Mathematical and Computational Applications

MDPI

Article

Intelligent System for Data Tracking in Image Editing Company

Kimlong Ngin [1,2,*], Yen Yat [3], Rufai O. Hafeez [3], Sijjad Ali Khuhro [1] and Mariia Sushchenko [4]

[1] Department of Computer Science and Technology, University of Science and Technology of China, Hefei 230026, China; Sijjadali786@mail.ustc.edu.cn

[2] Department of Computer Science and Technology, Royal University of Phnom Penh, Phnom Penh 12150, Cambodia

[3] School of Public Affairs, University of Science and Technology of China, Hefei 230026, China; yy2014@mail.ustc.edu.cn (Y.Y.); yemi@mail.ustc.edu.cn (R.O.H.)

[4] School of Information Science and Technology, University of Science and Technology of China, Hefei 230026, China; mariias@mail.ustc.edu.cn

* Correspondence: nginkimlong@mail.ustc.edu.cn

Received: 12 September 2017; Accepted: 30 October 2017; Published: 9 November 2017

Abstract: The success of data transaction in a company largely depends on the intelligence system used in its database and application system. The complex and heterogeneous data in the log file make it more difficult for users to manage data effectively. Therefore, this paper creates an application system that can manage data from the log file. A sample was collected from an image editing company in Cambodia by interviewing five customers and seven operators, who worked on the data files for 300 images. This paper found two results: first, the agent script was used for retrieving data from the log file, classifying data, and inserting data into a database; and second, the web interface was used for the viewing of results by the users. The intelligence capabilities of our application, together with a friendly web-based and window-based experience, allow the users to easily acquire, manage, and access the data in an image editing company.

Keywords: intelligent system; complex data; heterogeneous data; log file; image editing company

1. Introduction

Data has become the fuel of development of the twenty-first century [1]. The amount of data is increasing every day and is becoming more complex in some companies. Traditionally, the companies expected that data would be used to answer questions about what to do and when to do it. Nonetheless, facing the sheer amount of data and greater complexity, the company should enhance their adaptability and dynamics by developing a data-driven management mechanism to grasp the opportunity and create value from it [2]. Martínez-Rojas et al. explained that the ability to manage data is increasingly becoming an essential issue in a society based on the real work of the company [3]. Meanwhile, most of the image editing companies in Cambodia are facing data management problems that need to be solved in order to provide specialized reports of heterogeneous and complex data as they rapidly increase on a daily basis (e.g., the amount of client and operator (designer) images, source information data (file name, directory, file extension, etc.), amount of time spent on designing images, cost, and so on). Most of these data are coming from different sources with diverse formats [4]. Particularly, some operators need corporation and so they can work on editing the same image file, whereas some operators work for multiple clients on image data [5]. So, the manager of the company cannot easily track data without the system. The managers, therefore, are facing myriad questions of what sorts of data to collect, code, and store, and how the data could be analyzed and interpreted into an insightful value [6].

To solve these problems, we developed an intelligent system for an image editing company that could enable effective data tracking [7]. The system provides opportunities and new tracking methods for sustainable management of the data and images [8]. Moreover, the system is also an innovative tool for data analysis, query, and report generation that can help company managers in making good decisions [9].

The previous work of Wang et al. discussed the document model in the image editing system [10]. Similarly, Sun et al. discussed a solution to solve the consistency maintenance problem for shared documents in text editing systems by using hybrid document models including CorelDraw, Adobe Photoshop, and so on [11]. However, these studies are based on documents that allow users to store and share their information, and therefore the data formats are unstructured and insecure. So, the company managers not only have to work manually to track data from the storage documents line-by-line, but also the data formats make it difficult for them to prepare a new standard document form which is able keep the majority of the data. Furthermore, the managers need to spend more time to enter the data into that document form. Another researcher, Lihui [5], has discussed cooperative image editing that allows multiple operators to simultaneously work on editing an image file in either synchronous or asynchronous cooperative modes by running a web-based system integrated with the web-services. The operators can share the same information about the image during editing. We observed that this work can help the manager to track the qualities of the image, operator's performances, and cooperative work. However, it cannot help the manager to manage the all of the major problems mentioned above.

This paper presents a web-based intelligent system designed to enable faster and more accurate results from managing and searching of complex and heterogeneous data in an image editing company, for which it can reduce the cost and time. The intelligent system for data transactions in an image company (ISDT), used data from the log file of the operators and clients to calibrate and then store data in a database which can be viewed on the web interface. The system is comprised of three large modules; namely, the system log file, the agent script of PHP and C#, and the data viewer. As shown in Figure 1, the system log file is responsible for storing all data information from the clients and operators by which the data are transmitted through the FTP and operator's machine. The agent script of PHP and C# is responsible for retrieving data from the log file to calibrate, classify, and store in the database system. The data viewer plays an important role in viewing data from the database on the user-friendly web interface.

2. Overview of ISDT System Workflow Diagram

A workflow of ISDT depicts a series of actions that define how the process of an image editing company should be accomplished. A workflow visualizes how tasks flow in a company, whether their machines or people move forward consistently [7]. It also encompasses the course of action taken by the modeler to design/construct a process model consisting of start and end event(s), activities, gateways, edges, and so on [12].

The workflow in Figure 1 expresses the work processing of a company. The client and controller respond to uploads and downloads of data from FTP. The client will upload the new data to FTP To-Do and receive the data from FTP Done that has been uploaded by the controller when the editing has been completed. Meanwhile, the controller has a close relationship with the operator in order to transfer the client's data to each other; the controller could transfer the new data to an operator and receive the data when the operator has completed editing.

The dotted lines in Figure 1 illustrate the inspiration for the ISDT visualization. It was perceived to represent the log file system which the agent script needs to implement in order to grab the data from the log file. The log file stores all important data from the clients and operators, and then when it is enough, the system is be able to run.

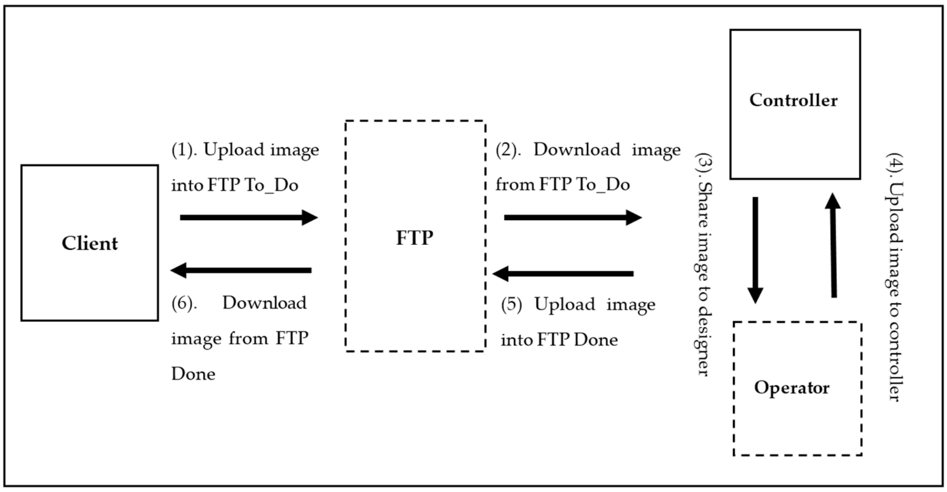

Client
(1). Upload image into FTP To_Do
(6). Download image from FTP Done
FTP
(2). Download image from FTP To_Do
(5) Upload image into FTP Done
(3). Share image to designer
Controller
Operator
(4). Upload image to controller

Figure 1. Intelligent system for data transactions in an image company (ISDT) workflow. The dotted lines represent the location that systems are implemented to absorb data.

