**ANY TIME ELECTRICITY BILL PAYMENT MACHINE CONTROLLER**

**ABSTRACT**

The "Any Time Electricity Bill Payment Machine Controller" is a system meant to provide clients with a quick and accessible way to pay their electricity bills at any time, without the need for manual assistance. This abstract provides an overview of the controller's primary features and functionalities.

The controller serves as the central control unit for an electrical bill payment self-service kiosk or machine. It combines numerous components to provide a seamless payment experience. A user interface, money processing capabilities, networking options, and rigorous security features are among these components.

Customers may easily engage with the machine thanks to the user interface. Users can enter bill information, choose their preferred payment methods, and view transaction summaries.

**LITERATURE SURVEY**

The literature review includes a thorough examination of existing research and studies in the topic. It intends to investigate various approaches, strategies, and algorithms used in the detection of fake news. The survey provides useful insights into the existing state of the art and highlights gaps and constraints that may be used to drive the proposed project.

**1. A Smart Electronics Energy Measurement, Monitoring, Billing and Payment System using IoT by P. Manikandan(2023)**

*This paper proposes a Smart Electronics Energy Measurement, Monitoring, Billing, and Payment System (SEEMBPS) that leverages the Internet of Things (IoT) technology to enable efficient and intelligent energy management.*

*The SEEMBPS utilizes IoT-enabled smart meters installed in households and businesses to measure and monitor energy consumption in real-time. These smart meters are equipped with advanced sensing capabilities, communication modules, and data processing capabilities, allowing them to collect and transmit energy consumption data to a centralized server.*

**2. Development of an ANN-Based Estimated Electricity Billing System by B.B Adetokun(2018)**

This paper presents the development of an Artificial Neural Network (ANN)-based Estimated Electricity Billing System (EEBS) that utilizes machine learning techniques to estimate electricity usage and generate accurate bills for consumers.

The EEBS leverages historical electricity consumption data, along with other relevant parameters such as weather conditions, time of day, and consumer behavior patterns, as inputs to train the ANN model.

# **3. *Smart Power; A Smart Card Electricity Payment System By M.C.S. Simpson(1996)***

*The system leverages smart card technology and integrates it with the existing electricity infrastructure to streamline the payment process and enhance user experience.*

*In the Smart Power system, consumers are issued smart cards that store payment information and account details. These smart cards are linked to the electricity meters installed in their premises.*

# ***4.Smart Electricity Measuring System by Surabhi Naik(2020)***

*The system offers various features to users, including real-time monitoring of energy consumption, historical data analysis, and energy usage alerts. Through a user-friendly interface, consumers can access their electricity usage information, view trends, and identify opportunities for energy optimization. Additionally, utilities can utilize the data provided by SEMS to analyze patterns, forecast demand, and implement demand response strategies.*

*SEMS also supports remote management and control of electricity consumption. It enables consumers to remotely monitor and control electrical devices, allowing them to optimize usage, reduce wastage, and ultimately lower energy costs. This feature is particularly beneficial in the context of smart homes and smart grids, where energy management is automated and adaptive.*

***USE CASE DIAGRAM FOR DESIGN AND IMPLEMENTATION OF*** ***ANY TIME ELECTRICITY BILL PAYMENT MACHINE CONTROLLER***

*In this use case diagram, there are three primary actors: User, Any Time Electricity Bill Payment Machine Controller, and Electricity Provider. The interactions between these actors are represented by use cases.*

***The main use cases identified are as follows:***

*Make Bill Payment: The user initiates the process by selecting an electricity provider and entering their customer ID and the amount to be paid. The payment is processed, and a transaction confirmation is generated.*

*View Bill Details: The user can view the details of their electricity bill, such as current charges, due date, and previous payment history.*

*Print Bill Receipt: The user has the option to print a receipt after making the bill payment. This receipt serves as proof of payment.*

