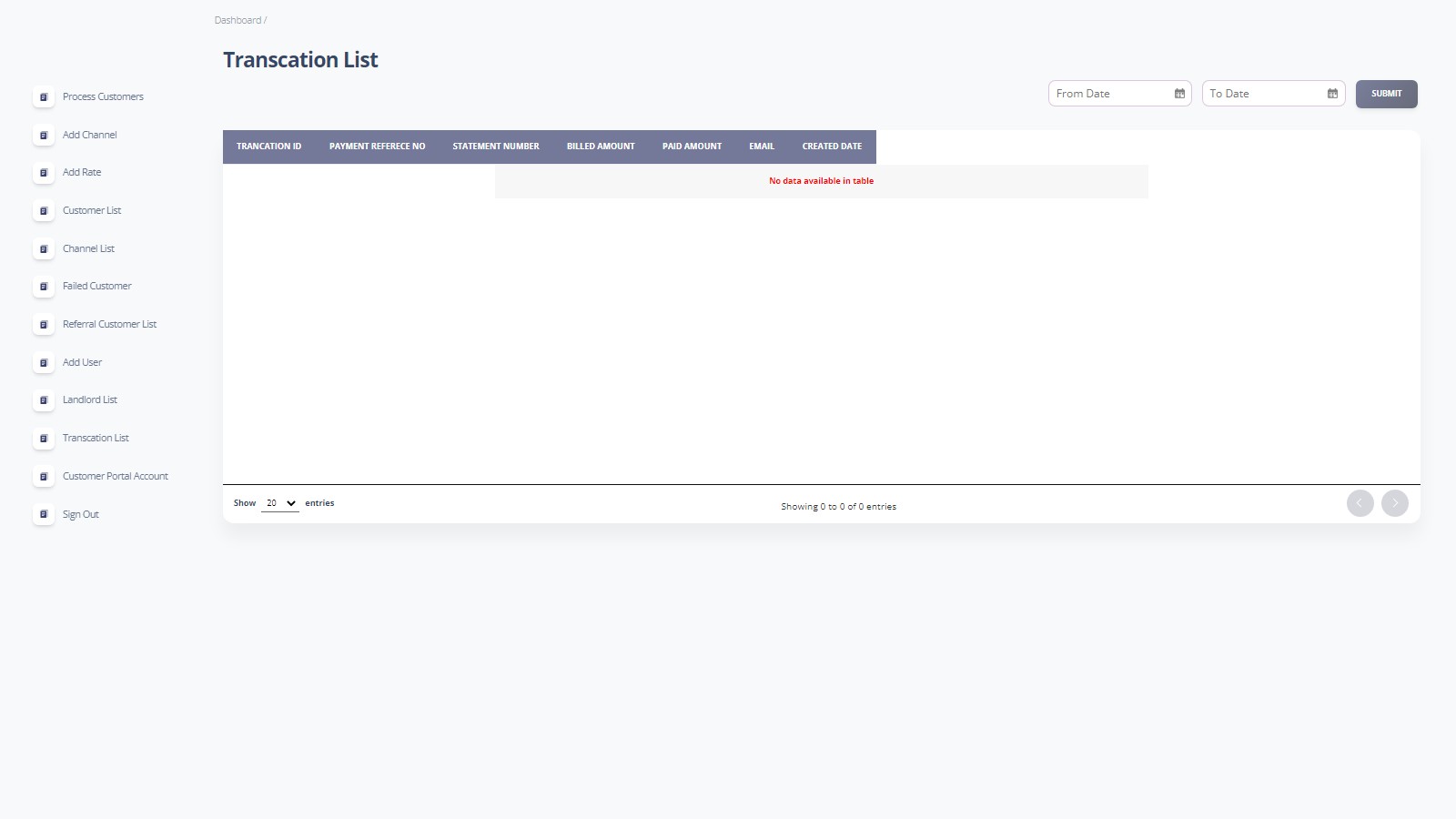


* **This module describes the information or list of landlords.**

Page | 37

Electricity Hub

#### Transaction List

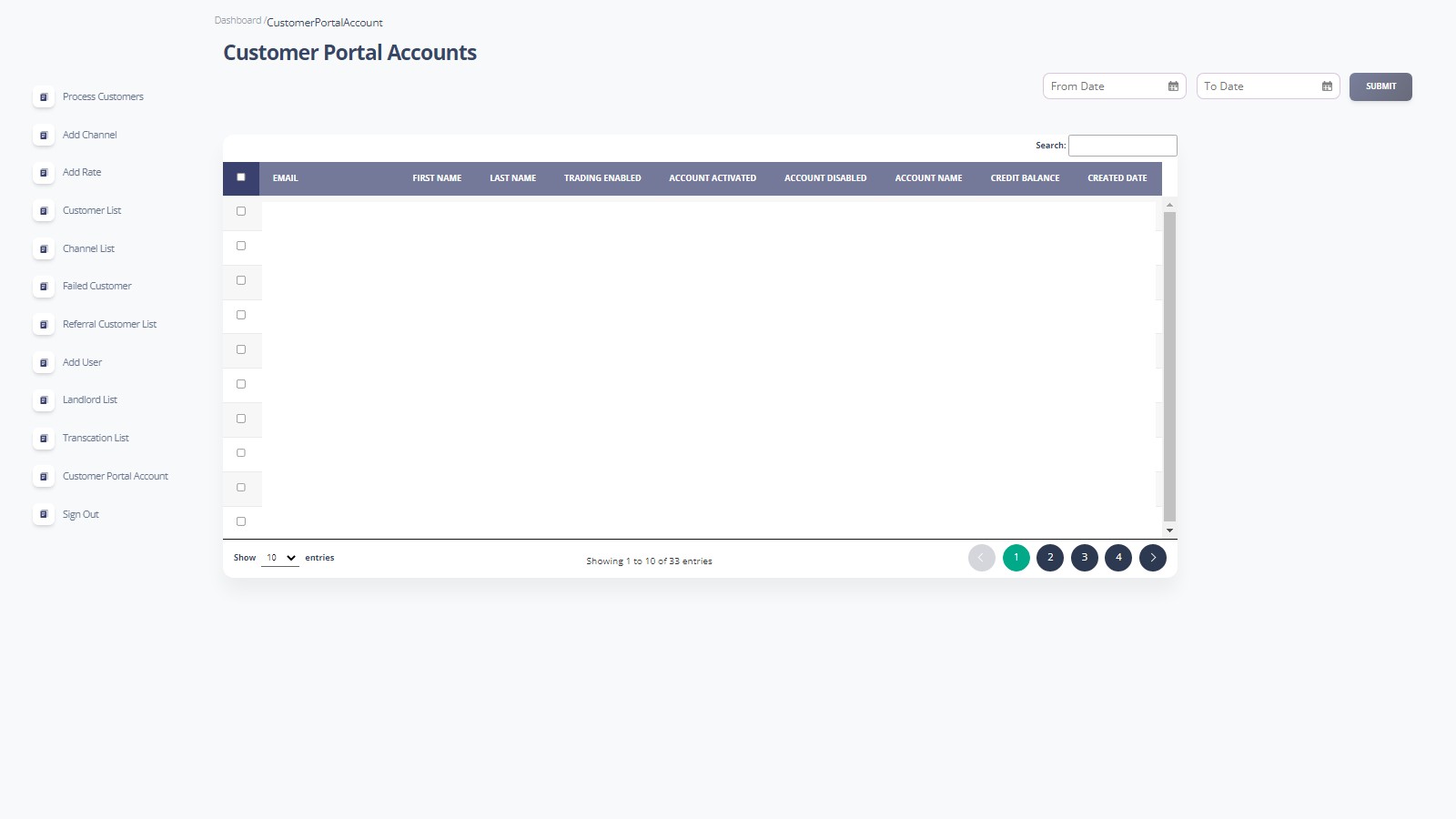


* **This module is created to describe the information or list of Transaction History.**

Page | 38

Electricity Hub

#### Customer Portal Accounts

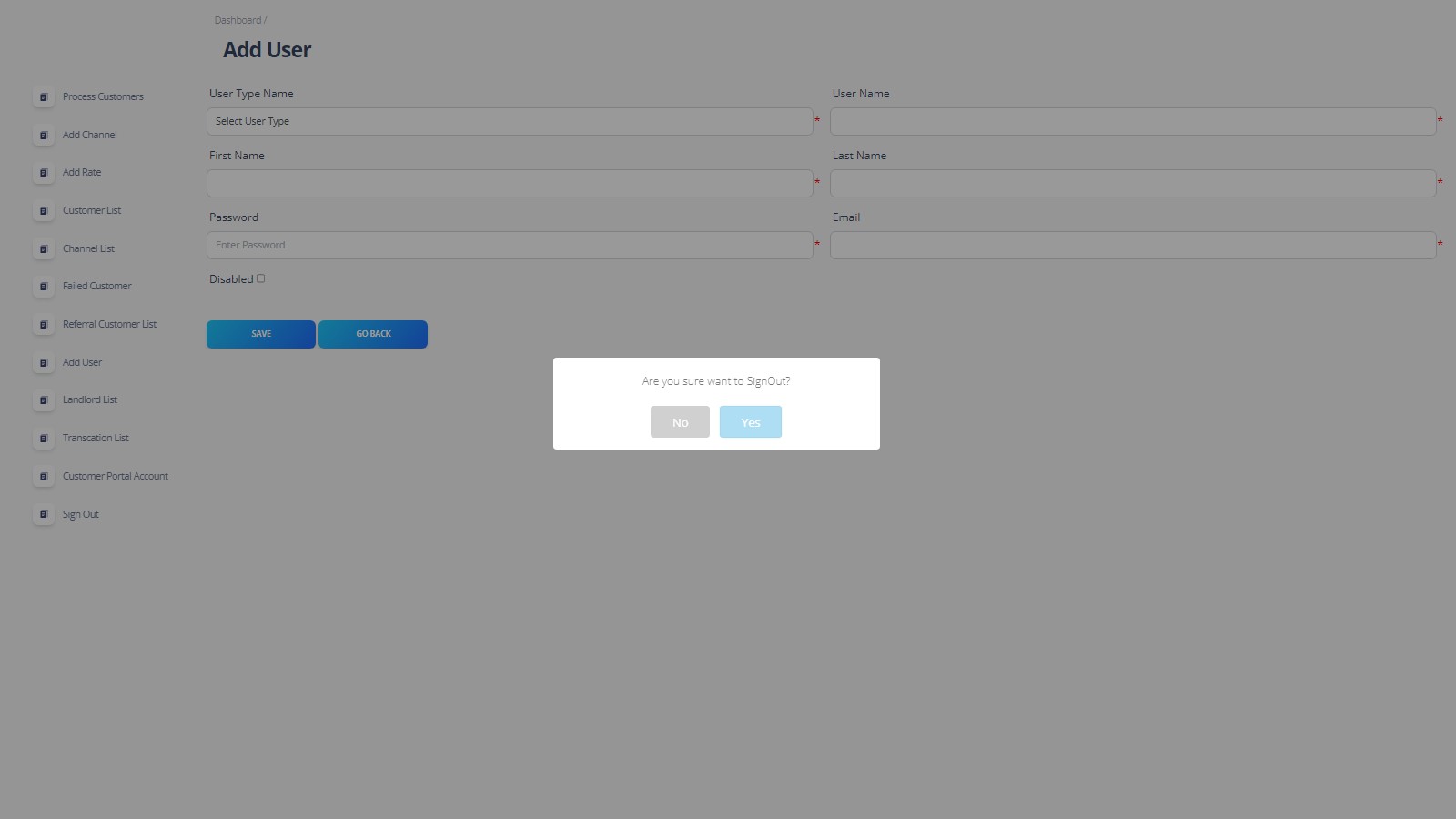


* **In this module admin can see the information about customer portal accounts.**

