ISSN (Online) 2278-1021
ISSN (Print) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering
Vol. 4, Issue 4, April 2015

Saasprin Saas based accounting made easy

Praveen Shukla[1], Kunal Singh[2], Monika Pandey[3], Narendra Kumar[4], Prof Seeta Yadav[5]

Army Institute of Technology, India[1,2,3,4]

Professor, Army Institute of Technology, India[5]

Abstrac: SaaS is used by most of the applications run under any project. The SaaS providers charge the companies and the SaaS users for the same. Bills and invoices are produced based on the usage and sent to the users. The accounting team solves the hassle of sorting and compiling the invoices and paying them. SAASPRIN is the software for the automation of this very process of attending and calculating the amount to be paid mentioned in these invoices. Considering the logical flow of data there are many factors which have been taken care of in designing of Saasprin. Analysis of which regular expressions to be used for scraping the relevant data from the invoices, which all methodologies can be employed for effective and self-explanatory representations of bill amounts and expenses are few among such factors .Also cost effective and fast cryptographic algorithm have been analyzed and used for better security and privacy. Saasprin is made as an open source software and can be customized as per the user requirements.

Keywords: Saasprin; Saas;Regular expression.

I. INTRODUCTION

We use SaaS i.e. software as a service for so many varying applications. Services like GitHub, Amazon Web Services, and LinkedIn are used widely used. The SaaS providing firms send their bills and invoices to the companies as per their type and duration of usage. Since different projects start at different times of the months the starting of accessing of SaaS by them also varies. In the end of the month the accounting department of the companies asks the project teams to give their invoices so that it can be paid on time for uninterrupted processing. Now this gets quite hectic for the accounting team to collect all the invoices, compile them, find out certain required pattern of expenditure e.g. which team is using the maximum of all, which SaaS provider is being used the most etc. and in the end pay the bills on time. The product we have tried to make solves this very problem so that it can be assist with the task of processing the invoices.

II. FUNCTIONING OF SAASPRIN APPROCHES

A. E-mail

First approach deals with the most common way of receiving the invoices. Invoices are sent to the registered e-mail id, from there the invoices are directed to a dummy e-mail box which is meant for compiling all the mails which has our concerned mails containing the invoices or bills. Different methods of doing so include use of MailGun. These are powerful APIs that enable you to send, receive and track email effortlessly. At the heart of MailGun is the API. Most of the MailGun service can be accessed through the Restful HTTP API without the need to install any libraries. However, Libraries have been written for many popular languages. We have to be sure to check out the additional capabilities provided by using the libraries. In addition to the API, MailGun supports the standard SMTP protocol. These can be in various formats like pdf, image extensions like jpeg etc. The next task is to scrape the relevant figures and data from these formats and send to the tables containing the records of amount. For different formats different methodologies are to be used for extracting the useful data. This is one of the important portion of Saasprin, ‘Scraping of data’.

B. Automation

Sometimes we also have to move to the websites of the providers and log into our accounts and then view our invoices. This process is observed for the second approach. Here the steps are recorded and an automated system repeats the steps for getting the invoices. Once after the invoices are collected at one place it is processed like any other invoice. The processing of invoices are defined in depth later in this paper. In another approach we also use Web APIs which helps in the process of scraping relevant data from HTML pages.

III. THE LOGICAL FLOW OF DATA

Now the logical flow of data will explain the exact processing in Saasprin.

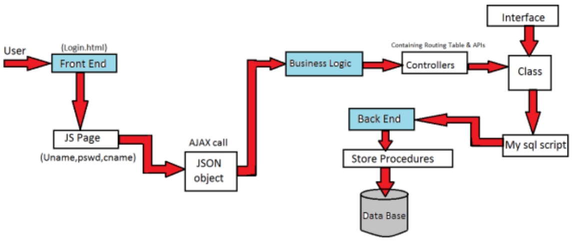

Figure 1: The Logical Flow of Data

As can be referred from the diagram the user completes the login action in the frontend with giving the input attributes username, company name, and password. The three attributes passed through JavaScript creates a JSON object. These three attributes fields are compulsory for creating an object. After object creation AJAX call is made for calling the API from the Business Logic. In the Business Logic, the API (with the routing tables) call the controller classes. These classes implement required interfaces as per the requirement. The controllers contains the API and as per the requirement which is mention in the

DOI 10.17148/IJARCCE.2015.4427

AJAX call it hits the specific API ,now this API redirect the flow to the SQL script . This SQL script is sent to the backend. Here in the backend the main database is referred through the store-procedure. The query fired on the database retrieves the required data, for which the user queried for. The retrieved data is sent through the backend to front-end and user will get the result as per their requests which is accessible to the user. During the flow of data there are many other factor which are to be taken care of. These factors include the involved controllers, the issue of security leakage etc.