3. ISDT System Structure

ISDT is an intelligent application with interactions between server-side and client-side. The server-side basically tells how the system works, updates, and changes. Meanwhile, the client-side or web interface can show data on a single record or multiple records. Data can be labeled and placed in tables, following a uniform "human-readable" format or stored in different formats. Data can be accessed at a static URL or be generated as a result from a search box on the user-friendly interface [13].

System Design

The system has been implemented on the basis of a client/server architecture [3] which includes two basic elements (a client-side and a server-side), by which both sides are interactive with each other. Figure 2 shows both the architecture and the technologies that are used in the development of the system. It is necessary to organize a "research demand and supply monitoring" system [14]. Firstly, the server level is implemented by means of C#, PHP, MySQL database, FTP client, and Adobe Photoshop. C# is used as the service that automatically runs inside the operator machine to retrieve data from the operator log file. PHP is used as the agent script that automatically runs on the server machine to retrieve data from the FTP log file, and MySQL is used as the database system that is responsible for storing data from the log file by the C# service and PHP agent script. The FTP client is installed on the server machine for client and controller transfers of data and pushing data information to be stored in the FTP log file. Meanwhile, Adobe Photoshop is installed on the operator machine to edit the images and push operator's actions to store in the Adobe Photoshop log file (operator log file). Secondly, the client level is implemented by means of CSS, CakePHP, HTML5, Ajax, and JSON. These technologies are used for designing the user interface and retrieving data from the database system for viewing on the user interface.

In the next section, the study will present the main module and functions that the system plays in its roles at the service-side and client-side.

4. The Module and Functionality

As mentioned above, ISDT uses both agent scripts to permit the log files of FTP and operator in order to start from the original data to the integrated access from which the information is contained in the common parts. This management is carried out through the use of three modules: Log file (FTP and Operator), Agent Script (C# and PHP), and data retrieving. Again, the server-side in Figure 2 shows the flow of the information in the three modules, which are detailed in next sections.

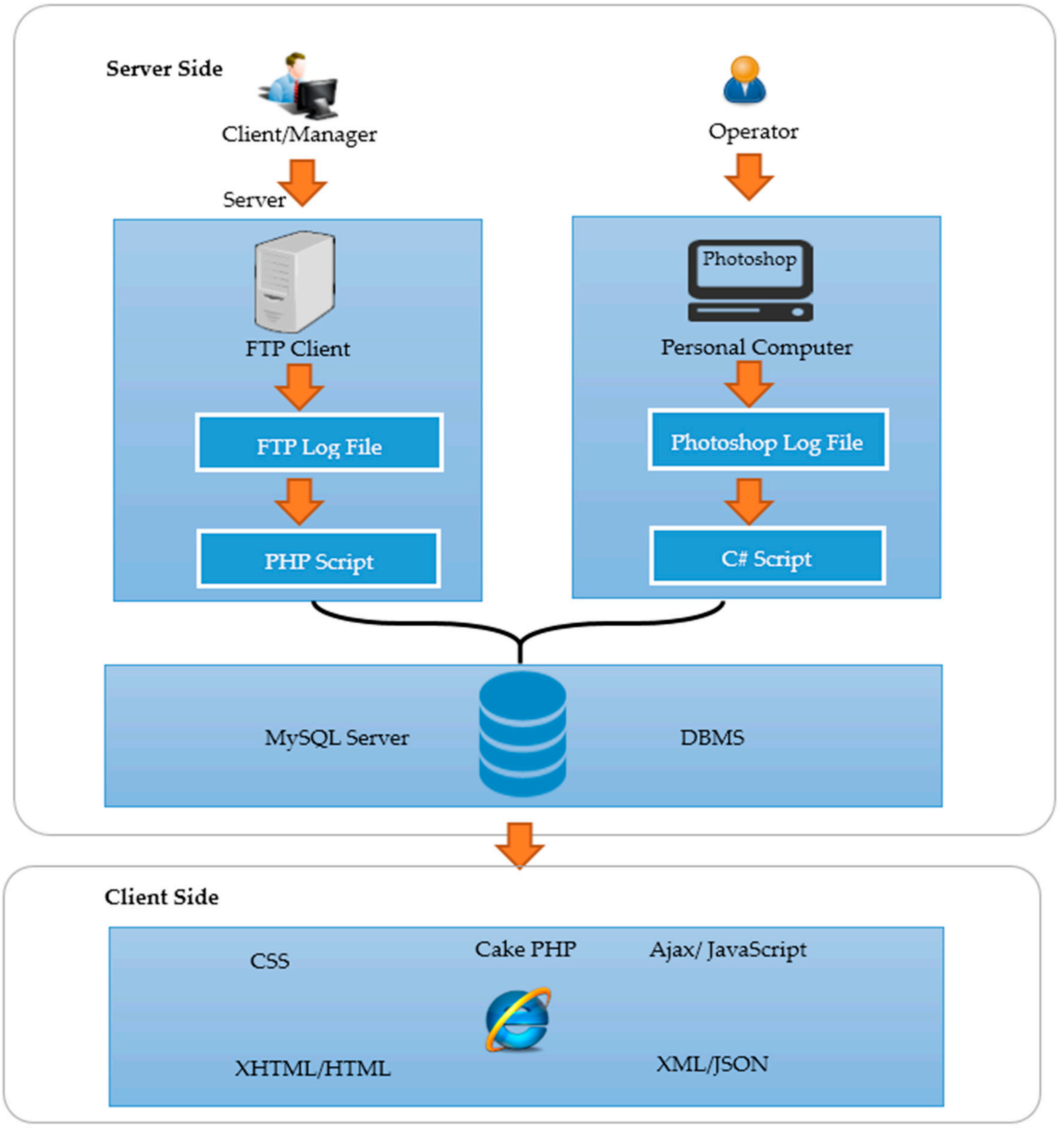


Figure 2. Server/Client architecture of the system.

4.1. Log File Module

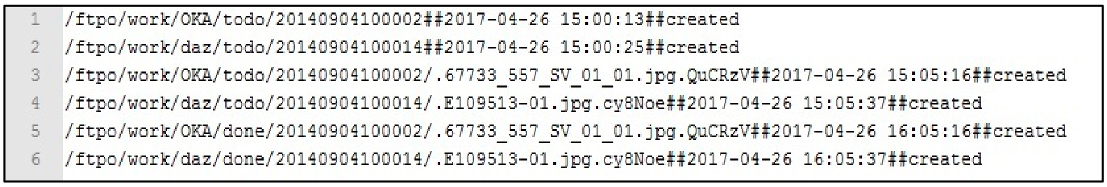
ISDT temporarily uses the log file to store data information, and it is used both within and beyond the human–computer interaction (HCI) community. An example of a field outside the HCI community where logging is helpful for researchers is Interactive Information Retrieval (IIR), which also uses collected log data to analyze and draw conclusions from users' behavior [15]. The log is an important storage file which is kept by most systems such as e-business websites and database systems in order to record their activities or transactions. The log file is usually a flat file, where each line contains at least a timestamp, a combination of one or more event identifiers, and the actual log message which contains information about the executed events [16]. This is the most important aspect of the ISDT system to store both client and operator data information in a database system.

