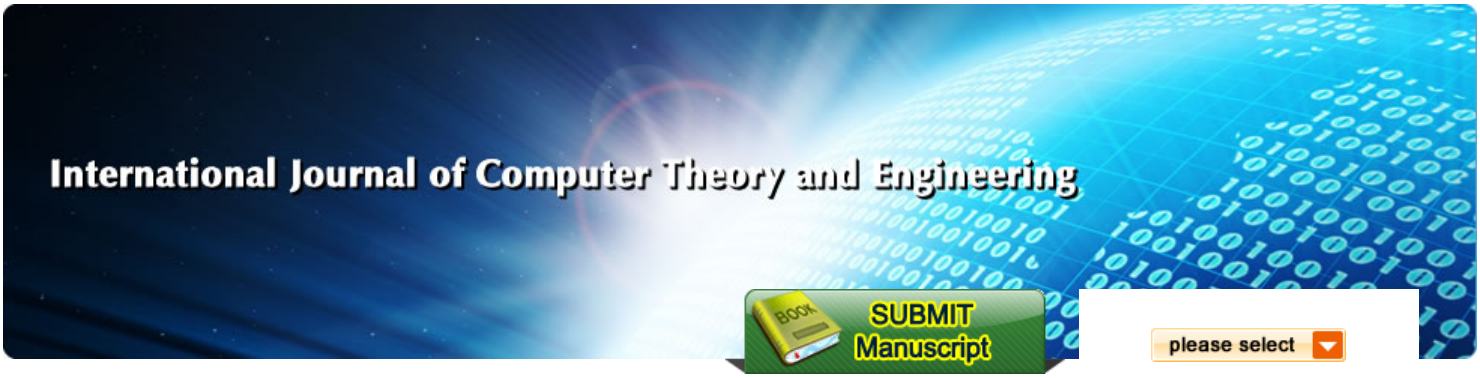
ISSN: 1793-8201



What's New

Jun 03, 2019 News! **Vol.9, No.5-Vol.10, No.3 have been indexed by EI (Inspec).** [Click]

Jan 19, 2020 News! Vol.12, No.1 has been published with online version. [Click]

Dec 11, 2019 News! Vol.11, No.6 has been indexed by Crossref.

General Information

ISSN: 1793-8201 (Print)

Abbreviated Title: Int. J. Comput. Theory Eng.

Frequency: Bimonthly

DOI: 10.7763/IJCTE

Editor-in-Chief: Prof. Wael Badawy

Executive Editor: Ms. Mia Hu

Abstracting/Indexing: Index Copernicus, Electronic Journals Library, Google Scholar, Ulrich's Periodicals Directory, Crossref, ProQuest, WorldCat, EBSCO, and EI (INSPEC, IET), Cabell's Directories.

E-mail: ijcte@iacsitp.com

Editor-in-chief

Prof. Wael Badawy

Department of Computing and Information Systems Umm Al Qura University, Canada

I'm happy to take on the position of editor in chief of IJCTE. We encourage authors to submit papers concerning any branch of computer theory and engineering.

READ NOW

International Journal of Computer Theory and Engineering (IJCTE) is an international academic open access journal which gains a foothold in Singapore, Asia and opens to the world. It aims to promote the integration of computer theory and engineering. The focus is to publish papers on state-of-the-art computer theory and engineering. Submitted papers will be reviewed by technical committees of the journal and association. The audience includes researchers, managers and operators for computer theory and engineering as well as designers and developers.

All submitted articles should report original, previously unpublished research results, experimental or theoretical, and will be peer-reviewed. Articles submitted to the journal should meet these criteria and must not be under consideration for publication elsewhere. Manuscripts should follow the style of the journal and are subject to both review and editing.

Featured Article

Object-Oriented Research Framework for the Fireworks Algorithm with the Focus on the Travelling Salesman Problem

Robert Ehni and Carsten Müller

The fireworks algorithm is a lately developed algorithm based on fireworks in the night sky. It is a swarm intelligence algorithm with a broad range of use. In this paper the algorithm is introduced and discussed. Furthermore, an implementation of the algorithm for optimizing the travelling salesman problem is introduced. Prior to implementation the adjustments to the algorithm will be provided.... [Read More]

Interested in being a peer-reviewer?

