

Developing a Scalable, Cross-Facility-Based Therapeutic Decision Reminder Engine

Cheng-Yi Yang, Ming-Chieh Tsai, Ray-Jade Chen, Yu-Sheng Lo, Lan-Ying Kang, Chien-Tsai Liu

Abstract: Medication errors are expensive and sometimes costly to healthcare systems, patients and their families, and clinicians. The National Health Insurance of Taiwan provided a NHI-PharmaCloud web-based database that contains the most complete drug histories of patients across different healthcare facilities in Taiwan. Several research studies have reported that adoption of a computerized physician order entry system (CPOE) with clinical decision support (CDS) function by medication prescribers can prevent medication errors and improve medication safety in patients; however, new drugs and drug-related knowledge are being quickly and frequently released. Thus, the same pharmacological mechanism of different drugs can be hard to detect; furthermore, there is a lack of patient-centered, integrated medication history that is necessary to prescribe the correct medication. Therefore, prevention of medication errors remains difficult for the clinical staff. To flexibly add more checking rules and improve drug-use safety among patients, we developed a scalable, expandable decision reminder solution (Therapeutic Decision Reminder Engine; TDRE) based on the NHI-PharmaCloud in the Taipei Medical University Hospital, which would support the clinical staff and more efficiently prevent medication errors. In addition, to ensure an effective TDRE performance in future implementations, we also conducted load testing to determine the maximum workload that TDRE could support. The test results showed that TDRE was effective and able to support our expected service quantity.

Index Terms—NHI-PharmaCloud, CPOE, CDS, medication errors, load testing

Manuscript received July 19, 2016; revised August 2, 2016. This work was supported by the Industrial Technology Research Institute of Taiwan (ITRI) project: F301AR9A10.

Cheng-Yi Yang is with Healthcare Service Division, Medical Informatics Dept., Industrial Technology Research Institute, ITRI, Hsinchu, Taiwan corresponding author to provide phone: +886 35918408 fax: +88635915007; e-mail: CY.Yang@itri.org.tw)

Ming-Chieh Tsai is with Healthcare Service Division, Medical Informatics Dept., Industrial Technology Research Institute, ITRI, Hsinchu, Taiwan. (MingChiehTsai@itri.org.tw)

Ray-Jade Chen is with Taipei Medical University Hospital, Taipei, Taiwan (rayjchen@tmu.edu.tw)

Yu-Sheng Lo is with Graduate Institute of Biomedical Informatics, Taipei Medical University, Taipei, Taiwan (Loyusen@tmu.edu.tw)

Lan-Ying Kang is with Information Technology Apartment, Taipei Medical University Hospital, Taipei, Taiwan (145095@h.tmu.edu.tw)

Chien-Tsai Liu is with Graduate Institute of Biomedical Informatics, Taipei Medical University, Taipei, Taiwan (ctliu@tmu.edu.tw)

I. INTRODUCTION

Medication errors include prescription faults and prescription errors [1]. If the prescriber fails to consider medication prescription rules or ignores a patient's personal information (i.e., pregnancy, allergies, drug-use history, etc.), duplicated medication, drug-drug interactions, or other medication errors may occur; occasionally, it may even lead to an adverse drug event (ADE) [2]. Research statistics [2, 3] in the United States indicate that, on average, medication errors harm at least 1.5 million patients, with approximately 400,000 preventable adverse events occurring every year and at least one medication error per day. Medication errors are expensive and sometimes costly to healthcare systems, patients and their families, and clinicians [3]. Moreover, one of the medication errors, the duplication of medication, not only harms patients [1, 4, 5] but also causes serious environmental pollution and increases the financial burden on health care systems as well as wastes medical and social resources [6-8]. Therefore, prevention of medication errors is a very critical issue in hospitals [2]. Because most of the medication errors occur at the step of prescribing, many studies suggested that computerized physician order entry (CPOE) system combined with clinical decision support (CDS) function can prevent medication errors and improve drug-use safety among patients [2, 5, 9, 10]; however, medication errors still occur. Several researchers believe that the adoption of CDS has actually resulted in more medication errors due to several reasons [3, 5, 11, 12]. First, the design of CDS checking algorithms is not able to recognize medication orders properly. For example, new drugs and drug-related knowledge are being released quickly and frequently. Moreover, CDS does not recognize new drugs with the same pharmacological mechanism and oral and intravenous forms of a medication as the same drug [13]. Furthermore, the CDS system has medication recognition rules usually hard-coded that are not easy to update. Second, when patients are transferred or encountered across different facilities, physicians do not have access to their integrated medication history, necessary to give a correct medication order [13]. Therefore, if the CPOE system with a CDS function is adopted without a patient-centered medical history, medication errors will continue to occur [8, 12]. In Taiwan, several patient-centered, nationwide electronic health record (EHR) infrastructures have been implemented, including the NHI-IC Card (the Smart Card) [4], NHI-PharmaCloud [4, 14], My Health Bank [14], and the National Electronic Medical