Page | 39

Electricity Hub

#### Sign Out



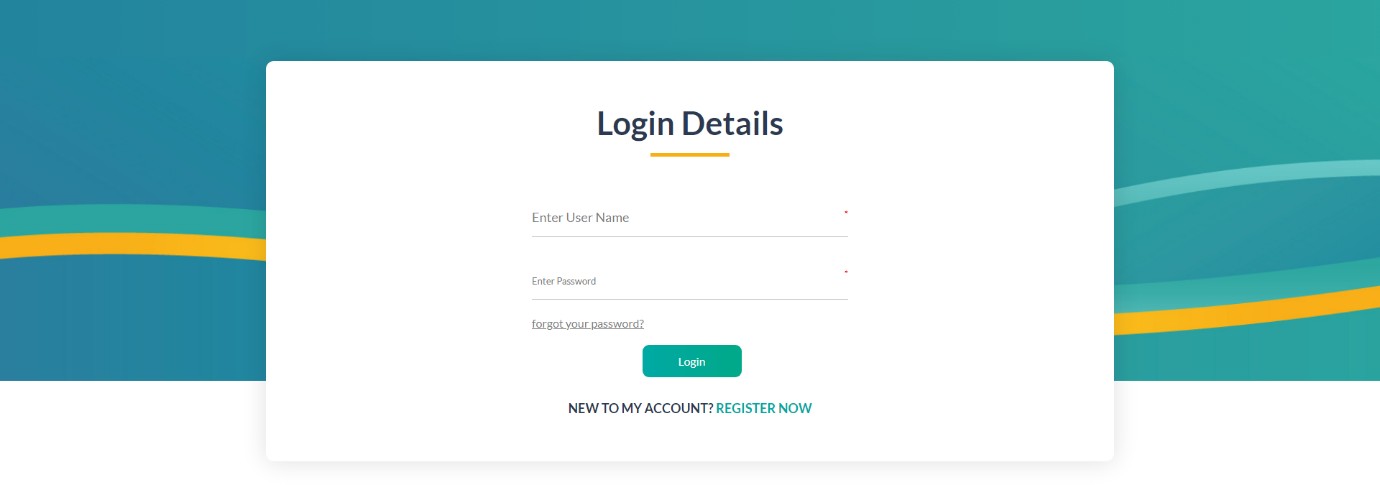
* **This is the last module of admin side, In this module admin can sign out from their current Logged In account.**

Page | 40

Electricity Hub

**User Side**

#### User Login

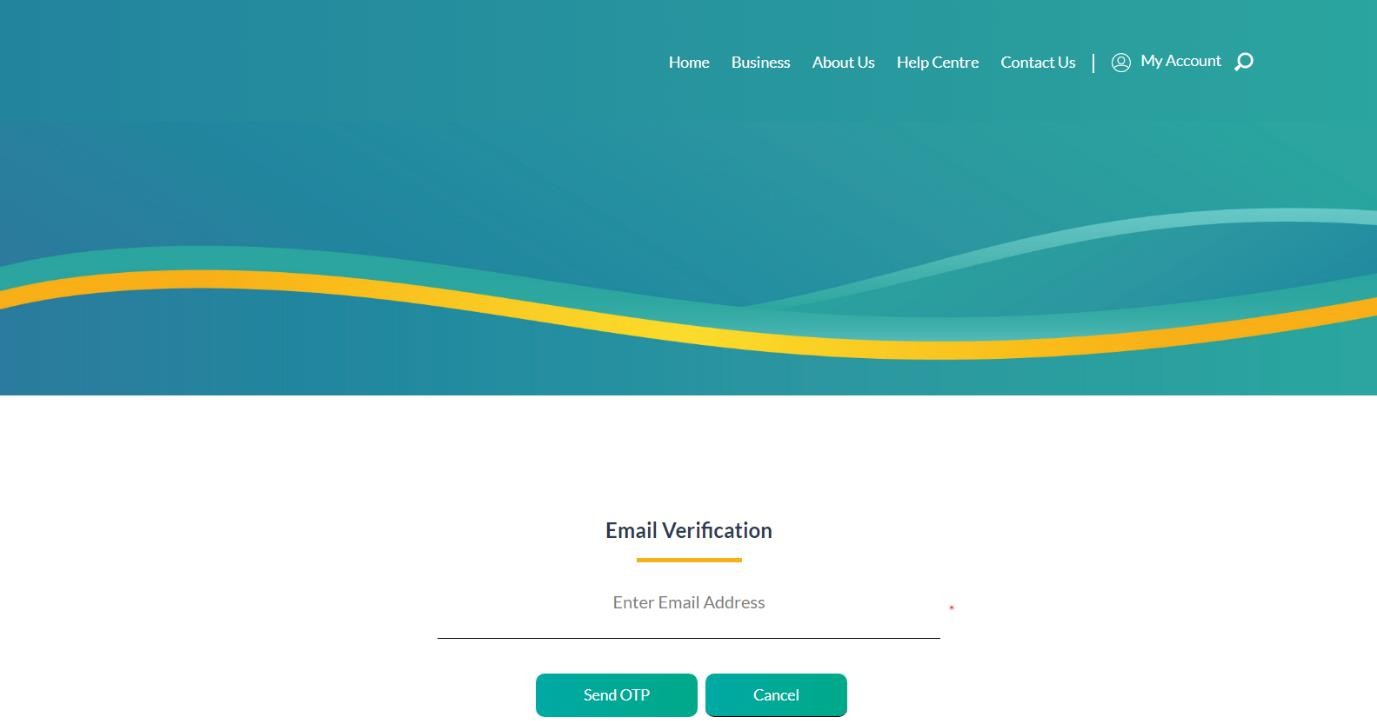


* **This is the first page or module where User can log in by entering required details or they can create their new account.**

Page | 41

Electricity Hub

#### Email Verification



* **This is the second step of log in module where user have to verified by entering their email address.**

Page | 42

Electricity Hub

#### Verification Code

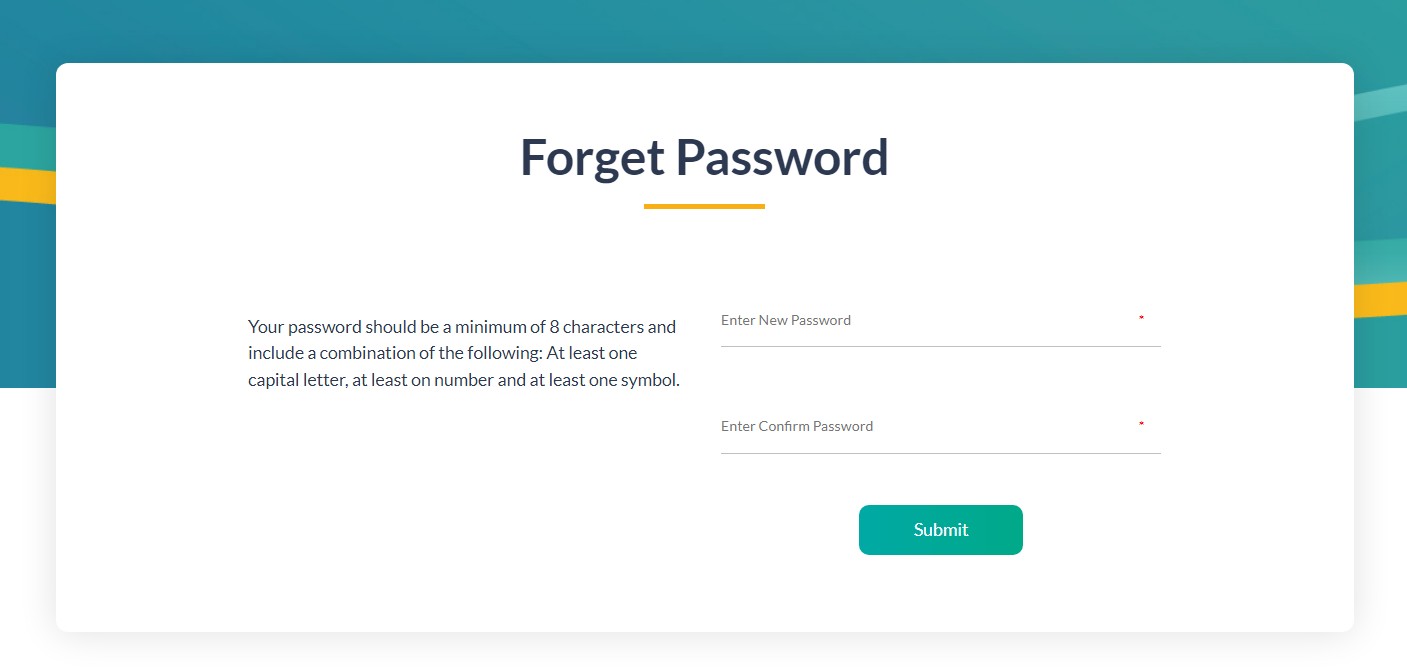


* **In this module user can process by use given verification code to them in their specified email.**

