RFID Data Streams Processing Using Complex Event Processing to Enhance Students Performance

MONA MQBAS, HAYA AL-DOSSARY, MARIUS NAGY
College of Computer Engineering and Sciences
Prince Mohammed Bin Fahd University
SAUDI ARABIA
Mona.m.abdullah@gmail.com, haya.a.d@windowslive.com, mnagy@pmu.edu.sa

Abstract: - This paper illustrates a new RFID-based model to autonomously monitor students' attendance in universities to enhance students' performance. The system is mainly focusing on processing the primitive events generated by the attached RFID sensor network. The underlying processing is performed via an instance of ESPER complex event processing engine. The system collects RFID primitive events and feeds them as inputs into ESPER CEP engine in order to filter events, eliminate redundancy and generate sophisticated meaningful complex events. New meaningful events are stored in as students' attendance records. They constitute an efficient source to generate statistics, monitor students' absences and send warning letters to students. As a result, the system will save the instructor's time and give the students the opportunity to receive an accurate feedback from their instructors regarding their attendance.

Key-Words: - Event, CEP, RFID, Artificial Intelligence, Filtering, Data

1 Introduction

In some universities around the world, students' attendance is done manually by the instructors during every lecture. This action might create inconveniences. For instance, in some cases the instructor might forget to take attendance or by mistake he/she might mark the wrong student absent in case of having many students with similar names. Moreover, the instructor needs to check for each student in the class if he reached 5%, 10% or 15% absences and notify the registration office to send a warning letter to the student.

For the purpose of enhancing the quality of service provided by universities and educational institutes, we are suggesting using Radio Frequency Identifier (RFID) technology which will be applied to get clear and accurate records of student's attendance. When a student attends a class, the RFID reader in this particular class room will read the student's information, which is stored in the RFID chip that is installed on the student's ID, and send this information to the CEP engine. The engine will filter the streams of data to get rid of duplication and redundancy. Furthermore, the system will send warning letters to the students automatically based on their rate of absences and create statistics about the attendance. These statistics show and predict the students' interests on the major and the courses.

In this system, simulated RFID events will be sent as streams of data to the CEP engine through the corresponding Input Adapter. Then the engine will run the predefined EPL rule (query) against this input stream to filter the coming events and eliminate redundancy. Whenever an event is meeting the specified EPL rule, the engine will invoke the system (which is written in Java) to retrieve the corresponding student schedule from the database (Oracle Database 11g Express Edition) to verify that this particular student is present and mark him/her as present in the Attendance Table within the database. Later and based on the resulting analysis of these records, the system will detect absent students, generate statistics and send warning letters to students.

2 Related Work

RFID is a very powerful technology. However, RFID generates massive streams of primitive data. Sometimes those data streams contain duplicated, corrupted or meaningless data records [11]. Several researchers proposed models and systems using CEP to filter the data and to generate meaningful complex events. One of these models is BCEPS system [3] which consists of a physical devise layer, RFID data processing layer, event processing engine, application layer and data storage layer. BCEPS has a priority-based event scheduling

algorithm. The system rearranges the events according to event priority. Another study [16] verified the accuracy of BCEPS model using PTSA event scheduling algorithm and TSTLA tracking and locating algorithm. Moreover, another complex event processing based on RFID model is described in [4]. The model consists of four modules (Event Monitoring module, Buffer module, Event Processing module and the Event sending/subscribe module).

Based on the Time Automata theory, Liu, Zhang and Wang [15] proposed a new RFID complex event processing method GEEP. This method improves the efficiency of Real-Time Locating Systems (RTLS) by extracting more accurate complex events. They used the TA to model a diagram of RFID data streams to derive the constraint between object and location.

