Real-Time Monitoring Tool for Inpatient Antibiotic Consumption

by

Veli Oğuzalp Bakır

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computational Sciences and Engineering

KOÇ UNIVERSITY

January, 2020

Real-Time Monitoring Tool for Inpatient Antibiotic Consumption

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Veli Oğuzalp Bakır

and have found that it is complete and satisfactory in all respects,

and that any and all revisions required by the final

examining committee have been made.

Committee Members:

Assoc. Prof. Mehmet Gönen (Advisor)

Prof. Önder Ergönül

Prof. Serap Şimşek Yavuz

Date:

To my dearest family

ABSTRACT

Inappropriate antibiotic use poses significant health threats on both local and global scale. Therefore, for the aim of improving antibiotic usage, antibiotic management programs are being implemented by health institutions worldwide. The main objectives of these programs are to ensure that prescribing complies with certain criteria, to monitor the trends of antibiotic consumption and to make corrective interventions when necessary. In order to achieve these objectives, several types of medical information must be evaluated simultaneously. However, software used for the evaluation of this information in healthcare facilities is often insufficient to meet the needs such as processing large data sets and managing relationships between various data groups.

STEWARD, a web application to be utilized in antibiotic consumption monitoring, is introduced in this thesis. The application has an easy-to-use interface and allows users to perform in-depth investigations without the need for technical knowledge. Moreover, it helps users evaluate antibiotic use accuracy by combining information stored in several data sets. Since the application is developed using open source technologies, it is free to use, and at the same time, advanced in terms of technical capabilities. When synchronized with central data management systems of health centers, it allows for real-time antibiotic consumption monitoring, which is particularly valuable at patient level antibiotic stewardship.

The proposed application is field tested using the data of İstanbul American Hospital. This way, the capabilities of the application have been affirmed and the tool is shown to be able to operate integrally with existing healthcare systems. And finally, antibiotic consumption trends in İstanbul American Hospital are examined and observations are presented.

ÖZETÇE

Uygunsuz antibiyotik kullanımı yerel ve küresel ölçekte önemli sağlık tehditleri oluşturmaktadır. Buna bağlı olarak, antibiyotik kullanımını iyileştirmek amacıyla dünya genelinde sağlık kuruluşları tarafından antibiyotik yönetim programları yürütülmektedir. Bu programların temel işlevi reçetelemenin belirli kriterlere uygun yapılmasını sağlamak ve antibiyotik tüketim eğilimlerini takip ederek gerektiğinde iyileştirici müdahaleler yapmaktır. Bu iki işlevin yerine getirilebilmesi için de birçok tıbbi verinin bir arada değerlendirilmesi gerekir. Ancak sağlık kuruluşlarında bu verilerin değerlendirilmesinde kullanılan yazılımlar genellikle farklı veri kümeleri arasındaki ilişkileri yönetme ve büyük veri kümelerini işleme gibi ihtiyaçları karşılamada yetersiz kalmaktadır.

Bu tezde antibiyotik kullanımının takip ve analizinde kullanılmak üzere geliştirilen bir web uygulaması olan STEWARD tanıtılmaktadır. Uygulama, kullanımı kolay bir arayüze sahiptir ve kullanıcıların teknik bilgiye ihtiyaç duymaksızın detaylı incelemeler yapmasına olanak sağlar. Bununla birlikte, birçok ilişkili veri kümesinin birlikte değerlendirilerek antibiyotik kullanımının uygunluğuna ilişkin çıkarımlar yapılmasına yardımcı olur. Uygulama tamamen açık kaynak kodlu yazılımlar kullanılarak geliştirildiği için kullanımı ücretsizdir ve aynı zamanda teknik özellikler yönünden gelişmiştir. Sağlık kuruluşlarının veri tabanları ile eşlendiğinde gerçek zamanlı antibiyotik kullanım takibine olanak tanır.

Geliştirilen bu uygulama İstanbul Amerikan Hastanesi'nin verileri kullanılarak sınanmıştır. Böylece uygulamanın kabiliyetleri somutlaştırılmış ve mevcut sistemlerle bütünleşik şekilde çalışabileceği gösterilmiştir. Ayrıca İstanbul Amerikan Hastanesi'ndeki antibiyotik tüketim eğilimleri incelenmiş ve gözlemler sunulmuştur.

ACKNOWLEDGMENTS

First, I would like to express my sincere gratitude to my thesis advisor Assoc. Prof. Mehmet Gönen for his remarkable guidance throughout my research. I highly appreciate his support, energy and immense knowledge which have been a significant source of motivation in all my graduate studies. I have always felt very fortunate to find the chance to work with him.

I am very grateful to Prof. Önder Ergönül for his generous support and supervision in several projects throughout my master's years, and to Prof. Serap Şimşek Yavuz for taking part in my thesis jury.

I would like to thank my friends Onur Dereli, Banu Ulusoy and Çiğdem Ak, and all Machine Intelligence and Data Analysis in Science Laboratory (MIDASLAB) members for their support and friendship.

I would also like to thank Bahar Madran and Özgür Özmen from American Hospital for their contributions in this research.

Finally, I would like to thank my beloved parents and my family for their endless support and encouragement through all these years of education far away from them. Writing these very sentences would not have been possible without their support.

This thesis has been supported by the master's scholarship (2210-A) from the Scientific and Technological Research Council of Turkey (TÜBİTAK).

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

NOMENCLATURE

ASP	Antibiotic Stewardship Program
ABR	Antibiotic Resistance
ACM	Antibiotic Consumption Monitoring
EHR	Electronic Health Records
PCT	Procalcitonin
CRP	C Reactive Protein
DMS	Data Management System
RDBMS	Relational Database Management System
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JS	JavaScript
JSON	JavaScript Object Notation
JDBC	Java Database Connectivity

Chapter 1

INTRODUCTION

Antibiotic resistance is recognized as one of the biggest threats to global health, food security and development today [WHO, 2018]. It can affect anyone regardless of age and country, and as it spreads, antibiotics become less effective to treat infections, which leads to longer hospital stays, higher medical costs and increased mortality. Although antibiotic resistance is a natural phenomenon, misuse and overuse of antibiotics accelerate the process. For these reasons Antibiotic Stewardship Programs (ASPs) are being conducted at both national and hospital-level. The main purpose of these programs is to ensure that prescription of antibiotics is performed according to certain criteria.

To measure the impact of these programs and evaluate antibiotic consumption trends, ASP personnel investigates antibiotic consumption data. Typically, the analyses are conducted intermittently and retrospectively using general-purpose data analysis software. As the amount of the data to be examined grows and the required analysis detail deepens, this software becomes insufficient to process the data. Therefore, even when the analysis is possible, it would still be problematic because of the time it will take. These problems can be overcome by using software specially designed for utilization in ASP investigations.

Designing such software requires good level of programming knowledge. Typically, ASP professionals cannot design such software due to lack of technical ability. Although there are several commercial antibiotic stewardship products, health centers, especially in Turkey, cannot afford their high licensing costs. In this thesis, we present the software we specially designed for monitoring antibiotic consumption in health centers. This software allows users to analyze antibiotic consumption in detail without the need for technical knowledge, thanks to its simple user interface. With the help of graphical representations and interactivity, it provides ease of use in antibiotic consumption monitoring. Also, utilizing open-source technologies, it is free to use, scalable, and advanced in terms of technical features.