Page | 43

Electricity Hub

#### Forget Password

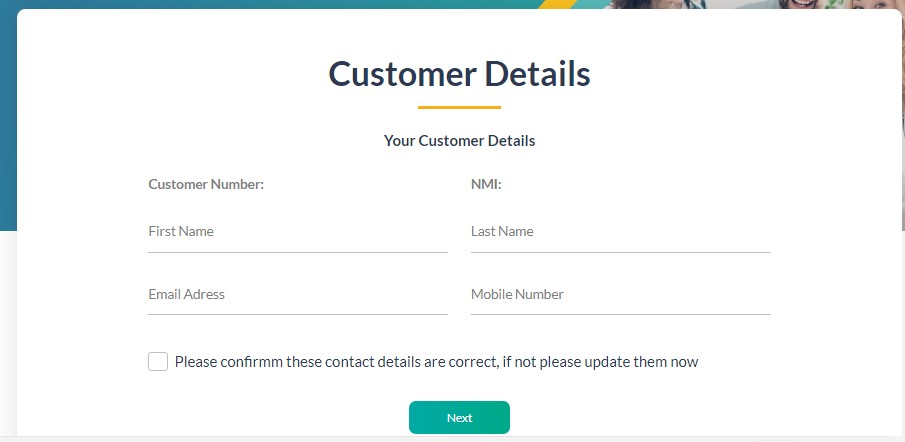


* **This module is created to set new password to their forgotten account password.**

Page | 44

Electricity Hub

#### Customer Details



* **This module is created to set the details of customers or users for their account.**

Page | 45

Electricity Hub

#### Verification Code

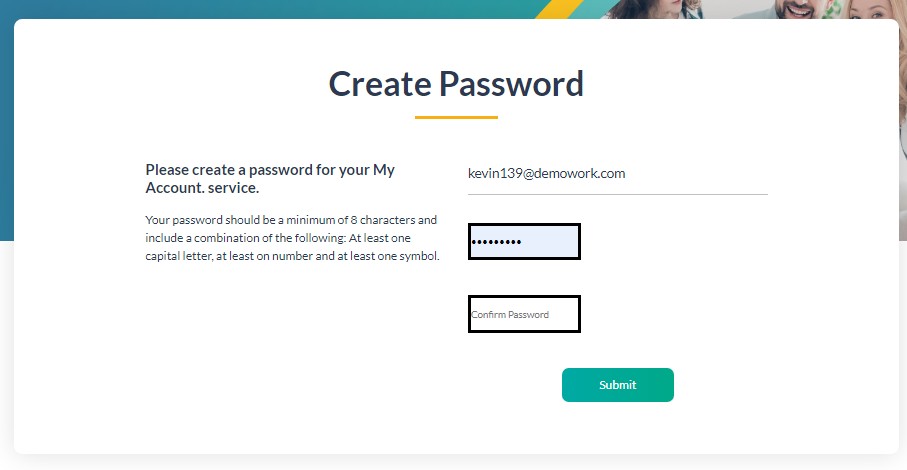


* **This step is applied or set by our team to verify that the account making user is exist in real life.**

Page | 46

Electricity Hub

#### Create Password

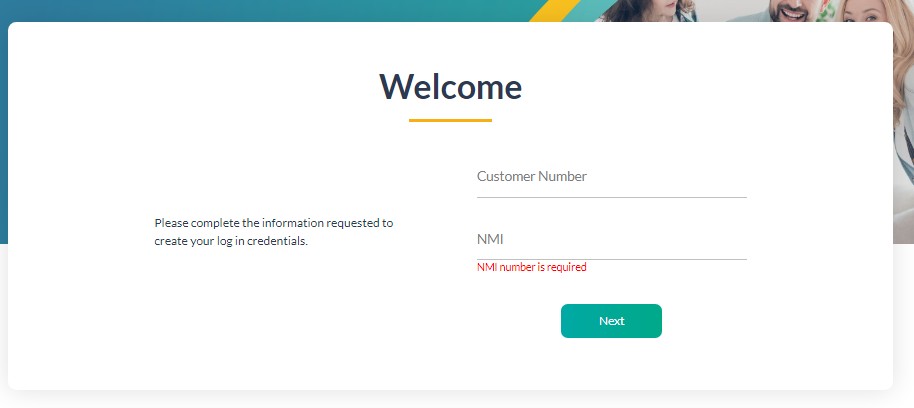


* **By using this module user can set or create password for their account.**

Page | 47

Electricity Hub

#### Active Account



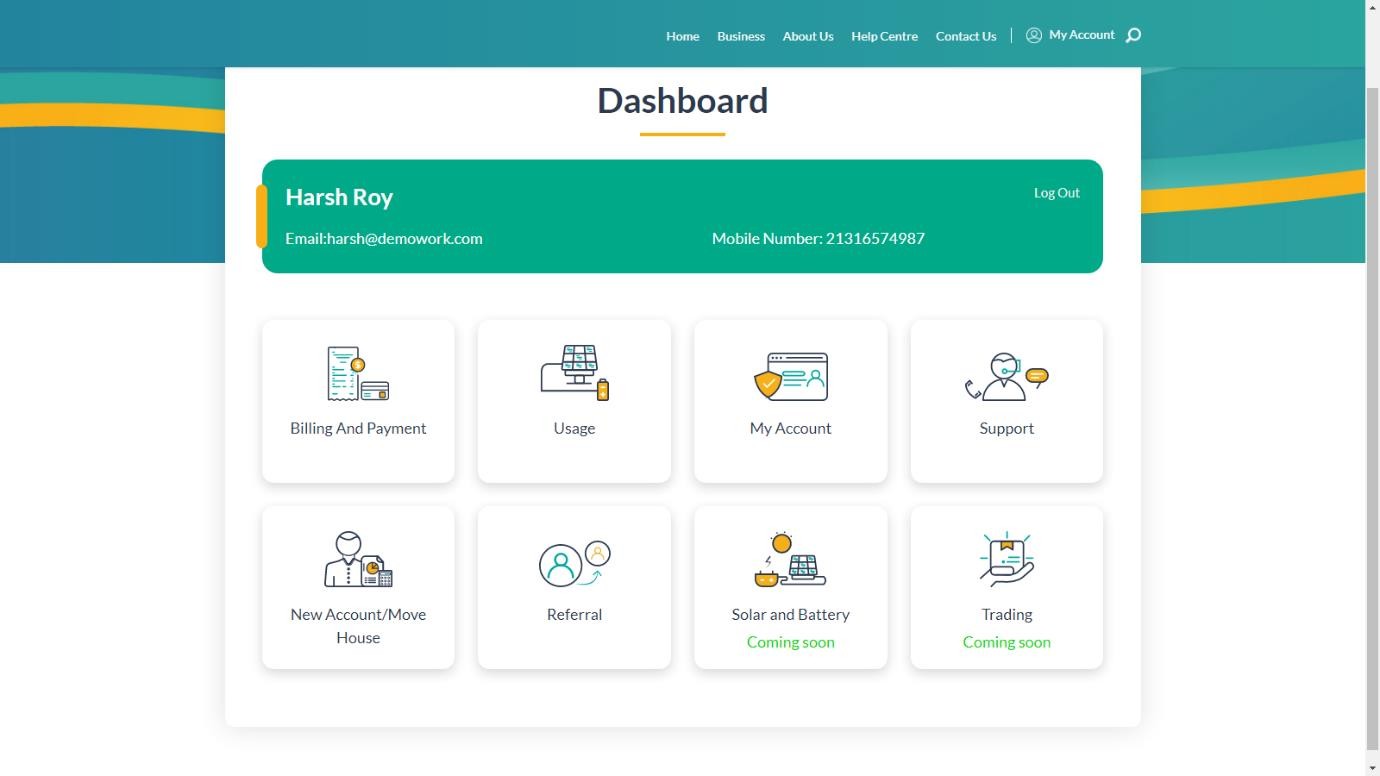
* **This module is the final step where user can enter to the application by using their provided details.**

