

An Event Driven Utility Billing System

Nilotpola Sarma, Roll no - 1501034

Vision

What is the product, on a high level?

The product is a Web Application which implements an utility billing system for services like electricity supply or water supply which is based on recording events on the data rather than storing the bill itself. In this product Consolidation/Correction becomes easy, Redundancy is removed and proper audit trail is always available.

Whom is it for?

It is for any utility service provider where billing is done periodically(say monthly) based on consumption, achieve full automation in their billing system including easy correction

What problem does its solve?

It solves the following problems present in the commonly used billing systems:

1. Lack of an Audit Trail (Corruption): No data update or delete is done in the project.
2. Retrospective Effect of Correction/Revision (Consolidation): Change in tariff can be modified from earlier date.
3. Unavailability of History of Consumer.
4. Redundancy.

What alternatives are available? Available alternatives may include any non-Event-Driven Utility Billing System. Most existing systems do not store the previous information of the customers in the form of an event. However they store the previous bills. Since they do not intend to have an event-based system, they allow direct editing of bills or other parameters like meter parameters etc. Because of these, the events cannot be retrieved from the system as some of the events are lost/overwritten. As such, calculation of bill when retrospective event exist can not be done automatically and hence manual intervention is required. This puts restrictions on tariff revision also.

Why is this project compelling and worth developing? It involves the use of Checkpointing and Rollback using a event based approach to Database Management. Hence it is compelling and worth developing.

The top-level objectives

differentiators: the product has an audit trail, easy correction and no redundancy.

target customers: any utility service provider.

scope of the product: it is a step toward fully automated utility billing system.

Competitive analysis:

Cost of storing the Events:

For one consumer, one row of events table occupies : at most 5×8 bytes. (suppose)

therefore billing information of one year occupies : $12 \times 5 \times 8$ bytes

therefore billing information of 10 years occupies : $10 \times 12 \times 5 \times 8$ bytes = 4.8kB(Which is very less).

For 1,00,00,000(one crore) such customers data size will be 480,000,000 = .48GB!!

What is novel about your product: It is free from data modification and fully automated.

Software Architecture

It is clear that the system can be built, making excellent use of the available resources and technology.

What is the product architecture? It is based on a Relational Database with a total of eleven

Events Table:

eventId	consumerId	eventTime	entryTime	enteredBy	eventType

Meter Reading Table:

eventId	reading

Payment Table:

eventId	amount	mode

Consumer Type Change Table:

eventId	newType

Load Change Table:

eventId	newLoad

Meter Change Event Table:

eventId	newMeterNo	newMeterReading	oldMeterReading

Status Table:

consumerId	load	meterReading	readingDate	consumerType

relations among which the Events Table is the base relation and other relations are inherited from them. Here I shall put to use the object oriented feature of the Java Language.

The components / modules that will interact in the system: A set of eleven relations shall interact amongst themselves through the base relation which is the Events Table through the eventId. The diagram of the relations in the database is shown on the left:

Consumer Table:

consumerId	consumerName

Billing Table:

eventId	billingDate	dueDate	amount

Tariff Change Table:

eventId	wefDate	Slab	consumer Type	rate

Users Table:

userId	name	designation	role

What is interesting about this project from a technical point of view? It is a very simple idea and uses tools like JEE and basic Database concepts.

Optionally, what languages/toolkits do you propose to use for the development? I propose to develop this in Java (Java Enterprise Edition) using an MVC framework(Spring).

Challenges and Risks

What is the single most serious challenge you see in developing the product on schedule? The time available for creating the product may become a bottleneck.

How will you minimize or mitigate the risk? I plan to remove certain features like change of meter etc from the product if time does not permit.

XXXX