Chapter 2 discusses the motivations behind ASPs and it broadly explains the mechanisms of antibiotics and antibiotic resistance. It also examines ASPs and antibiotic consumption monitoring systems from several perspectives. In Chapter 3, special requirements for antibiotic consumption monitoring task, technological structure of the proposed tool and its technical/practical features are explained. Then, in Chapter 4, the proposed tool is field tested with the data obtained from American Hospital of İstanbul. Real-life use cases of the proposed monitoring tool are presented and antibiotic consumption trends in American Hospital are investigated. Finally, in Chapter 5, the conclusions drawn from this thesis study and future research directions are listed.

Chapter 2

ANTIBIOTIC STEWARDSHIP PROGRAMS AND MONITORING TOOLS

By strict definition, the term “antibiotic” refers to an agent or substance that is produced by or derived from a microorganism that kills or inhibits the growth of another living microorganism while the term “antimicrobial” refers to an agent or substance derived from any source (microorganisms, plants, animals, synthetic or semi-synthetic) that acts against any type of microorganism, such as bacteria (antibacterial), mycobacteria (anti-mycobacterial), fungi (antifungal), parasite (anti-parasitic) and viruses (antiviral) [WHO, 2019]. However, note that, in this thesis the term “antibiotic” refers to an antimicrobial agent with the ability to kill or inhibit bacterial growth.

Antibiotic stewardship has been defined as “the optimal selection, dosage, and duration of antimicrobial treatment that results in the best clinical outcome for the treatment or prevention of infection, with minimal toxicity to the patient and minimal impact on subsequent resistance” [Doron and Davidson, 2011].

This chapter discusses the motivations behind antibiotic stewardship programs (ASPs) and antibiotic consumption monitoring (ACM). It gives information about the mechanisms of antibiotics and antibiotic resistance. Also, current monitoring practices, data requirements, different approaches in ASP interventions, different scopes in ASP assessment and different types of software that can be used for ACM are presented.

2.1 Motivation

Since the discovery of first antibiotic in the first half of 1900s, antibiotics have saved millions of lives. Starting with penicillin, many novel classes and mechanisms of antibiotics have been developed for the prevention and treatment of infections caused by various bacteria and fungi. On the other hand, starting with the very first use antibiotics, bacteria and fungi started to develop immunity to these drugs, as well. Parallel to the overuse and misuse of antibiotics over the course of decades, more and more microorganisms have become immune to antibiotics. This phenomenon is called antibiotic resistance (ABR) and has been identified as one of the biggest global public health threats of the 21st century, causing at least 700,000 annual mortality cases around the world [Hassouna and Deshpande, 2018]. It is estimated that ABR will be entailed to the death of 10 million people in 2050 if the current trend remains [O'Neill, 2016]. As "antimicrobials are ... the only class of drug therapy that affects not only the patient to whom it is prescribed but other current and future patients as well" [Schiff et al., 2001], health institutions have the responsibility to ensure correct and efficient antibiotic use. Thus, aiming to prevent inappropriate use of antibiotics and to combat resistance, ASPs have been promoted globally.

In addition to health-related problems, antibiotic resistance is forecasted to cause a cumulative economic cost of 100 trillion USD in 2050s. However, there are several studies concerning the financial impact of ASP interventions showing that a significant portion of these costs can be curbed [Glowacki et al., 2003, McGregor et al., 2006, Agwu et al., 2008].

The next section gives a broad description about the mechanisms of antibiotics and ABR to set the ground for the explanation of key aspects of ASPs and monitoring tools.

2.2 Mechanism of Antibiotics and Antibiotic Resistance

Antibiotics are special molecules used in prevention and treatment of bacterial and fungal infections in people, animals and plants. Antibiotics target specific parts or processes of microorganisms to disarm. For example, some antibiotics are designed to stop protein synthesis, some to prohibit DNA replication and some to spoil the cell wall of infectious microorganisms [CDC, 2019]. Therefore, each antibiotic has a different impact on different microorganisms. In other words, not all antibiotics are effective on all kinds of infections.

The potency of antibiotics is different. Therefore, the amount of antibiotic to be used in a treatment is determined by the characteristics of the antibiotic and the infected organism. Excessive antibiotic use causes side effects in the infected organism and harms beneficial bacterial flora. On the other hand, when an insufficient amount of antibiotic is used, the surviving microbes reproduce and lead to the recurrence of infection.

In addition, the duration of activity of antibiotics varies because antibiotic molecules are destroyed in the body at different speeds. Therefore, some antibiotics lose their effect in one day, while others lose it in eight hours. When the amount of antibiotic molecules in the blood falls under a certain threshold the antibiotic needs to be retaken to continue treatment. And as soon as the infection is removed from the organism, the antibiotic intake must be stopped.

The antibiotics can be administered in many ways depending on the conditions of the infected organism and the severity of infection [CDC, 2019]. The route of administration of antibiotics plays a significant role in the rise of antibiotic resistance. Table 2.1 explains the five antibiotic administration methods [Zhang et al., 2013].

To sum up, a successful treatment requires the right dose of the right antibiotic administered at the right time for the right duration through the right route.

When these requirements are not satisfied, the chance for microorganisms to develop antibiotic resistance increases. For instance, when an ineffective antibiotic is used in the treatment, not only the microbes survive but also, they acquire adapta-

Method	Organisms	Conditions
Intravenously (directly into a vein)	Inpatients (more often) Outpatients Pets Livestock	For more serious infections
Orally (by mouth)	Inpatients Outpatients (more often) Animals	For more common infections
Intramuscularly (through injection into a muscle)	People Animals	Alternative to oral method
Topically (directly to the skin)	People Animals	For skin, eye and ear infections
Spraying (applied by trucks or airplanes)	Plants	For fruit trees and crops

Table 2.1: Different Routes of Antibiotic Administration

tions to fight back against this antibiotic. These adaptations can be passed to other microorganisms via mobile genetic elements, making them resistant to this antibiotic, as well [CDC, 2019]. This scenario also applies when an effective antibiotic is used but the duration is not right.

2.3 Data Requirements for Antibiotic Stewardship Programs

Effective ASPs require proper quality indicators (QIs) that can be used to measure appropriateness of antibiotic use in the treatment of bacterial and fungal infections. Monitoring and improving these indicators are the main aims of these programs. Bosch et al. lists nine QIs as follows: (1) take two blood cultures, (2) take cultures from suspected sites of infection, (3) prescribe empirical antibiotic therapy according to local guideline, (4) change empirical to pathogen-directed therapy, (5) adapt antibiotic dosage to renal function, (6) switch from intravenous to oral, (7) document antibiotic plan, (8) perform therapeutic drug monitoring, and (9) discontinue antibiotic therapy if infection is not confirmed [Van Den Bosch et al., 2015]. Thus, the information about these steps, namely, infection indicators such as laboratory test results and fever status, and detailed antibiotic administration data must be available to the program personnel and to the monitoring tool. Considering this information, the program personnel should detect the misuse and overuse cases, and take the necessary actions.

2.4 Different Intervention Approaches of Antibiotic Stewardship Programs

There are two main intervention methods listed in the literature: restrictive and persuasive [Davey et al., 2017]. The methods which directly restricts the prescription of certain antibiotics or require approval from an infection specialist are categorized as restrictive methods. The duration of use of these antibiotics may be specified in by the infection specialist, as well.