*Process Payment and Generate Transaction Confirmation: The Any Time Electricity Bill Payment Machine Controller handles the payment processing, verifies the payment, updates the payment status, and generates a transaction confirmation for the user.*

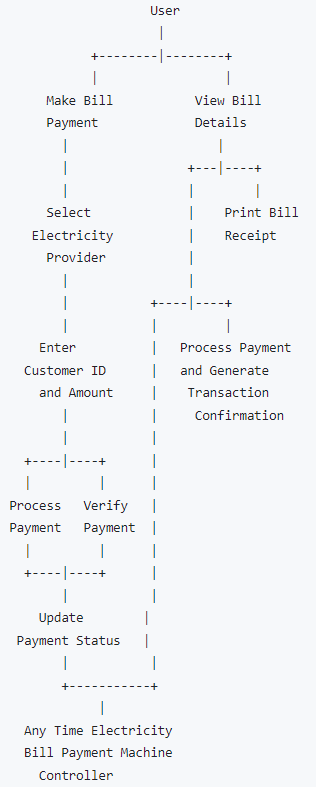
*Select Electricity Provider: The user can select the electricity provider they are associated with for the bill payment.*

*Enter Customer ID and Amount: The user enters their customer ID and the amount they wish to pay for their electricity bill.*

*Verify Payment: The Any Time Electricity Bill Payment Machine Controller verifies the payment details provided by the user, ensuring that the customer ID and payment amount are accurate.*

*Update Payment Status: After successful payment verification, the Any Time Electricity Bill Payment Machine Controller updates the payment status for the user's electricity bill.*

*These use cases represent the major functionalities and interactions involved in the design and implementation of an Any Time Electricity Bill Payment machine controller.*

**

***Fig 1.1 Use Diagram of Electricity Bill Payment Machine Controller***

***CODE IMPLEMENTATION***

*Algorithm for the given Python code:*

*Import the required libraries:*

*json*

*os*

*smtplib*

*email.mime.multipart*

*email.mime.text*

*datetime*

*mysql.connector*

*clearscreen (assuming it is a custom module)*

*logout (assuming it is a custom module)*

*Establish a connection to the MySQL database using the provided credentials.*

*Read the configuration parameters from the "config.json" file.*

*Define a function named bilEmailHome that takes userid and logintime as parameters:*

*a. Clear the screen.*

*b. Read the content of the "billEmailnotAdmin\_message.txt" file.*

*c. Print the formatted message with the retrieved parameters.*

*d. Read user input.*

*e. If the user input is not in the specified function tuple, clear the screen and call the bilEmailHome function again.*

*f. If the user input is "01#02", call the sendmailtocustomers function and pass userid and logintime as parameters.*

*g. If the user input is "00#01", call the logout function and pass userid as a parameter.*

*Define a function named sendmailtocustomers:*

*a. Define the SMTP server port and server address.*

*b. Get the login credentials from the configuration parameters.*

*c. Get the current date and time.*

*d. Execute a SQL query to retrieve email and consumer details from the "customer" table for the current month.*

*e. Iterate over the retrieved data:*

*Set the email subject as "Your electricity bill has been generated for the month {month} ({consumername})".*

*Execute a SQL query to retrieve the units consumed for the current month and consumer number.*

*Try to open the bill file based on the units consumed and consumer number.*

*If successful, read the bill file content as the email body.*

*Connect to the SMTP server, login, and send the email to the corresponding email address.*

*Print a success message if the email is sent successfully.*

*Otherwise, print an error message.*

*f. Print the number of emails sent and the number of errors encountered.*

*g. Wait for two seconds and then call the bilEmailHome function with the appropriate parameters.*

*The SQL\_MODE and START TRANSACTION statements seem to be unrelated to the code provided and can be ignored.*

