THE DESIGN AND IMPLEMENTATION OF DATABASE SERVICE FOR REAL-TIME MEDICATION MONITORING SYSTEM

Ventsislav Venkov, Rosen Ivanov

Abstract
This article presents the design and implementation of a Web service for assistance in process of medication prescription. The service is a part of a Real-Time Medication Monitoring (RTMM) system. Some of the most important errors in the process of medication intake occur in the process of medication prescription. This is the reason to focus on research and analysis of this process, and develop a service that helps medical staff to prescribe more precise medication therapies. One of the most important information about medication prescription is the reactions between different drugs. These reactions can be positive or negative, and the main objective of this research is to limit the negative reactions and increase the positive ones. The Web service is implemented using RESTful Java servlets and NoSQL database deployed on the Google's public cloud infrastructure. The experimental results show that the developed service can be a useful assistant of doctors when they prescribing medication therapies.

Key words
Medication prescription assistance, Real time medication monitoring, NoSQL databases

1. Introduction

This article presents the design and implementation of database service for a Real–Time Medication Monitoring (RTMM) system. RTMM is a process where the prescription and medication intake are automatically monitored and eventually controlled. It is important to report errors because accumulation of insignificant errors can cause later to ones that are more serious (Aronson,2009,pp.513-521). Another important problem is inappropriate medications prescription. Research in this field shows that about 20% of the people in Europe intake at least one potentially inappropriate medication, and in the USA and Canada this reaches 40% (Fialová et al.,2005.pp.1348-1358). The RTMM systems should to reduce medication prescription errors and improve precision in the process of medication intake to improve the quality of the treatment, decrease side effects, and optimize costs on patient's treatment.

The functionality of RTMM systems is realized using several Web services and some kind of smart mobile device that support this functionality and/or give access to the services (Checchi et al.,2014,pp.1237-1247). Such devices, for example drug dispensers, sensor-augmented pillboxes (Lee and Dey,2014pp.2259-2268), wearable electronics (Kalantarian et al.,2015,pp.1-6; Kalantarian et al.,2016,pp.43-52) and camera-based modules (Bilodeau and Ammouri,2011,pp.377-389) are specially designed and they increase the system's price. The up-to-date smart phones can take such role and it is cheaper and easier solution to design and develop such systems (Hayakawa et al.,2013,pp.37-52; Park et al.,2016,pp.178-185). This solution is preferred when the system should be accessible to people with disabilities or elderly people.

2. Related work

The fundamental research in the field of RTMM has been done in the last several years (Lee and Dey,2014,pp.2259-2268; Mistry et al.,2015,pp.e177-e193; Hanina et al.,2016). The major part of RTMM systems are focus on patient reminder for medication intake (Tran et al.,2014,pp.536-543) and support some kind of the feedback necessary for self-regulation of medication taking (Lee and Dey,2014,pp.2259-2268). The part of research projects in this field are in developmental phase, and are still to be completed. This determines the topicality as well as the enormous social significance of this problem. It is so significant because it affects the most vulnerable parts of the society in the greatest extent. These are the children and the elderly people. For example, it is estimated that elders represent 6.4% of the world's population, and 60% of them have prescribed medications (van Vliet et al.,2006,pp.79-93).

There are still problems that the currently existing research in this area has not been able to solve until now. *First*, the exact determination whether the patient took the necessary medication remains one of main problems. *Second* problem is the inability one RTMM system to encompass all kinds of medication packages (pills, inhalers, injections, syrups, etc.). *Third* problem is protection of information is critical in RTMM systems, because these systems collect, store, and process sensitive personal data. *Finally*, but not least, part of the RTMM systems allow to realize medication prescription, but they are not able to detect prescription errors.

In the proposed research we are focused on two problems – security and prescription errors. RTMM systems contain personal data that should be protected (Tong et al.,2014,pp.419-429; Trifirò et al.,2014,pp.551–561; Menditto et al.,2016,pp.253–265). At this stage, Directive 95/46 /EC of the European Parliament and of the Council of 24.10.1995 guarantee the protection of personal data in the European Union. This directive will be replaced entirely by 28.05.2018 of regulation 2016/679 of the European Parliament and the Council, which was adopted on 27.04.2016. According this regulation personal data should cover all data related to the patient that reveal information about physical or mental health of the patient in the past, the present or the future. Personal information refers to information associated with illness, injury, risk of disease, medical history and clinical treatment. From the list of medications that a person is taking, adopting or accepting indirectly can be judged for his or her illness. This data after 2018 will be classified as personal information for the countries of the European Union.

Another problem remains medication prescription errors. They can be classified such as knowledge–based errors, rules–based errors, and memory–based errors (Aronson,2009,pp.513-521). Our objective is to develop an expert system that cooperates with the doctors on the process of prescribing medical treatments. Such systems should be able to find *generic* (pharmaceutical drug that is equivalent to a brand-name product in dosage, strength, quality, performance, and intended use) and *biosimilar* (biologic medical product which is almost an identical copy of an original product that is manufactured by a different company) medications and check for *incompatible* and *complementary* drugs. Complementary drugs are a term used for a wide variety of health care practices that may be used along with standard medical treatment. Natural products like herbs, dietary supplements, and probiotics are complementary drugs. Functionality of the proposed service will be developed for reporting when prescribing incompatible drugs. Also, there will be designed a functionality that offers suggestions to the doctor about any complementary medicaments to the treatment, that can be prescribed as well. For example, when prescribing antibiotics the system will propose probiotics and vitamins to be included as complementary treatment, and when prescribing two antibiotic drugs in one treatment the system will notify that this can cause harm to the patient, so the doctor has to approve manually such a treatment after being notified.