4.1.1. FTP Log File Module

A. FTP Log File

The FTP client is the best communication tool for client and company [17]. It provides a secure method (via SSL, TLS, FTPS, HTTPS, or SFTP) to upload/download files to and from FTP servers [18].

Data from FTP will be stored in the FTP log file. As shown in Figure 3, an FTP log file stores the main data information such as a username that connects with user's account, Timestamp, Description: Status, Command, Trace, and the separator ("/", "##") [19].



```
/ftpo/work/OKA/todo/20140904100002##2017-04-26 15:00:13##created
/ftpo/work/daz/todo/20140904100014##2017-04-26 15:00:25##created
/ftpo/work/OKA/todo/20140904100002/.67733_557_SV_01_01.jpg.QuCRzV##2017-04-26 15:05:16##created
/ftpo/work/daz/todo/20140904100014/.E109513-01.jpg.cy8Noe##2017-04-26 15:05:37##created
/ftpo/work/OKA/done/20140904100002/.67733_557_SV_01_01.jpg.QuCRzV##2017-04-26 16:05:16##created
/ftpo/work/daz/done/20140904100014/.E109513-01.jpg.cy8Noe##2017-04-26 16:05:37##created
```

Figure 3. Data in FTP log file which is generated by the server.

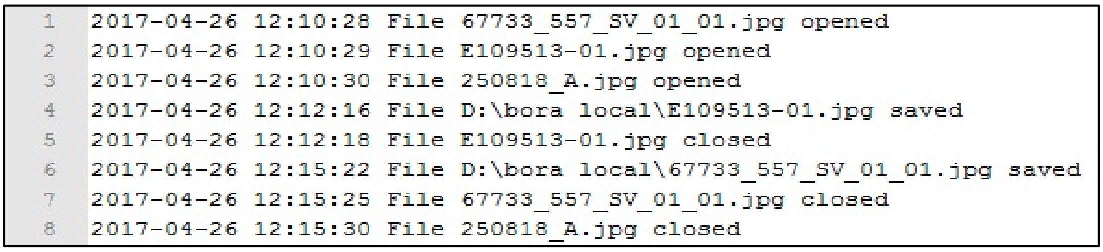
B. FTP Log File Calibration

1. Directory: (Client's information) has client name, FTP folder (to-do, done), directory name, and image name. Directory is separated by the separator "/".
    - FTP folder "to-do": for the client to upload the new projects (images) to a company so that the projects will be controlled by a company controller.
    - FTP folder "done": for the controller to upload the finished projects to the client.
2. Timestamp shows the date and time that client and controller have uploaded the data to the FTP-client. Timestamp is separated by the separator "##".
3. Status: data that have been uploaded by the client and controller to the FTP-client will show "created" status. It is separated by the separator "##" after Timestamp.

4.1.2. Operator Log File Module

A. Operator Log File Data

Adobe Photoshop is the most powerful software used for editing the images. Except for these functions, Adobe Photoshop also has a function to record the operator's actions in a log file. As shown in Figure 4, an operator log file recorded the important data such as Timestamp, File name, and Status. Particularly, the operator name will be the name of the log file. Operator log file data were separated by a space and a breaking line for new actions.



```
2017-04-26 12:10:28 File 67733_557_SV_01_01.jpg opened
2017-04-26 12:10:29 File E109513-01.jpg opened
2017-04-26 12:10:30 File 250818_A.jpg opened
2017-04-26 12:12:16 File D:\bora local\E109513-01.jpg saved
2017-04-26 12:12:18 File E109513-01.jpg closed
2017-04-26 12:15:22 File D:\bora local\67733_557_SV_01_01.jpg saved
2017-04-26 12:15:25 File 67733_557_SV_01_01.jpg closed
2017-04-26 12:15:30 File 250818_A.jpg closed
```

Figure 4. Data in the Operator log file which is generated by Adobe Photoshop in the operator's machine.

B. Operator Log File Calibration

1. Timestamp shows the date and time of images that the operator edited, from beginning to end. The Timestamp is separated by a space, and it stands on the first and second columns from the left edge. The first column is the date and the second column is the time.
2. Image Name: the file that the operator has edited in Adobe Photoshop. It stands in the fourth column.
3. Status has three types that are significant in one event processing an image. They are:

- Opened: means the image has been opened by the operator in Adobe Photoshop.
- Saved: means the image has been edited by the operator in Adobe Photoshop.
- Closed: means that editing has been completed for the image and it has been closed by the operator from Adobe Photoshop.

4.2. Agent Script or Service Module

As mentioned earlier, the agent script is the most important element and the core of the system. The agent script of ISDT operates in three main steps. The first step of script identification is to extract some features from a set of log files; the second step is to use those features for classifications [20], and the last step is to deliver data to a database system. The agent script or service of ISDT will be explained in the next sections.

4.2.1. Agent Script for Log File

The PHP script is one of the scripts that execute the system. It is one of the first languages that many web programmers have to learn [21]. Web script provides two advantages. One is the ability to manipulate Web pages to accomplish the goals and the other is regarding data collection [21]. PHP, in particular, has been widely employed in servers around the world as part of the LAMP (Linux-Apache MySQL-PHP) platform [22]. This study aimed at developing a PHP script of ISDT so as to execute at the server-side as well as to retrieve data from the FTP log file and insert data into a database system. The PHP script is executed by using Cron Jobs (Task Scheduling) that runs in the background. The Cron Jobs are used for scheduling tasks to run on the server [23]. There are two Cron Jobs which are running in the ISDT system. Firstly, it runs a PHP script retrieving data from FTP log files every 15 min. Secondly, it runs a PHP script to clean the contents of FTP log files at 12:00 am. Aside from this, the system has a significant C# service that is executed in the operator's machine. To create this service, we need to create an application that is installed as a service [24]. This service will be installed into individual operators' machine systems, where it can be automatically started when the computer boots [24], and is executable without a user interface [25]. The C# service will execute after the operator closes Adobe Photoshop to retrieve all important data from the log file (Figure 4) and enter the data into the database. Meanwhile, the log file content will be removed after the service has finished its retrieval of all data from the log file.

The agents of PHP script and C# service are the most important for the system to collect data and then spread out to the whole system. This will be discussed in detail in the next section about the process of script and services, data classifications, and data delivery for the system.