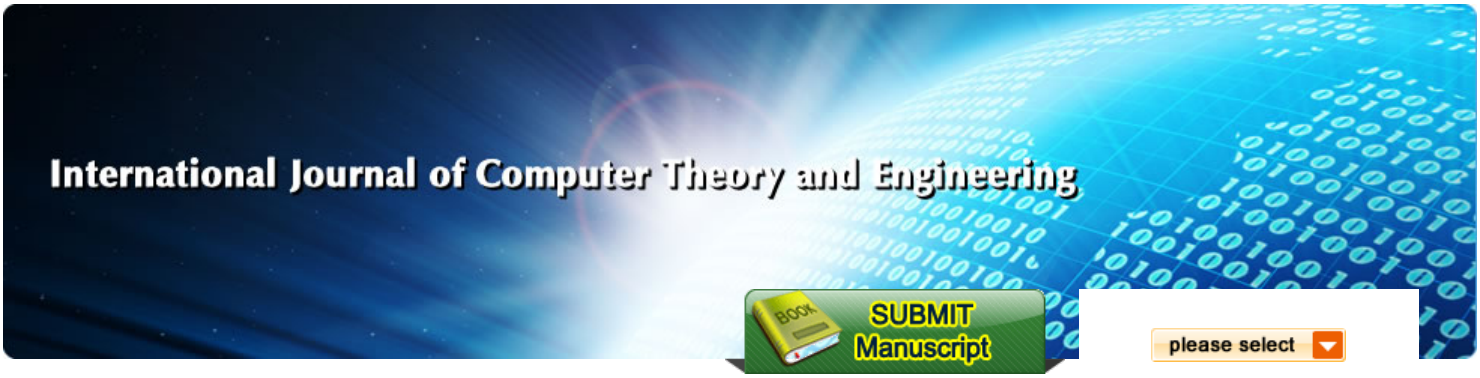
Send your curriculum vitae along with the application form to the editor, then your application will be processed ...

Frequently Asked Questions?

Q1. How to submit my research paper? What's the process of publication of my paper?

Engineering Village · crossref · ProQuest · doi SYSTEM · Electronic Journals Library · IET Inspec · EBSCO

IJCTE
ISSN: 1793-8201



Editor in Chief

Prof. Wael Badawy
Department of Computing and Information Systems, Umm Al Qura University, Canada

Editorial Board Members

Prof. Ayyaswamy Kathirvel, Karpaga Vinayaga College of Engineering and Technology, India
Prof. Chii-Jen Chen, Department of Computer Science and Information Engineering, Yuanpei University, Taiwan
Prof. K. Balasubramanian, European University of Lefke, Turkish Republic of Northern Cyprus, Turkey
Prof. Malik Sikander Hayat Khiyal, Preston University, Islamabad, Pakistan
Prof. Nabil Mohareb, United Arab Emirates university, UAE
Prof. Oscar Mauricio Caicedo Rendon, University of Cauca, Colombia
Prof. Vilas Lambture, Sir Visvesvaraya Memorial Engineering College, Nashik, India
Prof. Weimin He, University of Wisconsin-Stevens Point, USA
Prof. Yilun Shang, Tongji University, China
Prof. V. Thiagarasu, Gobi Arts & Science College, India
Dr. Ainuddin Wahid Abdul Wahab, Department of Computing University of Surrey Guildford, UK
Dr. Fayza A. Nada, IT Department, College of Applied Sciences, Al Rustaq, Sultanate of Oman
Prof. Kewen Zhao, University of Qiongzhou, China

What's New

Jun 03, 2019 News! **Vol.9, No.5-Vol.10, No.3 have been indexed by EI (Inspec).** [Click]

Jan 19, 2020 News! Vol.12, No.1 has been published with online version. [Click]

Dec 11, 2019 News! Vol.11, No.6 has been indexed by Crossref.

General Information

ISSN: 1793-8201 (Print)
Abbreviated Title: Int. J. Comput. Theory Eng.
Frequency: Bimonthly
DOI: 10.7763/IJCTE
Editor-in-Chief: Prof. Wael Badawy
Executive Editor: Ms. Mia Hu
Abstracting/Indexing: Index Copernicus, Electronic Journals Library, Google Scholar, Ulrich's Periodicals Directory, Crossref, ProQuest, WorldCat, EBSCO, and EI (INSPEC, IET), Cabell's Directories.
E-mail: ijcte@iacsitp.com

Editor-in-chief

Prof. Wael Badawy
Department of Computing and Information Systems Umm Al Qura University, Canada

I'm happy to take on the position of editor in chief of IJCTE. We encourage authors to submit papers concerning any branch of computer theory and engineering.

International Journal of Computer Theory and Engineering

CONTENTS

Volume 12, Number 2, April 2020

- **Software Design and Testing**

- **Data Analysis and Intelligent Business**

Distance and Price Validation System for Free Shipping on E-Commerce Using Batch Processing

Stephanie Pamela Adithama, Agitha Pramesti Sembiring, and Eduard Rusdianto

***Abstract*—PT XYZ is an e-commerce company that applies free shipping services. Every month the company will receive a shipping transaction invoice from third-party logistics, in the Excel file. The operational division will perform a manual validation process between distance and price with data in the Excel file. With manual processes, several problems arise, such as long processing time and errors caused by human factors. The validation system will use batch processing to process extensive data without interruption and will be executed using the Spring Batch framework. The programming language used is Java with a Spring Boot framework. The system can accept an Excel file that contains a collection of invoice notes and then process them, resulting in a report. The average amount of data an employee can do per day with a time of two hours is 500 data. After the system is implemented, the validation process takes 120 seconds for 500 data. The percentage for time reduction is 98.33%.**

***Index Terms*—Batch processing, e-commerce, free shipping, spring batch, validation system.**

I. Introduction

E-commerce is a process that supports customers, provides services and commodities, manages business transactions, and maintains bonds between providers, customers, and vendors with telecommunications network devices [1]. Indonesian e-commerce in the last ten years has grown by around seventeen percent, with a total e-commerce business reaching 26.2 million units. The impact of e-commerce on traditional business models is to accelerate the generation of new industries, reduce buyer costs, and accelerate the cycle of new product development [2].