The main objective of this article is the design and implementation of a Web service that receives, stores, and retrieves medical and system data in a secure manner and restricts unauthorized access. The information is stored in a non–relational database which schema is specifically designed. The service is Web based and makes queries using typical HTTPS requests. Another key functionality is the medication prescription module. Its role is to reduce the error rate in the drug prescription process and to increase the quality of the prescribed treatment by assisting the doctor in this process. The proposed service extends our prior work at medication monitoring in home environments (Venkov and Ivanov,2016,pp.151-158).

The remainder of this article is organized as follows. In Section 3, we describe base system architecture. Section 4 describes the design of the database. Section 5 discuses implementation of the proposed service and provide the results and analyze them, and finally Section 6 concludes the article and presents some ideas for future work.

3. System architecture

The architecture of our RTMM system is shown in Fig. 1. The system performs the following main actions: 1) Doctors enter new medication intake prescriptions; monitor their regular intake; receive medication intake history of their every patient. 2) Patients have access to the service through an app for a smart phone with an integrated Near Field Communication (NFC) module. Patients gain access through their patient ID NFC cards. They can get reports on medicaments, current and daily medication intakes. 3) System administrator controls whole system functionality.

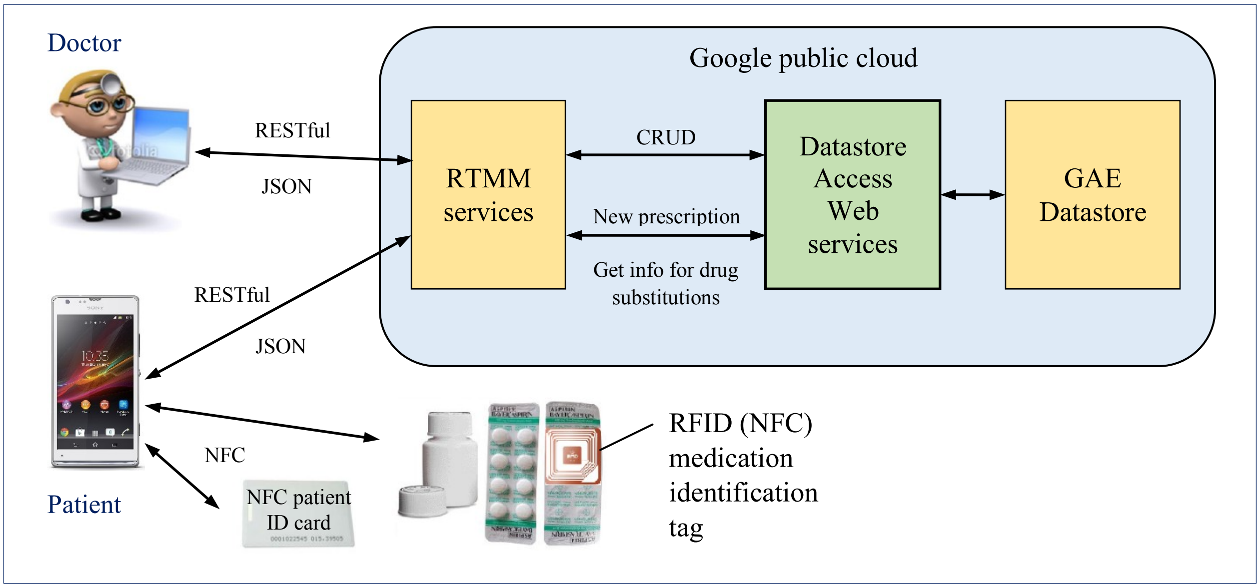


Fig.1. Cloud–based medication monitoring system architecture

Following is a description of the new functionality that complements the functionality of our RTMM system. The access to the database is encapsulated so that querying database tables is done only through requests to the Web services. The encapsulation and isolation of the database is one of the main steps to secure the database. Next, the access to the web services is done solely through HTTPS secure communication channels. The CRUD (Create, Read, Update, and Delete) operations are developed for every database table, along with two RESTful Web services that are created for implementing the developed main functionalities.