4.2.2. Data Classifications and Deliveries in the Log File

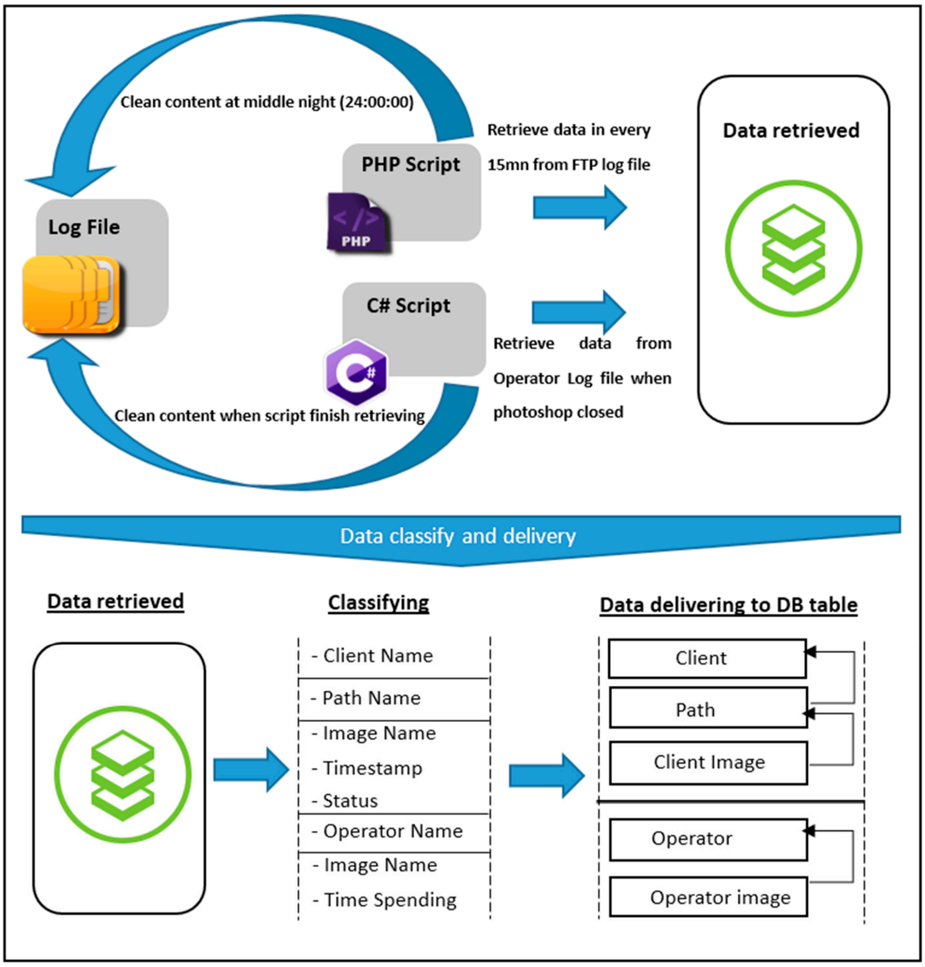
Data classifications can be used to group data before sending them to the database. System classification—covering the complete data classifications of the log file—is intended to act as a standard through pooling and tabular lists; various requirements give a breakdown of the data [26]. To classify, data from the log file is compiled by the ISDT agent script to analyze the data based on the database structures. Figure 5 illustrates an example of the agent script module that works on a log file which belongs to a given data classification and delivery.

The classifications and delivery processes of the data are as follows:

1. Retrieved data from the log file: ISDT's agent script performs a semantic and syntactic pre-processing of the log file in order to retrieve the terms of the important data related to the system's demands. Data retrieval has two types: one type is retrieved from the FTP log file using the agent of PHP script, and another type is retrieved from the operator log file using C# service.
2. Data classification: the process of organizing data into categories for its most effective and efficient use [26]. It involves multiple tags and labels, defining the types of data, confidentiality, and integrity [27]. Once the data in the pre-processing stage has been carried out and the best

elements of the database table are obtained, data retrieved from the FTP log file will generate the main elements such as "client name", "path name", "image name", "timestamp", and "status". Meanwhile, data retrieved from operator log file will also receive the main elements such as "operator name", "image name", and "time spending".

3. Data delivery: data is delivered into a database from which any collection of data or information is organized for a rapid search and retrieval by a computer [28]. Data retrieved from the FTP log file will be delivered to the three main tables (client, path, client image). Data from the FTP log file, such as "client name", will be stored in the *client* table. Meanwhile, "path name" and "client id" will be stored in the *path* table, where the *path* table has a relationship with the *client* table through the "client id". Particularly, "image name", "timestamp", "status", and "path id" will be stored in the *client image* table, where this table has a relationship with the *path* table through the "path id".



Clean content at middle night (24:00:00)
PHP Script
Retrieve data in every 15mn from FTP log file
Data retrieved
Log File
C# Script
Retrieve data from Operator Log file when photoshop closed
Clean content when script finish retrieving
Data classify and delivery
Data retrieved
Classifying
Data delivering to DB table
- Client Name
- Path Name
- Image Name
- Timestamp
- Status
- Operator Name
- Image Name
- Time Spending
Client
Path
Client Image
Operator
Operator image

Figure 5. Schema of our intelligence agent script module.

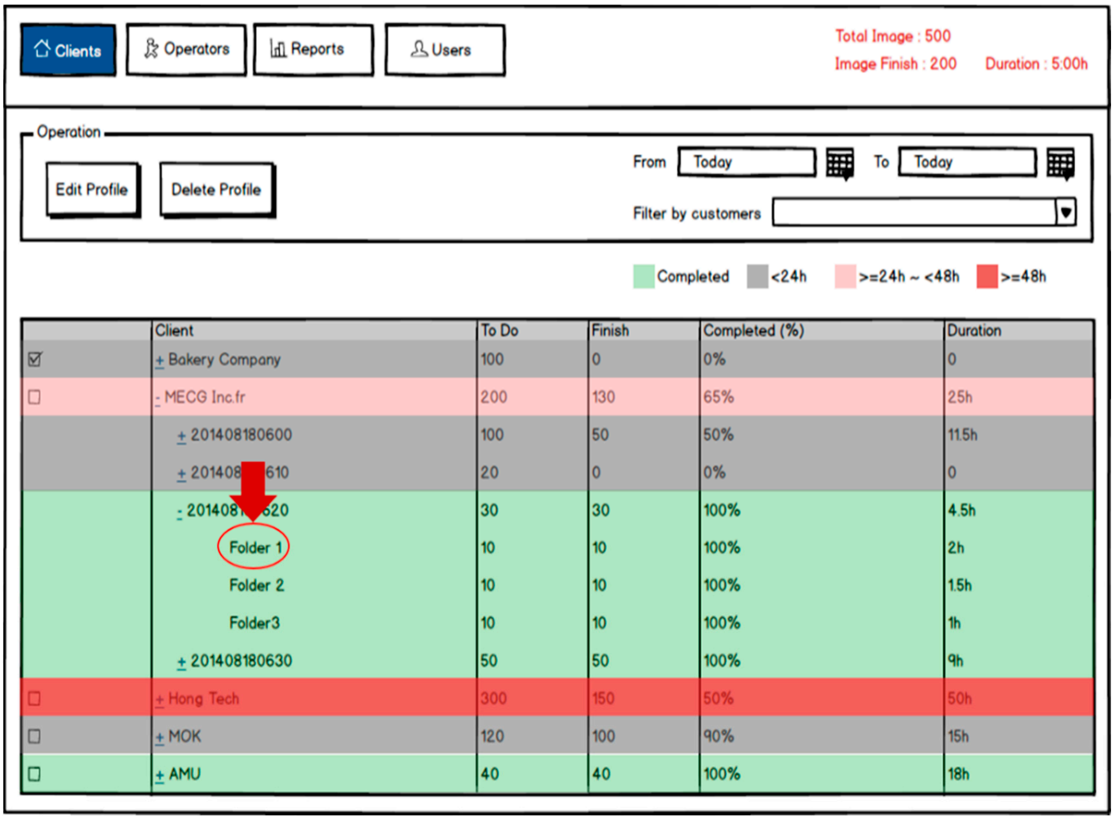
Data retrieved from the operator log files will be delivered to two main tables (*operator, operator image*), whereas the data from the operator log files (e.g., operator name) will be stored in *operator* table. Likely, "image name", "time spending", and "operator id" will be stored in the *operator image* table, where the *operator image* table has a relationship with the *operator* table through the "operator id". Firstly, agent script and service retrieve data from the log file and then they start to classify data to squeeze and to get the main element. Lastly, they store data in a database by following the specific structure. The agent script and service work very smoothly to absorb data automatically from the log file of the operator and the FTP client of the server-side.