PT. XYZ is e-commerce, with a large number of customers and increasing transactions. To deliver goods to customers, companies generally use services from third party logistics. Third-party logistics is an external logistics service provider company that offers one or several contract-based or agreement-based logistics activities [3]. To add higher value among other e-commerce, the company applies free shipping service to transactions that meet the requirements. The company will bear the shipping fee. Third-party logistics that works with the company will send a shipping transaction invoice, in an Excel file. The operational division will validate the data. This process includes checking the distance and price data in the billing file manually. Validation is done for distance and price data only. The system only handles transactions from third-party logistics that do not calculate the weight in shipping, such as GoSend and Grab. The checking process will take the origin and destination longitude-latitude location from the Excel file. Then put it into Google Maps to get the distance between the origin and destination location.

The supervisor examines data from the validation process carried out by the operational division. If it is valid, the supervisor forwards the data to the financial division for checking. If there is invalid data in the process of checking by the supervisor or financial division, then the data is returned to the operational division to do the validation process again.

The validation process at this time is not efficient. The validation process on data will be done many times before it can be said to be valid. Another obstacle that occurs is missing data and errors due to human inaccuracy. Over time, the company's transaction data will be even higher. This significant transaction is a tough challenge for operational division employees to achieve the overall target validation process. Therefore, this research builds a system that helps the operational division to process distance and price validation more efficiently.

This validation system is built based on a website to adapt user needs and make it easier to access. After the company implements this system, the validation process carried out by the operational division can run more efficiently and be easily used.

Manuscript received July 22, 2019; revised October 10, 2019.

Stephanie Pamela Adithama, Agitha Pramesti Sembiring, and Eduard Rusdianto is with the Department of Informatics Engineering, Universitas Atma Jaya Yogyakarta, Babarsari Street No. 43, Yogyakarta, Indonesia (email: stephanie.pamela@uajy.ac.id, 150708418@students.uajy.ac.id, eduard.rusdianto@uajy.ac.id).

II. Literature Review

A. Related Work

In a study of the implementation of online testing with a batch processing system, there was a need for speed to do extensive data processing and management. The online test will involve many participants, thus requiring technology, namely batch processing, to get efficient results. The scope of the problem in the research is how to design a system that can process extensive participant data efficiently and ensure the data is valid. System testing was carried out on 50-100 participants with different questions. The conclusion is the batch processing system helps in accelerating the online testing implementation process. The error rate in question randomization is far more accurate, and right on target [4].

In the study of report file generation systems using the batch processing method, many companies needed reports to evaluate an issue. To prepare these reports, many companies still performed this process manually. In this research, the report will be created automatically by the system. Reports

DOI: 10.7763/IJCTE.2020.V12.1262

on companies are in the form of extensive data. To be able to display large data on the report, the system with a batch processing method optimizes in making reports. The results of this study are report-producing applications in files that can speed up the process of reading reports through the batch processing method [5].

B. Electronic Commerce

Electronic Commerce, or commonly known as e-commerce, is the trading of products or services using computer networks, such as the internet. Business transactions that occur in e-commerce are business-to-business (B2B), business-to-customer (B2C), customer-to-customer (C2C) or customer-to-business (C2B). Through economic growth and technological development, the process of buying and selling transactions from traditional turned into e-commerce. E-commerce has the advantage of fast access, a wider choice of goods or services, and broad coverage [6].

C. Batch Processing

Some applications at large companies at this time, work on commands that can be executed without a display or user interface to run. These commands are usually run periodically and process large amounts of data. Batch processing is a process carried out on several cases simultaneously [7]. Batch processing is a solution for operations that involve extensive data. Batch processing is data processing without interaction and interruption [8]. Batch processing will group data into a particular set, and then it will set the amount of data to be stored in the database. Batch processing will take or read a collection of data. This data will be processed immediately, so the memory needed is not too large. Proper memory usage will have an impact on application performance, so the process of execution time can be more optimal.

D. Spring Framework

Software application development without proper tools and technology will reduce the productivity of making applications. The Spring Framework is a Java-based open-source framework created to facilitate the development of Java software. As a Java-based framework, Spring also applies object-based frameworks. Spring Framework makes it easy for developers to develop website applications, stand-alone applications, enterprise applications, and so on. Spring Framework also supports several modules such as Spring Security, Spring Boot, and Spring Batch [9].