***CODE:***

*import json*

*import os*

*import smtplib*

*from email.mime.multipart import MIMEMultipart*

*from email.mime.text import MIMEText*

*from datetime import datetime*

*from os import path*

*from os import path, getcwd*

*import time*

*import mysql.connector as c*

*from clearscreen import clear*

*from logout import logout*

*connection = c.connect(host='localhost', database='electricity\_bill', user='root', password='')*

*db = connection.cursor()*

*#Opening of config.json file*

*THIS\_FOLDER = path.dirname(path.abspath(\_\_file\_\_))*

*my\_file = path.join(THIS\_FOLDER,'files','config\_file', 'config.json')*

*with open(my\_file, 'r') as c:*

*params = json.load(c)["params"]*

*def bilEmailHome(userid,logintime):*

*'''This is the bill generation department homepage function'''*

*mydate = datetime.now()*

*clear() #Clear the screen*

*billGenAdmin\_message = open('files/messages/billEmailnotAdmin\_message.txt','r').read()*

*funcAdminTuple = ('01#02','00#01')*

*print(billGenAdmin\_message.format(params['company\_name'],userid,logintime,datetime.now(),mydate.strftime("%B")))*

*userinput = input()*

*if userinput not in funcAdminTuple:*

*clear() #Clear the screen*

*bilEmailHome(userid,logintime)*

*else:*

*if userinput=='01#02':*

*sendmailtocustomers()*

*sendmailtocustomers(userid,logintime)*

*elif userinput=='00#01':*

*logout(userid)*

*def sendmailtocustomers():*

*def sendmailtocustomers(userid,logintime):*

*port, smtp\_server = 465, 'smtp.gmail.com'*

*login, password = params['email'], params['password\_email']*

*mydate = datetime.now()*

*db.execute(f'SELECT email,consumername FROM customer WHERE month="{mydate.strftime("%B")}"')*

*db.execute(f'SELECT email,consumername, consumerno FROM customer WHERE month="{mydate.strftime("%B")}"')*

*data = db.fetchall()*

*message = MIMEMultipart()*

*message["from"] = login*

*for x,y in data:*

*error,emailno = 0,0*

*for x,y,z in data:*

*message["subject"] = f"Your electricity bill has been generated for the month {mydate.strftime('%B')} ({y})"*

*db.execute(f'SELECT unit\_consumed FROM customer WHERE month="{mydate.strftime("%B")}" AND consumerno="{z}"')*

*unitsConsumed = db.fetchall()[0][0]*

*try:*

*with open(path.join(getcwd(),'files','customerBillFolder',f'{unitsConsumed}{z}.txt'),'r') as bill:*

*body = bill.read()*

*with smtplib.SMTP(smtp\_server, port) as server:*

*server.login(login, password)*

*server.sendmail(message["from"], x, body)*

*print(f"Email (BILL) sent to {y}")*

*print()*

*emailno+=1*

*except:*

*print('There was some error!')*

*print()*

*error+=1*

*print(emailno, " Email sent!")*

*print("With ",error," errors!")*

*print()*

*print("Now please wait for two seconds!")*

*time.sleep(2)*

*bilEmailHome(userid,logintime)*

*SET SQL\_MODE = "NO\_AUTO\_VALUE\_ON\_ZERO";*

*START TRANSACTION;*

*SET time\_zone = "+00:00";*

*A picture containing text, screenshot, font

Description automatically generated*

***FIG 1.2 Creation of Table***

***CONCLUSION***

*The introduction of smart and creative electricity bill payment systems has dramatically altered the traditional billing process. These innovative solutions provide several benefits to both consumers and utility corporations, revolutionizing the management and payment of power bills.*

*The controller typically integrates various components, including a user interface, payment processing capabilities, connectivity options, and security features. It enables customers to interact with the machine, input their bill details, select payment methods, and complete transactions.*

*The development of smart electricity bill payment systems enhances convenience, accuracy, and efficiency in managing electricity billing and payment processes. These systems contribute to the transition towards smarter energy management, promoting sustainable practices, and empowering consumers and utilities alike.*