5. Results

5.1. The Dataset and Screenshots

The greatest strength of ISDT is its ability to generate and present reports of the complex and heterogeneous data used in the image editing industry. The users can access data information from the database through the use of different filters or can manually search for the desired information by browsing the different structures of the system. The screenshots below show that client's data and operator's data that can be viewed with the most accurate results through the analysis of the ISDT system. In the experiments, two main data sets are used to evaluate the performance of the proposed method. The first data set was collected from the client log file—300 images uploaded by five clients. All of those images have different extensions, such as .tiff, .jpg, .jpeg, .pdf, .png, etc. The data were uploaded in different paths and folders in the FTP-client. The client data are shown in Figures 6 and 7 on the web interface. Figure 6 shows the client name, path, total number of files, and duration, while Figure 7 shows the detail of each image that is coherent with Figure 6. The web interface view of Figure 7 is the sub-web interface view of Figure 6.

Another data set is from the operator log file that was collected from seven operators' machines (Figure 8). The operator data can be viewed on web interface as shown in Figures 9 and 10. Figure 9 shows the details of operator information and the total number of image files (to-do, done), and Figure 10 indicates the details of each image that was edited by each operator. The web interface view of Figure 10 is the sub-web interface view of Figure 9.



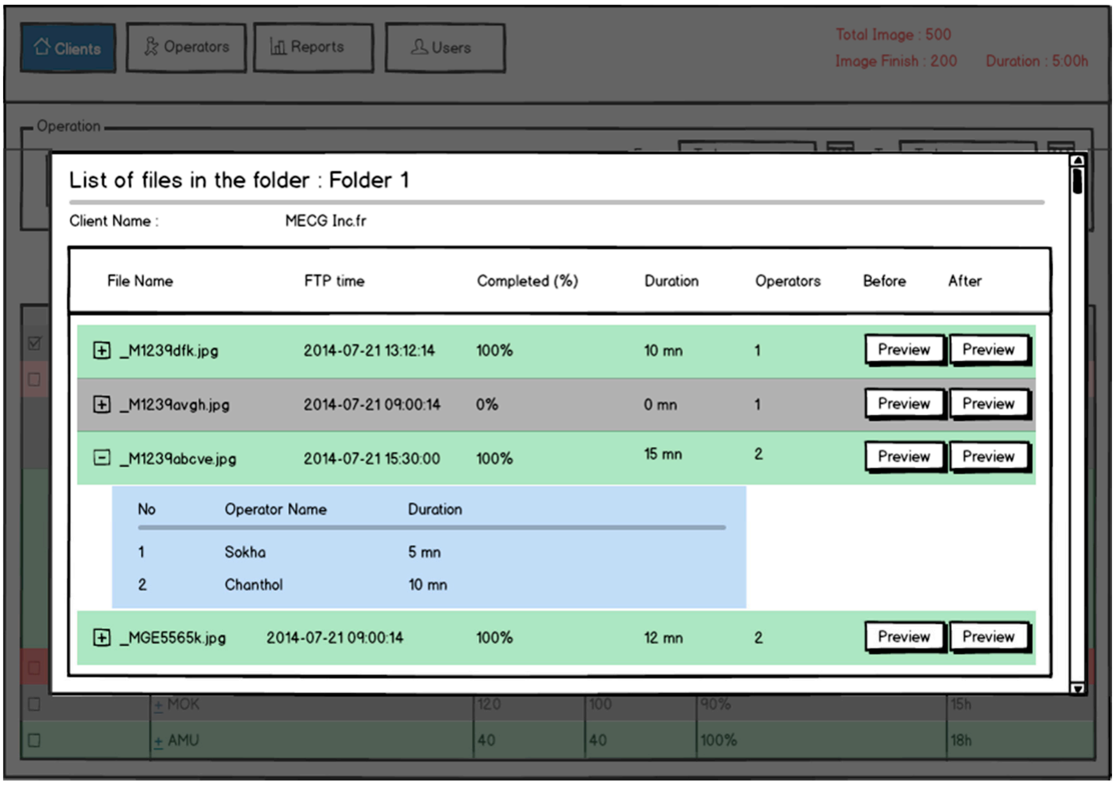
Total Image : 500
Image Finish : 200 Duration : 5:00h

From Today To Today

🟩 Completed ▨ <24h ▨ >=24h ~ <48h 🟥 >=48h

	Client	To Do	Finish	Completed (%)	Duration
☑	+ Bakery Company	100	0	0%	0
□	- MECG Inc.fr	200	130	65%	25h
	+ 201408180600	100	50	50%	11.5h
	+ 201408 610	20	0	0%	0
	- 2014081 620	30	30	100%	4.5h
	Folder 1	10	10	100%	2h
	Folder 2	10	10	100%	1.5h
	Folder3	10	10	100%	1h
	+ 201408180630	50	50	100%	9h
□	+ Hong Tech	300	150	50%	50h
□	+ MOK	120	100	90%	15h
□	+ AMU	40	40	100%	18h

Figure 6. Example screenshot of a client's result on the web interface.



Total Image : 500
Image Finish : 200 Duration : 5:00h

List of files in the folder : Folder 1

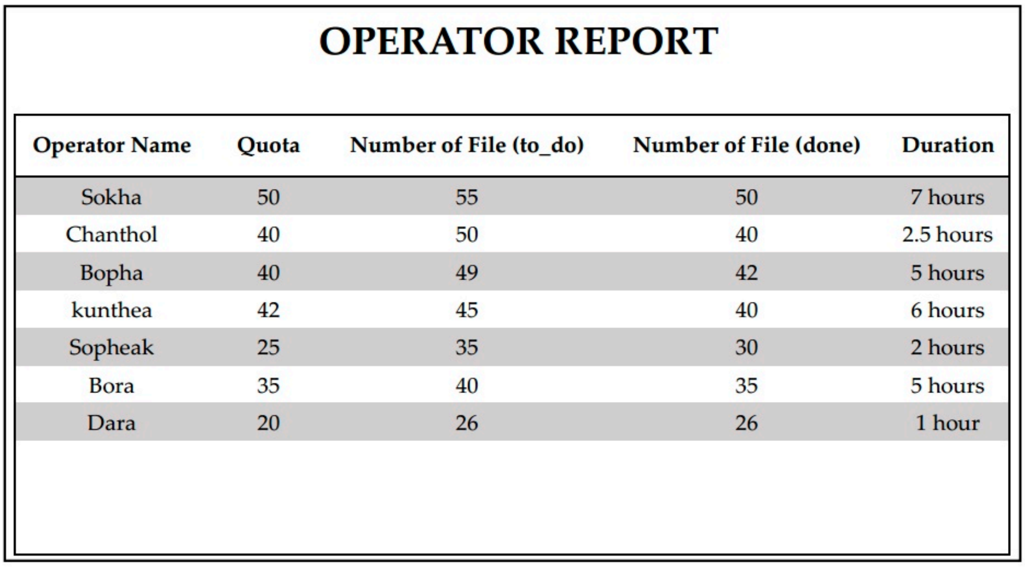
Client Name : MECG Inc.fr

File Name	FTP time	Completed (%)	Duration	Operators	Before	After
⊞ _M1239dfk.jpg	2014-07-21 13:12:14	100%	10 mn	1	Preview	Preview
⊞ _M1239avgh.jpg	2014-07-21 09:00:14	0%	0 mn	1	Preview	Preview
⊟ _M1239abcve.jpg	2014-07-21 15:30:00	100%	15 mn	2	Preview	Preview
⊞ _MGE5565k.jpg	2014-07-21 09:00:14	100%	12 mn	2	Preview	Preview