This research is focused on Web services that give new functionality. The *first service* performs creation of new medication intake prescription after checking for existence of incompatible drugs that are already prescribed. *Second* service performs a check whether complementary medications to the main treatment exist, but they are not prescribed. If there is an attempt to prescribe incompatible drug a notification is sent to the doctor to manually confirm or cancel the prescription of the medication. If complementary drugs exist for the prescribed one, a notification is returned as a reminder about them. Every response contains structured data that the requesting resource can process timely. The name of the medication to be checked is passed as a parameter to the Web resource. The response is JSON object that contains the information about the desired drug or an error message that explains the kind of error or the lack of such a medication. The format of request/responses to/from Web services are as following:

1. Query to create a new medication intake schedule

The only way to store new drug intake schedule is with RESTful POST requests. The data needed to save the new schedule is passed as a JSON object and contains the following parameters:

- patientId – this is the unique identification code of the patient for whom the drug is intended. This code is obtained from the NFC patient ID card.
- startDate – date from which the treatment must begin.
- endDate – date to which the treatment must end.
- medicationIntake – a list with medications for intake. This object contains following information: medicationId, dosage, medUniqueId, timeOfDay, and otherConsiderations (“before meals”, “with lots of water”, “before sleep”, etc.).

If an NFC tag is associated to the drug's container (bottle, box, etc.) there will be saved the unique ID of the tag as identification code of the medicine - MedUniqueId. The object MedicationIntake is a list of medication that specifies at what times of the day the patient should ingest his medication and other considerations about the intake.

The response is a JSON object that contains the following parameters:

- status – the main code of the execution result. Only two options are available - OK or ERR.
- errorCode – this code classifies the error, for example COMPLEMENTARY_EXIST_FOR_DRUG, CONFLICT_DRUG, INVALID_DATA.
- message – the message is a string parameter with custom text that shows the execution result in plain text.
- result – an object that contains the result in JSON format. The result is passed if a conflicting drug exists in past medication schedules. When complementary drugs exist for the prescribed schedule, this parameter contains a list of the available complementary drugs.

2. Query for drug substitutions. The parameters to the service are passed as JSON object using POST request:

- medicationName – the medication for which is requested to find substitution drugs.
- medicationType – clarifies the target drugs types. To this parameter can be assigned one of following values: GENERIC, BIOSIMILAR, or ORIGINAL.

The response is in JSON format:

- status – the main code of the execution result (OK or ERR).
- errorCode – this code classify the error, for example COMPLEMENTARY_EXIST_FOR_DRUG, ERR_CONFLICT_DRUG, INVALID_DATA.
- message – the message is a string parameter with custom text that describes the execution result in plain text.
- result – an object contains the result of the request. It may be a single drug, a list of drugs or null if no drugs fall into the requested category.

4. Database design

The main decision that needs to be made is whether the database should be relational or non–relational (NoSQL). This refers to the structure of the data itself, and, of course, the volumes of data that will be stored. Relational databases are usually used for strongly related data which needs to be stored in a way that these relations persist in the database and minimum amounts of extra memory is used. Relational databases inevitably drop productivity with growing of data volumes. Non–relational databases are commonly used when database schema should remain simple and without much complexity. This, along with other characteristics of the NoSQL databases, increases productivity and the amounts of information do not have significant influence on the performance of the database. The necessity of designing and developing cloud–based RTMM system arises by the fact that the problems in the process of prescribing and following medication intakes have increasingly growing importance and influence on economy and society (Peron et. al.,2011,pp1-10; Seidling et al.,2013,pp.25-36). Furthermore, cloud–based solutions, because of their advantages, are gaining increasing popularity (Griebel et al.,2015). Nevertheless, the ability to work on state level determines the large volume of data about doctors, patients and patients' treatments information. Recording, extracting, and processing large volumes of data are extremely facilitated when operating in the cloud. The cloud environment that Google provides – GAE, is chosen to be the infrastructure for our research. The database chosen is Google's NoSQL database named Datastore. The database is designed programmatically with the native API that Google provides – Datastore API. Choosing cloud infrastructure does not limit the choice of database architecture, but the fact that the data volumes theoretically can belong to big data makes NoSQL a logical choice for database architecture. Its advantages on working with big data volumes compared to the SQL architectures are undeniable (Bhogal and Choksi,2015,pp.393-398).

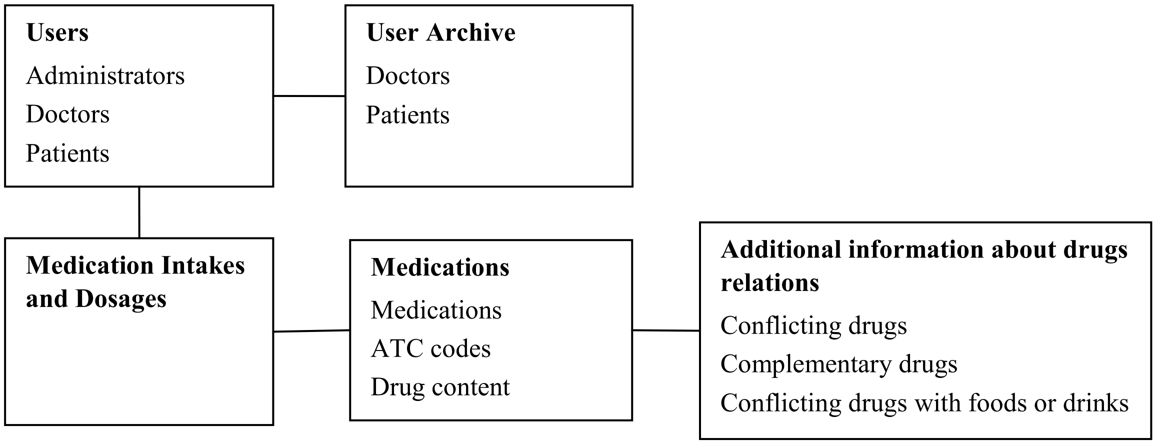


Fig 2. Database main modules

The main problems when working with data in the cloud are scalability, provision of adequate data security and privacy in the database and data confidentiality (Bertino and Sandhu,2005,pp.2-19); Nayak and Mishra,2015,pp.1749-1752). The tables are conditionally divided into five main modules that define the functionalities of the cloud–based RTMM system (Fig. 2).