On the other hand, the methods which aim to positively change the prescribing behavior of the prescriber by providing information support are categorized as persuasive methods. This method, using the means of educational support, reminders, auditions and feedback, aims to prevent unnecessary and inappropriate prescriptions without explicitly restricting. There are also cases which restrictive interventions contain persuasive components.

The proposed tool in this thesis is aimed to be utilized in persuasive interventions, providing information support to the prescriber and helping ASP personnel in ACM.

2.5 Different Scopes of Antibiotic Stewardship Programs

The ASPs can be examined in three main scopes, hospital level (including primary care), national level and international level.

As discussed in Section 2.3, health centers should monitor and assess antibiotic consumption according to certain quality indicators. Misuse and overuse cases should be detected and analyzed so that purposeful interventions can be designed accordingly, and antibiotic consumption is improved. Even though the objectives of antibiotic stewardship programs are similar, the methods of implementation of these programs may vary from one health center to another. Health centers may develop and follow their own guidelines and employ different intervention approaches to improve antibiotic consumption. In fact, the proposed tool is developed to be utilized in the hospital level stewardship programs, and its primary aim is to provide information support to stewardship managers regardless of what guidelines they follow.

However, monitoring and improving antibiotic consumption only in a few health centers is not enough to eliminate the problems described in Section 2.1. Although hospital level improvement of antibiotic use is the fundamental step to combat ABR, promising outcomes can be achieved only if the stewardship requirements are fulfilled nationwide or even globally. Therefore, national and international stewardship committees must monitor country-wide and world-wide antibiotic consumption trends and make restrictive and persuasive interventions when necessary.

As an example, in Turkey in 2003, Ministry of Health started a national antibiotic restriction program prohibiting the use of several broad-spectrum antibiotics (i.e., carbapenems, glycopeptides, piperacillin/tazobactam, ceftazidime and cefepime) without approval of infectious diseases specialists [Hassouna and Deshpande, 2018]. Moreover, for some other antibiotics, the maximum duration of utilization without approval is limited to 72 hours. As a result of the restrictions, it is observed that the consumption of restricted drugs is reduced in certain hospitals.

On the other hand, international level interventions can be exemplified as declarations issued by political unions such as Political Declaration of the High-Level Meeting of the United Nations General Assembly on Antimicrobial Resistance, global reports, reviews, guidelines and action plans prepared by international organizations such as the World Health Organization and the World Organisation for Animal Health, and global conferences, meetings, congress and forums gathered to raise awareness, exchange information and collaborate in combatting ABR [FAO et al., 2017].

2.6 Types of Monitoring Software

In this section, the potential types of software to be used for antibiotic consumption monitoring are discussed. Advantages and disadvantages of each software type is explained.

2.6.1 Desktop Applications

A desktop application can be described as a software that is designed to be run on a desktop computer. Special desktop applications can be designed for antibiotic consumption monitoring, however, designing desktop applications require a good level of knowledge in computer operating systems. The main advantages of desktop applications can be listed as:

- can be directly connected to the database system,
- does not encounter problems caused by internet connection,
- being suitable for visually intensive applications.

The main disadvantages can be listed as:

- required to be installed on each computer,
- updating the applications is inconvenient as each computer must be separately updated,
- often require IT support for installment and update,

- may require a fixed location for database connectivity,
- same application package does not work on different operating systems,
- performance problems caused by unsatisfied system requirements.

Desktop applications are most suitable for analysis of complex systems and visually intensive tasks. They constitute a good fit for antibiotic consumption monitoring task as well, however, since desktop applications use the processing unit of the computer that they are installed on, computationally demanding processes may cause performance problems. Also, maintenance of the system is often burdensome.

2.6.2 Mobile Applications

Another type of software available to be employed in monitoring tool design is mobile applications. A mobile application is the software designed to be run on mobile devices such as smartphones and tablets. As the prevalence of smart mobile devices in the community has increased substantially in the last decade, the number of tools facilitating mobile software development has also increased. The main advantages of mobile applications can be listed as:

- can work online and offline,
- easier to build than desktop applications,
- faster than web applications with similar features,
- often does not require IT support.

The main disadvantages can be listed as:

- required to be installed on each mobile device,
- small screen size compared to computer screens can be restrictive,
- same application package does not work on different mobile systems,

- performance problems may occur due to low processing power compared to computers.

Mobile applications are most suitable for simple data collection and tracking tasks where mobility is an advantage. Also, they are suitable for providing information about policies and guidelines to the prescriber. There are a number of available mobile applications aiming to provide decision support to improve antibiotic stewardship such as Imperial Antibiotic Prescribing Policy (IAPP) and Sanford Guide to Antimicrobial Therapy [Imperial College London, nd, Sanford Guide, nd]. However, mobile apps are not a good fit for antibiotic consumption monitoring purposes because the small screen size is restrictive for this task. The users often prefer to take advantage of a larger screen using which they can get richer visual support and utilize multiple information sources simultaneously.

2.6.3 Web Applications

Web applications are another type of software that can be utilized in antibiotic consumption monitoring. A web application is a software that can be accessed and manipulated online using a web browser. The application runs in a server computer and users can connect to this computer via internet. The main advantages of web applications can be given as:

- they do not need to be installed,
- accessible from any device connected to the internet, regardless of the operating system,
- any update on the server becomes immediately effective for all users, no explicit update required,
- computations are done on the server computer thus no performance issues depending on the user's device,
- easier to build than desktop and mobile applications.

The main disadvantage can be given as:

- require internet connection.

Web applications are suitable for supporting ACM tasks since they can be accessed from any device with internet connection (e.g., desktop computers, tablets, smartphones) currently available to the personnel. Also, because the computations are handled at the server side, no performance problems arise due to potentially insufficient processing capacities of available devices. Since they do not require installment and update, they do not bring additional burden to IT units. Also, the internet connection requirement is not an obstacle since most, if not all, health centers in Turkey have internet connection because of several requirements (e.g., to transfer healthcare statistics to the Ministry of Health of Turkey and to access centralized doctor appointment system).

The advantages and disadvantages considered; web applications are the most favorable of these three potential application types to employ for ACM support in health centers. Thus, we developed a web application to be utilized in antibiotic stewardship activities, and the next chapter presents detailed information about the design and features of this application.

Chapter 3

A REAL-TIME MONITORING TOOL FOR ANTIBIOTIC CONSUMPTION

In this thesis, we present a web application which we designed to be utilized in real-time antibiotic consumption monitoring. This way, we aim to help prevent inappropriate antibiotic use in health centers by providing information support to physicians and ASP personnel. Note that the proposed tool is designed to monitor and analyze the antibiotic use of inpatients, and outpatients are out of the scope of this project.

In very broad terms, the application processes the antibiotic consumption data according to the inputs obtained from the user and presents graphs and tabular information in an interactive way. System backend is developed using R programming language. Elasticsearch search engine is used for database operations. The functional components and the user interface are combined as a web application using the R package named Shiny, and the web application is deployed to the Web via Shiny Server software which is installed on a high-performance computer. Since the whole set of software utilized in this web application is open source, the tool is completely free to use.

Special requirements for antibiotic consumption monitoring task, design of the application and its technical features are explained in detail in the following sections.

3.1 Special Requirements for Antibiotic Consumption Monitoring

Data management systems used in health centers often do not provide data processing and visualization services for specific needs such as detailed antibiotic consumption monitoring. They are mainly designed for handling daily operational procedures such as storing and displaying health records and generating essential medical and financial reports and statistics. ASP personnel, on the other hand, need to analyze antibiotic consumption trends utilizing multiple sources of information usually stored on separate relational tables in the database. Therefore, they need special tools using which multiple information sets can be examined simultaneously and in relation to each other.