No	Operator Name	Duration
1	Sokha	5 mn
2	Chanthol	10 mn

□	+ MOK	120	100	90%	15h
□	+ AMU	40	40	100%	18h

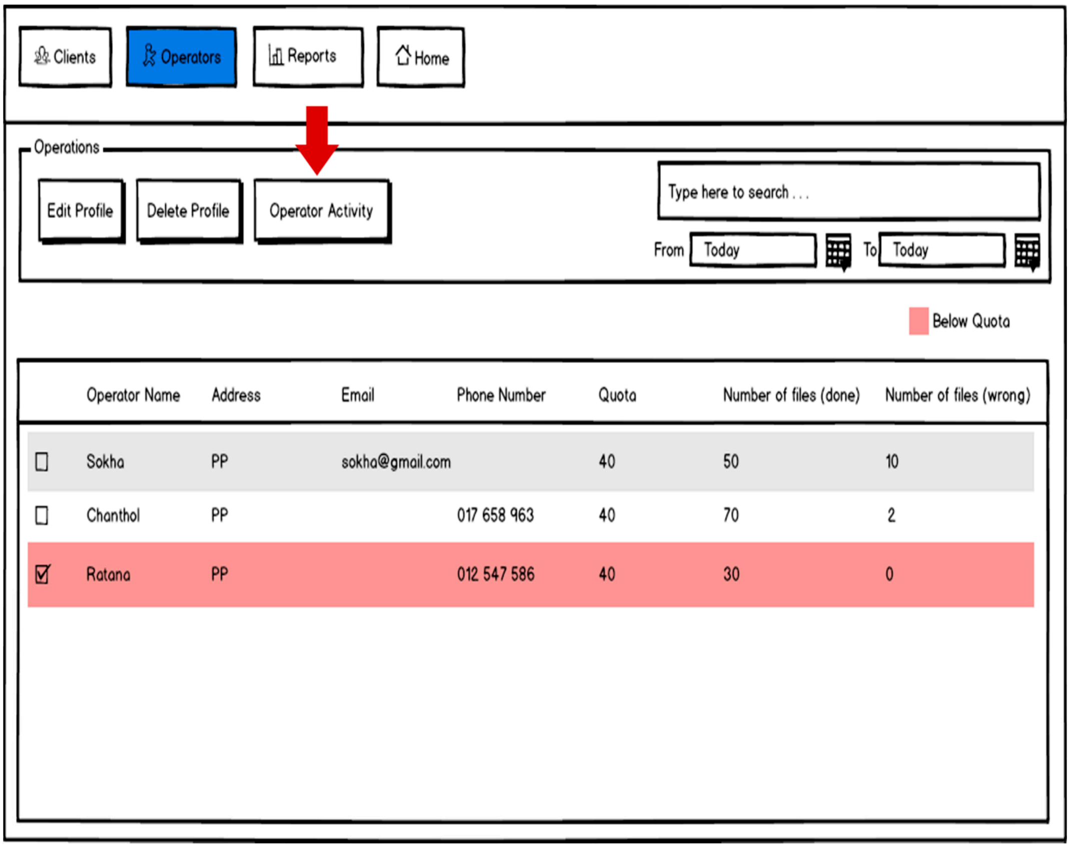
Figure 7. Example screenshot of client's detailed results on the web interface.



OPERATOR REPORT

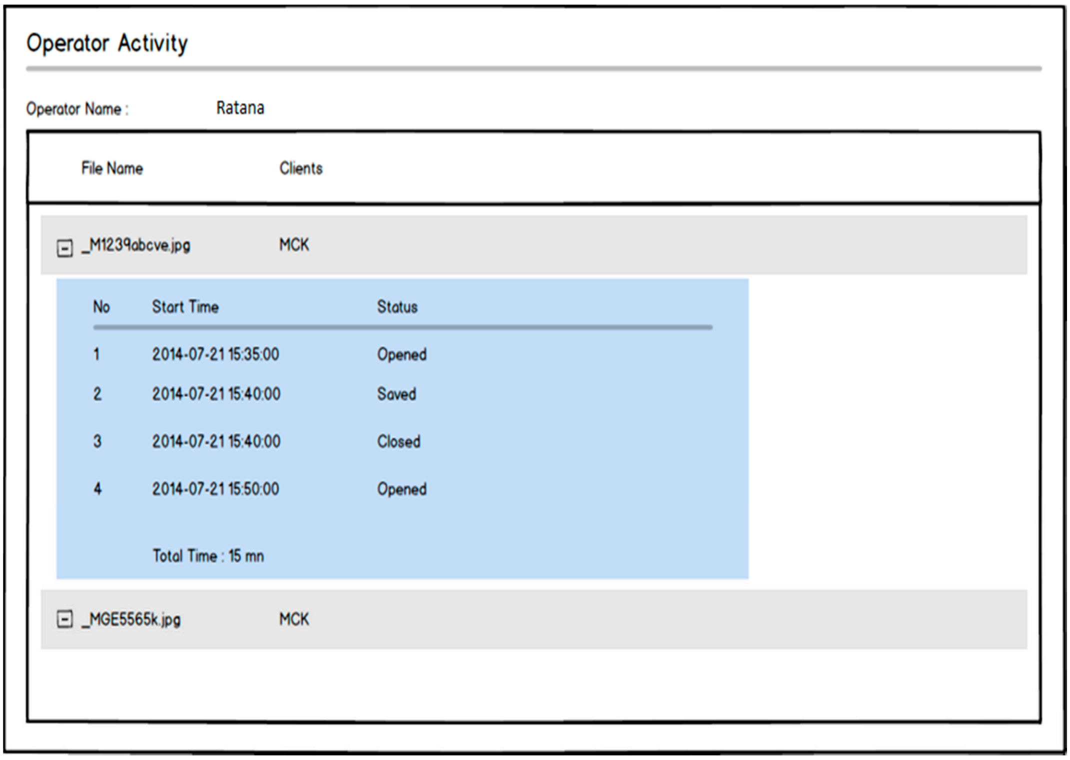
Operator Name	Quota	Number of File (to_do)	Number of File (done)	Duration
Sokha	50	55	50	7 hours
Chanthol	40	50	40	2.5 hours
Bopha	40	49	42	5 hours
kunthea	42	45	40	6 hours
Sopheak	25	35	30	2 hours
Bora	35	40	35	5 hours
Dara	20	26	26	1 hour

Figure 8. Screenshot of operator's report.



	Operator Name	Address	Email	Phone Number	Quota	Number of files (done)	Number of files (wrong)
□	Sokha	PP	sokha@gmail.com		40	50	10
□	Chanthol	PP		017 658 963	40	70	2
☑	Ratana	PP		012 547 586	40	30	0

Figure 9. Example screenshot of an operator's result on the web interface.