Record Exchange Centre (EEC) [15]. Of these, the NHI-PharmaCloud contains the most complete personal drug history across different healthcare facilities. NHI-PharmaCloud is a web-based database provided by the National Health Insurance of Taiwan initiated in July 2013. The National Health Insurance of Taiwan is a single-payer national insurance healthcare system that provides a comprehensive insurance coverage rate of over 99% for the majority of Taiwan's 23 million inhabitants [15-18]. Due to its high coverage rate, NHI-PharmaCloud records almost all medications used by each patient [18] and provides the authorized clinical staff members at NHI-contracted facilities an access to most recent 3-month medication records of their patients. These records include the patient ID, drug names, the drug-related Anatomical Therapeutic Chemical Classification (ATC) code, the prescription date, the location of drug prescription, and quantities and the estimated surplus quantity of drugs that can be accessed [4, 17, 18]. NHI-PharmaCloud supports the batch mode that downloads reserved patients' medication information before a medical encounter. Hence, if we adopt the CPOE system with CDS based on a cross-facility, patient-centered medication history for real-time order inspection, alongside the scalability to extend drug knowledge and recognition rules, this system can decrease prescription errors, eliminate the waste associated with duplicate orders, and improve patient-drug safety. Therefore, the objective of this study was to develop a scalable, expandable decision reminder solution based on NHI-PharmaCloud to support the clinical staff and to more efficiently prevent medication errors.

II. FRAMEWORK DESIGN

To add more flexible medication order checking rules, and to improve patient drug-use safety, we developed a Scalable, Cross-Facility-Based Therapeutic Decision Reminder Engine (TDRE) based on NHI-PharmaCloud. In this study, we collaborated with Taipei Medical University Hospital and used the batch mode to retrieve patients' medication information from NHI-PharmaCloud using their patient ID and consent duration. We used these retrieved datasets with TDRE to check the physician's order (prescriptions) in real time during a patient encounter. In addition, to ensure TDRE performance in future implementations, we also conducted load testing to determine how the maximum workload TDRE would be able to support once it is deployed on a basic hardware system.

The TDRE system was developed using the Java programming language with Spring MVC, Spring REST web service framework [19], and Drools (Business Rules Management System) [20]. Fig. 1. describes the six main components of TDRE: extensible reminder logic, the management module, the PharmaCloud adapter, a local repository, a patient history access module, and the TDRE adapter. In the following, we discuss the details of TDRE and the functions of each component.

A. Extensible reminder logic

The extensible reminder logic component was developed with the Drools Business Rules Management System [20] to modularize the related reminder logic to be more scalable and expandable. In this study, we developed four reminder logics: the duplicated medication reminder, maximum dose reminder, pregnant drug reminder, and high risk drug reminder components. In this study, we present the duplicated medication reminder component as an example to describe the mechanism of the extensible reminder logic. To act in concert with the NHI medical duplicated checking scheme, the duplicated medication checking logic is shown in Fig. 2.

B. The Management module, PharmaCloud adapter, Local Repository, and Patient history access module

The management module provides the hospital administrative staff with the patients' ID and consent duration as parameters to enable the PharmaCloud adapter. The PharmaCloud adapter invokes the NHI VPN to retrieve the patients' three-month medication record via batch mode prior to the patient's medical encounter. The retrieved medication information is deposited into the local repository. The local repository was designed with the PostgreSQL [21] relational database, and the patient history access module was developed with Java and the Hibernate data access model [22] to support the extensible reminder logic in queries of the patients' medication information from the local repository.