1) Users – the system supports three types of users – Doctors, Patients and Administrators. Doctors and Patients have separate tables in the database. The Administrator profiles are hardcoded in the system.
2) User Archive – tables for recording changes in the users' profiles of the Doctors and the Patients are developed. The changes of both doctors and patients are performed by the doctor's profile and are confirmed or denied by an administrator.
3) Medication Intakes and Dosages – in this module are recorded the schedules of the medication treatments for patients. The full history of every patient's treatments is recorded into this module. User app uses this data to notify users when the time comes for medication intake or shows the schedule for the whole treatment.
4) Medications – contains information about prescribed medication. The Anatomical Therapeutic Chemical (ATC) Classification System is integrated in the project. It is used for the classification of active ingredients of drugs according to the organ or system on which they act and their therapeutic, pharmacological and chemical properties. In addition, there are additional relations between generic and original drugs. This data is of key importance for the implementation of the functionalities of module 5.
5) Additional information about relations between drugs – in this module clinically significant medication relation is reflected. These relations can be divided in 2 sections – desired and adverse (Ament et al.,2000,pp.1745-1754). Relations can be between two or more drugs as well as between drugs and foods or drinks. Information about couples of medications that form reactions between them is recorded. Doctors are to be notified when they attempt to prescribe treatments with two or more drugs that have negative reactions between them as well as when prescribing treatments with the possibility to prescribe complementary drugs.

After the notification, the doctor will still be able to prescribe the desired drug if he or she is confident in the result of the prescribed therapy, but the digital copy of this act will be a psychological barrier to him if harm to the patient is done as well as handy assistant. Because Web services in GAE apps share Datastore the database is identified as stand–alone app. The main purpose, besides programming code isolation is also data isolation from the rest of the system. Another part of the security system is the protection of information exchanged over the communication channels. Communication between the database service and the other parts of the system is done through encrypted HTTPS communication channel. Authentication of users is verified using Hash

Message Authentication code (HMAC) mechanism. For cryptographic hash function, we use SHA-1 (HMAC-SHA1).

As mentioned one of the objectives of our project is securing the database and adding functionalities to assist doctors on prescribing medication treatments in non–hospital environment. An attempt is made to optimize processes where errors can eventually occur (Aronson,2009,pp.513-521). The main functionalities of developed services are as following:

1) An indication that two or more drugs are to be prescribed, that have conflicting medication content and require explicit confirmation from the doctor. If an attempt to prescribe such a combination of medications is made, the system denies to record this prescription and shows notification to the doctor about the eventual error and the doctor has to manually approve or erase the conflicting combination.

2) An indication that prescribed drug has complementary medications. These drugs support the positive effects of the main treatment or suppress the negative ones. The system does not submit the request, but show the doctor notification about the existence of complementary drugs that can improve the quality of the treatment. After this notification, the doctor needs to choose which medicines to select in addition to the previous prescription and submit it again.

3) Enquiry for possible medication substitutions. This functionality is handy for getting quick references about medicine substitutions in cases where different medicine with the same active ingredients needs to be chosen because of price, quality, etc.

4) Software architecture is designed so that the database is isolated and secured from external access.

For the implementation of the described functionalities there are used the following new tables in the database (see Fig. 3):

- **ATC codes** – the ATC codes are drugs classification method developed by the World Health Organization.
- **Medication** – the information about medical drugs that are authorized to be used by the European Medicines Agency (EMA).
- **MedContents –** contains data about medications active ingredients.
- **MedContentConflict** – contains information about proven conflicts between active ingredients of two or more drugs or ingredients with foods or drinks.
- **MedContentComplementary** – for each drug there can be context for unlimited number of complementary drugs.

Tables “Medication” and “ATC codes” are populated programmatically as data is exported from the EMA web site. Data is recorded for 955 drugs and 714 medication contents. From the exported data is retrieved information whether a drug is generic or biosimilar drug, but there is no information if the medication is an original one as well as which is the original drug of a generic or a biosimilar one. An algorithm is implemented that checks if a drug is generic or biosimilar and if another drug exists with the same content but not marked as generic or biosimilar. If such a drug exists – it is chosen to be the original medication. When creating the record the IDs of the original and generic drugs are assigned to the corresponding generic, biosimilar or original drug.