#### User can make their account active by entering their details in this module.

Page | 48

Electricity Hub

#### Dashboard

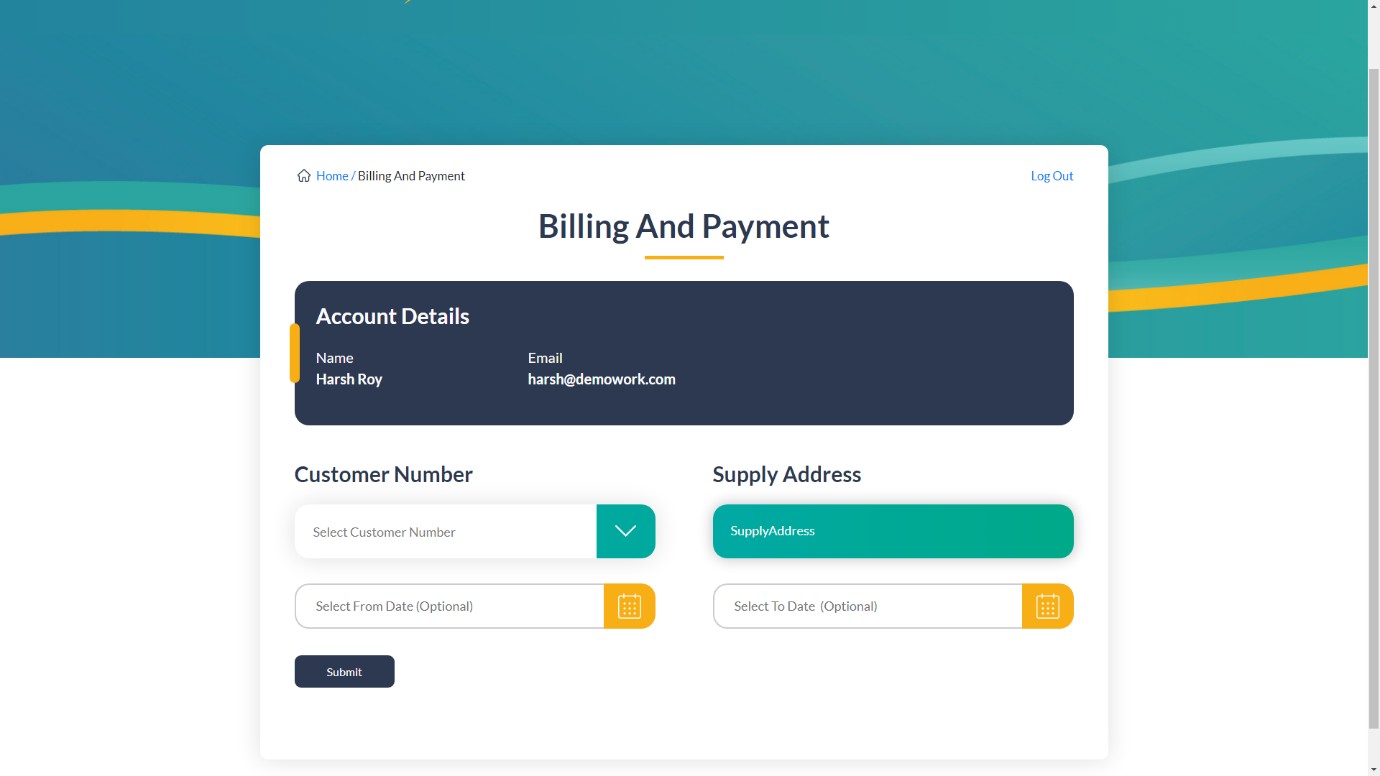


* **This page store all the modules in Dashboard.**

Page | 49

Electricity Hub

#### Billing And Payment

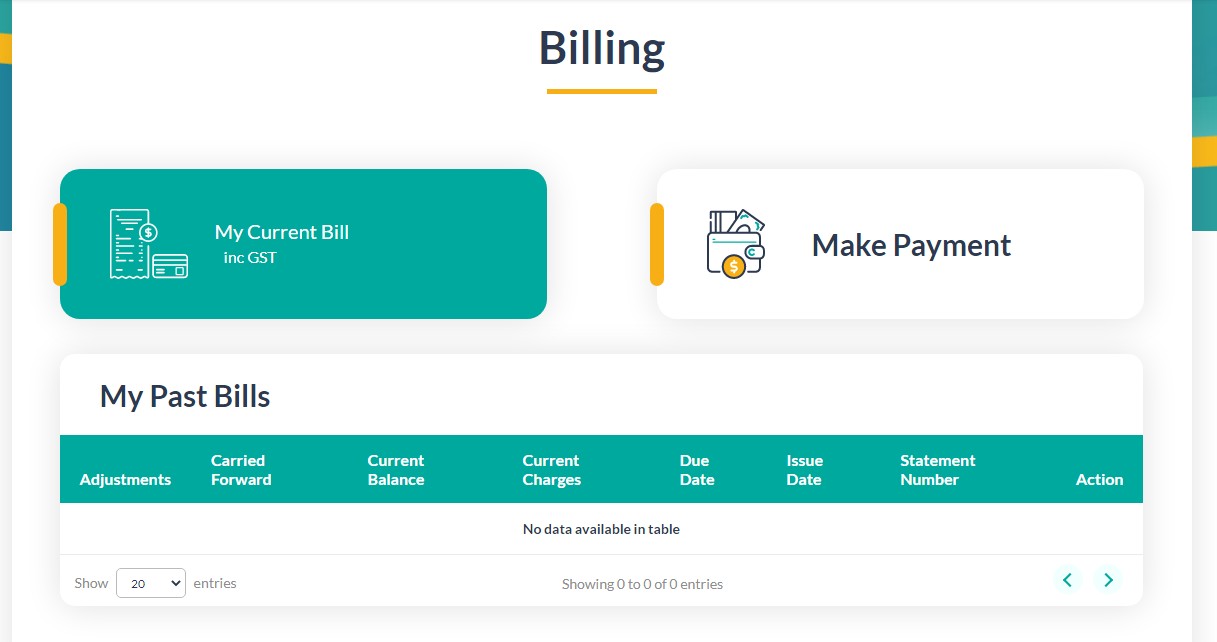


* **This module helps customers to make their electricity payment and get the history of their previous billing.**

Page | 50

Electricity Hub

#### Billing

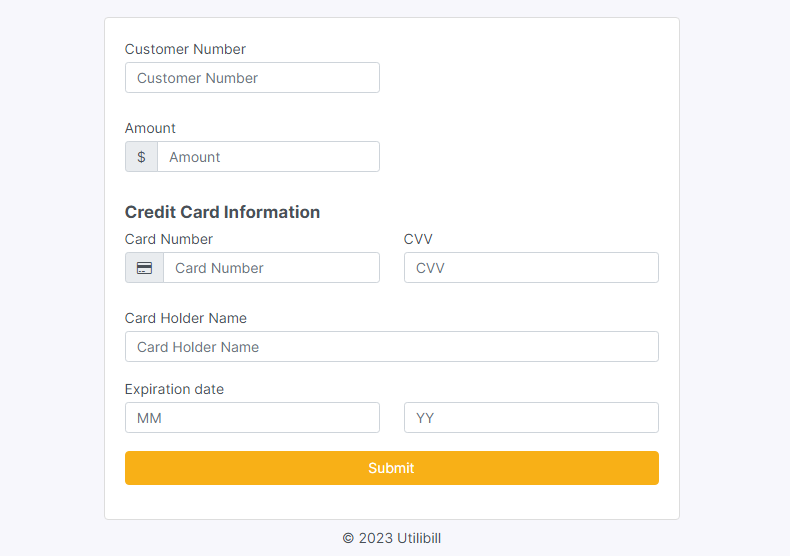


* **User can get all the detail of their previous bills and their current bill in this module.**

Page | 51

Electricity Hub

#### Payment

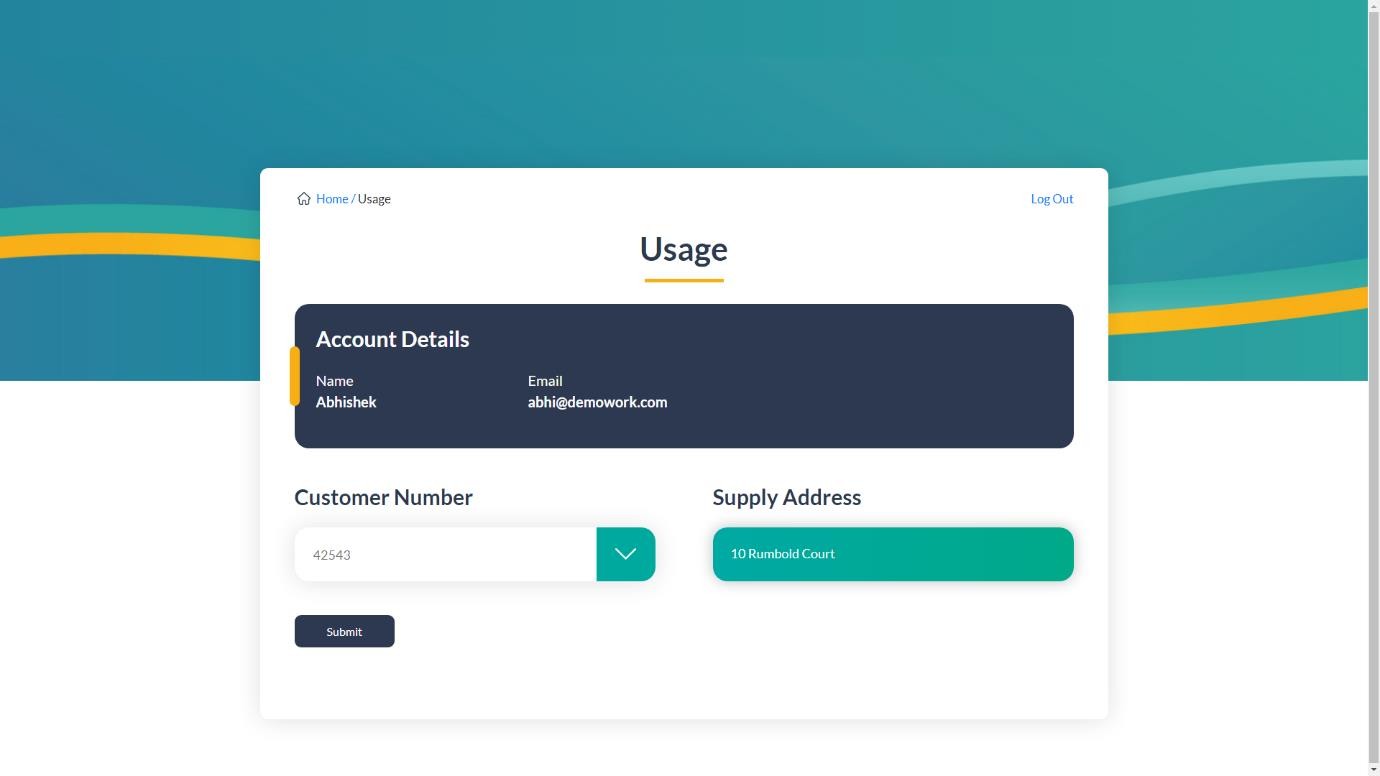


* **In this module user can make payment of their current and previous pending bills.**

Page | 52

Electricity Hub

#### Usage

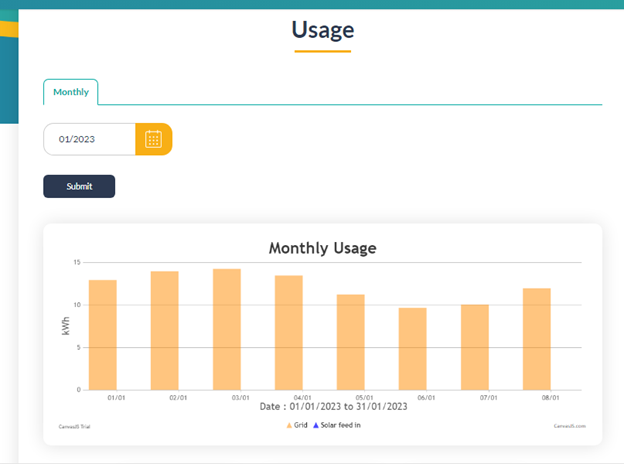


* **This module used to get the electricity usage details by entering required detail.**

Page | 53

Electricity Hub

#### Monthly Usage

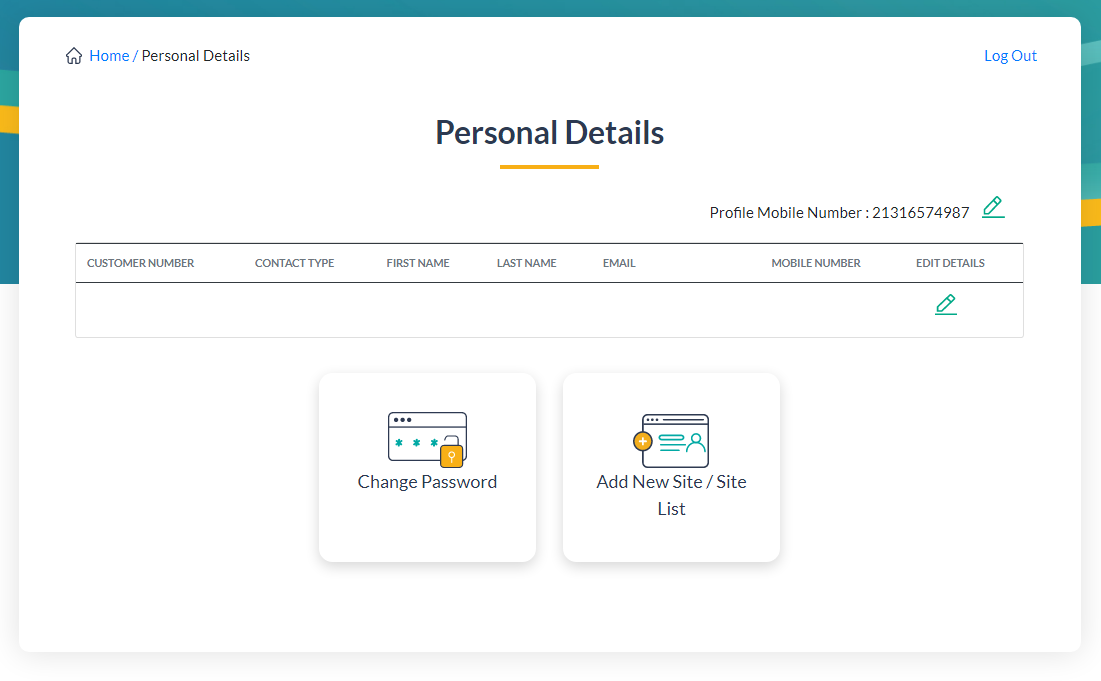


* **This module shows the monthly electricity usage which have used by customers after entering their details.**