E. Spring Boot

Spring Boot is a Java-based framework that is used to create a microservice. Spring Boot is designed to avoid complex XML configurations and reduce time to develop applications. Spring Boot provides a flexible way to configure Java Beans, XML configurations, and transactions that occur in the database. In Spring Boot, everything is automatic, so there is no need to spend time configuring one by one. Spring Boot is used primarily in making RESTful APIs and website pages. One of the advantages of using Spring Boot is the use of JSON APIs in communicating between servers [9].

F. Spring Batch

Spring Batch provides functions that developers need to develop applications using batch processing. Spring Batch is an open-source framework for batch processing by executing various jobs. The features implemented by the Spring Batch include data validation, output format, and the ability to handle large data sets [8].

Spring Batch provides classes and APIs to read or write resources, manage transactions, repeat jobs, and partition techniques to process large amounts of data. Spring Batch handles the items in the chunk. A Job will read and write data into a smaller chunk. Chunk or commonly known as chunk processing is a specific batch-oriented algorithm that contains a set of execution paths [10].

III. Result and Discussion

In Fig. 1, there are three roles in this system's business process, namely: operational division, supervisor, and financial division. The operational division handles the validation process of a file that contains a collection of invoices shipping transactions. The supervisor is the person who is responsible for the operational division. The finance division handles finance in the company.

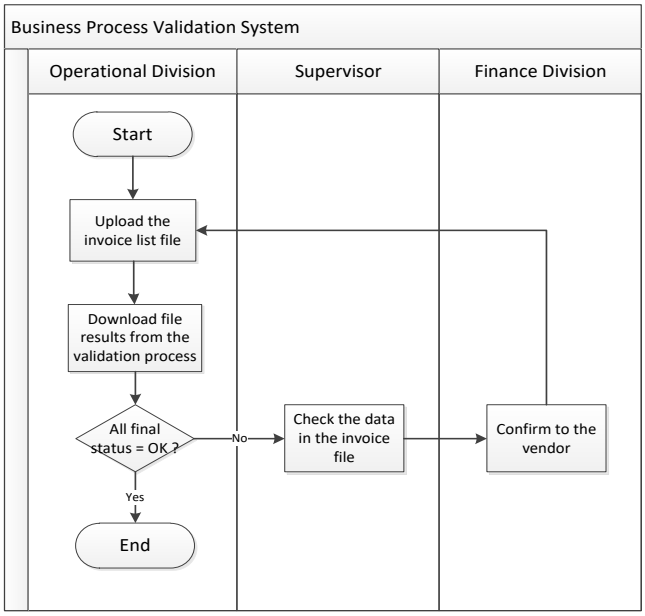


Fig. 1. Business process validation system.

In the business process diagram, the operational division will upload an Excel file that contains a collection of invoices shipping transactions for the validation process. Then the user downloads the report file results of the validation process and checks each final status on an invoice. If there is a final status that is not OK, then the user will proceed to the supervisor for checking. Status not OK, i.e., invalid data related to operational rules, such as distance or calculation price of third party logistic. The supervisor then proceeded to the financial division for confirmation to the vendor. After checking, the data will be uploaded again until all final statuses are OK.

This distance and price validation system is a web-based system, with object-oriented programming, and uses Java

programming language. This system is built using the Spring Boot framework for back-end websites. Data processing uses Spring Batch, which provides functions to run batch processing. This system uses the PostgreSQL DBMS and requires internet connectivity to access the Google Maps API.

This system consists of four parts: management of third-party logistics rules, invoice list management, validation process, and history of processed invoices. Users can log in, manage third party logistic rules, manage invoices, validate distances and prices, manage airway bill history, and make distance tolerance.