Operator Activity

Operator Name : Ratana

File Name	Clients
⊟ _M1239abcve.jpg	MCK

No	Start Time	Status
1	2014-07-21 15:35:00	Opened
2	2014-07-21 15:40:00	Saved
3	2014-07-21 15:40:00	Closed
4	2014-07-21 15:50:00	Opened

Total Time : 15 mn

File Name	Clients
⊟ _MGE5565k.jpg	MCK

Figure 10. Example screenshot of operator's detailed results on the web interface.

5.2. Speed

The processing time and speed of ISDT are presented in Table 1. In the table, the unit of speed is seconds. The value in the second row is the average runtime of the images in one week. It is assumed that in one day the clients upload between 100 and 500 image files and the average time running to view data is 2 s. The experiments were run on a desktop computer with Intel Core i3 4.00 GHz CPU and 4 GB RAM. It is obvious that the proposed method is highly efficient and met the requirements of the image editing company.

Table 1. The speed and number of images in one week.

100–500	500–1000	1000–1500	1500–2000	2000–2500	2500–3000	3000–3500
≈2	≈4	≈6	≈8	≈12	≈15	≈20

6. ISDT Evaluation

Based on empirical analysis, this section describes a preliminary evaluation of the ISDT application system to check the validation:

- To show that the ISDT system can accurately and effectively retrieve complex data from the log file.
- To show that the ISDT system removes a manual job by providing the best report for the users.

To achieve these objectives, the system was tested in two steps: (1) to assess the real installation of the ISDT application system, so as to measure the system model in an image editing company; and (2) to receive feedback from users and conduct a survey [3].

6.1. Measurement Module

To measure the accuracy of the ISDT system, we asked the users to use the real system in the company by:

Observation 1. Testing both scripts (C# script and PHP Script).

1. The customers and a company manager upload any image through the FTP-client to observe the C# script's ability and accuracy (running on the server machine).
2. The operators work on a customer's image that was transmitted by the controller to observe the PHP script's ability and accuracy (running on each operator's machine).

Observation 2. ISDT identified the results of tests.

We gathered system results by testing with the five customers and seven operators who worked with 300 image data files in the MaskClip Company in Cambodia. Table 2 shows the results of customer data retrieved by the application system from the database. The five customers uploaded 300 images through the FTP client, and seven operators joined to edit all of the images. Table 2 highlights that the customer OKA uploaded 105 images and then two operators (ID numbers 1 and 2) joined to edit the images, and 86% of the images were completed after spending 9.5 h. Another customer, Lumizy, uploaded 94 images, and then another two operators (ID numbers 3 and 4) joined to edit the images; 87% of the images had been completed after 11 h. Likely, customer Bex uploaded 55 images, and then two operators (ID numbers 5 and 6) joined to edit the images, and 91% of the images had been completed after around 4 h. Moreover, customer Boo uploaded 20 images and then one operator (ID number 6) edited the images, and 95% of the images had been completed after 1 h. Furthermore, customer Cam uploaded 26 images and one operator (ID number 7) edited the images, and 100% of the images had been completed after 1 h. In Table 3, 50 images had been edited by the operator Sokha in approximately 1.5 h; whereas, another 40 images were edited by the operator Chanthol in just around 1 h and 20 min. Meanwhile, another 42 images had been edited by the operator Bopha by using a longer time of 5 h, and another 40 images were edited by the operator Kunthea by spending the longest time of 6 h. Likewise, another 30 images were edited by the operator Sopheak by spending 2 h, while another 35 images were edited by the operator Bora in 2 h. Lastly, another 26 images were edited by the operator Dara by spending around 1 h.

These results tell the different times consumed and the different levels of completion of the image editing in accordance with the number of images and the capacity of the operators. The results also emphasize the capacity assessment of the system by which the data are obtained by interacting with real users.

Table 2. The results of customer reports by testing on five customers.

ID	Cus	Num Opt	OPT ID	To-Do	TD	Completed (%)
1	OKA	2	1, 2	105	9.5 h	86%
2	Lumizy	2	3, 4	94	11 h	87%
3	Bex	2	5, 6	55	≈4 h	91%
4	Boo	1	6	20	≈1 h	100%
5	Cam	1	7	26	1 h	100%

ID = Customer ID, Cus = Customer name, Num Opt = number of operator, OPT ID = Operator ID, To-Do = number of file, TD = total of duration.

6.2. Comparison of Result with the Questionnaire

The second type of evaluation of ISDT was done by users who had used the system. The users can provide feedback regarding their satisfaction with the system by answering the survey questionnaires, which is an input for a simulation of the reasoning process and to compare the outcome of the process with survey answers. The questionnaire consisted of 11 questions regarding the functions and friendliness of the ISDT. For each question, users were asked to tick a box to make a comparison to the real system result.

The results demonstrated that Table 2 column 6 (TD), Table 3 column 5 (Finish), and column 6 (D) are noteworthy. The "time spending" to edit images is not optimal for some operators. It can be observed that the operators with ID numbers 1, 3, 4, and 6 spent a very long time to complete the editing of the images compared to operators 2, 5, and 7. The different times spent by the ISDT system operators may be due to their knowledge and experience of the system processes which were running on their machines. However, another result in Tables 2 and 3 shows satisfaction with the results and usefulness of the system, with the accurate and effective time for the users.

Table 3. The result of operator report by testing on seven operators.

ID	OPT Name	To-Do	Finish	D
1	Sokha	55	50	7 h
2	Chanthol	50	40	2.5 h
3	Bopha	49	42	5 h
4	Kunthea	45	40	6 h
5	Sopheak	35	30	2 h
6	Bora	40	35	5 h
7	Dara	26	26	1 h

ID = operator id, OPT Name = operator name, To Do = project file, Finish = finished file, D = duration.

7. Discussion

The ISDT system uses a technique to store data information in the log file temporarily before it is converted into the database. The intention to get data from operators and servers (e.g., clients and controllers) who are cooperating in the company is the most important attribute of the whole system, in relation to the effects of ISDT on the log file. The ISDT system could be a useful system for explaining the intentions of tracking and managing on heterogeneous and complex data from the log files in an image editing company and provide the most accurate results for users. However, it is unlike some other studies, where systems used the log file to track failure prediction [29], to perceive behaviors of users [30], to analyze frequency mining, and to predict time consumption [16].

The tracking failure is less time while tracking the behaviors of users is less efficient because of the complexity and heterogeneity, and it is also difficult to allocate specific data. Therefore, there is a lack of data management as compared to a log file. The log file can easily be stored in a computer database system for easy access by the user interface.

The system for data tracking emphasizes the application of algorithms to pattern analysis and prediction. This study has created a data tracking processing system to which it could emphasize that the processing of data not only requires the right technologies and analytical tools/techniques [31], but also requires the consideration of the real work of an organization. This will drive the transformation from complex jobs to knowledge and business value processes.

8. Conclusions and Future Work

We have developed a fully automatic and reliable intelligent system for data tracking in an image editing company. Our system is based on three phases of processing. The first phase is the log file system, where the data of operator and client can be stored. The second phase is the agent script segmentation, where the agent script is the most important element of a system that runs at the back-end (server-side) to retrieve data from the log file, and then classify and deliver data to the database system. The agent script has two types that are running at server-side; namely, the first is the PHP script that automatically runs on the server machine for retrieval of data from the FTP log file; and the second is the C# script that automatically runs on the operator machine to retrieve data from the operator log file. In the third phase, the web interface provides a user-friendly way to handle data on a web browser by the users.