For the purpose of enhancing students' performance and providing a better quality of education, universities and education institutions around the world are implementing intelligent systems to improve the delivery of education and knowledge. For instance, Patel [12] described the obstacles a lecturer in any college or university would face by manually taking the students attendance; starting with the time consuming problem and ending with the probability that "some students may call his/her friend as "present" even though this student is currently absent". In this paper, the author described the RFID technology and proposed a web-based management system to monitor students' attendance records automatically. The system consists of a main server, a RFID-reader attached to a PC and a camera installed in every classroom. The system will collect the students' attendance statuses and their pictures then compare this data with the data stored in the university's database to create a record of the present students.

Agrawal and Bansal [13] identified and described the advantages of implementing RFID technology as a part of an attendance tracing system. Moreover, they proposed a system which was developed in C# using Microsoft Visual Studio. They integrated RFID technology along with a special object counter to overcome the false/fake attendance issue. The system will track all the RFID tags in the class room and the object counter will count the number of students in the room. Then the tow numbers will be compared, it they are equal, the system will send the RFID tag data to middleware to start filtering and processing.

R. Patal, N. Patal and Gajjar [14] proposed a real time intelligent system based on RFID hardware to record students' attendance. According to the authors "the aim of the system is to read the tags in class room and not read the students away or outside from the class room". Furthermore, the system contains a middleware which filters row RFID data and removes redundancy. Then the filtered data will be processed by the system in comparison to the corresponding data in the main database to create new attendance records.

In their paper, Rahman, Abassi and Shaikh [10] developed a model to build a smart university using RFID technology. The implementation of the proposed system will manage monitoring attendance records, switching control of electrical items and security locks of rooms. The model consists of a RFID sensor network, database server and an end user application to search profiles, view attendance records according to a specific date and generating graphs of each ID number records.

3 Concepts

3.1 RFID-Radio Frequency Identification

Radio Frequency Identification is a technology similar to Bar code technology; however, RFID has the advantage of the remote tracking. An RFID system or network consists mainly of readers and tags. RFID tag is the part of the system which stores data and uses radio frequency electromagnetic fields to transfer data signals. The other part of the system is the reader or the transceiver. The reader scans the coming data and passes them to another central device for processing. The middleware (software for reading and writing tags) and the tag can be enhanced by data encryption for security-critical application at an extra cost, and anti-collision algorithms may be implemented for the tags if several of them are to be read simultaneously [17].

3.2 CEP-Complex Event Processing

Complex Event Processing is applied to the concept of filtering, processing and correlating streams of primitive duplicated data; and outputting sophisticated meaningful events. CEP integrates data from various sources in order to generate complex conclusions and results for real-time applications and situations through triggering actions. CEP involves rules to aggregate, filter, and match low-level events, couples with actions to generate new, higher-level events from those events [5].

ISBN: 978-1-61804-307-8

3.3 EPL- Event Processing Language

Event processing language (provided by Oracle Cooperation) is a language used to declare queries to run by Esper CEP engine. It is very similar to SQL, but the main difference is that SQL query runs on tables (entities) and EPL query runs on data streams. EPL has SELECT, UPDATE, INSERT INTO, WHERE, HAVING, GROUP BY clauses and they work in the same way SQL clauses do. EPL rules allow the engine to filter and correlate the coming events in the data streams by running EPL statements on the raw data. An example of EPL syntax is:

select select_list
from stream_def [as name] [, stream_def [as name]] [,...]
[where search_conditions]
[group by grouping_expression_list]
[having grouping_search_conditions]
[output output_specification]
[order by order_by_expression_list]
[limit num_rows] [3]

3.4 Esper CEP Engine

Esper CEP engine is an Event Stream Processing and Complex Event Processing Engine. Esper engine is capable of triggering actions and events when real-time event condition occurs among event streams. Esper is a lightweight kernel written in Java which is fully embeddable into any Java process. It enables rapid development of application that process large volumes of incoming messages or events [2].

