iJRASET
International Journal For Research in
Applied Science and Engineering Technology

# INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

**Volume:** 9 **Issue:** VIII **Month of publication:** August 2021

**DOI:** https://doi.org/10.22214/ijraset.2021.36224

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

# Digital Billing and AI-based Expense Tracking System

Gaurav Raj[1], Bhagyesh Dholaria[2], Devanshu Mistry[3], Harsh Vaghasiya[4], Prof. Pramod Mundhe[5]
*[1, 2, 3, 4, 5]Computer Science and Engineering Department, School of Computer Engineering and Technology, MIT World Peace University, Pune, India*

***Abstract: Digital billing is the next big step towards sustainable devolvement and promotion of the idea of self-budgeting. As conventional printed paper invoices are prone to damage and being misplaced, an electronic solution would allow them to preserve and exchange invoices without manual intervention. This approach would ensure shorter payment delays, fewer errors and reduced printing costs. In this paper we present our idea of a unified web-based digital billing system for exchanging invoices between vendors and customers. Our application allows the users to manage all of their invoices at a single place and make use of our in-built machine learning model to predict trends in their sales and purchases.***

## I. INTRODUCTION

The traditional paper printed invoices being used are a major issue in the longer run as they make use of a lot of paper and printing procedures, this causes an increase in the invoicing expense and incurs a huge toll on the environment. In this paper we present a unified web-based digital billing system for exchanging bills between vendors and customers.

Our platform would allow both the vendors and the customers on board, where they can manage their sales and purchases respectively. The invoices generated between the vendor and the customer act as a link between them for each of their transactions. A machine learning model is integrated onto the platform allowing the vendors and customers to forecast their future sales and purchases respectively. The model works upon the user's historical transaction data to predict their future transactions.

The major features present in our model are the Vendor's Sales Prediction and the Customer's Expense Prediction. We studied and executed multiple prediction algorithms according to our platform's use case.

First, we tried ARIMA (Auto Regressive Integrated Moving Average), one of the easiest machine learning algorithms to perform time series forecasting. But while integrating it with our system we found that the long term forecasts were eventually becoming a straight line and hence had trouble predicting.

Then, we tried to implement LSTM (Long Short Term Memory) which is well-suited to classify, process and make predictions based on time series data, it also takes into account if any lags of unknown duration are present between important events in a time series. We found the output to be over fitted and not up to the par. Lastly, we shifted to Facebook Prophet. It is a procedure for forecasting time series data based on an additive model, where nonlinear trends are integrated with yearly, weekly, daily seasonality and holiday effects. This algorithm yielded us the best results on our platform. We utilized Facebook Prophet for both Vendor and Customer's forecasting model.

Considering the participants, Digital Business operations are generally divided into B2C (Business-to-Consumer), B2B (Business-to-Business), and B2G (Business-to-Government). Our system aims at providing a digital answer to B2C (Business-to-Consumer) and B2B (Business-to-Business) exchange of invoices.

We hope to drastically reduce the carbon footprint of paper printed invoices and alleviate its impact on the environment by implementing our digital solution and ideally completely seize the usage of papers for invoices. Our solution would also promote the idea of personal money management or self-budgeting within individuals. As this would be a great help for the younger generation by familiarizing them with the idea of self-budgeting.

The 2020 COVID-19 pandemic has also given us a big reason to promote digital billing in order to reduce the spread of virus via physical contact through the printed bills. As they increase the chances of individuals getting infected with the virus during an exchange.

The remainder of the paper is organized as follows. In Section II, functional and non-functional specifications, the system architecture and database model specification are explained. In Section III, Machine Learning Model for Sales and Expenditure Forecasting is presented. Section IV describes system development, the use of the application and XSLT transformations. In Section V the future scope and improvements are presented.

## II. SYSTEM ARCHITECTURE

### *A. Functional Specifications*

*1)* Authentication of user whenever he/she logs into the system.
*2)* Registered email of the user acts as a unique identifier across the system.
*3)* Generation and viewing of a Digital Invoice for registered vendors and customers.
*4)* Users can configure their personal and store's details.
*5)* Vendors can utilize the Machine Learning Model to predict their future sales.
*6)* Vendors can manage their store's inventory stock.
*7)* Customers can utilize the Machine Learning Model to predict their future purchases.
*8)* The generated bill can't be altered or deleted once created and can only be accessed by the linked vendor and customer.
*9)* The bill can be viewed easily in the application interface and can be exported in both PDF and Image format.
*10)* The Fig. 1 contains a Use Case diagram depicting the basic functionalities available to a registered vendor and customer on the platform.