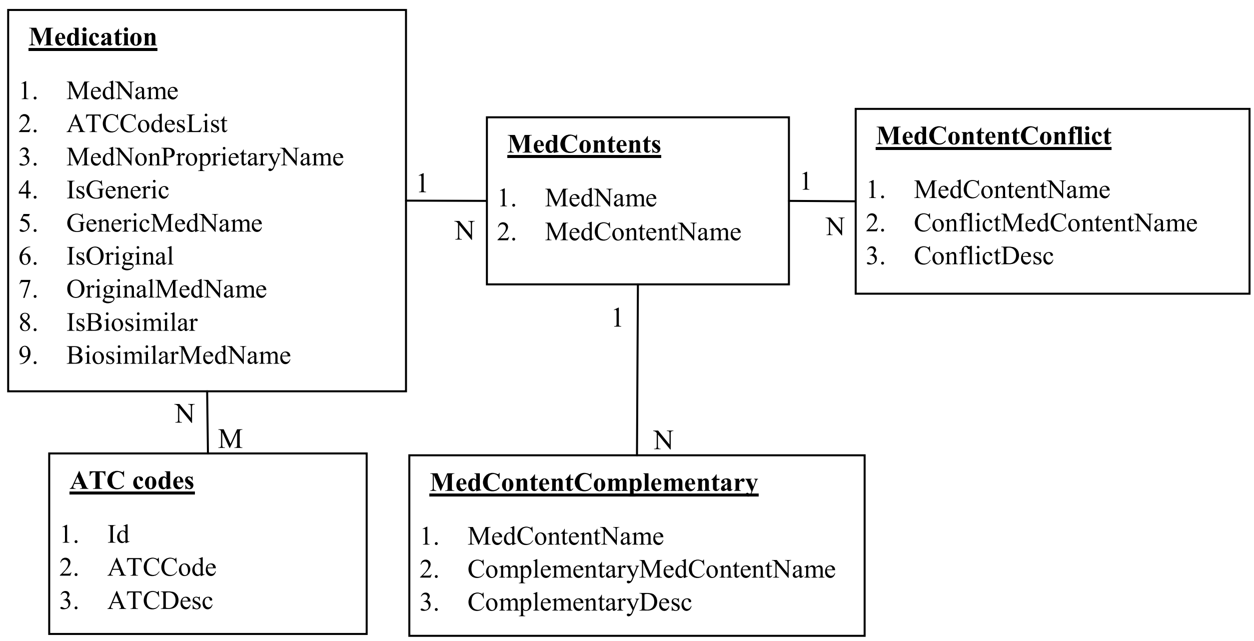


Fig 3. Database tables and relations

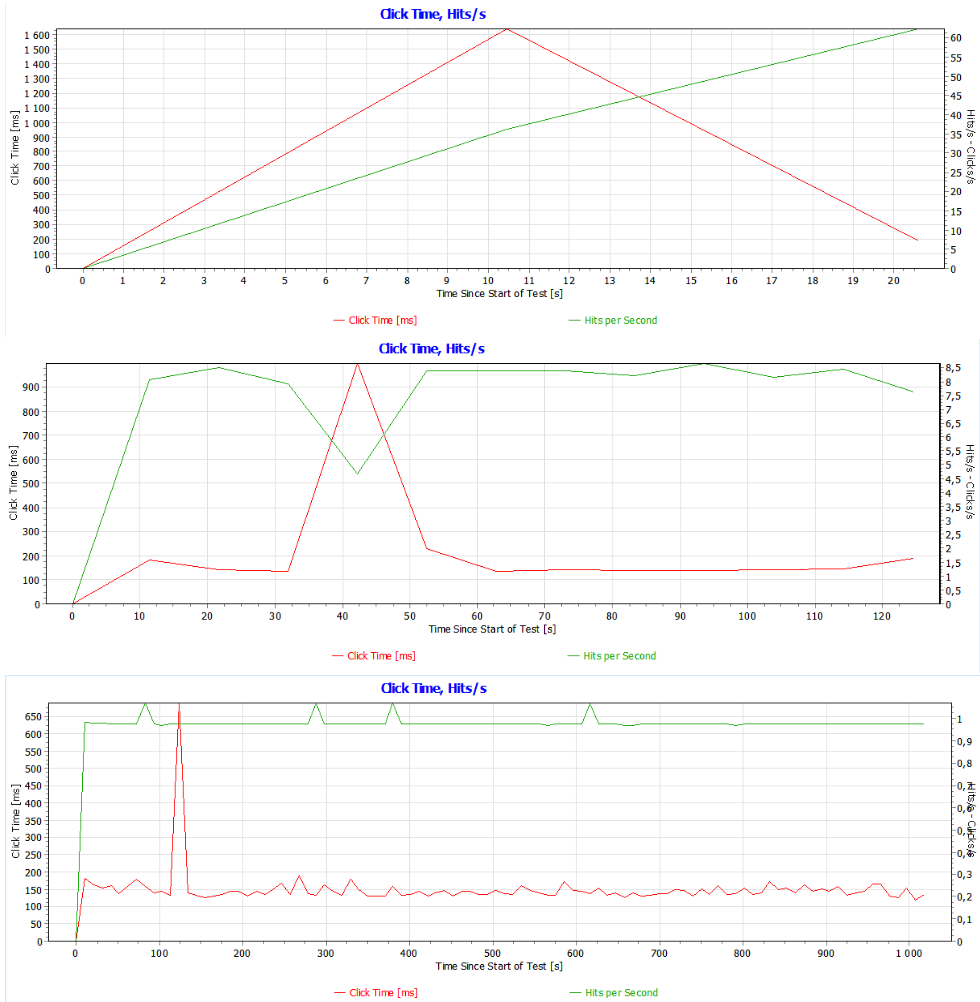
5. Experimental results

In this section, we analyze computational efficiency of the proposed Web services. We use Webserver Stress Tool v.8.x Enterprise Edition from Peassler AG to test the Web services load performance. The efficiency of the system functionality is initially tested in local environment. We implemented Web services functionality using laptop with Intel Core I5-2400 (3.10GHz) CPU and 8GB DRAM, OS Windows 7 Ultimate x64, and GAE plugin for IDE Eclipse Neon.3 Release (4.6.3). After the successful local implementation and testing, the application is deployed and tested in GAE's cloud infrastructure. Experiments are conducted using two RESTful Web services that are created to implement the proposed system functionality.

We test three types of simulated user activity: 1) 100 simultaneous users, 10 requests per user, and 1 second between clicks, 2) 10 simultaneous users, 100 requests per user, and 1 second