C. The TDRE Adapter

The TDRE Adapter includes the Controller and the Model. The Controller was built with the Spring MVC Framework and Spring REST web service [19]. The client can query their patient's medication information using the patient's ID and ATC code to check if a new order will be a duplicated medication order. The Model includes several Fact classes for the extensible reminder logic to check the rules. The REST WS of TDRE can provide an implementer to integrate with the CPOE in the hospital or a smartphone app.

III. INFORMATION WORKFLOW

In this section, we provide two scenarios, retrieving NHI-PharmaCloud patients' drug history and determining therapeutic duplication, to describe the interactions and workflow between the components of TDRE.

A. Retrieving NHI-PharmaCloud patients' drug history [Fig. 3]

1. The clinical staff logs in to the manager module.
2. The clinical staff uses the manager module to initiate the PharmaCloud adapter to download the last three months of cross facility drug history.
3. The PharmaCloud adapter sends the batch request message to NHI PharmaCloud; the message includes a

list of the patients' IDs and these patients' consent durations.

4. NHI PharmaCloud asynchronously responds to the request sending the information back to the PharmaCloud adapter.
5. The PharmaCloud adapter receives the returned batched query results and stores these records in the local repository.

B. Determining therapeutic duplication [Fig. 4]

1. The clinical staff uses CPOE or APP to check if the order for the patient is a therapeutic duplication (Fig. 4, Step 1).
2. The CPOE or APP invokes the rest web service of the TDRE adapter and posts the patient's ID and drug ATC code (Fig. 4, Step 2).
3. The TDRE adapter invokes the extensible reminder logic and passes the ID and ATC code (Fig. 4, Step 3).
4. The extensible reminder logic invokes the patient history access module to query the local repository using the patient's ID to return the patient's three-month drug history back to the extensible reminder logic (Fig. 4, Steps 4, 5, 6, and 7).
5. The extensible reminder logic uses the information from Steps 3 and 7 to run the therapeutic duplicated checking logic.

IV. PERFORMANCE TESTING DESIGN

Because the time in a clinical medical encounter is limited, the system performance must be efficient and react in real time. According to the previous studies [23, 24], the benchmark of request-response times is 2 seconds. Therefore, in this study, we conducted load testing to realize the maximum affordable workload of TDRE for a request-response time of less than 2 seconds. The results could indicate a need to make trade-offs in the deployment settings. Because retrieving the NHI-PharmaCloud patients' drug history is done in batch mode prior to the medical encounter, we did not run load testing for this scenario. Instead, we focused on load testing for determining therapeutic duplication.

A. Test Environments

We developed a Client to simulate CPOE and APP to initiate the patient's medication order to TDRE and to check the medication against the duplicated medication order rules (Fig. 5). To simulate a random query to TDRE (following the Poisson distribution), we adopted the Uncommons Maths [25] Java library API in the Client. To record every transaction and its timestamp, we added log4j logger in all the source code (including TDRE and the Client); then, we calculated the request-response time of each transaction through the log. We deployed TDRE on a PC server with an Intel i5-3317U 1.70

GHz CPU, 2 GB RAM, Microsoft Windows 7 Professional 64-bit, postgresql 9.3, and JDK 1.8. We deployed both the Client and TDRE in this hardware environment.

B. Test Case Design

As we performed load testing, the Client increased the workload gradually and we calculated the test results from the transaction log. In Fig. 5, A is the timestamp of the request initiated by the Client, and B is the timestamp of the response received from TDRE. B minus A equals the request-response time of each transaction (called Ttotal). The workload was increased until the test result shown in $T_{total} > 2$ seconds, such that the performance objective was no longer satisfied. Therefore, the previous test values became the maximum affordable workloads.