4 System Architecture

Our system consists of five layers which are illustrated in Fig.1:

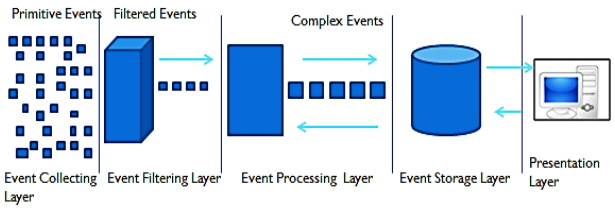
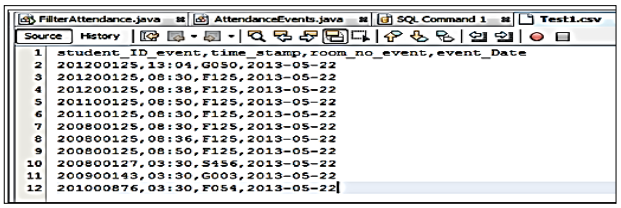


Fig.1. System architecture

4.1 Event Collecting Layer

This layer contains RFID readers and tags which are responsible of generating the primitive events. Since a real RFID system is not implemented yet for this system, a CSV file (refer to Fig.2) had been used to simulate the RFID system events. This CSV file is populating the engine with the student IDs, the attendance time stamp and the class room number.

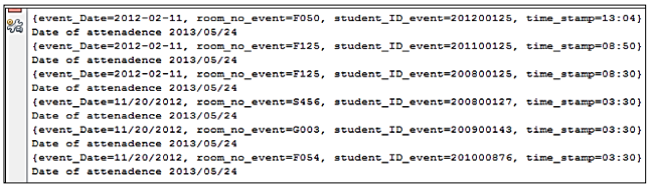


	student_ID_event,time_stamp,room_no_event,event_Date
1	201200125,13:04,G050,2013-05-22
2	201200125,08:30,F125,2013-05-22
3	201200125,08:38,F125,2013-05-22
4	201100125,08:50,F125,2013-05-22
5	201100125,08:30,F125,2013-05-22
6	200800125,08:30,F125,2013-05-22
7	200800125,08:36,F125,2013-05-22
8	200800125,08:50,F125,2013-05-22
9	200800127,03:30,S456,2013-05-22
10	200900143,03:30,G003,2013-05-22
11	201000876,03:30,F054,2013-05-22

Fig.2. Primitive attendance events CSV file

4.2 Event Filtering Layer

The filtering in this layer is performed by the CEP engine. The CEP is running an EPL query to select only the unique student IDs within one hour (refer to Fig.4). This process is done in order to completely eliminate events redundancy. Even though the engine is capable of generating 500,000 events per second, the output had been limited to be generated one by one. The filtered events are illustrated in Fig. 3.



```
{event_Date=2012-02-11, room_no_event=F050, student_ID_event=201200125, time_stamp=13:04}
Date of attenadence 2013/05/24
{event_Date=2012-02-11, room_no_event=F125, student_ID_event=201100125, time_stamp=08:50}
Date of attenadence 2013/05/24
{event_Date=2012-02-11, room_no_event=F125, student_ID_event=200800125, time_stamp=08:30}
Date of attenadence 2013/05/24
{event_Date=11/20/2012, room_no_event=S456, student_ID_event=200800127, time_stamp=03:30}
Date of attenadence 2013/05/24
{event_Date=11/20/2012, room_no_event=G003, student_ID_event=200900143, time_stamp=03:30}
Date of attenadence 2013/05/24
{event_Date=11/20/2012, room_no_event=F054, student_ID_event=201000876, time_stamp=03:30}
Date of attenadence 2013/05/24
```

Fig.3. Filtered events

4.3 Event Processing Layer

In this layer of the system, processing is performed over three phases based on the existing of the event:

4.3.1 Phase 1: Insert Attendance Status as Present

When the CEP engine generates the filtered events, the system will be invoked to retrieve each student's schedule in the output stream then it will compare the timestamp and room number. This process is to verify that this specific student had attended the class at the required time and room. Later, the system will insert a new record in the attendance table marking this student present in the corresponding course and date. This process is illustrated in Fig.5 and the complex events (attendance records) are illustrated in Fig.4.