General-purpose data analysis and visualization tools like MS Excel are commonly available at health centers, and they are widely used by ASP personnel. However, these tools often require the data to be extracted from the database and formatted properly. When the data to be analyzed is significantly large, extraction and formatting processes become burdensome. Moreover, these tools are prone to performance issues depending on the size of the data. Some of these tools offer direct connection to data management systems, as well, in which case no explicit data extraction and formatting is required. However, a good level of technical knowledge in query languages and database operations is required for utilizing this feature, which most ASP personnel does not have.

There are a variety of commercial tools available specially designed for conducting and evaluating antibiotic stewardship, as well. With simple user interfaces, these tools do not require technical knowledge. To name a few: MedMined by BD [MedMined, nd], TheraDoc by Premier HealthCare [TheraDoc, nd], Sentri7 by Wolters Kluwer [Sentri7, nd] and PathFinder by Vecna [Vecna Healthcare, nd]. These products are able to identify the major stewardship concerns such as antibiotic administration route switches (e.g., from intravenous to oral), drug-bug mismatches, dosage adjustments and resistance. Most of them are able to integrate with infection control tools. Also, they are able to employ customizable alert mechanisms. However, they do not necessarily save time. The increased information flow requires interpretation and triaging, leading to substantial number of nonactionable alerts [Hassouna and Deshpande, 2018]. Moreover, the licensing costs of these tools are a high burden that most of the health centers cannot afford.

To sum up, there is a need for antibiotic consumption monitoring tools which are specially designed to provide information support to physicians and ASP personnel. Already available data management systems and general-purpose data analysis and visualization tools are insufficient to fulfill this need. Although there are a variety of commercial products specially designed to fulfill this need, health centers usually are not able to afford them because of high license fees.

In this thesis, we present a real-time ACM tool, which employs a bunch of open source software, namely, R Programming Language, Shiny Server and Elasticsearch Search Engine. Utilizing these open source software makes the tool completely free and at the same time advanced in terms of technical capabilities.

3.2 System Design

The proposed ACM tool in this thesis is a web application. Section 3.2.1 explains how the proposed web application operates in detail, and Section 3.2.2 presents the capabilities and advantages of this web application.

3.2.1 Workflow

Workflow schema of the system is given in Figure 3.1. The components of this workflow are explained below.

1. Information Transfer

   Firstly, when developing an information transfer pipeline for an ACM tool, one needs to have a good understanding of the database structures and data type standards used for storing electronic health records (EHRs). The most commonly used EHR standards around the world are Health Level Seven (HL7) standards which has been developed by the not-for-profit organization Health Level Seven International. HL7 and its members provide a framework (and related standards) for the exchange, integration, sharing, and retrieval of electronic health information. These standards define how information is packaged

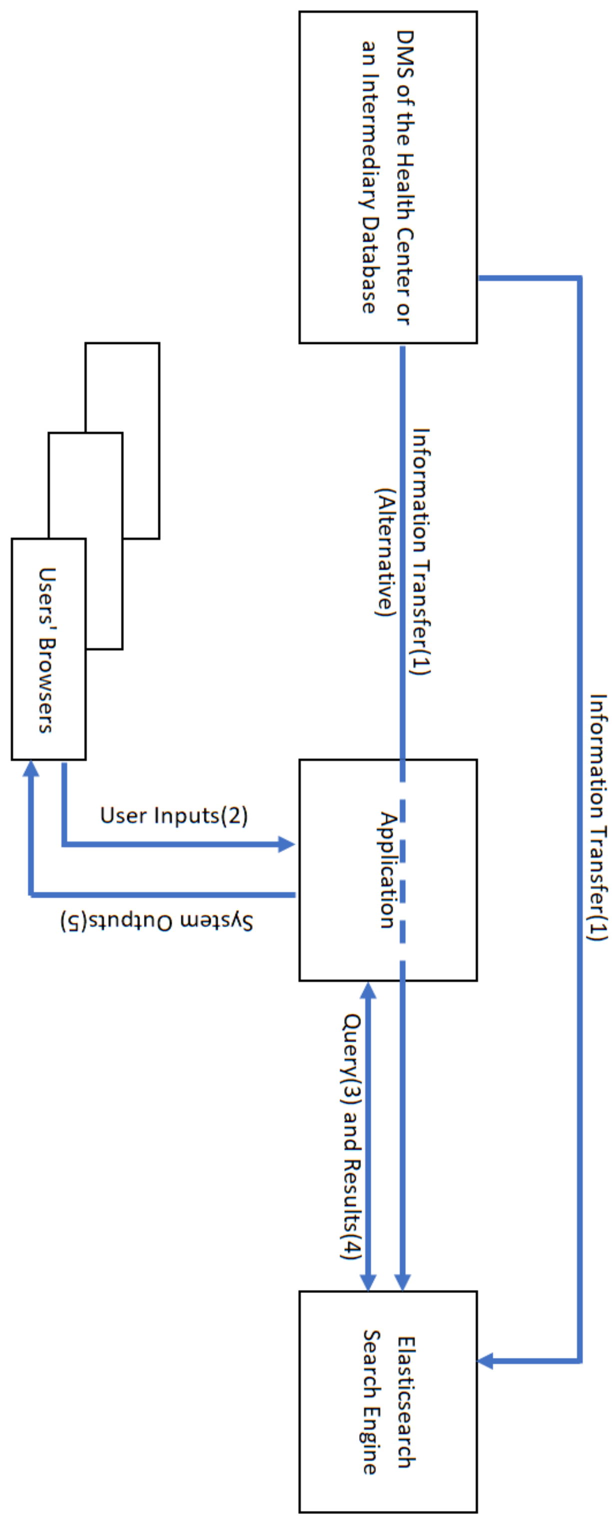


Figure 3.1: Workflow Schema

and communicated from one party to another, setting the language, structure and data types required for seamless integration between systems [HL7 International, nd]. Turkeys National Health Information System (NHIS) is implemented based on HL7 standards, as well [Kose et al., 2008]. Therefore, aiming to develop a monitoring tool to be used comprehensively, we ensured conformity to these standards in the design of the proposed application.

In accordance with these standards, EHRs are stored in so called relational databases instead of flat file databases. These two types of database systems are described as follows.

A **flat file database** is basically a plain text file in which the lines store different records and the lines are delimited by some special character separating different record fields. In other words, a flat file database is a standalone table whose information is stored in a simple and portable file, and it does not require special software to be used.

A **relational database**, however, consists of multiple tables which are linked to some others by record field references. Some fields (i.e., columns) of these tables are "indexed" so that it allows quick search in the database. Relational databases usually require special software for data access and data operations. This software is called relational database management system (RDBMS), and well-known examples are Oracle, MySQL and PostgreSQL.

The information transfer module of the proposed application is designed to be compatible with the special structure of relational database systems. The technical details of this module will be explained later in this section.

The application is designed to run in real-time, meaning when a new EHR (e.g., hospitalization of a new patient or administration of an antibiotic) is created, it is immediately processed by the application and it becomes available in the monitoring system. In other words, the time required for a new medical record to be transferred to the monitoring system is negligible when the necessary conditions are met, thanks to the design of the system.