Page | 54

Electricity Hub

#### Personal Details

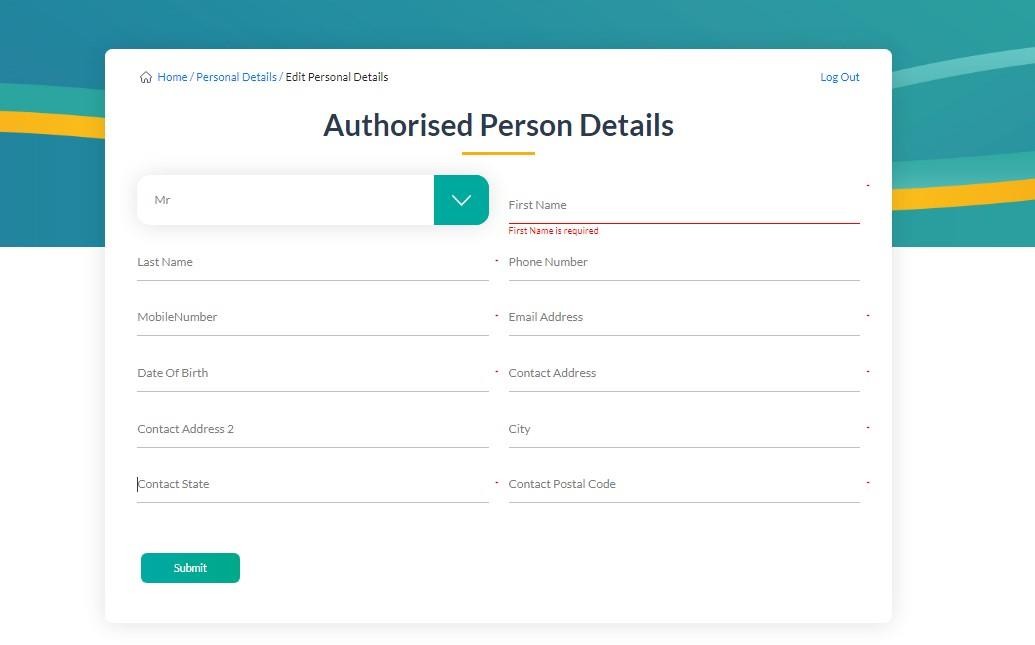


* **This module is created to show the personal details of customers.**

Page | 55

Electricity Hub

#### Authorized Person Details

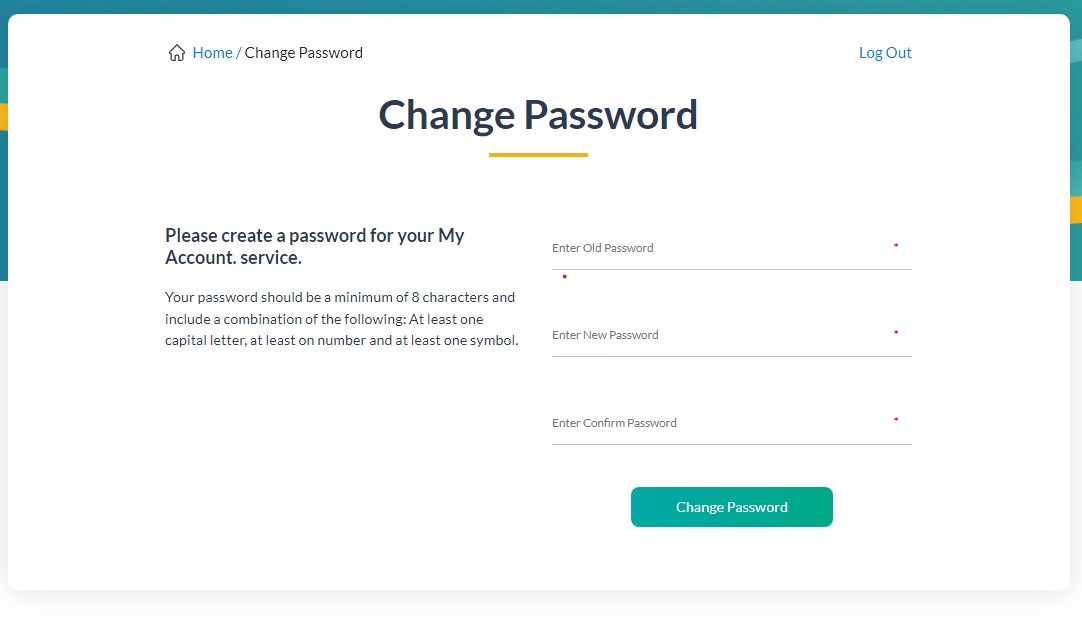


* **User can edit their personal detail by using this module.**

Page | 56

Electricity Hub

#### Change Password

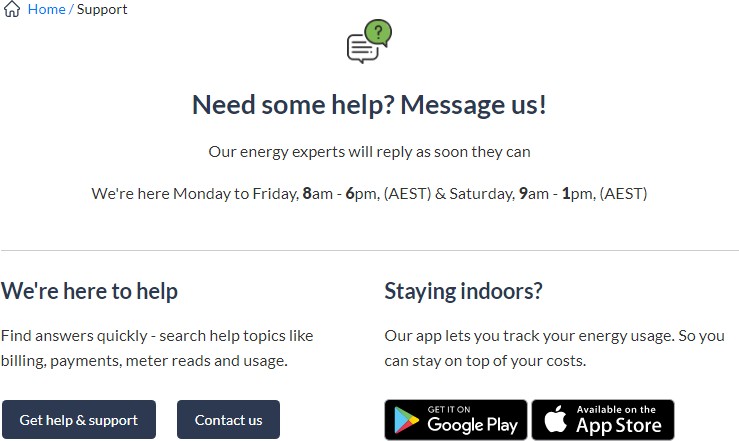


* **User can change their Current or Old Password using this module.**

Page | 57

Electricity Hub

#### Support

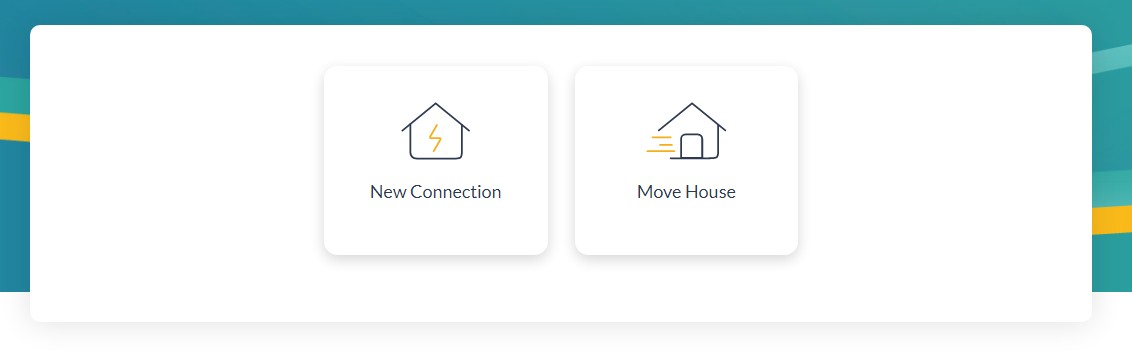


* **User can get any type of help about this application by using this module.**

Page | 58

Electricity Hub

#### New Connection/Move House

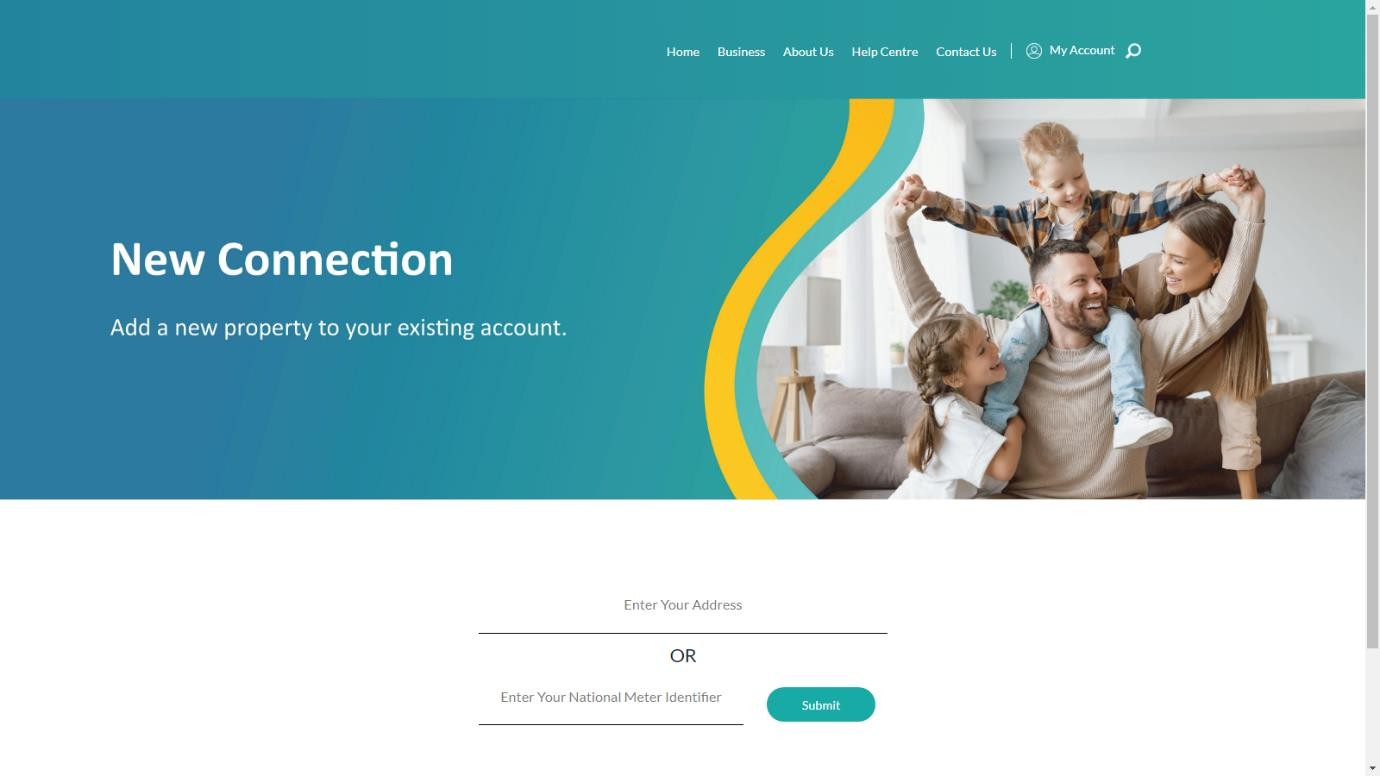