V. PERFORMANCE TESTING RESULT

Table I shows the load testing results for TDRE. We sent requests from the Client to TDRE starting from 1000 requests per minute and increased the workload incrementally. Following Test number 8, where the Client sent 1200 requests to TDRE, the request-response time soared, rising to values much greater than our benchmark of 2 seconds. Therefore, the maximum affordable workload of TDRE is 1100 requests per second. From the test results we can infer that TDRE would be able to provide 528,000 ($1100 \times 60 \times 8 = 528,000$) requests during an eight-hour working day.

TABLE I. LOAD TESTING RESULTS.

| Test number | Requests/minute | Request-response time (seconds) |
|-------------|-----------------|---------------------------------|
| 1 | 500 | 0.112 |
| 2 | 600 | 0.123 |
| 3 | 700 | 0.142 |
| 4 | 800 | 0.164 |
| 5 | 900 | 0.161 |
| 6 | 1000 | 1.438 |
| 7 | 1100 | 1.472 |
| 8 | 1200 | 3.364 |

VI. DISCUSSION AND CONCLUSION

To prevent medication errors, we developed a scalable, expandable intelligent decision reminder solution, TDRE, based on NHI-PharmaCloud to support clinical staff and to more efficiently prevent medication errors. The test results show that TDRE worked properly and is able to fulfill our expected goal. Field-testing at Taipei Medical University indicates that the average service volume (inpatient plus outpatient) is approximately 6000 to 7000 encounters per day. Therefore, TDRE should be able to support this service

quantity. In future work, we have several issues we would like to focus on.

1. We would like to examine the effectiveness of TDRE after implementation in real clinics, including improvements in saving time during encounters and how many medication errors were captured and blocked by TDRE.
2. We also want to continue to add more drug hazard prevention rule modules into the extensible reminder logic to expand the reminder scope of TDRE.
3. We would like to incorporate other patient-centered nationwide EHR data, including the NHI-IC Card (the Smart Card) [4], My Health Bank [14], and EEC (the National Electronic Medical Record Exchange Centre) [15], to provide a more comprehensive and complete patient service.
4. We will focus on the UX design to allow the clinical staff to invoke TDRE more efficiently.

5. We would like to examine if the reminder system design disturbs the medical encounter in the clinic in any way. Multiple studies have indicated that excessive alerting might cause decision support overload and alert fatigue, causing the clinical staff to ignore or override alerts[5, 13]. Therefore, we need to balance providing the reminder with disruptions to the medical encounter to ensure that the reminder will be useful.

ACKNOWLEDGMENT

This work was supported by the Industrial Technology Research Institute of Taiwan (ITRI) project: F301AR9A10. The authors would like to express our appreciation to the Uncommons Maths open source software developers. We also like to thank Kent Chen for the partial technical support.