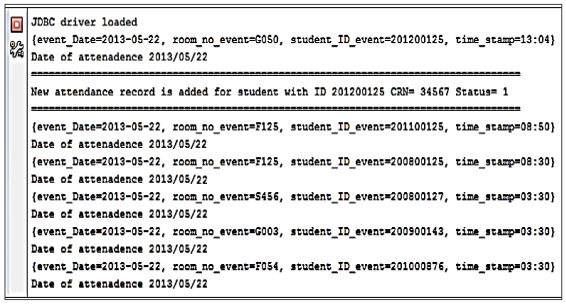


Fig.4. Generating complex events (attendance records)

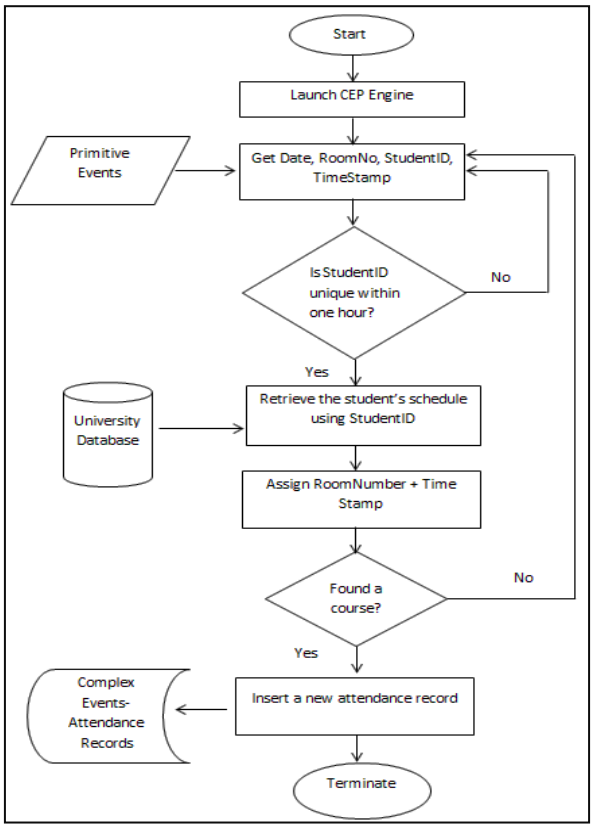


Fig.5. Insert attendance status as present flow chart

4.3.2 Phase 2: Insert Attendance Status as Absent

A timer will be set every day to call InsertAbsent() method at the end of the day. After the end of the last class session, the system will retrieve all the courses that have a class period on that day – for example: if the day is Monday, the system will select all the courses which have lectures on Monday. Then it will retrieve all enrolled students in these courses and do not have any attendance records for that particular day in the attendance table. The system will mark these students absent and insert these records in the attendance table.

4.3.3 Phase 3: Send Warning Letters

During this phase of the system performing process, a timer will be set to wait for a period of two weeks to check the number of absences for each student. The system will retrieve the data from the attendance table and count each student absences. It will check if the student had reached 5%, 10% or 15% of absences and will insert these records in the warning letters table. A notifying email will be sent automatically to the student to notify him/her about his/her absences percentage. When a student reaches 15%, the system will send an email to the instructor to notify him/her that this student needs to be dropped from the course.

4.4 Event Storage Layer

All of the generated complex events – attendance records- are being stored in a local client version of Oracle Database 11g. This database may be considered as an attendance view of what a university database might look like. Fig.6 is illustrating the structure of the database that is used for this system.

4.5 Presentation Layer

This layer is implemented by a Java application. This application has been created using the Swing API in NetBeans. The main aim of creating this application is to give a user-friendly access to the automatically generated attendance records in the database. Once the authorized admin login to the application, he/she can have a full access to modify any student's attendance status to present, absent or excused absent. Furthermore, the admin can generate statistics based on the attendance records corresponding to each student, instructor and college. Also, dynamic PDF reports can be created based on the statistics.