between clicks, and 3) 10 simultaneous users, 100 requests per user, and 10 second between clicks. The "click" is a simulated mouse click of a user sending a URL request to the server and immediately requesting any necessary redirects. All tested web resources return correct information. Request execution times vary from 0.14 to 1.13 seconds. In all tests, reported number of errors is zero, which proves that the developed RESTful services execute correctly. For Test 1, the average clicks time (the time a user had to wait until his click was finished) of all URL requests is 721ms. Minimum click time is 249ms and maximum click time is 1131ms. As shown in Fig. 4a, the click time for time interval from 1 second to 10.5 seconds increase up to 1640ms and decrease to 190ms after 21 seconds since start of test.

Table 1. Web server load performance - stress test results

Test No.	Errors	Min Click time [ms]	Max Click time [ms]	Avg. Click Time [ms]
1	0	249	1131	721
2	0	191	232	198
3	0	143	170	150



a)

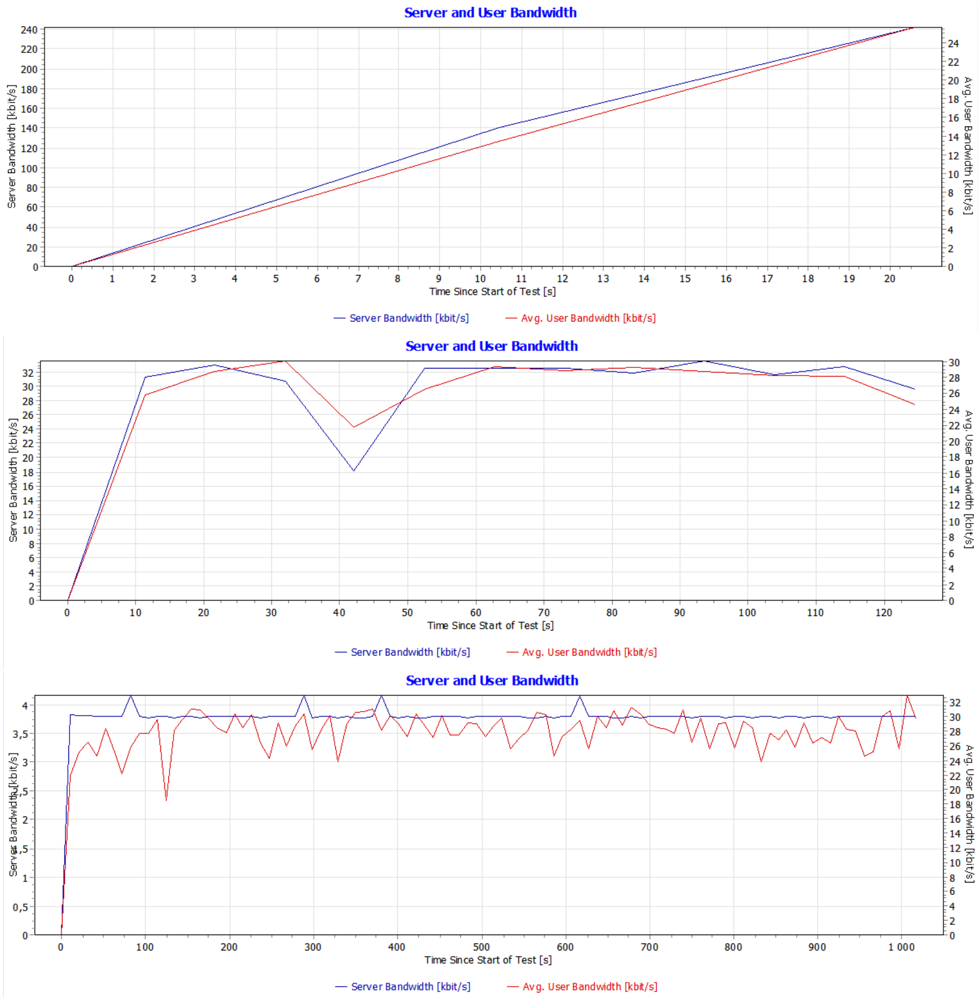
b)

c)