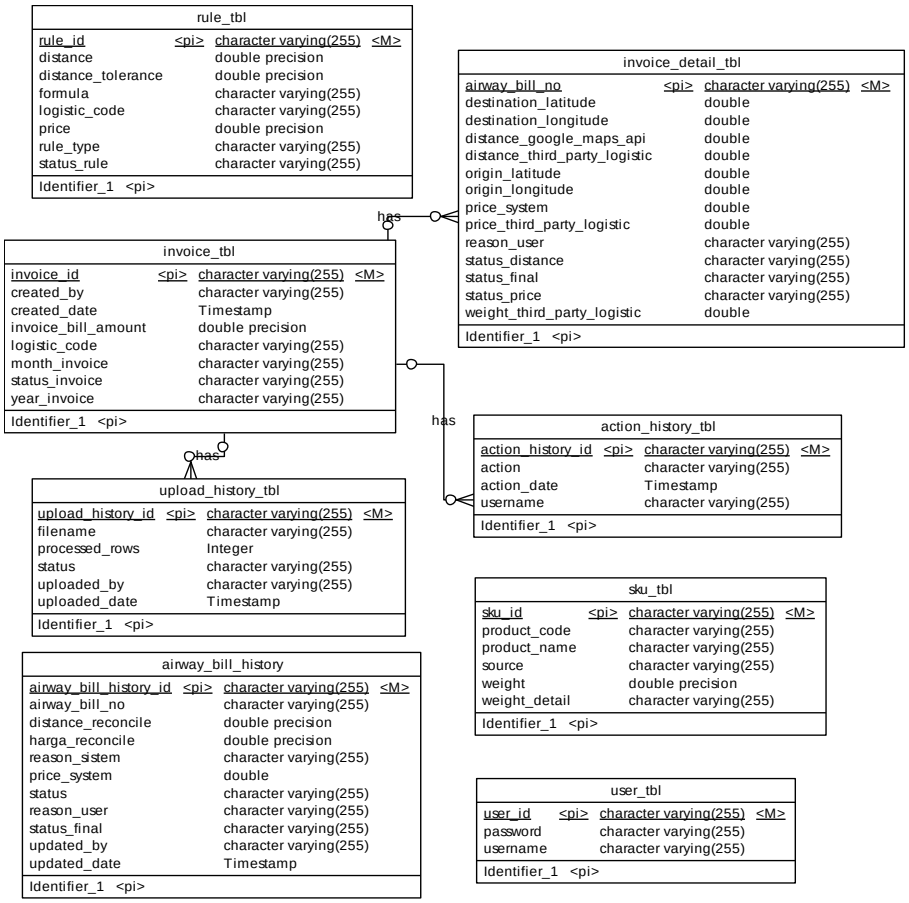


Fig. 2. Entity relationship diagram.

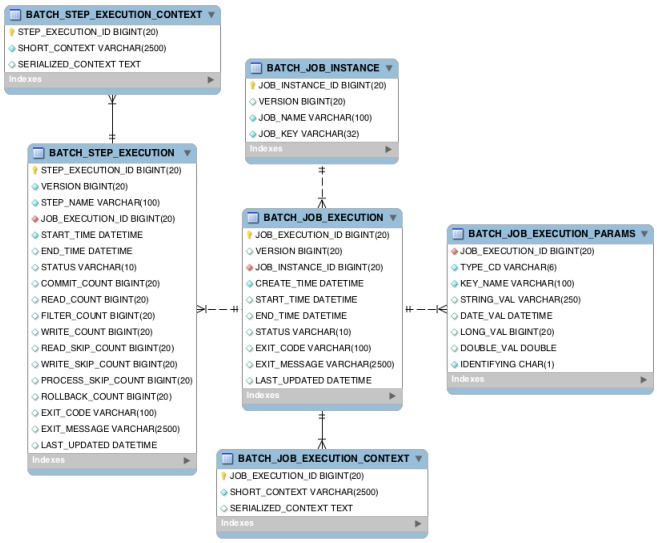


Fig. 3. Entity relationship diagram for spring batch.

Fig. 2 is an Entity Relationship Diagram (ERD) that displays tables used to store data according to business processes in this validation system. Fig. 3 shows the ERD Spring Batch. The ERD in Fig. 2 and Fig. 3 has no relationship because of the ERD in Fig. 3 is created automatically when the system implements the Spring Batch framework. Spring Batch requires these tables to store data such as Job and Step when executing batch processing.

This validation system uses the Spring Batch framework to manage the functions of batch processing. This batch processing is applied to invoice list management and validation processes, with the following steps: 1) Step. The Step consists of sequential stages in a batch. The Step is used to determine and control batch processing. The sequential stages include three functions, namely, ItemReader, ItemProcessor, and ItemWriter. ItemReader is used to read data and prepare data from a variety of different input types. In this validation system, the input comes from the file uploaded by the user. Every input data will be used to calculate the distance through the Google Maps API with the parameters of the longitude and latitude of the destination and origin location. The distance calculation results are used to calculate prices. The results will be compared with the data in the excel file, and the ItemProcessor function executes this process. ItemWriter has the task of writing the output and saving in the database. 2) Job. A job contains configuration or settings to run batch processing through the Spring Batch. This Job will include the call Step that was created and used to run or shut down the process or Step.

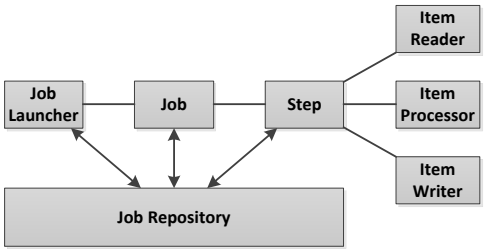


Fig. 4. Spring batch component.

Fig. 4 is a component of the Spring Batch, namely Job, Step (Reader, Processor, Writer), and Job Launcher. Job is a batch process. Job Launcher will execute Job. The Job will have a step that contains the essential stages in the process of managing data such as reading, processing, and writing data into the database. All of these components are used to manage data in Excel files. Exchange of data between the front-end and back-end on the system using the Hypertext Transfer Protocol (HTTP) request and Representation State (REST) is used to produce data and consumes data.