ISBN: 978-1-61804-307-8

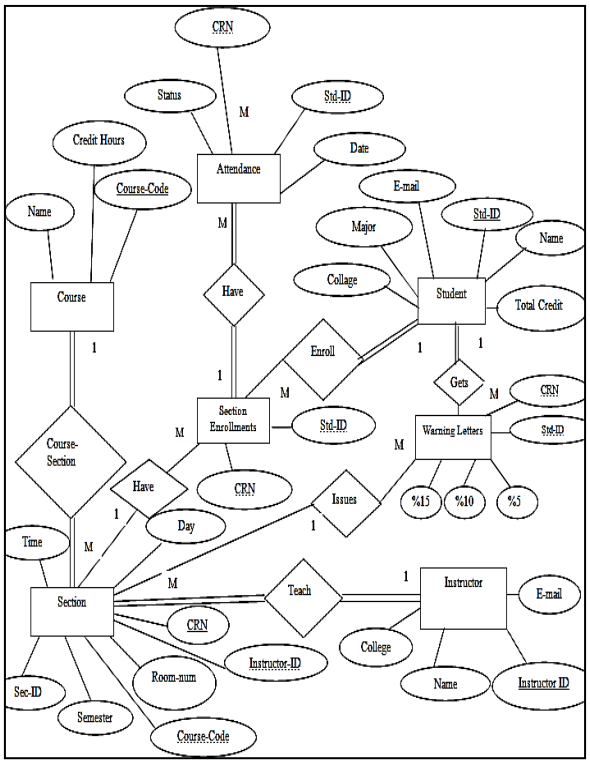


Fig.6. Entity Relation Diagram (database structure)

5 Features Description

5.1 Reading Input Streams

The simulated attendance records are stored in a CSV file. They are ordered in the same order of prosperities in the POJO (Plain Old Java Object) Java Bean which provides the setter-methods and getter-methods to access event properties. A CSVInputAdapter is used to create an input steam of all of the events coming from the CSV file and to pass them down to the CEP engine.

5.2 Filtering Events

As soon as the CSVInputAdapter instance passes the events stream to Esper CEP engine, the engine starts running the EPL query ("SELECT DISTINCT student_ID_event , time_stamp, room_no_event,event_Date from "+"cep.AttendanceEvents.std:firstunique(student_ID _event).win:time(50 min)output every 1 events") against streams of students attendance records events . This query is selecting only the first unique entering Student ID within a window size of 50 minutes. When an event meets the specified condition in the EPL statement, the engine will output one filtered event at a time. Later the UpdateListener class will be invoked through passing the event object.

4.3 Processing Events

Most of the events processing and complex events generating happens within the UpdateListener class. First the event properties will be extracted from the event object - using event.get(PropertyName) method - and their values will be assigned to variables. Then for each student ID and based on the room number, the CRN of the particular course he/she is attending will be retrieved from the SECTION_ENROLLMENTS table in the database. The event time stamp will be compared against the starting time of the class period, if the comparison is true, a new attendance record will be inserted in the attendance table with the student ID, course CRN, current day date and a 1 for the attendance status.

Along with the lunching of the system, a timer is set to count 16 hours (from 8 a.m. till 12 p.m.). When the time is almost midnight and with the end of the last class period, the system will retrieve all courses which have a class period in this particular day (e.g. Monday). The courses are retrieved as a one CRN Result Set object. Inside the while loop, which is iterating over the CRNs, another sql statement will be executed to retrieve all the enrolled students in the corresponding course. Every student ID will be checked if it has any attendance record (with the same CRN and date) in ATTENDANCE table in the database. If not, a new record will be inserted for this student to mark him/her absent in that particular course and day.

5.4 Check Absents Percentages

Within the system a timer is set to check every two weeks the number of absences of each student. Number of absences and the related course CRN will be retrieved from ATTENDANCE table through a sql statement. Furthermore, the number of lectures per week of the corresponding course will be retrieved too. Then the system will check the number of absents and the number of lectures to conclude whether this student had reached a 5%, 10% or 15% of absences. In the case of reaching any of these percentages, a new record will be inserted into WARNING_LETTERS table to indicate that a warning letter will be send to this student and to the instructor of the course in case of reaching 15%.