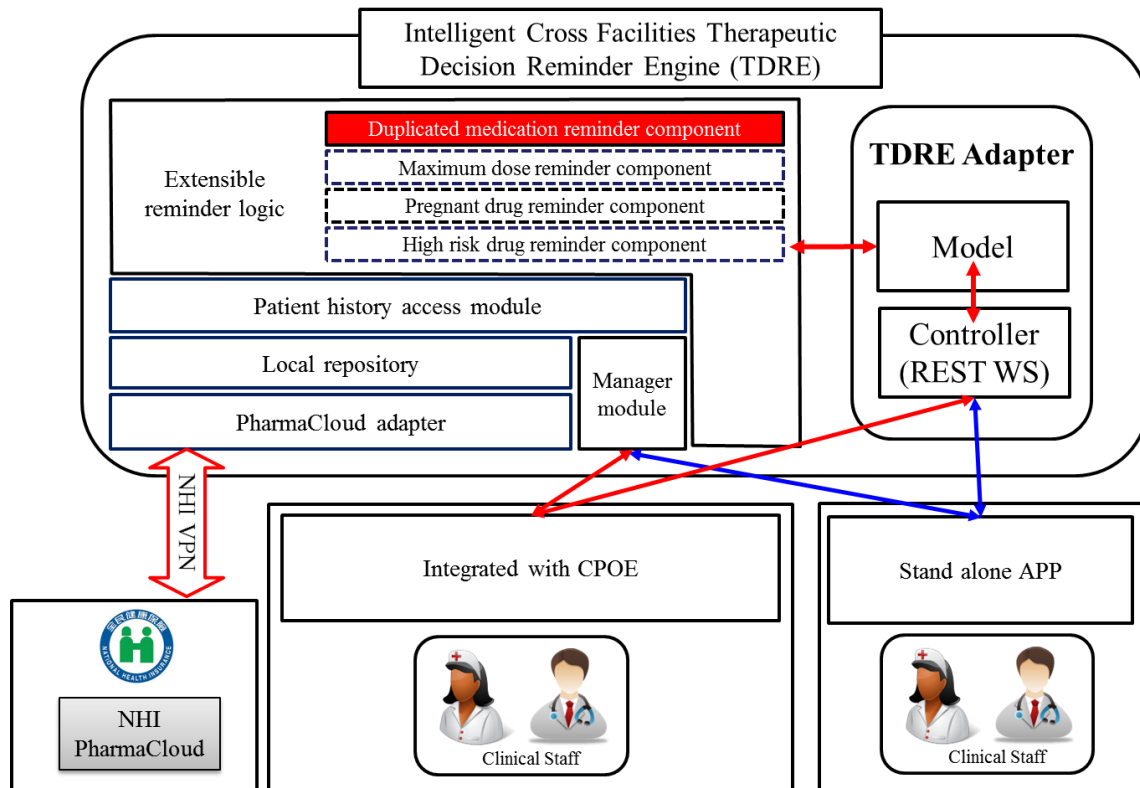


Fig. 1. The Intelligent Cross Facilities Therapeutic Decision Reminder Engine, or TDRE.