For real-time monitoring to be feasible, naturally, the information transfer must be continuous and instantaneous. To ensure these, one can suggest connecting the application to the data management system (DMS) directly, so that the delays caused by information transfer will be minimized. However, since the application requires continuous data flow from the DMS, it charges an additional burden to the system. Considering the performance issues that it may cause; direct connection scenario may not be preferred.

Alternatively, an intermediary database consisting only of the EHRs required for the monitoring system can be created. This way, the application connects to this mirror database which is synchronized with the main database, thus no additional burden on the main DMS is charged.

If none of these scenarios are feasible for a health center, EHRs can be transferred to the monitoring system periodically, as a collection of files. For example, the EHRs can be extracted from the DMS and transferred to the application on a daily basis. Scheduling the information transfer at off-peak times like early mornings is usually a good practice which prevents the performance issues with the DMS. Applying this scenario, however, the ACM managers must be aware of that some of the features of the application (e.g., antibiotic misuse alert) will no longer function as originally intended, since the system does not run in real-time.

Once the database connection is established, the application keeps track of new and updated records. It processes and formats these records, and indexes into an Elasticsearch database (illustrated as "Information Transfer(1)" in Figure 3.1). In the case of non-continuous information transfer, after the necessary files received, the application inserts the records to Elasticsearch database all at once (illustrated as "Information Transfer(1) (Alternative)" in Figure 3.1).

Elasticsearch is a distributed, open source search and analytics engine for all types of data, including textual, numerical, geospatial, structured, and unstructured [Elastic, nd]. It is built on a Java-based search library called Apache Lucene and it was first released in 2010 by Elasticsearch N.V. (now known as Elastic). It is fast and scalable, and it is the central component of the Elastic Stack, a set of open source tools for data ingestion, enrichment, storage, analysis, and visualization.

In Elasticsearch, the documents (i.e., records) are stored in JSON (JavaScript Object Notation) format, and an index basically is a collection of documents and the relationships among them. The first step to use Elasticsearch is "indexing" the data. Indexing is the process of formatting and inserting the data into Elasticsearch. In addition, during this process, Elasticsearch generates a data structure called inverted index, which constitutes the main technology that gives Elasticsearch a substantial search speed.

The reason that Elasticsearch is used in this thesis is because it is fast and scalable. Moreover, it runs in near real-time, in other words, the time required between indexing a document and having it searchable is negligible. Also, it is accompanied with other open-source technologies such as Logstash, and this makes Elasticsearch easier to use.

In order to establish a fast and continuous data flow pipeline from database of a health center into Elasticsearch, we suggest using Logstash. **Logstash** is another product by Elastic, which is used as a data flow bridge between Elasticsearch and other databases. It provides a pipeline that collects and transforms data, and indexes it into Elasticsearch [Elastic, nd].

The rest of this subsection presents an example configuration of Logstash with the JDBC (Java Database Connectivity) input plugin to keep Elasticsearch synchronized with a RDBMS such as MySQL. Note that this part is intensive in terms of technical terminology and written to provide information about technical steps of synchronizing Elasticsearch with other databases. Non-technical readers may prefer skipping to the next subsection.

In this example scenario, basically, Logstash need to periodically visit the main database (i.e., RDBMS of a health center), and track the records which are new or updated since the previous visit. Although conceptually straightforward, this task requires certain conditions to be met.

First of all, the "id" field of the main database and "_id" field of Elasticsearch must be synchronized, so that Logstash can link the records in the main database with the ones in Elasticsearch. This way, when Logstash detects a new record or an update in the main database, it manipulates the corresponding document in Elasticsearch.

Secondly, records in the main database are required to have a field which stores the creation or update time. Utilizing this field, Logstash will track only the records that are created or updated after the previous visit to the main database. Without this condition, Logstash would need to check the whole set of records in the database to identify the changes, and this would take a substantial amount of time in each visit.

When these conditions are satisfied, Logstash can successfully track the changes in the main database and synchronize Elasticsearch accordingly.

Consider a scenario where we utilize Logstash to synchronize Elasticsearch and a table in some database with the following specifications:

name of the database:	`x_hospital_database`,
name of the table:	`antibiotic_administration_table`,
name of the update time field:	`update_time`.

We will specify `update_time` field in the Logstash pipeline, so that Logstash can filter the records whose `update_time` values are later than its last visit to `antibiotic_administration_table`.

Note that the table needs to contain `id` field as explained previously in this section. Also, the table will, naturally, contain several other fields, yet these fields are not mentioned since they do not play a role in the configuration of Logstash.

We can configure the Logstash pipeline using the piece of code below [Elastic, 2019, Marquardt, 2019].

```
input {
  jdbc {
    jdbc_driver_library => "<jdbc driver library path>"
    jdbc_driver_class => "<jdbc driver class>"
    jdbc_connection_string => "jdbc:<db type>://<db host>/x_hospital_database"
    jdbc_user => <username>
    jdbc_password => <password>
    tracking_column => "update_timestamp"
    use_column_value => true
    tracking_column_type => "numeric"
    schedule => "*/2 * * * * *"
    statement => "SELECT *,
                    UNIX_TIMESTAMP(update_time) AS update_timestamp
                 FROM antibiotic_administration_table
                 WHERE UNIX_TIMESTAMP(update_time) > :sql_last_value
                    AND update_time < NOW()
                 ORDER BY update_time ASC"
  }
}
filter {
  mutate {
    copy => { "id" => "[@metadata][_id]"}
    remove_field => ["id", "@version", "update_timestamp"]
```

```
  }
}
output {
  elasticsearch {
      index => "x_hospital_es_index"
      document_id => "\%{[@metadata][_id]}"
  }
}
```

Note that, for illustration purposes, a MySQL statement is employed as the `statement` parameter above, and the syntax may differ for other RDBMSs.

Here, `tracking_column` parameter is set as `update_timestamp` which is basically a numerical representation of `update_time` generated by the MySQL function `UNIX_TIMESTAMP()` [MySQL, nd]. In each visit to the database, Logstash will synchronize the data in Elasticsearch and store the `update_timestamp` value of the last accessed record as an internal variable i.e., `.logstash_jdbc_last_run`. Then, in the next visit, it will use this value to detect the new changes in the database.

`WHERE` condition in `statement` specifies the filtering criteria. `:sql_last_value` returns the value of variable `.logstash_jdbc_last_run` allowing for the use of this value in the criteria statement. The function `NOW()`, as the name implies, returns the current date and time [MySQL, nd]. Then, as specified by the statement, Logstash will get the records whose `update_time` is between "now" and the time of the last visit. This data will be inserted into Elasticsearch, and `.logstash_jdbc_last_run` variable will be updated. One may think that `update_time < NOW()` piece of the criteria statement is redundant, however, it prevents skipping the changes which occur during processing.

The `schedule` parameter specifies the time between the concurrent visits to the database. It uses the Cron time string format, and the specified piece of string i.e., `*/2 * * * * *` implies that the system will initiate a visit to the database every 2 minutes [Elastic, 2019].

In `filter` section of the code, we copy the values of `id` field of `antibiotic_administration_table` into a metadata variable named "_id" (i.e., `[@metadata][_id]`. These values will be used to match the records in Elasticsearch and the main database. Also,in this part, the fields `id`, `@version` and `update_timestamp` are removed from the documents since these fields are unnecessary to include in the Elasticsearch.