The ISDT system has been evaluated by a set of real users. The interaction of these users with a dataset comprised of the real projects from which it provided information about the performance of the ISDT. Additionally, responses provided by the users to the questionnaire offered useful feedback about their satisfaction with the proposed system. The results confirm the capability of ISDT to retrieve data from the log file and then to classify and store the data in the database, and finally, to view data on the web interface. The results also assert the ability of ISDT to achieve a high percentage of success and a reasonable satisfaction by the users with the proposed tool.

In the future, improvement of the ISDT system will be carried out in the following directions:

1. Plan to provide WebRTC via web-services that connect to browsers and mobile applications with real-time communications (RTC) [32].
2. Provide multiple users' registration for those who want to use ISDT system to manage their job.
3. FTP is less secure, and so the improvement of the FTP by allowing the client and company to transfer data through the system interface by which the data would be stored in PHP or C# library.
4. Build a data backup and recovery system which could protect the data.
5. Improving the system architecture would also be needed so as to decrease the time consumed and improve the accuracy of the results.

Author Contributions: All authors contributed equally to this paper. All authors read, revised and approved the final script.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baur, A.W. Harnessing the social web to enhance insights into people's opinions in business, government and public administration. *Inf. Syst. Front.* **2017**, *19*, 231–251. [CrossRef]
2. Jie, S.; Amankwah-Amoah, J.; Wang, X. A multidisciplinary perspective on big data in management research. *Int. J. Prod. Econ.* **2017**. [CrossRef]
3. Martínez-Rojas, M.; Marín, N.; Miranda, M.A.V. An intelligent system for the acquisition and management of information from the bill of quantities in building projects. *Expert Syst. Appl.* **2016**, *63*, 284–294. [CrossRef]
4. Heterogeneous Crowdsourced Data Analytics. Available online: http://ieeeaccess.ieee.org/special-sections-closed/heterogeneous-crowdsourced-data-analytics/ (accessed on 29 October 2017).
5. Integrating AJAX and Web Services for Cooperative Image Editing. Available online: http://ieeexplore.ieee.org/document/4216105/ (accessed on 29 October 2017).
6. Mazzei, M.J.; Noble, D. Big data dreams: A framework for corporate strategy. *Bus. Horiz.* **2017**, *60*, 405–414. [CrossRef]
7. Grierson, H.J.; Corney, J.R.; Hatcher, G.D. Using visual representations for the searching and browsing of large, complex, multimedia data sets. *Int. J. Inf. Manag.* **2015**, *35*, 244–252. [CrossRef]
8. Liu, M.S.; Zhou, Y. Application of artificial intelligence technology in asphalt concrete mixer control system. In Proceedings of the 4th International Conference on Fuzzy Systems and Knowledge Discovery, Haikou, China, 24–27 August 2007; pp. 159–163.
9. Yoon, T.E.; Ghosh, B.; Jeong, B.K. User Acceptance of Business Intelligence (BI) Application: Technology, Individual Difference, Social Influence, and Situational Constraints. In Proceedings of the 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 6–9 January 2014; pp. 3758–3766.
10. Wang, S.; Wu, C.; Gao, L.; Yao, Y. Research on Consistency Maintenance of the Real-time Image Editing System based on Bitmap. In Proceedings of the 2014 IEEE 18th International Conference on Computer Supported Cooperative Work in Design, Hsinchu, Taiwan, 21–23 May 2014.
11. Sun, D.; Sun, C.Z. Context-Based Operational Transformation in Distributed Collaborative Editing Systems. *IEEE Trans. Parallel Distrib. Syst.* **2009**, *20*, 1454–1470. [CrossRef]
12. Claes, J.; Vanderfeesten, I.; Pinggera, J.; Reijers, H.A.; Weber, B.; Poels, G. A visual analysis of the process of process modeling. *Inf. Syst. Bus. Manag.* **2015**, *13*, 147–190. [CrossRef]
13. Gregg, D.G.; Walczak, S. Exploiting the information web. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2007**, *37*, 109–125. [CrossRef]
14. Rabiei, M.; Hosseini-Motlagh, S.M.; Haeri, A. Using text mining techniques for identifying research gaps and priorities: A case study of the environmental science in Iran. *Scientometrics* **2017**, *110*, 815–842. [CrossRef]
15. Woodruff, J.; Alexander, J. MultiLog: A tool for the control and output merging of multiple logging applications. *Behav. Res. Methods* **2016**, *48*, 1296–1307. [CrossRef] [PubMed]
16. Fageeri, S.O.; Ahmad, R. An Efficient Log File Analysis Algorithm Using Binary-Based Data Structure. *Procedia Soc. Behav. Sci.* **2014**, *129*, 518–526. [CrossRef]
17. Why FTP Is No Longer Up to the Job. Available online: http://www.sciencedirect.com/science/article/pii/S1353485810700556 (accessed on 29 October 2017).
18. Free FTP Client Software for Windows. Available online: http://www.coreftp.com/ (accessed on 29 October 2017).
19. Server Logs. Available online: https://wiki.filezilla-project.org/Server_logs (accessed on 29 October 2017).
20. Arabnejad, E.; Moghaddam, R.F.; Cheriet, M. PSI: Patch-based scrip tidentification using non-negative matrix factorization. *Pattern Recognit.* **2017**, *67*, 328–339. [CrossRef]
21. Andress, J.; Linn, R. Introduction to web scripting with PHP. In *Coding for Penetration Testers*, 2nd ed.; Syngress: Waltham, MA, USA; pp. 151–175.
22. Amanatidis, T.; Chatzigeorgiou, A. Studying the evolution of PHP web applications. *Inf. Softw. Technol.* **2015**, *72*, 48–67. [CrossRef]
23. Managing CronJobs with PHP. Available online: https://code.tutsplus.com/tutorials/managing-cron-jobs-with-php--net-19428 (accessed on 29 October 2017).
24. Introduction to Windows Service Applications. Available online: https://docs.microsoft.com/en-us/dotnet/framework/windows-services/introduction-to-windows-service-applications (accessed on 29 October 2017).

25. System. Service Process Namespace. Available online: https://msdn.microsoft.com/en-us/library/system.serviceprocess(v=vs.110).aspx (accessed on 29 October 2017).
26. Data Classification. Available online: http://searchdatamanagement.techtarget.com/definition/data-classification (accessed on 29 October 2017).
27. What Is Data Classification? A Data Classification Definition. Available online: https://digitalguardian.com/blog/what-data-classification-data-classification-definition (accessed on 29 October 2017).
28. Database. Computer Science. Available online: https://www.britannica.com/technology/database (accessed on 29 October 2017).
29. Fronza, I.; Sillitti, A.; Succi, G.; Terho, M.; Vlasenko, J. Failure Prediction Based on Log Files Using Random Indexing and Support Vector Machines. *J. Syst. Softw.* **2013**, *86*, 2–11. [CrossRef]
30. Huang, Y.F.; Hsu, J.M. Mining web logs to improve hit ratios of prefetching and caching. *Knowl. Based Syst.* **2008**, *21*, 62–69. [CrossRef]
31. Tian, X.; Liu, L. Does big data mean big knowledge? Integration of big data analysis and conceptual model for social commerce research. *Electron. Commer. Res.* **2017**, *17*, 169–183. [CrossRef]
32. WebRTC Enabled Multimedia Conferencing and Collaboration. Available online: http://ieeexplore.ieee.org/document/6840017/ (accessed on 29 October 2017).