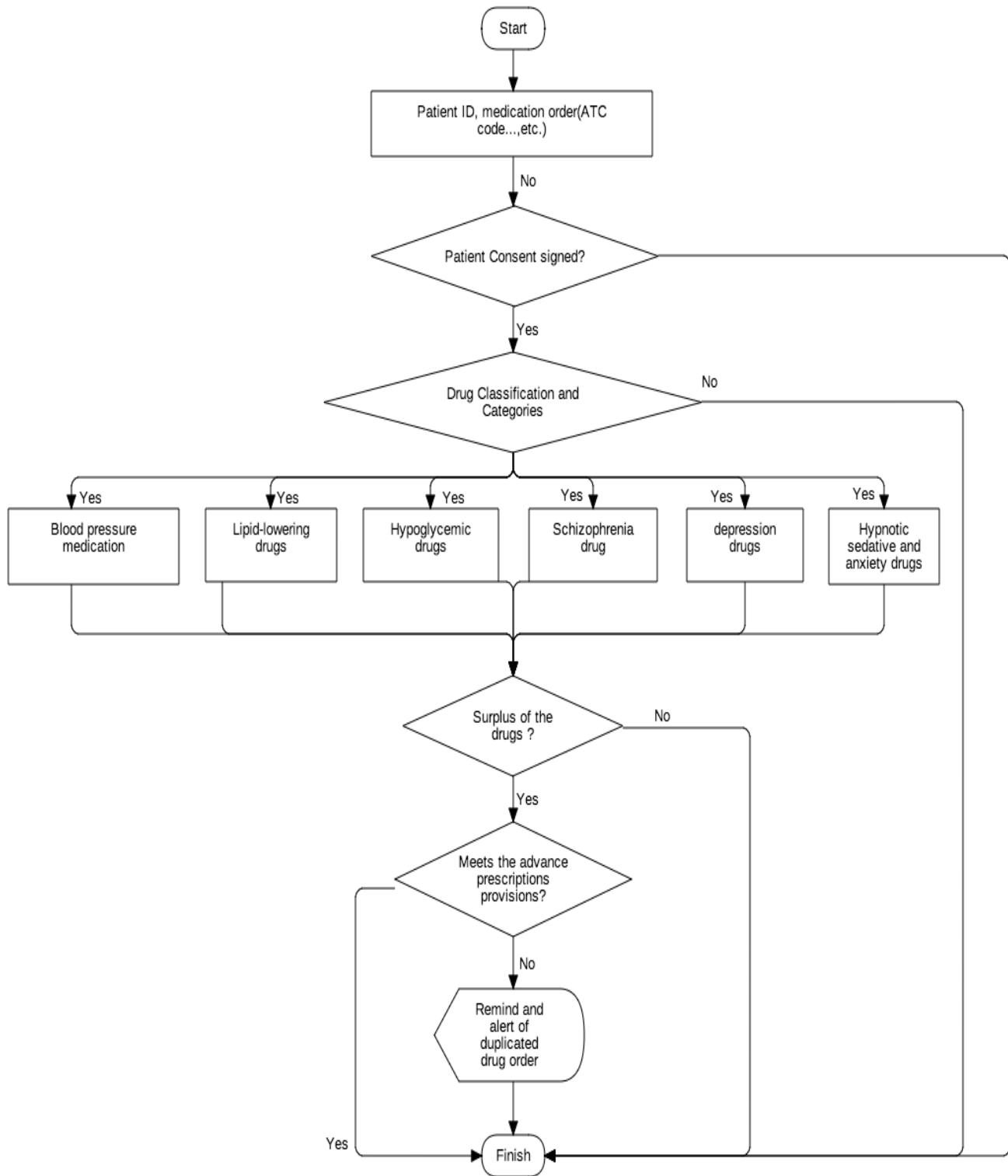


Fig. 2. The duplicated medication checking logic.

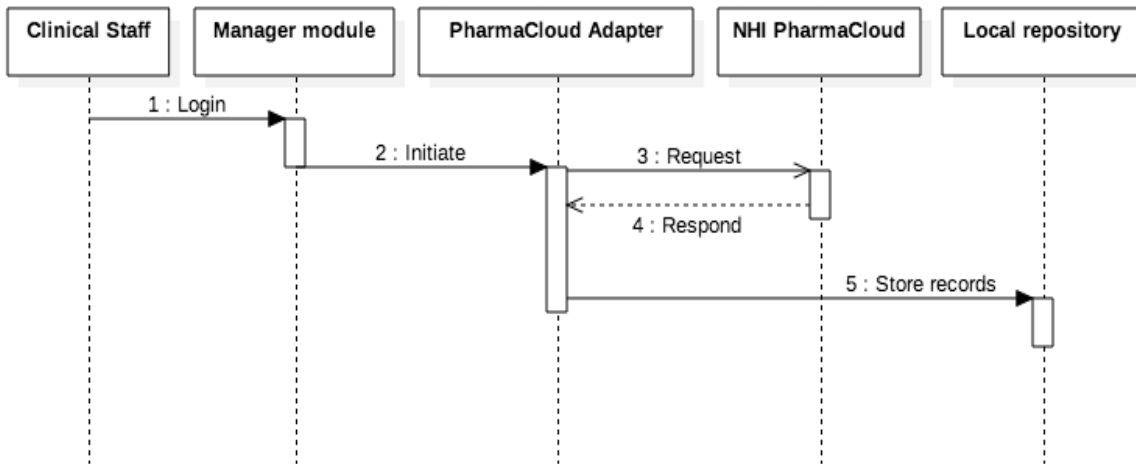


Fig. 3. Information workflow to retrieve the drug history of NHI-PharmaCloud patients.