Use case diagram shown in Fig. 1 represents functional requirements for creating and receiving digital invoices for vendors and customers respectively and methods that it includes or extends. It can be seen that the application checks session validity and user privileges almost every time an action is about to be executed. Besides that, every method that connects to the database validates the entered data and executes an SQL query. After committing the data into the database, invoice is generated and can be viewed by both customer and vendor.

### *B. Non-Functional Specifications*

*1)* Verification and validation of the input data at multiple levels such as before storing in the database or execution of an action.
*2)* Verification and validation of user privileges or session before the loading an interface or execution of any action
*3)* The session expires after a defined time.
*4)* The Machine Learning model requires at least 15 invoices both for the vendor and the customer to work effectively.
*5)* Cross-platform compatibility and can be run on any device with a web browser.
*6)* Need active internet connection for usage.
*7)* Derivation Queries are carried out during non-active hours on the database.
*8)* Easy-to-use interface.

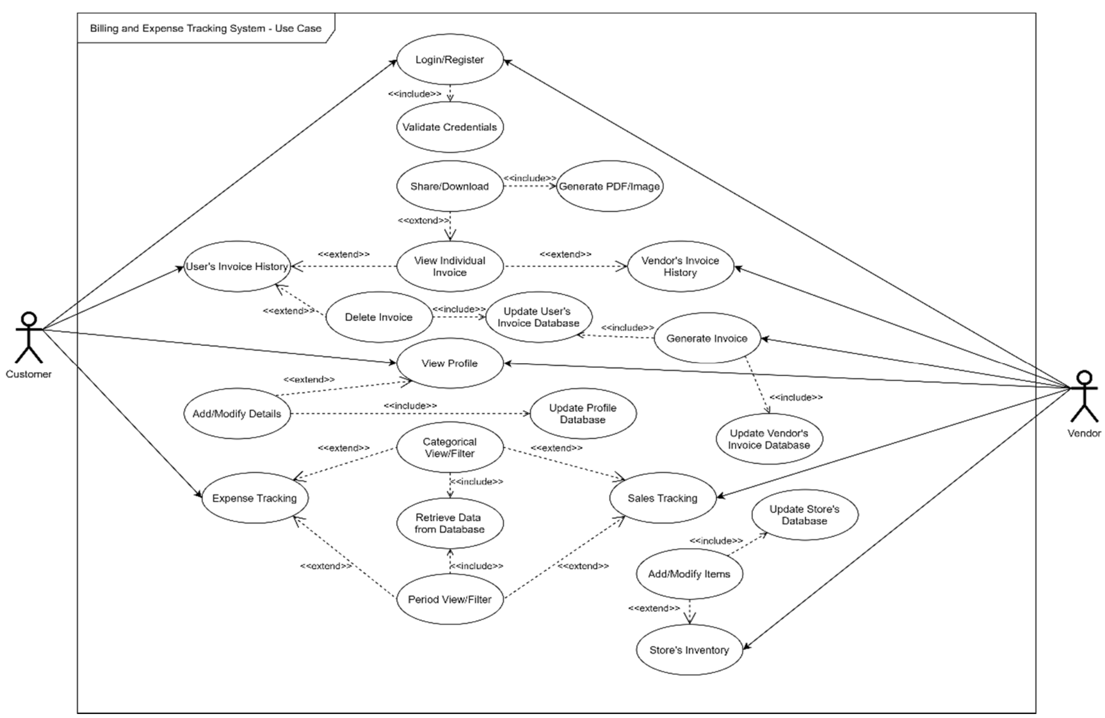


*Figure 1: Use Case Diagram*

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

### *C. System Architecture Diagram*

The Digital Billing system consists of a central web application connected directly to a web service, as shown in Fig. 2. The application also has different interfaces for different types of users (i.e. vendors and customers). The vendors and customers are linked to each other for every transaction between them via the generated invoice. This invoice is preserved in our database and can only be accessed by the respective vendor or customer linked to the invoice.

The database is hosted on AWS RDS platform. The data can be accessed from AWS with SQL-Alchemy Interface and can be sent to React.JS using FastAPI. Similarly when a user wants to save some data in the database (inventory management, invoice generation, etc.) the data can be sent through FastAPI and will be validated and committed to the database. Monthly Expenditure Prediction for customer and Sales Prediction for vendors will be carried out by Machine Learning Algorithms. ML engine is already trained by different datasets to predict the desired output for a particular user.