A. Use of Controllers and Drivers

Drivers can be called as an extension to controllers. We have used PhantomJS as our headless browser. PhantomJS is a scripted, headless browser used for automating web page interaction. PhantomJS gives a JavaScript API that enables automated navigation, screenshots, and behavior of users and assertions making it a common tool used to run browser-based unit tests in a headless system like a continuous integration environment. PhantomJS is based off of Web kit making it a similar browsing environment like the common existing web browsers. It is open-source software released under the BSD License. Since PhantomJS is using Web Kit, a real layout and rendering engine, it is capable of capturing a web page as a screenshot. Since PhantomJS can render anything on the web page, it can be employed to convert contents not only in HTML and CSS, but also SVG and Canvas.

Web browsers can read HTML files and compose them into visible or audible web pages. Browsers without displaying the HTML tags and scripts, use them to interpret the content of the page. So HTML is used in our application files as controller. We have also scraped data from the HTML copy of invoices. As is known HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language. The Hypertext Transfer Protocol (HTTP) is an application protocol for distributed, collaborative, hypermedia information systems. HTTP is the foundation of data communication for the World Wide Web. Hypertext is structured text that uses logical links (hyperlinks) between nodes containing text. HTTP is the protocol used for the exchange or transfer of hypertext. The main motive behind using this protocol has always been its basic utility. Whenever your web browser fetches a file (a page, a picture, etc.) from a web server, it does so using *HTTP* - that's "Hypertext Transfer Protocol". HTTP is a request/response protocol, which means your computer requests for some file (e.g. "I want the file 'A.html'"), and the web server sends back a response ("Here's the file", followed by sending the file itself). That request which your computer sends to the web server contains all sorts of (potentially) required information.

B. Factors to be considered while using ASP.net

- Application files or Websites written in ASP.NET might contain security vulnerabilities that remain invisible to the owner. So detection of security vulnerabilities is one among the crucial issues to be considered. The algorithm which performs a scanning process for all website or application files is required. Our scanner tool relies on studying the source code of the application depending on ASP.NET files and the code behind files (Visual Basic VB and C sharp C#). A program written for this purpose is to generate a report that describes most leaks and vulnerabilities types (by mentioning the file name, leak description and its location). The suggested algorithm will help organization to deal with and fix these vulnerabilities and improve the overall security.
- Also ASP.NET AJAX framework provides the facility of JSON serialization features to the ASP.NET web-services from client-side JavaScript, even using third-party JavaScript-libraries like jQuery.

ASP.NET supports all the common and popular browsers like Google Chrome, Opera, Netscape and Internet Explorer.

IV. PROCESSING OF INVOICES

The main challenge and most important step is the processing of invoices. Processing here refers to the scraping reading the relevant important data and recording them into tables. For this we use tabular structure for storage according arranged in rows and columns for the fields. The scraping here is performed with the use of regular expression. It proves to be the most the efficient measure for recognizing various data patterns and also sorting and saving important data. Study of various regular expression has been performed for choosing the best one to be implemented.

A. Defining the Determinstic Finite Automata

Pattern matching is important in several critical network services such as intrusion detection and pattern matching. As the complexity of rules to be considered increases, conventional string matching techniques are replaced by advanced regular expressions. To keep up with line rates, deal with denial of service attacks and provide predictable resource provisioning, the design of such engines must allow examining payload traffic at several gigabits per second and provide worst case speed guarantees. While regular expression for pattern matching is a well-studied theoretical concept, the memory requirement for the same have always been a “to be solved” issue. This is especially true for DFAs representing complex regular expressions present in practical rule-sets. That’s why we have used a novel method to reduce the DFA memory requirement and still provide worst-case speed guarantees. We have used regular expressions not only for scraping data from invoices or from their HTML formats but also for the validation purposes in the scenarios like “log in” time (contact no., email id etc.). So memory requirement and consumption have been considered here also. We focus on reducing memory usage of composite DFAs by compressing transitions. Based on the observations that more than one states may move to the same next state. This is observed many a times especially at the time of string matching and string acceptances. These algorithms aim at reducing redundancy between states.