And finally, `output` section of the code tells Elasticsearch to index the specified documents, matching `document_id`s stored in the specified metadata variable.

Note that, in the proposed configuration, when a document is deleted from the main database, the deletion will not be propagated to Elasticsearch. To fix this problem, one can employ a "soft delete" approach and include a binary valued field in the main database to indicate whether a record is deleted or not. In this scenario, when a record is deleted, the value of the deletion indicator field will change, thus Logstash will update this record in the Elasticsearch. So that the deletion information will be transferred to Elasticsearch.

2. User Inputs

One of the main advantages of ACM tools is that they provide user interfaces for triggering various predefined actions. These actions may employ user-defined parameters. Utilizing these parameters, for example, users can determine the characteristics of the patient group to be monitored or adjust the style of the visual components.

The proposed application offers a variety of actions which employ user inputs. Users enter these inputs from any web browser using the interface generated by the application. Some actions are designed to be explicitly triggered by the user, while others are automatically triggered when predefined conditions are satisfied. When explicit triggering is required, usually a trigger button is included in the interface. When an action is triggered, the user inputs are sent to the application to be processed (illustrated as "User Inputs(2)" in Figure 3.1).

Input operations takes place on the user's (the client's) device, thus they are called client-side operations. On the other hand, functional processes take place on the server machine, thus they are called server-side operations.

Various types of inputs are utilized in the application such as date, free text, numeric and set of options, each of which has special characteristics and visual representations. These input types and actions are explained in detail in Section 3.2.2.

3. Query and Results

   Most of the actions in the application, naturally, requires retrieval and processing of the information stored in the Elasticsearch database. As an action requiring information retrieval is triggered, the application sends a query, a piece of text with special syntax, to the database (illustrated as "Query(3)" in Figure 3.1). This query is dynamically created according to the corresponding user inputs and specifications of the action.

   When a query is submitted, Elasticsearch Search Engine collects the data and processes it as described in the query. The resulting data is sent back to the application in JSON format (illustrated as "Results(4)" in Figure 3.1).

4. System Output

   As the application obtains the query result from Elasticsearch in JSON format, the predefined procedures for the corresponding action start. Usually, it is more convenient to transform the data into a tabular format since it is easier to observe and manipulate the data this way. Then, this transformed data is either employed in a tabular format as is, or it is embedded into various charts depending on user needs.

After the operations with the data are finalized, an HTML file enriched with CSS and JavaScript components is generated by the server-side of the software and it is sent to the client-side. Users' browsers process this HTML file and project the output to the screen (illustrated as "System Output(5)" in Figure 3.1).

5. System Orchestration

   This section explains how the information flow in the system is regulated.

   Web applications comprise at least an application server and a web server, occasionally accompanied by a database. In this thesis, the term "application" is used to refer to the combination of these three components.

   A typical web application runs as follows. Users access to the system through web browsers. They trigger actions using the interface. These action requests are transferred to application server via web server. Then, application server performs the requested action and sends results back to users via web server.

   In the presented web application, the application server is basically a high-performance computer running scripts written in R programming language. These scripts store the procedures constituting the functionality of the application. Various R packages (i.e., collections of R functions, data, and compiled code in a well-defined format) are utilized in these procedures. These packages provide a strong codebase for technically demanding tasks such as communicating with the database and generating interactive visualizations.

   On the other hand, the web server of the proposed application is a computer running Shiny Server which is an open source back end program specifically designed to host Shiny apps. When necessary, Shiny Server provides a controlled environment for shiny applications to be accessible only within the organization, which is essentially advantageous for the proposed application due to information security concerns. Shiny Server hosts an application at a specified web address and automatically starts the application when a user visits the address. When the user leaves, Shiny Server automatically stops the application. It creates separate sessions for each user, so that multiple users can access and use the application at the same time without interfering each other [Allen, 2014].

3.2.2 Features and Advantages

The application consists of three functional components: "Patient Tracking", "System Tracking" and "Alerts". We will refer to these functional components as "tabs".

Patient tracking tab is designed to monitor antibiotic administration records and laboratory test results in the patient level. Users enter either a Patient ID or an Episode ID whose records to be retrieved. Users also determine the date range to be focused and thresholds below which the test results will be regarded as normal. According to these inputs, the system displays episode information, antibiotic use records and laboratory test results. Episode information can be described as the general hospitalization information such as Patient ID, Episode ID, Hospitalization Unit, Gender, Hospitalization Date and Discharge Date. Antibiotic use records include Date, Time, Active Ingredient, Commercial Name and a value indicating appropriateness of antibiotic use. Laboratory test records include Date, Time, Test Type, Result and a value indicating if the corresponding result is in the normal range.

Antibiotic use records and test results are shown in both tabular and graphical format. Tables allow filtering and sorting. Graphs help observing the trends in the records easily.

System tracking tab is designed to monitor the antibiotic administration trends in the system level. Users determine the characteristics (e.g., Gender, Hospitalization Unit, Administered Antibiotic Classes) of the study group. The application collects the antibiotic administration records and test results of the specified group, and generates graphs showing antibiotic use trends. The corresponding data is also displayed

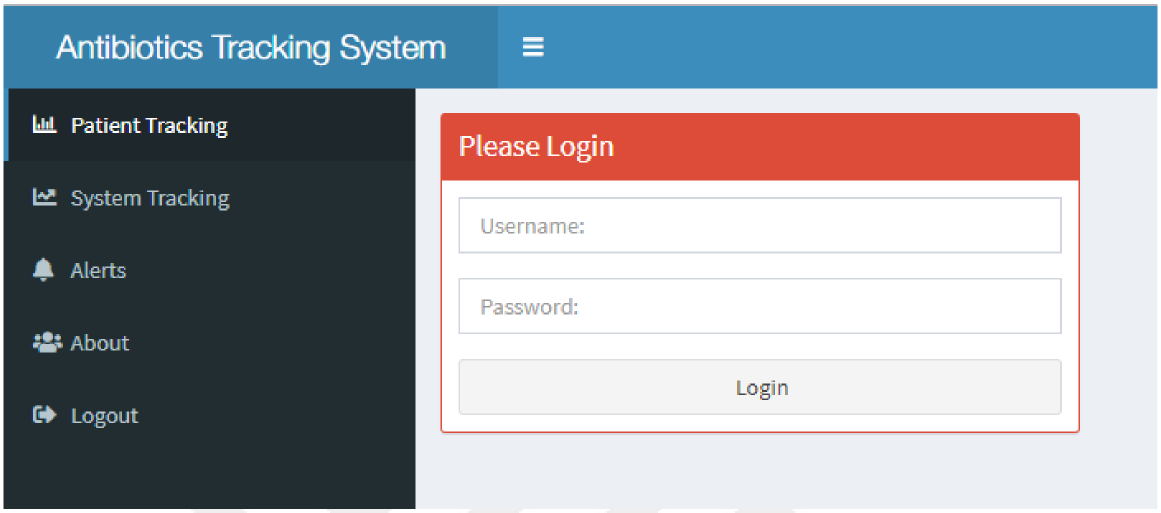


Figure 3.2: Sidebar and Login Panel

in the tabular format.

Alerts tab is designed as a daily reporting system which indicates the potential antibiotic misuse cases. Prescribers/physicians can check this tab periodically and reassess whether the specified antibiotic administrations are indeed appropriate. If inappropriate, the physician discontinues the administration.