* **User can have the option to get new connection and move house.**

Page | 59

Electricity Hub

#### New Connection

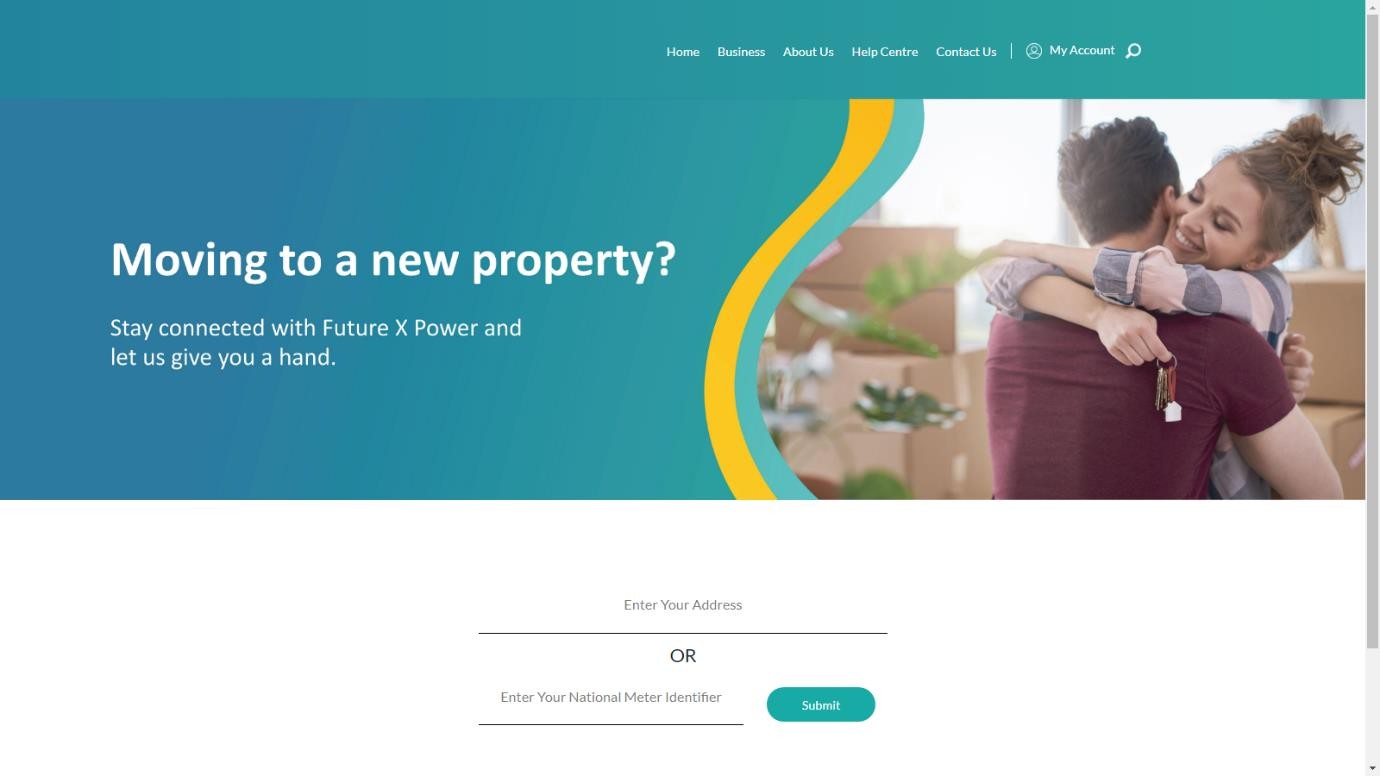


* **User can apply for new connection by using this module.**

Page | 60

Electricity Hub

#### Move House

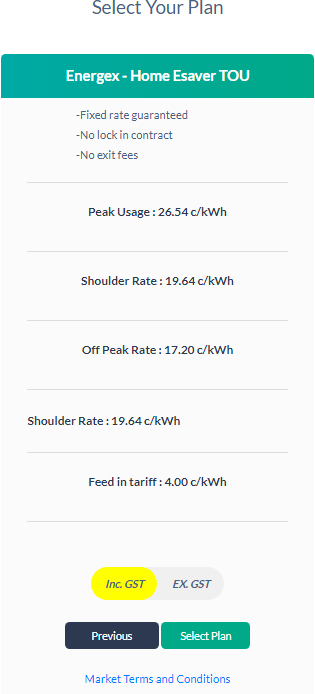


* **User can move their current connection to their new house or to new work place.**

Page | 61

Electricity Hub

#### Select Plan Include GST



* **User can select their plan from Included GST option.**

Page | 62

Electricity Hub

#### Select Plan Exclude GST

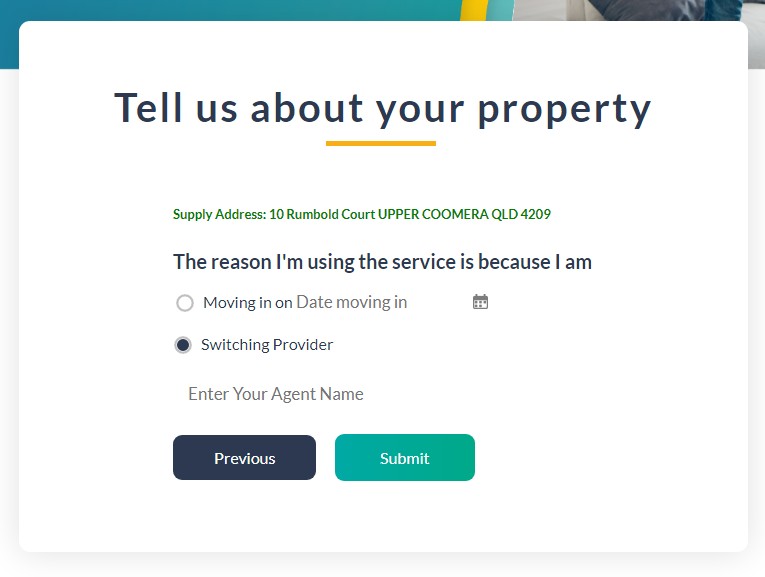


* **User can select their plan from Excluded GST option.**

Page | 63

Electricity Hub

#### Tell Us About Your Property

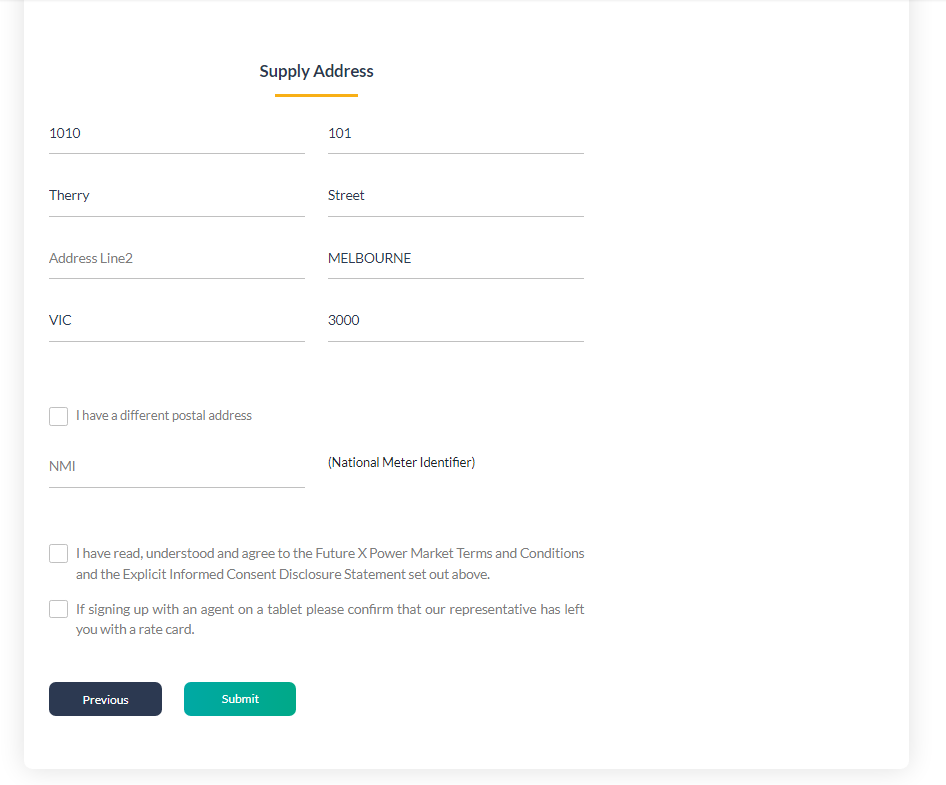
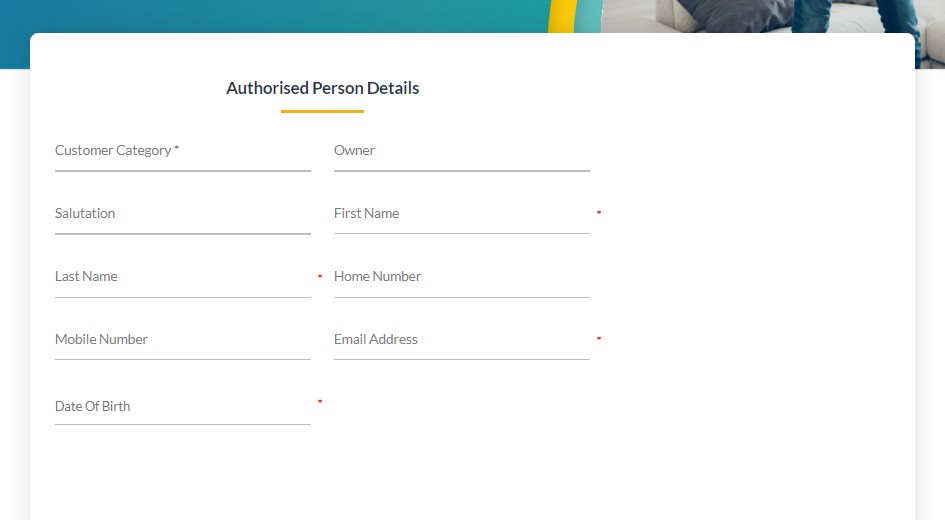