Batch configuration is the central part for configuring each batch processing component. Fig. 5, Fig. 6, Fig. 7 are pieces of code found in batch configuration.

```
public Job importInvoiceDetailJob(
String originName, String invoiceId) throws Exception {
  return jobBuilderFactory.get("importInvoiceDetailJob ")
                .incrementer(new RunIdIncrementer())
                .flow(stepInvoice(originName,invoiceId))
                .end()
                .build();}
```

Fig. 5. The code in method job.

The code in Fig. 5 is a Job component that will execute batch processing. The Job name is importInvoiceDetailJob, which will contain a collection of steps that will be run by JobLauncher. This method will have two parameters, originName, and invoiceId. Both parameters will be used to read files uploaded to the server, through stepInvoice (originName, invoiceId). For each Job, it will have a unique id, which will distinguish between each Job. The identifier (id) will be formed automatically with incrementer(new RunIdIncrementer ()).

```
public void JobInvoiceLauncer() throws Exception {
   jobLauncher
       .run(invoiceBatchConfiguration
          .importInvoiceDetailJob
              (originName,invoiceId), newExecution());
}
```

Fig. 6. The code in method job launcher.

The code in Fig. 6 is a Job Launcher, and this method is called when batch processing is executed. In running a batch job, there are at least two things needed by the Spring Batch, namely Job and Job Launcher. The Job will contain what command will be executed, while Job Launcher will execute the specified Job. In the implementation, when the user uploads the file, the Job Launcher will be called. This Job Launcher will execute Job importInvoiceDetailJob.

The code in Fig. 7 is a Step component that contains steps such as reading data, performing processes, and writing data into the database. Reader(originName, invoiceId) is the function to read data in a file. Processor(processor()) is the function that every row in a file will traverse, such as calculating and searching through the Google Maps API. Writer(writers()) is the function to write back into the database. Chunk in the Step method is an approach in dividing data into a certain number of batches. The expected results in the use of chunk are all data entered directly into the system, but broken down into smaller parts. In the implementation, the data divided into chunks, each of which consists of 200 amounts of data.

```
public Step stepInvoice(String originName, String
invoiceId) throws Exception {
    return stepBuilderFactory.get("stepInvoice")
          .<InvoiceDetail,InvoiceDetail>chunk(200)
          .reader(reader(originName, invoiceId))
          .processor(processor())
          .writer(writers())
          .build();
}
```

Fig. 7. The code in method step.

The interface of manage invoice detail in Fig. 8 and Fig. 9 serves to manage invoices shipping transaction data. Such as viewing information on the dashboards, uploading Excel file, downloading result report files, and making changes to actions that are rejected and approve. The process that occurs is checking the Excel file. If the Excel file is not empty, the system will call the function to save the file to the server. Next, call the JobInvoiceLauncer() function to run batch processing.

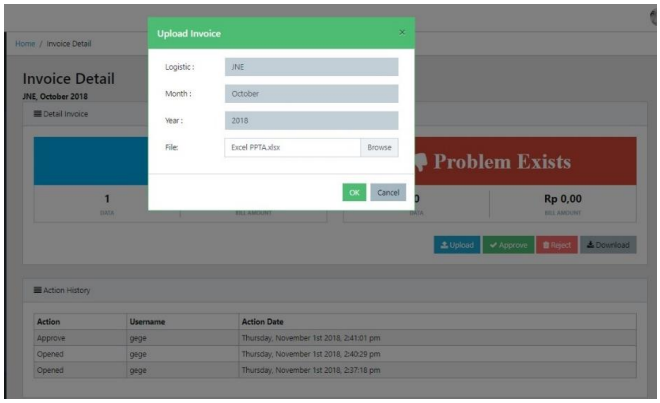


Fig. 8. Upload invoice file.

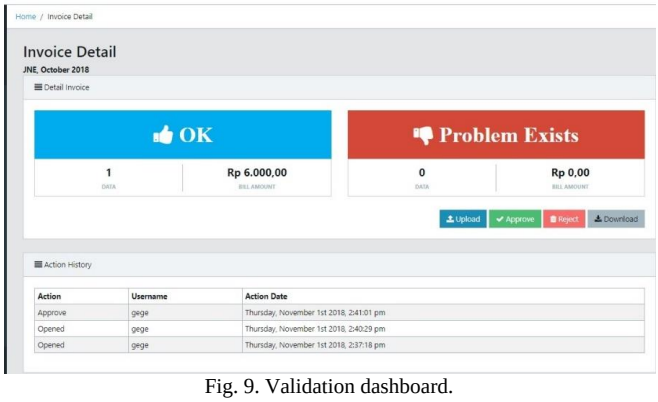


Fig. 9. Validation dashboard.