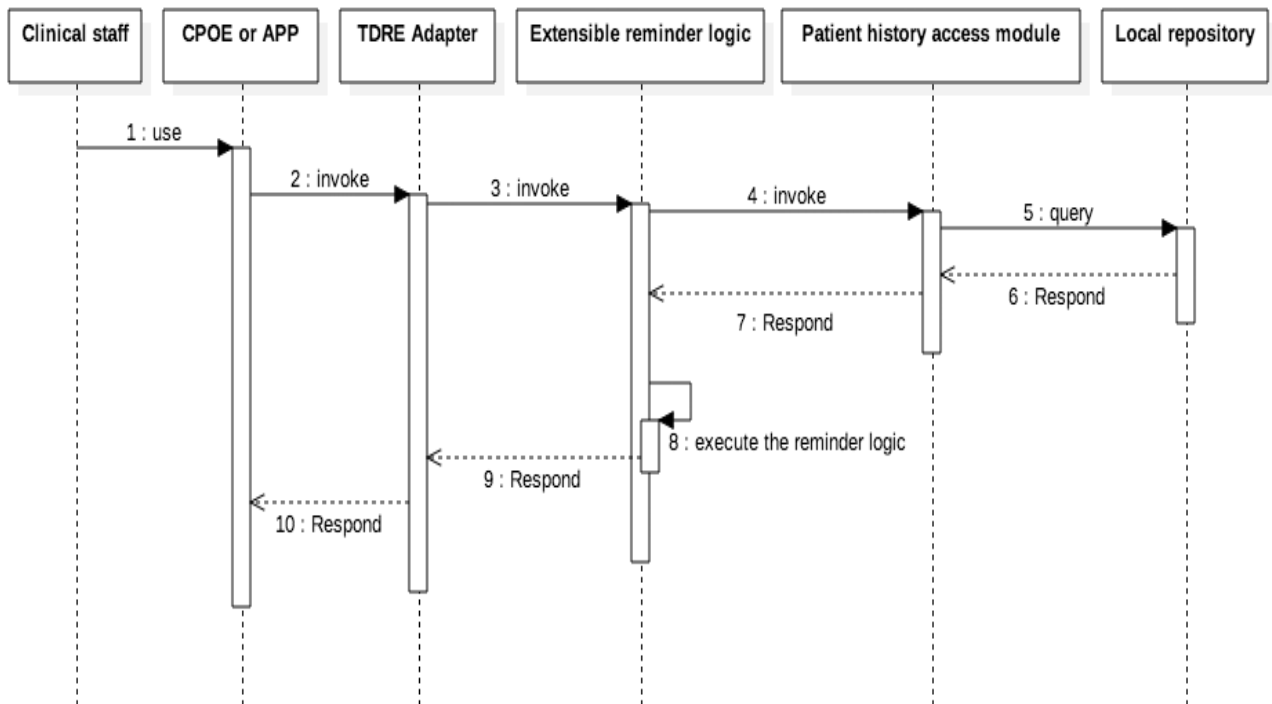


Fig. 4. Information workflow to determine therapeutic duplication.

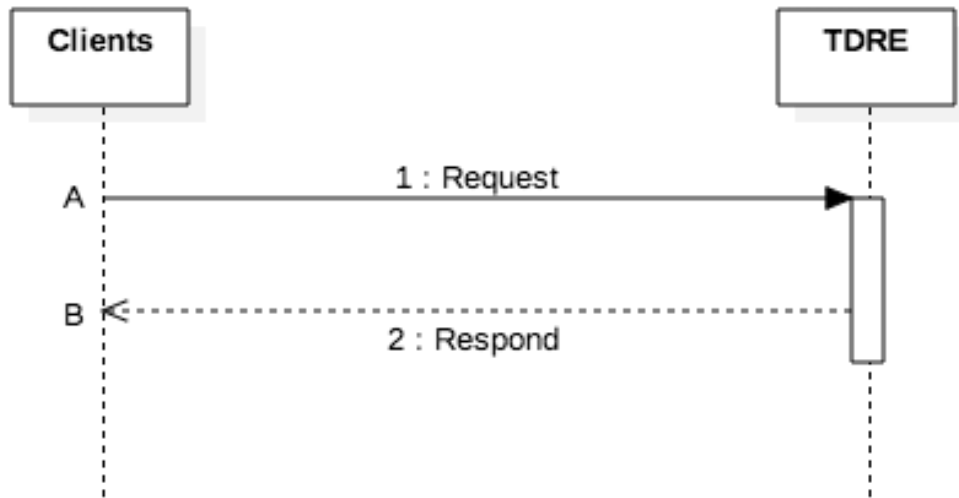


Fig. 5. The sequence diagram of the performance test case.