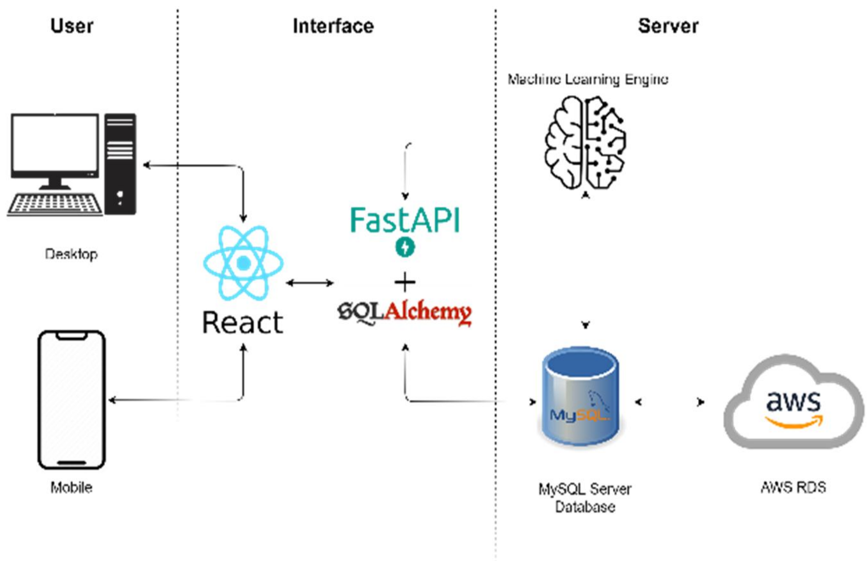


*Figure 2: System Architecture*

### *D. Database Model Specification*

The database model is shown in Fig.3, was designed keeping all the normalization rules in mind for efficient query operations. The end-user's details are stored in the *Vendor* and *Customer* table based on whether they are a vendor or not. The *Customer* table contains the user's details that are common to both a vendor and a customer, whereas the *Vendor* table contains details related to only the users who are a vendor. The *vID* in the *Vendor* table is foreign key that links it to the *uID* of the *Customer* table creating a one-to-one relationship. The *Items* table contains details pertaining to the store items inserted by the various vendors who are registered onto our platform. The item details in this table are contributed by the vendors themselves and are globally available to each one of the vendors. These items are used in the *InvoiceDetail* and *Inventory* table for defining the contents of an invoice and available stock of a vendor's store respectively. The generated invoice's data is stored in the *Invoice* and *InvoiceDetail* table. The *Invoice* table contains *invoiceID* as primary key and links the customer and vendor associated with that invoice via *uID* and *vID* as foreign keys creating a one-to-many relationship. It also contains the Date, Time and Total Amount of the generated invoice. The individual items of the invoice are stored in the *InvoiceDetail* table. This table links an invoice with individual items present in it using the *invoiceID* and *itemID* as foreign keys creating a many-to-many relationship. It also contains the quantity and price discount associated with the individual items in that invoice. The *Inventory* table defines the available stock for each registered vendor's store. The *vID* is linked with the *itemID* and *item_qty* in this table. Here, *item_qty* defines the quantity of that item present currently in the vendor's store. These item quantities individually are updated whenever an invoice is generated with respective items present in it or whenever the vendor adds more item stock manually.