Fig. 4. Protocol times for all requests: a) Test 1, b) Test 2, and c) Test 3

Server bandwidth is up to 240 kbit/s and client bandwidth is up to 24 kbit/s. (see Fig. 5a). Local CPU load does not change significantly during the requests and it is around 1.8%. The available system memory varies with 140 to 240MB. Tests 2 and 3 simulate URL queries from 10 users, each sending 100 requests at an interval of 1 second (Test 2) or 100 requests at an interval of 10 seconds (Test 3). As expected, in this case, the peak server load is 7.5 times less for Test 2 and 60 times less for Test 3. In Test 2, the average click time is 198 ms, and in Test 3 - 150ms.

Fig. 6 shows an example for a query for prescription of a new drug. Two variants of the answer are shown: 1) On successful prescription of the drug, and 2) The presence of a conflict between the new medication and the one that the patient is already taking. In last treatment, the doctor was prescribed antibiotic Augmentin, and now tries to prescribe Ospamox. The Ospamox ATC code is J01CA04 (amoxicillin) and Augmentin ATC code is J01CR02 (amoxicillin and enzyme inhibitor). Both ATC codes refer to beta-lactam antibacterial antibiotic penicillin (J01C). Augmentin contains two main ingredients - amoxicillin and clavulanic acid, but Ospamox is also based on amoxicillin (as trihydrate). In this case, the system returns error status and error message contents is “The drug Augmentin is recorded as conflict medicament content with Ospamox.” To find conflicts between the drugs we analyze the current and previous prescribed medication therapies for a period of one to three months ago. This period is different for each particular drug.



a) b) c)

Fig. 5. Server and user bandwidth: a) Test 1, b) Test 2, and c) Test 3

Request:

```
{
    "patientId": "8804182205",
    "startDate": "22.07.2017",
    "endDate": "05.08.2017",
    "medicationIntake":
    {
        "medicationId": "Ospamox",
        "dosage": "1000 mg",
        "medUniqueId": "04ccb0e23b2b80",
        "otherConsiderations": "after meals",
        "timeOfDay":  ["08:00","20.00"]
    }
}
```

Response 1:

```
{
    "result": {},
    "status":  "OK"
}
```

Response 2:

```
{
    "result": {
        "generic": false,
        "biosimilar": false,
        "original": true,
        "atcCode": "J01CR02",
        "genericMedName": "",
        "originalMedName": "",
        "name": "Augmentin"
    },
    "errorCode": "CONFLICT_DRUG",
    "errorMessage": "The drug Augmentin is
    recorded as conflict medicament content
    with Ospamox",
    "status": "ERR"
}
```


Fig. 6. Query for prescription of the new drug

6. Conclusions and future work

In this article, we presented how we designed and implemented a prototype service for medication prescription, which is part of a RTMM system. We developed two RESTful services. *Fist* service allows doctors to prescribe new medication. First attempts to create new intake with incompatible drugs are denied and the doctor that tries to prescribe the erroneous schedule is notified about the denial. If the doctor insists on prescribing such medication, he has to manually confirm that he received the notification and send the request again. *Second* service returns information about possible medication substitutions. This service gives information about a medication‘s original or generic alternatives. Another functionality integrated in this service is notifying the doctor if complementary drugs of the main medication exist and is “best practice” to prescribe such drugs, as well. Information about drugs that are authorized for use in the European Union and ATC medication classification codes are imported from the European Medicines Agency. The designed and implemented functionalities are tested in Google’s public cloud environment GAE. All aspects of the research execute as expected. The results show that the developed functionalities execute correctly in all simulated test scenarios. Request execution times vary from 0.14 to 1.13 seconds. The results of the experiments show that the developed system has potential to become a fully developed medical staff assistant.

The developed functionality for notifying doctors on prescribing drugs with known incompatibilities helps doctors to reduce errors in new therapeutic treatment schedule and adding new drugs to already existing schedule. The functionality for notifying doctors on prescribing drugs with known complementary drugs to the main medication theoretically improves the quality of the prescribed therapy, while at the same time reduces the side effects or undesired effects of the prescribed therapy.

Due to the fact, that GAE Datastore is relatively slow it is foreseen to use another NoSQL database, for example MongoDB deployed on Google Compute Engine. There will be experiments involving doctors to assess the usability of the service.

Acknowledgements
This work has been partially funded by Bulgarian Ministry of Education and Science (Project DH-07/10).

References

Ament, P. W., Bertolino, J. G., Liszewski, J. L. (2000). Clinically significant drug interactions. *American family physician*, 61(6), 1745-1754.

Aronson, J. K. (2009). Medication errors: what they are, how they happen, and how to avoid them. *QJM: An International Journal of Medicine*, 102(8), 513-521.

Bertino, E., & Sandhu, R. (2005). Database security-concepts, approaches, and challenges. *IEEE Transactions on Dependable and secure computing*, 2(1), 2-19.