REFERENCES

- [1] Aronson, J.K., *Medication errors: definitions and classification*. British Journal of Clinical Pharmacology, 2009. **67**(6): p. 599-604.
- [2] Agrawal, A., *Medication errors: prevention using information technology systems*. British journal of clinical pharmacology, 2009. **67**(6): p. 681-686.
- [3] Radley, D.C., et al., *Reduction in medication errors in hospitals due to adoption of computerized provider order entry systems*. Journal of the American Medical Informatics Association, 2013. **20**(3): p. 470-476.
- [4] Huang, S.-K., et al., *NHI-PharmaCloud in Taiwan—A preliminary evaluation using the RE-AIM framework and lessons learned*. International journal of medical informatics, 2015. **84**(10): p. 817-825.
- [5] Magid, S., C. Forrer, and S. Shaha, *Duplicate Orders: An unintended consequence of computerized provider/physician order entry (CPOE) implementation*. Applied clinical informatics, 2012. **3**(4): p. 377-391.
- [6] Bertie Hazell and R. Robson. *Pharmaceutical waste reduction in the NHS*. 2015; Available from: <https://www.england.nhs.uk/wp-content/uploads/2015/06/pharmaceutical-waste-reduction.pdf>.
- [7] Chien, H.-Y., et al., *Study of Medication Waste in Taiwan*. Journal of Experimental & Clinical Medicine, 2013. **5**(2): p. 69-72.
- [8] Edseth, B. *Efficient Medication Delivery & Monitoring*. 2011; Available from: <http://healthcare-executive-insight.advanceweb.com/Features/Articles/Efficient-Medication-Delivery-Monitoring.aspx>.
- [9] Bates, D.W., et al., *The impact of computerized physician order entry on medication error prevention*. Journal of the American Medical Informatics Association, 1999. **6**(4): p. 313-321.
- [10] Amarasingham, R., et al., *Clinical information technologies and inpatient outcomes: a multiple hospital study*. Archives of internal medicine, 2009. **169**(2): p. 108-114.
- [11] Services, U.S.D.o.H.a.H., *Duplicate medication order errors increase after computerized provider order entry is implemented*. RESEARCH ACTIVITIES, 2012. **377**: p. 10-11.
- [12] Wetterneck, T.B., et al., *Factors contributing to an increase in duplicate medication order errors after CPOE implementation*. Journal of the American Medical Informatics Association, 2011. **18**(6): p. 774-782.
- [13] Versel, N. *E-Prescribing May Increase Medication Errors*. 2011; Available from: <http://www.informationweek.com/healthcare/clinical-information-systems/e-prescribing-may-increase-medication-errors/d/d-id/1099529>.
- [14] Executive Yuan, R.O.C.T. *My Health Bank*. 2014; Available from: http://www.ey.gov.tw/pda_en/Dictionary_Content.aspx?n=A240F8389D824425&sms=D8F3EB15472D7847&s=D89F43811161A92D.
- [15] Li, Y.-C.J., et al., *Building a National Electronic Medical Record Exchange System—Experiences in Taiwan*. Computer methods and programs in biomedicine, 2015. **121**(1): p. 14-20.
- [16] Cheng, T.-M., *Reflections on the 20th anniversary of Taiwan's single-payer national health insurance system*. Health Affairs, 2015. **34**(3): p. 502-510.
- [17] Hsieh, P.-J., H.-M. Lai, and Y.-L. Hong, *Explaining Physicians' Acceptance and Resistance to the NHI Pharmacloud: A Theoretical Model and Empirical Test*. 2015.
- [18] Chao, C.-T., et al., *Cumulative cardiovascular polypharmacy is associated with the risk of acute kidney injury in elderly patients*. Medicine, 2015. **94**(31).
- [19] Software, P. *Spring Framework*. 2016; Available from: <https://projects.spring.io/spring-framework/>.
- [20] Hat, R. *Drools*. 2016; Available from: <http://www.drools.org/>.
- [21] PostgreSQL. *PostgreSQL*. 2016; Available from: <http://www.postgresql.org/>.
- [22] Hibernate. *Hibernate*. 2016; Available from: <http://hibernate.org/orm/>.
- [23] Yang, C.-Y. and C.-T. Liu, *Performance assessment and tuning for exchange of clinical documents cross healthcare enterprises*. Computer Standards & Interfaces, 2016. **47**: p. 52-61.
- [24] Nah, F.F.-H., *A study on tolerable waiting time: how long are web users willing to wait?* Behaviour & Information Technology, 2004. **23**(3): p. 153-163.
- [25] Maths, U. *Uncommons Maths*. 2012; Available from: <http://maths.uncommons.org/>.