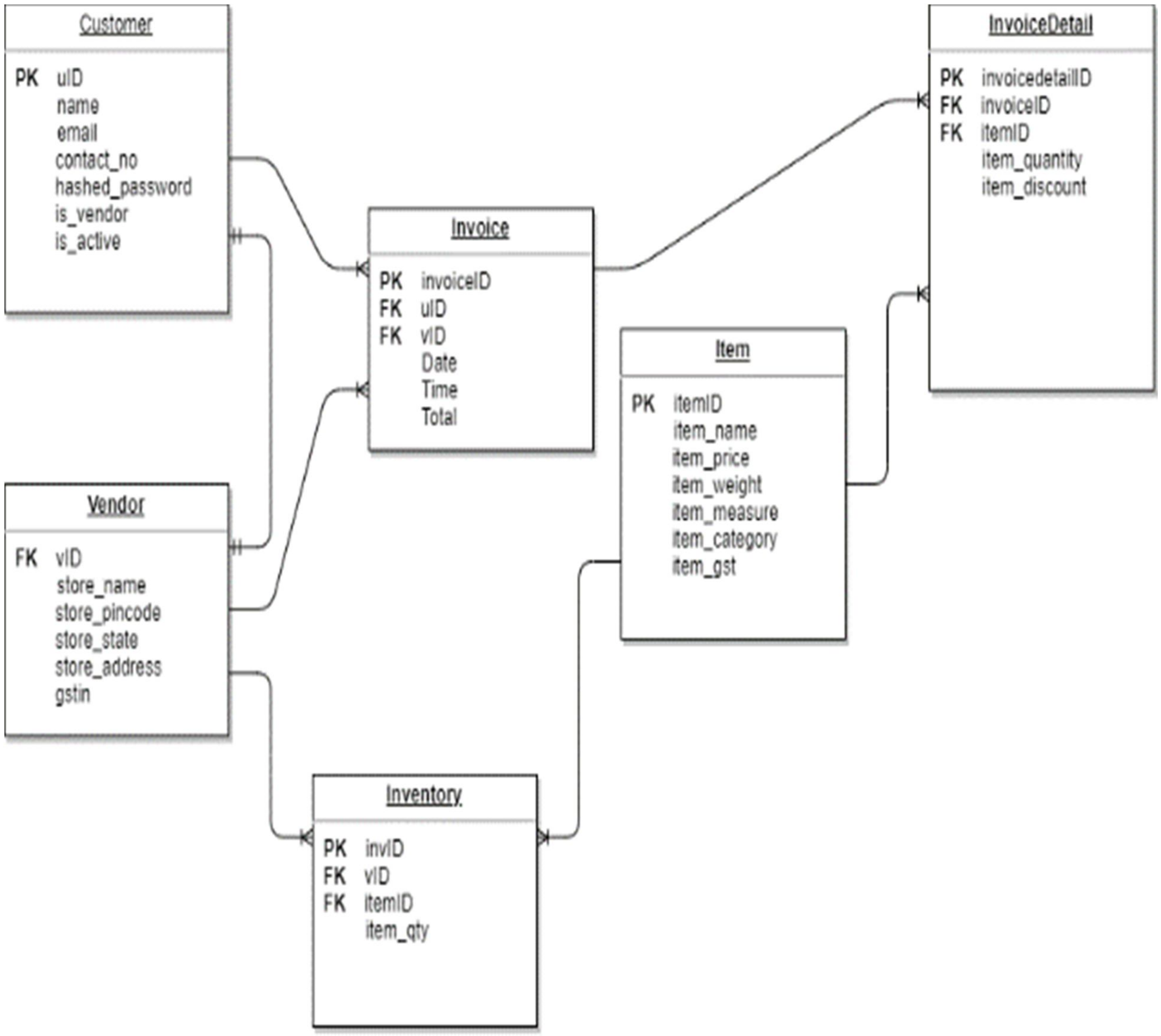


*Figure 3: Database Schema*

### *E. Sales and Expenditure Forecasting*

Forecasting is a common task that helps companies, sellers, and people with capacity planning, goal setting, and anomaly detection. Despite its importance, there are challenges associated with producing reliable and quality forecasts - especially when there are a variety of time series and expertise in time series modelling are relatively rare. For our application we have local vendors, shop owners and general public who generally follows a pattern

Prophet is a tool developed by Facebook for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects on target value. It works best with time series that have strong seasonal effects and several seasons of historical data. Prophet is robust to missing data and shifts in different trends, and typically handles outliers very well than other methods.

In order to have a quality trained model which can provide near-to-actual prediction over a wide range of patterns and data we are decomposing the formula and creating the dataset having every component variable.

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

The Prophet model has the form:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t$$

Where:

$g(t)$ is the trend function?

$s(t)$ is the periodic component (seasonalities)

$h(t)$ represents holidays/events which occur on a potentially irregular basis.

$\varepsilon_t$ is the error term (which is often assumed to be normally distributed).

1) *Trend Function*

The basic model for this term is:

$$g(t) = \frac{C}{1 + e^{-k(t-m)}}$$

Where:

$C$ is known as capacity.

$k$ is the growth rate.

$m$ is the offset parameter

Also, $\lim_{t\to\infty} g(t) = C$

2) *Seasonality*

The seasonal components are approximated by Fourier modes: The equation is as follows:

$$s(t) = \sum_{n=1}^{N} \left( a_n \cos\left(\frac{2\pi nt}{P}\right) + b_n \sin\left(\frac{2\pi nt}{P}\right) \right)$$

Where:

$P$ is the period.