Users navigate between these tabs from the left sidebar (shown in Figure 3.2). These tabs require signing in to be activated, since the information is only available for the authenticated users due to information security concerns. You can observe the login panel in the same figure.

Each tab has a special design with different input and output areas as discussed above. Figure 3.3 shows the input panel of System Tracking tab with example inputs.

The main advantages of the system are explained below.

1. Enables real-time monitoring

   Real-time monitoring feature has direct benefits on patient, community and management/financial sides. The tool is not only for retrospective analysis, but it also allows users to track patients' laboratory test results, diagnoses and

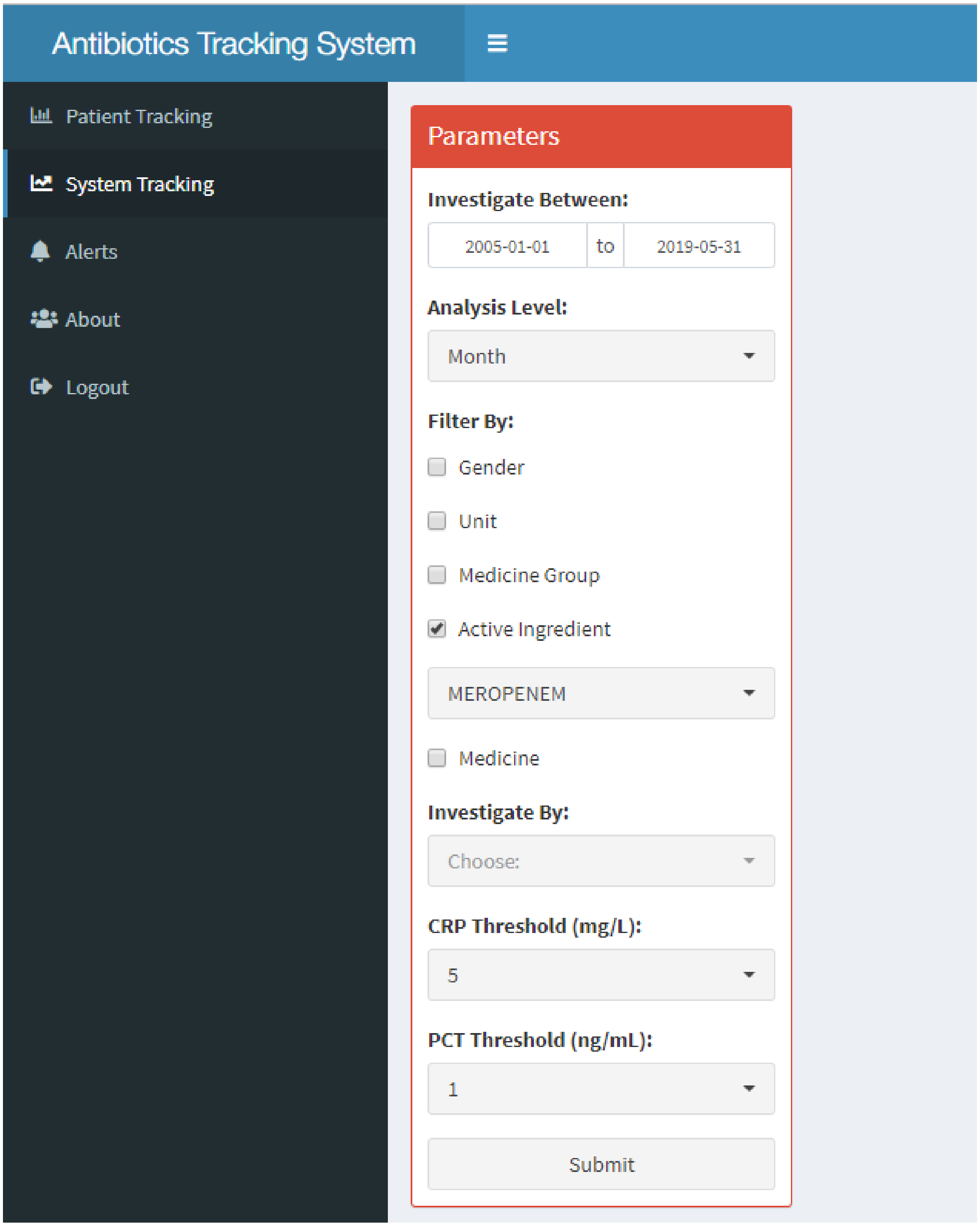


Figure 3.3: Input Panel

antibiotic consumption in a convenient way in real-time. Thus, it provides support for timely interventions such as suspension of unnecessary antibiotic use. As discussed in Section 2.1, misuse and overuse of antibiotics lead to antibiotic resistance, which has fierce impacts on both the patient's and the community's health. Financial benefits of real-time monitoring is also significant. By preventing unnecessary use of antibiotics, drug administration costs can be reduced significantly. Also, as discussed in Section 2.1, huge costs of antibiotic-resistant infections, and the morbidity and mortality attributable to them can be mitigated with the help of effective antibiotic stewardship and continuous monitoring.

2. Enables in-depth monitoring

The application allows filtering on the data, so it helps users conduct in-depth analyses to assess ASP performance in subgroups. One practical use case is filtering antibiotic uses by active ingredients and investigating the rate of misuse in the resulting subset. Figure 3.3 shows an example that the antibiotic uses are filtered as their active ingredient is to be meropenem. This feature of the application allows users to carry out analyses targeting specific subsets in the data.

Another feature assisting in-depth monitoring is "investigate" feature. It allows ASP managers to focus on a specific dimension of antibiotic consumption data and compare the categories within this dimension. An example use case is to investigate by gender, in which case the application displays antibiotic use statistics for male and female patients separately on the same graph. This allows ASP managers to easily compare ASP performances in selected subcategories.

Filter and investigate features can be used together to assess the performance in various subgroups. An example use case for this is to filter the data on active ingredients to be meropenem and investigate by units to observe the rate of misuse of antibiotics containing meropenem in different units in the health center. This helps users in benchmarking the statistics and adjust the targets of potential interventions accordingly.

3. Fast & scalable

   The application runs very fast thanks to its special design. Instead of using traditional relational database systems, the data is indexed in an Elasticsearch database. Nested indexing is preferred since it allows collecting all related information of a hospitalization episode within the same parent document. This allows fast reach to the queried information. Even if the query is in the scope of nested fields, all the information stored in the parent document is returned as fast as if the information is kept in a single denormalized document. Moreover, since Elasticsearch has a cache mechanism that recognizes previous queries, response time becomes even smaller starting from the second time a certain query is run.

   This tool is designed in a scalable manner because this project is aimed to develop a comprehensive ACM tool that can be used in health centers of any size. Unlike the traditional DBMS which may have performance problems in large scales, the presented tool utilizes Elasticsearch Search Engine which is able to handle big data operations without trouble.

4. Visual support & interactivity

   The tool does not merely return raw query results in the traditional tabular format, but it provides interactive time-series graphs and conditionally formatted interactive tables for convenient use.

   Laboratory test results are displayed as interactive time-series graphs. This allows convenient recognition of the trends in the test results. In addition, the cases with unnecessary antibiotic consumption are highlighted in the corresponding table for faster recognition. Similarly, the test results exceeding specified thresholds are highlighted in corresponding tables, and these thresholds can be reset from the input panel. For culture tests, the positive test results are highlighted. These tables allow sorting and filtering, as well.