DOI 10.17148/IJARCCE.2015.4427

Along with this it also serves the purpose of complexity in designing DFA for a particular problem and memory management. However, in this paper we try to obtain memory reduction by exploiting transitions redundancy between states and transitions distribution inside states. We have also thought about transitions compression from two-dimensions. We start our work with the observation that although each state may have relatively high number of different next-states, its transitions concentratively transfer to a two clusters concentratively. Based on the observation, we adjust transitions in every cluster separately by its extracting base value for each state, which can introduce more identical transitions between states. This can be understood in a better way if we think of the practical and physical aspects like memory consumptions.

B. Data Visualisation

Data visualization is viewed by many disciplines as a modern equivalent of visual communication. It is not owned by any one field, but rather finds interpretation across many (e.g. it is viewed as a modern branch of descriptive statistics Data vi Data visualization or data visualization is viewed by m Data visualization or data visualization is viewed by many disciplines as a modern equivalent of visual communication. It is not owned by any one field, but rather finds interpretation across many (e.g. it is viewed as a modern branch of descriptive statistics by some, but also as a grounded theory development tool by others). It involves the creation and study Of the visual representation of data, meaning "information that has been abstracted in some schematic form, including attributes or variables for the units of information" any disciplines as a modern equivalent of visual communication. It is not owned by any one field, but rather finds interpretation across many (e.g. it is viewed as a modern branch of descriptive statistics by some, but also as a grounded theory development tool by others). It involves the creation and study of the visual representation of data, meaning "information that has been abstracted in some schematic form, including attributes or variables for the units of information". Visualization or data visualization is viewed by many disciplines as a modern equivalent of visual communication. It is not owned by any one particular field, but rather finds interpretation across many platforms (e.g. it is viewed as a modern branch of descriptive statistics by some, but also as a grounded theory development tool). It includes the making and study of the visual representation of data, which means "information that has been abstracted in some schematic form, including attributes or variables for the units of information". Here the heads, are organizational devices that guide the reader through your paper. There are two types: component heads and text heads. There main functioning are as depicted by their names respectively.

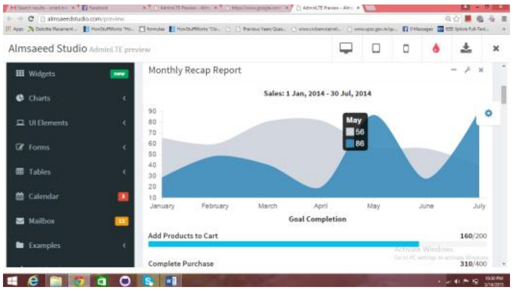

Figure 2: Dashboard Representing Comparative Reports for better Human Readability

CONCLUSION

The very idea of designing Saasprin emerged out of the need of an automated system which will reduce the tedious task of the accounting team. As mentioned earlier for collecting invoices for the SaaS availed by different teams performing projects, running at different time cycles of the month, it becomes too a hectic task. So the product will be a beneficial live tool helping each and every department and industry for their accounting and bill payment. Saasprin will also help team leads to have a track of their consumptions and further requirements. The dashboard is the self-explanatory interface telling monthly usages. Further line graphs are also presented comparing bill amounts of present and previous months. More over extrapolating the graph lines will also help in estimation of future consumptions.

FUTURE SCOPE

The modern era is the era of sharing knowledge. The era of open source. So we aspire to make Saasprin compatible to open source deployment in future. Thus Saasprin can also be customized and used according to respective requirements by professionals. Saasprin will also help team leads to have a track of their consumptions and further requirements. The dashboard is the self-explanatory interface telling monthly usages. Further line graphs are also presented in manners comparing bill amounts of present and previous months. More over extrapolating the line graphs will also help in estimation of future consumptions and bill amounts. Also if we have a clear idea about the above mentioned aspects, we can also aim for creating or searching for alternatives for betterment of the resource utilization. Representations like line-graphs, donuts and bar graphs can also be used for recording, estimating and keeping track of resource utilization. Thus Saasprin can be brought into use according to the specifications and requirements of the users. It can also be standardized after modifying from the generic useful format we have made.

REFERENCES

[1]. R. S. Boyer and J. S. Moore, "A Fast String Searching Algorithm," Communications of the ACM, pp. 762–772, 1977.

[2]. R. Rivest, "The RC5 encryption algorithm," in Fast Software Encryption, ser. LNCS, vol. 1008. Springer-Verlag, 1995, pp. 86–96.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[3]. D.Conte,P.Foggia,C.Sansone,andM.Vento. Thirty years of graph matching in pattern recognition. Int. Journal of Pattern Recognition and Articial Intelligence, 18:265–298, 2004. 1

[4]. J. E. Hopcroft, R. Motwani, and J. D. Ullman, Automata Theory, Languages and Compilation. Addison Wesley, 3rd ed., 2004.