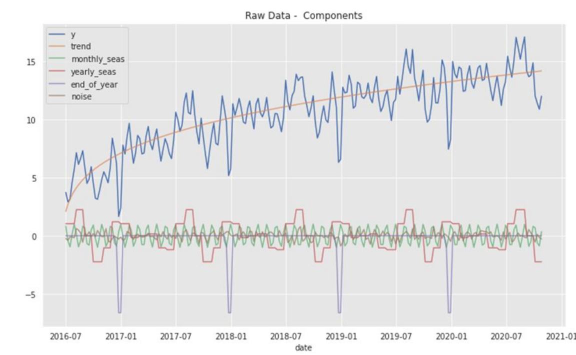


*Figure 5: Raw Data Components Graph*

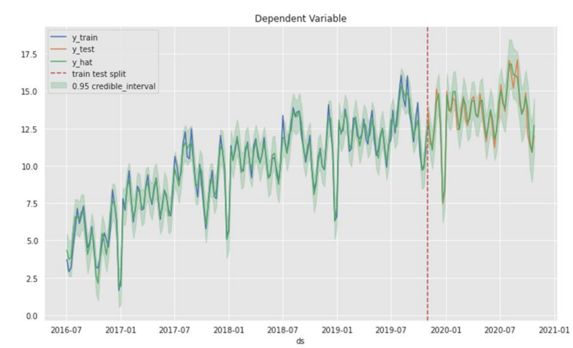


*Figure 5: Dependent Variable Graph*

3) *Holidays and Special Days:* In order to have the additive effect of holidays and other special days on forecasted values we need to specify the date-time formatted values. Python does provide libraries that enable us to retrieve the country wise list of holidays. We can add that data frame into the prophet model to make it work in addition to the seasonality.

Fig.4 shows the individual raw components of $y(t)$ and reflects that the additive model id functioning correctly. Fig.5 shows that the predicted value is coinciding well with the actual value and the predicted value is well placed within 95% credible interval.

## III. FUTURE SCOPE

We aim to bring this same platform onto the mobile devices natively as an Android / iOS Application. This would drastically improve the user base as most of the users would find accessing our services via mobile devices. Another ground-breaking feature we wish to integrate in future would allow the customers themselves to configure their shopping list using our globally available items list and exchange that information with the vendor directly via a QR Code. This feature would alleviate the vendors as they won't have to configure the invoice with each item requested by the customer. They can just scan the code from the user's device to get their complete shopping list and then approve it to generate the required invoice. In 2020, the Indian Government modified rules regarding the generation of electronic bills generated within India. They created a regulated framework which made it compulsory for every electronic invoice and e-way bill generated to be approved and compatible with their system. We aim to make platform directly compatible with this newly implemented framework, making it easier for the end-users to satisfy this regulation instated by the Indian Government. Bill. These upcoming features would make our platform more robust, widely acceptable and will be key to attract more people towards digital billing.

## IV. CONCLUSION

In this paper we presented our Digital Billing system, aimed at enabling enterprises, especially small and medium scale enterprises with a simple access to digital invoicing. A web application for generating, receiving and viewing digital invoices was developed.
Digital invoicing allows enterprises to benefit from reduced printing costs and fewer errors. Apart from generating invoices this application also consists of interactive dashboards which helps vendors and customers to keep track of their sales and expenses respectively.
The machine learning algorithms add a new dimension to this application by providing expenditure prediction for customers in order to promote self-budgeting. For vendors they can predict their future sales and manage their inventory according to forecasted demand. We hope this system makes the adoption of e-invoicing easier for people and alleviates the environment at the same time.

## REFERENCES

[1] Huang, Q., & Zhou, F. (2017, March). Research on retailer data clustering algorithm based on spark. In AIP Conference Proceedings (Vol. 1820, No. 1, p. 080022). AIP Publishing.
[2] Rey, T. D., Wells, C., & Kauhl, J. (2013). Using data mining in forecasting problems. In SAS Global Forum 2013: Data Mining and Text Analytics.
[3] Maingi, M. N. A Survey on the Clustering Algorithms in Sales Data Mining.
[4] Sastry, S. H., Babu, P., & Prasada, M. S. (2013). Analysis & Prediction of Sales Data in SAP-ERP System using Clustering Algorithms. arXiv preprint arXiv:1312.2678.
[5] Shah, N., Solanki, M., Tambe, A., & Dhangar, D. Sales Prediction Using Effective Mining Techniques.
[6] Flesch, B., Vatrapu, R., Mukkamala, R. R., & Hussain, A. (2015, October). Social set visualizer: A set theoretical approach to big social data analytics of real-world events. In Big Data (Big Data), 2015 IEEE International Conference on (pp. 2418-2427). IEEE.
[7] Asooja, K., Bordea, G., Vulcu, G., & Buitelaar, P. (2016). Forecasting Emerging Trends from Scientific Literature. In LREC.
[8] V. Singh, L. Freeman, B. Lepri, and A. Pentland, "Predicting spending behavior using socio-mobile features".