To access the database, it will use invoiceDetailRepository objects. In Fig. 10 is a function to retrieve data with a successful status based on the id. The business process will require data from the database, so it calls the

invoiceDetailRepository.getSumBillAmountSuccess (id) function and sends a response in the form of JSON with an OK message.

```
@Override
public Response getSumBillAmountSuccess(String id) {

    return Response.builder()
            .message("OK")
            .data(invoiceDetailRepository.getSumBillAmountSuccess(id))
            .build();
}

@Override
public Response getSumBillAmountProblemExist(String id) {

    return Response.builder()
            .message("OK")
            .data(invoiceDetailRepository.getSumBillAmountProblemExist(id))
            .build();
}
```

Fig. 10. Service invoice detail.

	A	B	C	D	E	F	H	I	J	K	L	M	N
1	Air waybill Number	Origin Longitude	Origin Latitude	Destination Longitude	Destination Latitude	Shipping Distance	Total Price (Rp)	Reason User	System Validation Distance (km)	Distance Status	Price Status	Final Status	System Validation Price (Rp)
2	BLIGO10001561	106.8036	-6.19013	106.826687	-6.178905	5.886	19500		6.54	ok	problem_	problem_	10000
3	BLIGO10001562	110.4354	-7.77685	110.433215	-7.773594	0.567	1150		0.63	ok	ok	ok	6000
4	BLIGO10001563	110.416	-7.77881	110.401237	-7.780946	3.456	4800		3.84	ok	ok	ok	6000
5	BLIGO10001564	110.4188	-7.78054	110.430537	-7.772922	2.709	3105		3.01	ok	ok	ok	6000
6	BLIGO10001565	110.3188	-7.80496	110.341589	-7.747522	7.515	11350		8.35	ok	ok	ok	13000
7	BLIGO10001566	110.3115	-7.7676	110.367547	-7.746839	9.846	14850		10.94	ok	ok	ok	17000
8	BLIGO10001567	110.4188	-7.78054	110.430537	-7.772922	2.709	3105		3.01	ok	ok	ok	6000
9	BLIGO10001568	106.8036	-6.19013	106.826687	-6.178905	5.886	5850		6.54	ok	ok	ok	10000
10	BLIGO10001569	110.3115	-7.7676	110.367547	-7.746839	9.846	14850		10.94	ok	ok	ok	17000
11	BLIGO10001570	110.416	-7.77881	110.401237	-7.780946	3.456	4700		3.84	ok	ok	ok	6000
12	BLIGO10001571	110.3188	-7.80496	110.341589	-7.747522	7.515	11350		8.35	ok	ok	ok	13000
13	BLIGO10001572	106.8036	-6.19013	106.826687	-6.178905	5.886	19500		6.54	ok	problem_	problem_	10000
14	BLIGO10001573	110.4188	-7.78054	110.430537	-7.772922	2.709	3105		3.01	ok	ok	ok	6000
15	BLIGO10001574	110.3115	-7.7676	110.367547	-7.746839	9.846	14850		10.94	ok	ok	ok	17000
16	BLIGO10001575	110.3188	-7.80496	110.341589	-7.747522	7.515	5850		8.35	ok	ok	ok	13000
17	BLIGO10001576	110.4604	-7.78981	110.463355	-7.783857	1.044	1300		1.16	ok	ok	ok	6000
18	BLIGO10001577	106.8036	-6.19013	106.826687	-6.178905	5.886	5850		6.54	ok	ok	ok	10000
19	BLIGO10001578	110.4188	-7.78054	110.430537	-7.772922	2.709	3105		3.01	ok	ok	ok	6000
20	BLIGO10001579	106.8036	-6.19013	106.826687	-6.178905	5.886	19500		6.54	ok	problem_	problem_	10000

Fig. 11. Validation result report in excel file.

Users can download the results of the validation process in Excel format. This report can be used to view detailed validation results. In Fig. 11, data located in the box shows the results of the validation process by the system. The left part is the original invoice data sent by the third party logistic. In the example, there are several invoice status still problematic. This problem occurs because the total price calculated by the system is lower than the cost of the bill. This result means that companies are required to pay more than they should.

This validation system test uses a unit test to test the functionality of a business process. Testing is using Junit 4 and Mockito as additional dependencies. Tests performed on all classes on the package controller and package service. The tests performed are Line Coverage and Method Coverage. Testing is said to be successful if the minimum coverage is 90%. The package controller will be tested on the JSON results returned by an API path. The package service contains business processes so that each function will be tested.

Coverage com.future.project in Backend

100% classes, 95% lines covered in package 'implementation'

Element	Class, %	Method, %	Line, %
ActionHistoryServiceImpl	100% (1/1)	100% (3/3)	100% (22/22)
AirwayBillHistoryServiceImpl	100% (1/1)	100% (2/2)	100% (27/27)
ExcelParserServiceImpl	100% (1/1)	100% (3/3)	86% (20/23)
GoogleApiServiceImpl	100% (1/1)	100% (1/1)	100% (5/5)
InvoiceDetailServiceImpl	100% (1/1)	100% (12/12)	95% (88/92)
InvoiceServiceImpl	100% (1/1)	100% (11/11)	94% (88/93)
RuleServiceImpl	100% (1/1)	100% (9/9)	98% (90/91)
SkuServiceImpl	100% (1/1)	100% (8/8)	89% (50/56)
UploadHistoryServiceImpl	100% (1/1)	100% (3/3)	100% (22/22)
UserServiceImpl	100% (1/1)	100% (2/2)	100% (15/15)

Fig. 12. Test result of package service.