ISBN: 978-1-61804-307-8

5.5 Sending E-Mails

When a new record is inserted into the WARNING_LETTERS table in the database, the value of the Mail attribute within this table is set to 0, which indicates that a mail has to be sent to the corresponding student. Then the system will select all the rows in this table where (Mail=0) and retrieve each student's email to send him/her the warning letter with the reached percentage of absences. After sending the mail successfully, the WARNING_LETTERS table will be updated by setting the value of the Mail attribute to 1.

5.6 Generate Statistics

For the purpose of generating graphical statistics based on the students' attendance rates, JFreeChart library had been used to create a chart object and dataset, and to plot this chart and save it as a png image. When the authorized admin login into the system, he/she will be able to choose whether to generate statistics based on a specific student, instructor or college performance (refer to Fig.7). Then the system will execute the related SQL statement, calculate the attendance percentage of each related course-based on the records in ATTENDANCE table -, round the Double percentage to the second decimals and populate the dataset to display a chart in a new panel and save it as illustrated in Fig.8. and Fig.9.

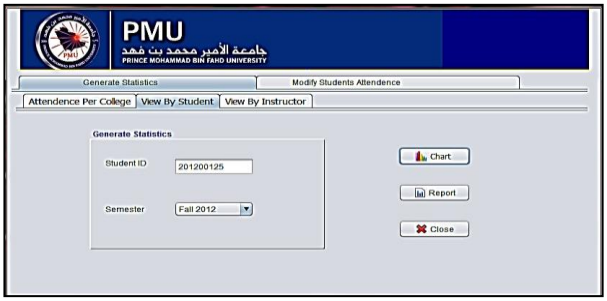


Fig.7. Generate statistics about students' attendance

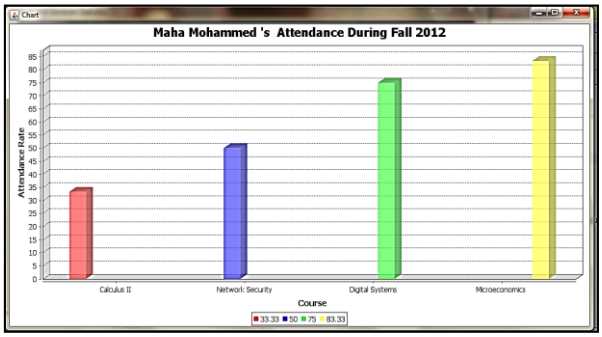


Fig.8. Graphical statistics based on a student attendance

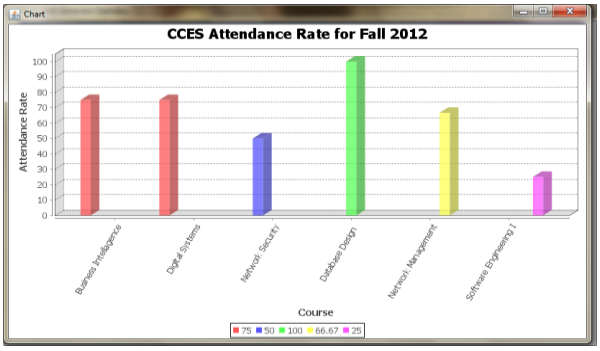


Fig.9. Graphical Statistics based on CCES Students' Attendance Rates

5.7 View Attendance Records

The application interface makes attendance records available for illustration and modification. To view a specific attendance record, the admin must enters a specific student ID in the text field, select a course form the dropdown list and choose a data from the JDateChooser as illustrated in Fig.10.

If the admin did not select a date, the system will select all the related records in ATTENDANCE table where STD_ID and CRN are matching the giving data by the admin Fig.11.

