Electricity Billing System Problem Statement

Group Members: Will Humphlett, DJ Harris, Jonathan Porco, Michael Duvall, and Swati Baskiyar

This project seeks to modernize the current electricity billing system. The major goal is to computerize and automate every step in the billing process to make it more seamless, accessible, and efficient. The primary function of this project is to track the units of power consumed by a given user and bill them accordingly. In turn, administrators of this project can view power consumed, by whom, and what they are required to pay.

By automating this process, billing errors are eliminated from previous workflows where the accounting was done by hand, and unneeded labor (e.g. calling to pay the bill, workers driving to consumer's house to record units consumed, etc) are scaled back in favor of an approach that minimizes human interaction. This also allows for seamless data sharing between the user and the administrator. However, with this data availability comes the need for high security provisions to safeguard it, as well as thorough debugging provisions to allow for investigation should something go awry.

This project will be constructed as a web application, built with Python and the Django Web-Framework. For database needs, we will be using SQLite, and hosting will be accomplished with Amazon Web Services. CI will be accomplished with GitHub and GitHub actionsn

The following are the classes of users we have defined for the purpose of explaining the use cases from their view

- Customer - A user that consumes electricity and is expected to pay for their service. They will have access to their data and their data alone
- Employee - A user that is able to view customer data and edit customer data under a few conditions. They are expected to aid the customer when needed.
- Administrator - A user that is given full permissions to see and edit customer data. They manage the application and make changes as needed.

The following are high-level use cases as well as their relevant data to further flush out the project and what it hopes to accomplish.

Automatic User Creation/Registration

- Previously, enabling electrical service for a customer would require the customer to contact a customer service representative who would have to create an account and schedule a worker to turn on power manually. This process is inefficient and can be automated. Now, a user can create an account themselves and the data they input can be validated automatically. Such data points include address, method of payment, and payment required. This allows both customers and administrators to view data relevant to a given user. Furthermore, this creates an account that the customer can access to view billing schedules, payment needed, detailed summaries, and other tools that seek to better inform the user without requiring manual input.

View Billing Statements

- Customers would be able to view billing statements, past, present, and future, to allow them to see what they have paid and what they will pay in the future. Because this data is stored in a database, retrieving this data is now an automated process instead of requiring communication with an employee who would then have to mail this data by hand. However, should an employee need to adjust details about a customer's billing statements, they have the capability to edit these database entries as needed (e.g. address, payment method for the customer view and amount due, power consumed for the statement view)

Bill Summary / Usage Statistics

- In addition to providing billing statements on a month to month basis, this project seeks to generate detailed billing summaries to give customers and employees alike a quick and detailed view on any given customer's power consumption. Such information includes cumulative payments made, max/min power consumption throughout the year, changes in cost per unit, etc. Customers will only be allowed to view the billing summary attached to their account, while employees will be able to view the billing summary for any given customer. The billing summary will not be saved off as a document, rather it will be generated based on the current data in the database as it is requested.

Administration Tools

- A suite of administrative tools will be provided to allow admins to perform investigation and maintenance on the project as needed. Such use cases include updates to the project, adjustment of data in the database, and general updates as new features/enhancements are requested. When these use cases are exercised, administrators will have full access to the debugging provisions required to ensure that these changes are stable, scalable, and do not change any existing behavior unless explicitly intended. Assuming these checks pass, then the administrator would be able to deploy their changes so they will be realized in production.

Turn On/Off service

- Previously, turning on or off electrical service to a given customer required a worker to manually trigger a device attached to the customer's home. Now that these devices are networked, we can take advantage and automate this process based on given criteria. When a customer registers an account, assuming they are in good financial standing, their service can be automatically enabled. Conversely, if they miss a set amount of payments, their service can automatically be disabled, pending the customer making good on their missed payments. Notifications of each of these state changes can be sent to the customer via the push notification system (see below).

Set and Adjust Prices

- In response to environmental conditions, administrators are given the ability to set and adjust prices for a given unit of power. Furthermore, the amount due for a given billing statement will also be editable by the administrator (e.g. discount) as needed.

Track Power Usage Over Time

- Modernizing the electricity billing system requires a customer's power consumption to be automatically tracked over time. This means that the networked device attached to a customer's home will be polled for units consumed, and this will be used in conjunction with the price per unit to generate a billing statement for a given customer in a given month. In practice for this project (as there are no real homes we can poll), we will generate normally randomized data to simulate power units consumed and store them per user to allow for billing statement/summary generation.

Push Notification System

- There are situations in which an administrator/employee would need to inform the customer immediately without waiting for the customer to log on to the billing system. In situations such as these, a push notification system is provided to allow for information to reach the customer in a timely fashion. Currently, outages, payment due, and payment overdue will each incur their own push notification automatically, as well as any manual notifications an administrator/employee deems necessary.

Permissions System

- Given the sensitive nature of the data being handled, a robust permissions system is needed to make sure that data is only accessible by those it is meant to be accessed by. Throughout this statement, we have roughly defined three classes of users. Customers, Employees, and Administrators. Customers should only be able to access data relevant to them and no one else. Employees should be able to access data of customers they are currently assisting, as well as general usage statistics that themselves are an aggregation of all user data. Employees will only be able to edit user data under strict requirements. Lastly, Administrators are given carte blanche access to the project code itself as well as all accompanying user data.

Accept Payments

- As this project seeks to modernize the current billing system, we intend to automate the payment process away from manual, paper billing. Thus, when a billing statement is issued, the customer is able to pay on-line with no employee aid using their preferred method of payment. Built into this payment system is a calculation of late fees, overdue balances, and any other pricing concerns needed to calculate a final amount due per customer per statement.