* **User can select the like that on which date they want to move to new connection or switch their provider to new provider by using this module.**

Page | 64

Electricity Hub

#### Authorized Personal Details

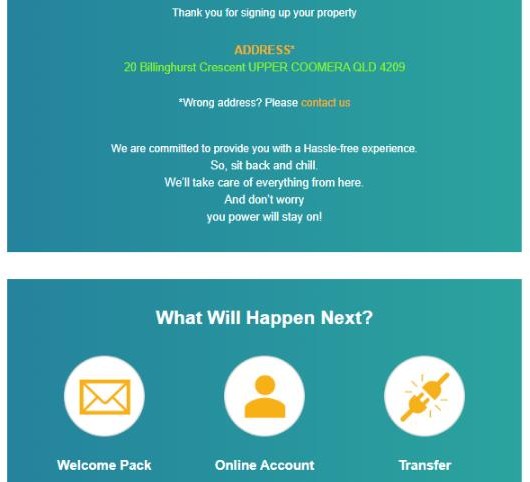


* **User can add their Personal Details and their supply address where they want their connection.**

Page | 65

Electricity Hub

#### Thank you for Signing up Your Property

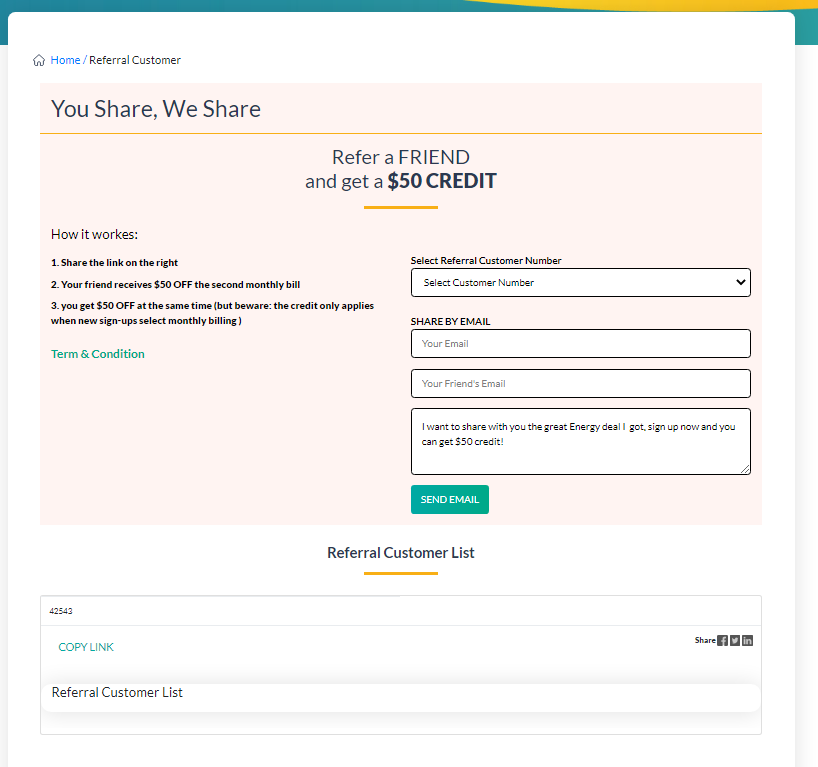


* **User will get the thank you massage when they will successfully get connected with us.**

Page | 66

Electricity Hub

#### Refer Customer



* **User can refer this service to any of their friends or their family members.**

Page | 67

Electricity Hub

# Agile Documentation

Page | 68

Electricity Hub

## 5.Agile Documentation

### Agile Project Charter

**Project Charter:** Electricity Hub

#### Background (Problem Statement):

Most of the time the Electricity Provider maintain all customers’ details and information about their selected plan in records. When the number of records increases, it is difficult to maintain the information of Customers and Connections. Maintaining the records manually leads to error prone and required more manpower and it consumes more time for processing the records.

#### Goals (Objectives):

The system being designed is economical with respect to the Admin and Providers point of view. In **Electricity HUB**, on Dashboard of Admin side, it displays information about Customers Plans, Customer Moving Date, Customer personal details, Customer NMI Number, Customer ID, Customer Selected Channel. Once the connection process is Done and user completely have their connection it will automatically display on Admin’s dashboard where Admin can approve the particular services for which users have applied.

#### Scope:

The design and Implementation of our **HUB** is to provide services to Customers. Admin can book new plans options where he can fill in the details of customers according to customer’s requirement and add prices or plans. All data is stored securely on SQL server and managed by Admin. Admin will be able to add details like customer’s information which is used to make new connections and move their current or old connections to new space.

#### Key Stakeholders

|  |  |
| --- | --- |
| **Project manager** | Satyabrata Lenka |
| **Project team members** | Nishant Raj, Ritik Raj |

Page | 69

Electricity Hub

#### Project Milestones

**Start Date:** 20-January-2024 **End Date:** 16-March-2024 **Invoicing Date:**  1-March-2024

|  |  |
| --- | --- |
| Constraints | * Users have knowledge about English language. * All the users must have an internet enabled devices. |
| Assumptions | * Electricity Hub is web application so all user can easily access services like Billing and Payment, make new Connection, See their monthly Power Usage, etc. * Admin and User first need to login in the system. |
| Risks and Dependencies | * Sometimes, User is making new connection and there is no network connection so, issues occurred. So, at that time User cannot upload their details. In this situation Customers have to contact the admin using call or any other communication device. Admin explains details about that Connection error problem to the developers. |

Page | 70

Electricity Hub

### Agile Product Roadmap

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **XYZ Company** | **February** | | | **March** | | | **April** | | |
| 1 | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** |
| **Admin** | **Process Customers Add Channel**  **Add Rate**  **Customer List Channel List**  **Failed Customer**  **Referral Customer List Add User**  **Landlord List Transaction List**  **Customer Portal Account Sign Out** | | | | | | | | |
| **User** | **Login**  **Registration**  **Billing And payment Usage**  **My Account Support**  **New/Move**  **House**  **Referral** | | | | | | | | |

Page | 71

Electricity Hub

### Agile Project Plan

**Project Name:** Electricity Hub

**Project Guide: Satyabrata Lenka**

**Project Deliverable:** Web Application

#### Scope Statement:

We have three actors on our **Electricity Hub** i.e., Admin and Customers. Customers can select their plans by their need. The admin will provide the services to the customers. So, this application will be treated as a bridge between the Customer and the Admin. Admin will manage the Customers which are registered in our Hub. All data is stored securely on SQL server and managed by Super Admin. When the customer is making new connection, the admin will get the notification related to their selected plans.

**Start Date:** 30-March-2023 **End Date:** 01-May-2023 **Overall Progress:** 95**%**

Page | 72

Electricity Hub

**Admin Side:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Responsible** | **Start** | **End** | **Days** | **Status** |
| Process Customers | Nishant  Ritik | 26-February-24 | 27-February-24 | 2 | Complete |
| Add Channel | Nishant  Ritik | 03-March-24 | 05-March-24 | 3 | Complete |
| Add Rate | Nishant  Ritik | 06-March-24 | 08-March-24 | 3 | Complete |
| Customer List | Nishant  Ritik | 10-March-24 | 10-March-24 | 1 | Complete |
| Channel List | Nishant  Ritik | 11-March-24 | 12-March-24 | 2 | Complete |
| Failed Customer | Nishant  Ritik | 13-March-24 | 14-March-24 | 2 | Complete |
| Referral Customer List | Nishant  Ritik | 17-March-24 | 17-March-24 | 1 | Complete |
| Add User | Nishant  Ritik | 18-March-24 | 20-March-24 | 3 | Complete |
| Landlord List | Nishant  Ritik | 21-March-24 | 22-March-24 | 2 | Complete |
| Transaction List | Nishant  Ritik | 01-April-24 | 01-April-24 | 1 | Complete |
| Customer Account Portal | Nishant  Ritik | 03-April-24 | 05-April-24 | 3 | Complete |
| Sign Out | Nishant  Ritik | 08-April-24 | 08-April-24 | 1 | Complete |

Page | 73

Electricity Hub

**User Side:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Task Name** | **Responsible** | **Start** | **End** | **Days** | **Status** |
| Login | Nishant Ritik | 09-April-24 | 10-April-24 | 2 | Complete |
| Registration | Nishant Ritik | 11-April-24 | 11-April-24 | 1 | Complete |
| Billing And payment | Nishant Ritik | 12-April-24 | 12-April-24 | 1 | Complete |
| Usage | Nishant  Ritik | 13-April-24 | 13-April-24 | 1 | Complete |
| My Account | Nishant Ritik | 14-April-24 | 14-April-24 | 1 | Complete |
| Support | Nishant Ritik | 15-April-24 | 15-April-24 | 1 | Complete |
| New Account/Move House | Nishant Ritik | 16-April-24 | 17-April-24 | 1 | Complete |
| Referral | Nishant  Ritik | 17-April-24 | 18-April-24 | 1 | Complete |

### Agile User Story

**Agile User Story Template**

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story ID** | **As a***<type of user>* | **I want to** *<perform some task>* | **so that I can** *<achieve some goal>* |
| 1 | Project manager | View a status report from each team member | Ensure the project stays on track. |
| 2 | Admin | Admin can manage all data of Customers. | So that our web application will be perfectly running and all functionality are used by  the admin and customer can easily work on it. |
| 4 | User | Reduce the paperwork of Service provided to customer. | User can easily use this application and easily get Electric Services on time. |