Fig. 12 is the result of testing by line coverage and method coverage on the package service. The test results show that the average percentage for class coverage is 100%. The average for the method coverage is 100%, and line coverage is 96.2%. Based on the average results obtained, it has met the company's testing standards.

Coverage com.future.project in Backend

100% classes, 100% lines covered in package 'controller'

Element	Class, %	Method, %	Line, %
ActionHistoryController	100% (1/1)	100% (3/3)	100% (4/4)
AirwayBillHistoryController	100% (1/1)	100% (1/1)	100% (2/2)
InvoiceController	100% (1/1)	100% (12/12)	100% (13/13)
RuleController	100% (1/1)	100% (7/7)	100% (8/8)
UploadHistoryController	100% (1/1)	100% (1/1)	100% (2/2)
UserController	100% (1/1)	100% (2/2)	100% (3/3)
WeightListController	100% (1/1)	100% (4/4)	100% (5/5)

Fig. 13. Test result of package controller.

Fig. 13 is the result of testing by line coverage and method coverage on the package controller. The test results show that the average percentage for class coverage is 100%. The average for the method coverage is 100%, and line coverage is 100%. Based on the average results obtained, it has met the company's testing standards.

The average amount of data an employee can process, per day with a time of two hours is 500 data. After the system is implemented, for 500 data, the validation process takes 120 seconds. After this validation system is implemented, the percentage for the time reduction is 98.33%.

IV. Conclusion

The distance and price validation system for free shipping was successfully implemented to replace the validation process that was previously still manual. System functionality can run well, according to user needs. This research can help the operational division of the e-commerce company reduces time and speed up the distance and price validation process. The operational division can avoid errors that occur due to human factors.

Currently, the system uses distance and price to calculate the final price. The system still needs to add parameters, which is the weight of the goods, to meet the needs of several other vendors working with the company.

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Stephanie Pamela Adithama and Agitha Pramesti Sembiring analyzed system requirements, developed and tested the system. Eduard Rusdianto designed the system from the requirement specifications and wrote documentation. Stephanie Pamela Adithama and Eduard Rusdianto wrote the paper.

References

[1] Y. A. Nanehkaran, "An introduction to electronic commerce," *International Journal of Scientific & Technology Research*, vol. 2, no. 4, pp. 2–5, 2013.

[2] Y. Liu, "Research on business model innovation of logistics enterprises," *Modern Economy*, vol. 7, no. 14, pp. 1720–1727, 2016.

[3] X. Yang, "Status of third party logistics – A comprehensive review," *Journal of Logistics Management*, vol. 3, no. 1, pp. 17–20, 2014.

[4] M. Bagir, "Implementasi online testing dengan batch processing sistem," *Seminar Nasional Telekomunikasi dan Informatika*, pp. 50–55, 2016.

[5] S. Syahputra, "Pembangunan sistem informasi penghasil berkas laporan menggunakan metode batch processing," *Thesis, UAJY*, 2018.
[6] S. Shahriari, M. Shahriari, and S. Gheiji, "E-commerce and its impacts on global trend and market," *International Journal of Research–Granthaalayah*, vol. 3, no. 4, pp. 49–55, 2015.
[7] N. Martin, M. Swennen, B. Depaire, and M. Jans, "Batch processing: Definition and event log identification," RWTH Aachen University, 2015.
[8] M. T. Minella, *Pro Spring Batch*, New York: Springer Science + Business Media, 2011.
[9] C. Walls, *Spring Boot in Action*, New York: Manning Publications Co, 2014.
[10] A. Cogoluegnes, T. Templier, G. Gregory, and O. Bazoud, *Spring Batch in Action*, New York: Manning Publications Co, 2011.

Stephanie Pamela Adithama was born in 1988 and received her master of engineering in information engineering from Universitas Atma Jaya Yogyakarta, Indonesia in 2013. Her area of interest includes information management, database, and business intelligent. She is a lecturer of Information Engineering Department in Universitas Atma Jaya Yogyakarta since 2014.

Agitha Pramesti Sembiring was born in 1997 and received her bachelor of computer in information engineering from Universitas Atma Jaya Yogyakarta, Indonesia in 2019. Her area of interest includes software engineering and information management. Now, she is an employee in startup company in Indonesia since 2019.

Eduard Rusdianto was born in 1972 and received his master of engineering in electrical engineering from Universitas Gadjah Mada, Indonesia in 2003. His area of interest includes information management, database, and software engineering. He is a lecturer of Information Engineering Department in Universitas Atma Jaya Yogyakarta since 1997.