Fig.10. View attendance record for a particular course in a specific day

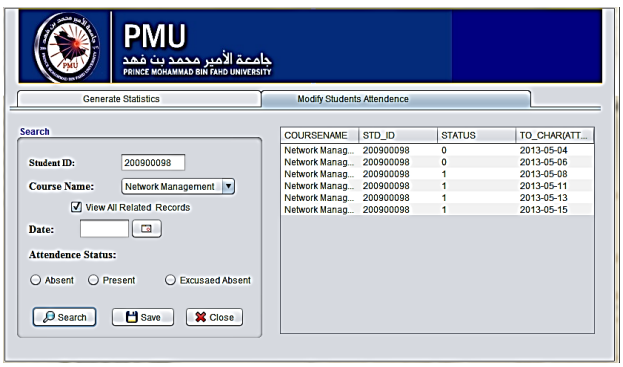


Fig.11. View all records of a particular student and course

ISBN: 978-1-61804-307-8

5.8 Modify Attendance Records

After selecting the specified attendance record, the admin can choose to change the status to Absent, Present or Excused Absent by choosing one of the related JRadiobuttons. Then the system will execute a SQL Update statement in order to commit the admin changes and refresh the table. As illustrated in Fig.12.

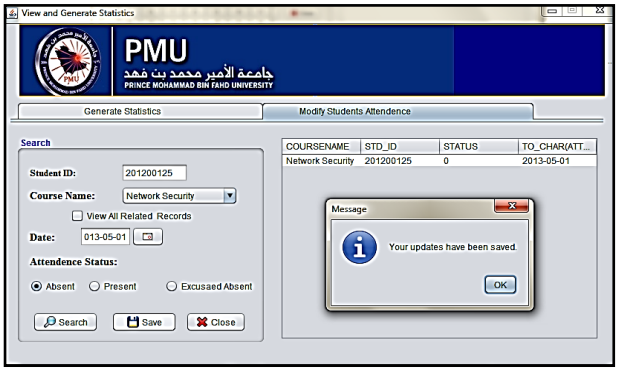


Fig.12. an attendance record status has been modified and updated

5.9 Generate PDF Reports

For the purpose of achieving this feature, IText Java library had been installed and imported into the system. Using IText library, Document class had been imported and an instance of this class called document had been created too. Further, this document instance had been used to create a separate PDF file for each student, instructor or college-refer to Fig.13- the admin chose and save in the specified path. It was used also to open the PDF file, write into it and insert a copy of the previously generated chart in it.