Page | 74

Electricity Hub

### Agile Release Plan

**Admin Side**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Start** | **End** | **Duration** | **Status** | **Release Date** |
| Admin | Process customer | 15-Feb-24 | 15-Feb-23 | 1 | Released | 15-Feb-23 |
| Admin | Add Channel | 16-Feb-24 | 18-Feb-24 | 3 | Released | 18-Feb-24 |
| Admin | Add Rate | 19-Feb-24 | 21-Feb-24 | 3 | Released | 21-Feb-24 |
| Admin | Customer List | 22-Feb-24 | 22-Feb-24 | 1 | Released | 22-Feb-24 |
| Admin | Channel List | 23-Feb-24 | 24-Feb-24 | 2 | Released | 24-Feb-24 |
| Admin | Failed Customer | 25-Feb-24 | 26-Feb-24 | 2 | Released | 26-Feb-24 |
| Admin | Referral Customer List | 27-Feb-24 | 27-Feb-24 | 1 | Released | 27-Feb-24 |
| Admin | Add User | 28-Feb-24 | 29-Feb-24 | 2 | Released | 29-Feb-24 |
| Admin | Landlord List | 01-March-24 | 01-March-24 | 1 | Released | 01-March-24 |
| Admin | Transaction List | 04-March-24 | 04-March-24 | 1 | Released | 04-March-24 |
| Admin | Customer Portal Account | 11-March-24 | 13-March-24 | 3 | Released | 13-March-24 |
| Admin | Sign Out | 18-March-24 | 18-March-24 | 1 | Released | 18-March-24 |

**User Side:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Task** | **Start** | **End** | **Duration** | **Status** | **Release Date** |
| User | Login | 22-Feb-24 | 22-Feb-24 | 1 | Released | 22-Feb-24 |
| User | Registration | 23-Feb-24 | 24-Feb-24 | 2 | Released | 24-Feb-24 |
| User | Billing And payment | 25-Feb-24 | 26-Feb-24 | 2 | Released | 26-Feb-24 |
| User | Usage | 27-Feb-24 | 27-Feb-24 | 1 | Released | 27-Feb-24 |
| User | My Account | 28-Feb-24 | 29-Feb-24 | 2 | Released | 29-Feb-24 |
| User | Support | 01-March-24 | 01-March-24 | 1 | Released | 01-March-24 |
| User | New Account/Move  House | 04-March-24 | 04-March-24 | 1 | Released | 04-March-24 |
| User | Referral | 11-March-24 | 13-March-24 | 3 | Released | 13-March-24 |

Page | 75

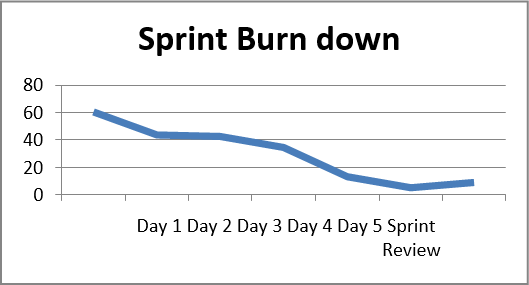
Electricity Hub

### Agile Sprint Backlog

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Backlog Item | Story Points | Responsible | Status | Origina l Estimat  e | Day 1 | Day 2 | Day 3 | Day 4 | Day 5 | Sprint Review |
| **Admin Panel Development** | **8** | Ritik | Completed |  |  |  |  |  |  |  |
| Designing Admin Panel |  | Ritik | Completed | 7 | 5 | 3 | 0 | 0 | 0 | 0 |
| Designing Admin Panel all functionality |  | Ritik | Completed | 3 | 1.5 | 1 | 5 | 0 | 1 | 0 |
| Coding for interacting with database |  | Nishant | Completed | 1 | 0.5 | 0 | 3 | 0 | 0 | 0 |
| Testing |  | Nishant | Completed | 0.5 | 1 | 2 | 3 | 1 | 0 | 0 |
| **User Panel Development** | **8** | Ritik | Completed |  |  |  |  |  |  |  |
| Designing User Panel |  | Nishant | Completed | 8 | 6 | 0 | 0 | 0 | 0 | 0 |
| Designing User Panel all functionality |  | Ritik | Completed | 3 | 1 | 3 | 3 | 3 | 0 | 0 |
| Coding for interacting with database |  | Nishant | Completed | 1.5 | 1 | 0.5 | 0.5 | 1 | 1 | 0 |
| Testing |  | Nishant | Completed | 2 | 0.5 | 0 | 0 | 0 | 0 | 3 |
| Total |  |  |  | 39 | 31.5 | 29 | 21 | 8 | 3 | 7 |

Page | 76

Electricity Hub



### Agile Test Plan

**Project Name:** Electricity Hub

**Written By:** Nishant Raj, Ritik Raj

**Tested By:** Nishant Raj, Ritik Raj

**Browser:** Google Chrome/Fire fox/MS Edge

**Version:** Visual studio 2019

**Tested On:** 16-April-24

#### Description:

The system is developed for electricity service to the customer. It maintains the customer with what plans they should go with as per their power usage. This is for both residential and commercial. This is an overseas client, we have two actors on portal i.e., User and Admin. The web application using Asp.NET 4.5 SQL Server 2019, Bootstrap, Kendo Controls, JavaScript and jQuery.

Page | 77

Electricity Hub

**Admin Side:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Date** | **Action** | **Expected Results** | **Actual Results** | **Pass?** |
| 1 | 27-Feb-23 | Process customer | Showing List of Customers. | Completed Successfully | Yes |
| 2 | 05-March-2023 | Add Channel | Add, Listing, Update, Delete Channels | Completed Successfully | Yes |
| 3 | 08-March-24 | Add Rate | Add, Listing, Update, Delete Rates | Completed Successfully | Yes |
| 4 | 10-March-24 | Customer List | Listing, Delete Customers List | Completed Successfully | Yes |
| 5 | 12-March-24 | Channel List | Listing, Delete Channel List | Completed Successfully | Yes |
| 6 | 14-March-24 | Failed Customer | Listing Failed Customer | Submitted Successfully | Yes |
| 7 | 17-March-24 | Referral Customer List | Listing Refer Customer | Submitted Successfully | Yes |
| 8 | 20-March-24 | Add User | Add, Listing, Update, Delete User | Submitted Successfully | Yes |
| 9 | 22-March-24 | Landlord List | Listing Landlord | Submitted Successfully | Yes |
| 10 | 24-March-24 | Transaction List | Admin get notification  when Users/Customers perform activity. | Submitted Successfully | Yes |
| 11 | 01-April-24 | Customer Portal Account | Listing Customers | Completed Successfully | Yes |
| 12 | 04-April-24 | Sign Out | Signing Out | Completed Successfully | Yes |

**User Side:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test #** | **Date** | **Action** | **Expected Results** | **Actual Results** | **Pass?** |
| 1 | 07-April-24 | Login | **Logging In Successfully** | Completed Successfully | Yes |
| 2 | 11-April-24 | Registration | **Registered Completely** | Completed Successfully | Yes |
| 3 | 12-April-24 | Billing And payment | **Make Payment Successfully** | Completed Successfully | Yes |
| 4 | 13-April-24 | Usage | **Get Power Usage** | Completed Successfully | Yes |
| 5 | 14-April-24 | My Account | **Get Account Details** | Completed Successfully | Yes |
| 6 | 15-April-24 | Support | **Get Help by the Application Handler** | Completed Successfully | Yes |
| **7** | 16-April-24 | New Account/Move House | **Easily move to new house and get new connection and make**  **new account** | Completed Successfully | Yes |
| **8** | 17-April-24 | Referral | **User can give reference to their friends and families.** | Completed Successfully | Yes |

Page | 78

Electricity Hub

### Earned – values and burn chart.

**Admin | User**

**Planned Value (PV) or Budgeted Cost of Work Scheduled (BCWS)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| WBS | Task Name | TBC | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| 1.1 | Admin | 7800 | 800 | 550 | 700 | 450 | 850 | 600 | 450 | 650 | 550 | 700 | 850 | 650 |
| 1.2 | User | 8700 | 950 | 900 | 450 | 650 | 900 | 800 | 550 | 550 | 850 | 700 | 1000 | 400 |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Total Budget Cost** | | **16500** | 1750 | 1540 | 1150 | 1100 | 1750 | 1400 | 1000 | 1200 | 1400 | 1400 | 1850 | 1050 |
| Cumulative Planned Value (PV) | |  | 1750 | 3200 | 4320 | 5450 | 7200 | 8600 | 9600 | 10800 | 12200 | 13600 | 15450 | 16500 |

**Actual Cost and Earned Value**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cumulative Actual Cost (AC) | 800 | 1950 | 4550 | 6550 | 10800 | 13600 | 14500 |
| Cumulative Earned Value (EV) | 1170 | 5025 | 10200 | 12180 | 13275 | 16050 | 16500 |

**Project Performance Metrics**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Cost Variance (CV = EV-AC) | 370 | 3075 | 5650 | 5630 | 2475 | 2450 | 2000 |
| Schedule Variance (SV = EV-PC) | -580 | 1825 | 5850 | 6730 | 6075 | 7450 | 6900 |
| Cost Performance index (CPI = EV/AC) | 1.46 | 2.58 | 2.24 | 1.86 | 1.23 | 1.18 | 1.14 |
| Schedule Performance Index (SPI = EV/PV) | 0.67 | 1.57 | 2.34 | 2.23 | 1.84 | 1.87 | 1.72 |
| Estimated Cost at Completion (EAC) | 11282 | 6403 | 7360 | 8873 | 13424 | 13981 | 14500 |

Page | 79

Electricity Hub

# Proposed Enhancement

Page | 80

Electricity Hub

### 6 Proposed Enhancement

In Today’s world moving to the online application so we initiate this project for the Electricity hub to make electricity services faster and low time consumer by using this Application. This system is useful for the Customers who have no time in this conflictual life. This Application will save the precious time of our customers and provide them service’s which they need at their doorsteps.

Page | 81

Electricity Hub

# Conclusion

Page | 82

Electricity Hub

### 7 Conclusion

It concludes that this system is user-friendly and easy to use. We can say that any Civilian can use this system to get their electricity connection and if they already have connection than they can easily move their connection to another new place by sitting at their house. Our application is mainly developed to help today’s generation.

Page | 83

Electricity Hub

# Bibliography

Page | 84

Electricity Hub

### 8 Bibliography

**Website**

* [https://www.google.com](https://www.google.com/)
* [www.stackoverflow.co](http://www.stackoverflow.con/)m
* [www.w3school.com](http://www.w3school.com/)
* [www.tutorialspoint.com](http://www.tutorialspoint.com/)
* [www.c-sharp-corner.com](http://www.c-sharp-corner.com/)

Page | 85