Bhogal, J., & Choksi, I. (2015). Handling big data using NoSQL. *Proc. of 29th IEEE International Conference on Advanced Information Networking and Applications Workshops*, pp. 393-398.

Bilodeau, G. A., & Ammouri, S. (2011). Monitoring of medication intake using a camera system. *Journal of medical systems*, 35(3), pp. 377-389.

Checchi, K. D., Huybrechts, K. F., Avorn, J., & Kesselheim, A. S. (2014). Electronic medication packaging devices and medication adherence: a systematic review. *Jama*, 312(12), 1237-1247.

Fialová, D. et al., (2005). Potentially Inappropriate Medication Use Among Elderly Home Care Patients in Europe. *JAMA*, 293(11), pp.1348-1358.

Griebel, L. et al., 2015. A scoping review of cloud computing in healthcare. *BMC medical informatics and decision making*, 15(1), p.17.

Hanina, A., Kessler, G., & Guan, L. (2016). Method and apparatus for monitoring medication adherence, U.S. Patent No. 9,486,720. Washington, DC: U.S. Patent and Trademark Office.

Hayakawa, M., Uchimura, Y., Omae, K., Waki, K., Fujita, H., & Ohe, K. (2013). A smartphone-based medication self-management system with realtime medication monitoring. *Applied clinical informatics*, 4(1), 37-52.

Kalantarian, H., Alshurafa, N., Nemati, E., Le, T., & Sarrafzadeh, M. (2015). A smartwatch-based medication adherence system. *Proc. of IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks* (BSN), 2015, pp. 1-6.

Kalantarian, H., Motamed, B., Alshurafa, N., & Sarrafzadeh, M. (2016). A wearable sensor system for medication adherence prediction. *Artificial intelligence in medicine*, 69, 43-52.

Lee, M. L., & Dey, A. K. (2014). Real-time feedback for improving medication taking. *In Proceedings of the 32nd annual ACM conference on Human factors in computing systems*, pp. 2259-2268.

Menditto, E. et al., 2016. Scaling up health knowledge at European level requires sharing integrated data: An approach for collection of database specification. *ClinicoEconomics and Outcomes Research*, 8, pp.253–265.

Mistry, N., Keepanasseril, A., Wilczynski, N. L., Nieuwlaat, R., Ravall, M., Haynes, R. B., & Patient Adherence Review Team. (2015). Technology-mediated interventions for enhancing medication adherence. *Journal of the American Medical Informatics Association*, 22(e1), e177-e193.

Nayak, A., Mishra, S. K. (2015). Cloud Database Security: A Survey. *International Journal of Advanced Research in Computer Science and Software Engineering*, 5(5), 1749-1752.

Park, Y. T., Lee, Y. T., & Jo, E. C. (2016). Constructing a Real-Time Prescription Drug Monitoring System. *Healthcare informatics research*, 22(3), 178-185.

Peron, E. P., Marcum, Z. A., Boyce, R., Hanlon, J. T., & Handler, S. M. (2011). Year in review: medication mishaps in the elderly. *The American journal of geriatric pharmacotherapy*, 9(1), 1-10.

Seidling, H. M., Lampert, A., Lohmann, K., Schiele, J. T., Send, A. J., Witticke, D., & Haefeli, W. E. (2013). Safeguarding the process of drug administration with an emphasis on electronic support tools. *British journal of clinical pharmacology*, 76(S1), 25-36.

Tong, Y., Sun, J., Chow, S. S., & Li, P. (2014). Cloud-assisted mobile-access of health data with privacy and auditability. *IEEE Journal of biomedical and health Informatics*, 18(2), 419-429.

Tran, N., Coffman, J. M., Sumino, K., & Cabana, M. D. (2014). Patient reminder systems and asthma medication adherence: a systematic review. *Journal of Asthma*, 51(5), 536-543.

Trifirò, G. et al., (2014). Combining multiple healthcare databases for postmarketing drug and vaccine safety surveillance: Why and how? *Journal of Internal Medicine*, 275(6), pp.551–561.

van Vliet, M. J., Schuurmans, M. J., Grypdonck, M. H., & Duijnstee, M. S. (2006). Improper intake of medication by elders—insights on contributing factors: a review of the literature. *Research and theory for nursing practice*, 20(1), 79-93.

Venkov, V., & Ivanov, R. (2016). Cloud-Based System for Real Time Medication Monitoring. *In Proceedings of the 17th International Conference on Computer Systems and Technologies*, pp. 151-158.

About the authors

Ventsislav Venkov received the Computer Systems and Technologies degree from the Technical University of Gabrovo, Bulgaria. He is currently a PhD student at the same university. The title of his dissertation is "Context-aware cloud-based service for real time medication monitoring", e-mail: venci.venkov@gmail.com

Rosen Ivanov received the Electronics Engineering degree from the University of Gabrovo, Bulgaria. He received the Ph.D. degree from the University of Sofia, Bulgaria. He is currently Associate Professor of Computer Systems and Technologies at University of Gabrovo. He is the author of five books and over than 65 scientific papers. His research interests include mobile communications, assistive technologies and digital signal processing, e-mail: rs-soft@ieee.org.