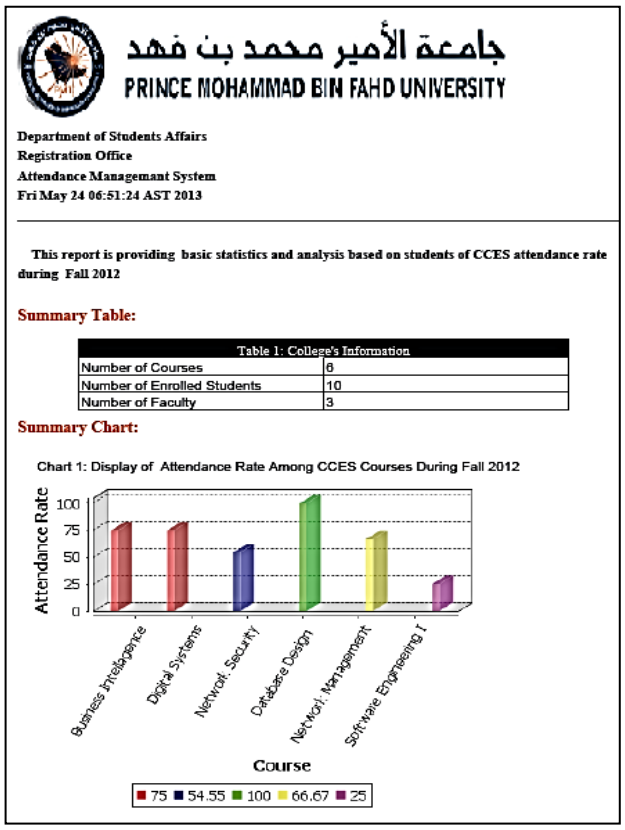


Fig.13. CCES PDF report

6 Conclusion

In this paper, we designed and implemented a complex event processing system based on simulated RFID data streams. The purpose of this system is to facilitate taking attendance and enhance the quality of service in universities and educational institutes through monitoring students' performance and attendance rates. An instant of Esper CEP engine is used to eliminate the redundancy from the RFID simulated primitive data streams and to generate sophisticated complex events (students' attendance records). The system proved its functionality through the accurate elimination of redundancy in primitive data streams and rapid generation of meaningful events (students' attendance records). It also replaces the traditional manual process of taking students' attendance with a rapid fully automatic one with a user friendly and easy to use interface.

Moreover, we plan to continue working on the system by implementing a prediction algorithm to mine the RFID filtered complex events generated by the system. The algorithm will use the complex events as a data set to predict whether the students are actually interested in the major they are studying and help them in making a better decision in this regards.

ISBN: 978-1-61804-307-8

References:

[1] Mark Palmer, "Seven Principles of Effective RFID Data Management", Progress Software-Real Time Division.

[2] E. Wu, Y. Diao and S. Rizvi, "High Performance Complex Event Processing over Streams". SIGMOD, Chicago, 2006, pp.407-418

[3] G Zhang and L Zhang, "Study of CEP-Based RFID Data Processing Model". SISIITA, 2008, pp.254-258.

[4] G. Wang and G. Jin, "Research and Design of RFID Data Processing Model Based on Complex Event Processing". ICCSSE, 2008 pp.1396-1399.

[5] D. Robins, "Complex Event Processing".

[6] C. Zang and Y. Fan "Complex Event Processing in Enterprise Information Systems Based on RFID"

[7] Supreet Oberoi , "Introduction to Complex Event Processing and Data Streams".

[8] G. Cugola and A. Margara, "Processing Flows of Information: From Data Stream to Complex Event Processing".

[9] E. Badr, "Reliability Solutions for a Smart Digital Factory using: (1) RFID based CEP (2) Image Processing based Error Detection (3) RFID based HCI" M.S. thesis, University of Stuttgrat, Stuttgart. Germany, 2011

[10] Y. Bai , F. Wang and P. Liu, "Efficiently Filtering RFID Data Streams," 2006

[11] G. Zhang, L. Zhang and X. Liu, "Study on CEP-Based BCEPS Model of RFID-Based RLTLUM System," *Journal of Software*, vol. 4, no. 6, August 2009, pp.605-61 3.

[12] U. Patel, "Student Management System based on RFID Technology," *International Journal of Emerging Trends and Technology in Computer Science*, vol. 2, issue. 2, November-December 2013, pp. 173-178.

[13] A. Rehman, A. Abbasi and Z. Shaikh, "Building A Smart University using RFID Technology," in *Proceeding of the International Conference on Computer Science and Software Engineering*, 2008, pp.641-644.

[14] R. Patel, N. Patel and M. Gajjar "Online Students' Attendance Monitoring System in Classroom Using RFID: A Proposed System Framework," *International Journal of Emerging Technology and Advance Engineering,* vol.2 ,no 2, February 2012, pp. 61-65

[15] A. Agrawal and A. Bansal "Online Attendance Management System Using RFID with Object Counter," *International Journal of Information and Computation Technology*, vol. 3, no. 3, 2013, pp. 131-138.

[16] Y. Liu, H. Zhang and Y. Wang, "RFID Complex Event Processing: Applications in Real-Time Locating System," *International Journal of Intelligent Science*, 201.2, pp. 160-165

[17] E. Zudor, Z. Kemeny , P. Egri and L. Monostori "The RFID Technology and It's Current Applications" *In proceedings of The Modern Information Technology in the Innovation Processes of the Industrial Enterprises, 2006, pp. 29-36.*