Chapter 4

IMPLEMENTATION OF THE TOOL USING REAL-LIFE DATA

In this chapter, we present the implementation of the proposed tool using the data from American Hospital of İstanbul. The main aim of this chapter is to present real-life use cases of the proposed monitoring tool. In addition, this chapter investigates the antibiotic consumption trends, and provides observations and recommendations on the ASP conducted in American Hospital of İstanbul.

American Hospital of İstanbul, which we will call as American Hospital from here on, is one of the oldest and well-known hospitals in Turkey. It was established in 1920 as the first non-profit private hospital in Turkey [American Hospital, nd]. The hospital today has 278 patient rooms, including 58 rooms reserved for intensive care patients, 62 intensive care beds and 12 operating rooms. Annually, 220,000 patients visit the hospital and more than 10,000 operations are carried out.

American Hospital started antibiotic stewardship activities (i.e., monitoring of antibiotic consumption) in 2012. ABIS (Turkish abbreviation for antibiotic monitoring system) program was established with the support of information technologies department of the hospital. This system has allowed stewardship personnel to monitor consumption automatically. With this system, antibiotic consumption within 24 hours (number of inpatients, number of limited antibiotics used during that day, prophylactic antibiotic information lasting longer than 24 hours and discontinued before 24 hours) have been reported to the infection control team on a daily basis. The team have documented monthly consumption amounts in Excel files and have assessed the trends at infection control committee meetings.

In addition, considering the recommendations of the reference institutions, a working group was established in 2017. An infectious diseases specialist, a pharmacist, an infection control nurse and a person from the management have been included in this working group. The group has convened at least quarterly to monitor antibiotic consumption and have planned and conducted effective studies to reduce consumption. It has worked in collaboration with infection control team and pharmacy department to monitor consumption.

Also, clinical instructions have been established to guide healthcare workers in specific situations concerning antibiotic consumption. Monthly bulletins and conferences have been organized to raise awareness. Meetings have been held with various departments with the aim of providing information about current antibiotic consumption behaviors.

However, ASP personnel of American Hospital does not have a direct access to laboratory test results and other medical information. Also, the antibiotic consumption reports available to them are insufficient to disclose the antibiotic consumption trends in low levels. Thus, the proposed tool has a potential to add significant value to the stewardship activities of American Hospital.

The next section provides information about the American Hospital data set to be utilized in the implementation, and Section 4.2 presents some analyses. Note that the data to be investigated in this chapter is provided as a set of external files, in other words, no continuous data flow mechanism is established.

4.1 Data Description

This section describes the data obtained from American Hospital, which consists of the following tables (i.e., data sets) stored in the relational database system of the hospital.

1. Inpatients Table

   The information stored in inpatients table (i.e., column names, descriptions, column types, sample values and the number of distinct values) is explained in

Column Name	Description	Type	Example	Distinct Count
PID	(Patient ID) A unique code identifying each registered patient.	Integer	781918, 780344	123,180
EID	(Episode ID) A unique code identifying each hospitalization episode. An episode includes all the actions starting from hospitalization to discharge. EID is the primary key of this table.	Integer	7814354, 7744362	208,509
Gender	Biological gender of the patient.	Character	MALE, FEMALE	2
Unit ID	A unique code identifying each hospital unit.	Integer	1174, 140078567	29
Unit	Description of each hospital unit.	Character	KAT 1 A, KAT 4 B3	29
Hospitalization Date	Starting date of an episode.	Date	2006-09-07 17:53:31	-
Discharge Date	Ending date of an episode.	Date	2018-09-08 16:32:00	-

Table 4.1: Inpatients Table

Table 4.1. The data starts from January 1, 2005, in other words, the data includes the patients who were discharged on January 1, 2005 and after. The data ends at June 19, 2019, in other words, the data includes the patients who were hospitalized on June 19, 2019 and before. There are 123,180 distinct patients, 208,509 distinct hospitalization episodes, 29 distinct units (not necessarily serving at the same time) in the data set. The primary key of this table is “EID”.

2. Antibiotic Administration Table

   The information stored in antibiotic administration table is explained in Table 4.2. This table consists of the antibiotic administration records between January 1, 2005 and June 19, 2019. Records of outpatients are not included in this table. The table is linked to the inpatients table by “EID” field, and the primary key of this table is “SID”.

3. Laboratory Test Results Table

   The information stored in laboratory tests table is explained in Table 4.3. This table consists of the laboratory test records between January 1, 2005 and June 19, 2019, like other tables. Although there are several other tests conducted in the hospital, only “C Reactive Protein” and “Procalcitonin” tests are included in this table as these are the main tests to survey the presence of infections. The table is linked to the inpatients table by “EID” field, and the primary key of this table is “SID”, as well.

4. Culture Test Results Table

   The information stored in culture tests table is explained in Table 4.4. Although, this table originally consists of fifteen columns in total, we do not include all of them in the application to prevent redundant memory use. The explanation table includes the ones with the practical use in the application. The culture test records in this table are between January 1, 2005 and June 19, 2019, and

Column Name	Description	Type	Example	Distinct Count
EID	(Episode ID) A unique code identifying each hospitalization episode. An episode includes all the actions starting from hospitalization to discharge.	Integer	7814354, 7744362	124,578
AID	(Action ID) A unique code identifying each parent action related to corresponding antibiotic consumption action.	Integer	2671389, 2672224	268,254
SID	(Subaction ID) A unique code identifying each antibiotic consumption action. SID is the primary key of this table.	Integer	53259387, 53259381	985,129
GRID	(Group ID) A unique code identifying each drug group.	Integer	67, 1044	119
Active Ingredient	Name of the active ingredient of the corresponding antibiotic.	Character	LEVOFLOKSASIN	112
MID	(Medicine ID) A unique code identifying each medicine used in the hospital. The numbering is specific to hospital.	Integer	14050, 11743	307
Name	Explicit name of the medicine, consisting of brand name, dose and form.	Character	TAVANIC 500 MG 1 FLAKON	307
Date-Time	Administration time of the corresponding medicine.	Date	2018-01-01 12:35:49	-

Table 4.2: Antibiotic Administration Table

Column Name	Description	Type	Example	Distinct Count
EID	(Episode ID) A unique code identifying each hospitalization episode. An episode includes all the actions starting from hospitalization to discharge.	Integer	7722630, 7693942	63,945
AID	(Action ID) A unique code identifying each parent action related to corresponding laboratory test action.	Integer	7724100, 7694420	234,238
SID	(Subaction ID) A unique code identifying each laboratory test action. SID is the primary key of this table.	Integer	158763840, 158042036	261,307
Test ID	A unique code identifying each laboratory test type.	Integer	302884, 10049959	2
Name	Explicit name of the laboratory test type.	Character	C Reaktif Protein (CRP), Prokalsitonin (PCT)	2
Result	Result of the corresponding test.	Numeric	1.6, 80.3	-
Unit	Explicit name of the medicine, consisting of brand name, dose and form.	Character	mg/L, ng/mL	2
Normal	The suggested interval for the corresponding test result to be regarded as normal.	Character	<8.5, <5.0	-
Date-Time	Collection time of the corresponding test sample.	Date	2018-08-28 11:40:53	-

Table 4.3: Laboratory Tests Table

Column Name	Description	Type	Example	Distinct Count
EID	(Episode ID) A unique code identifying each hospitalization episode. An episode includes all the actions starting from hospitalization to discharge.	Integer	52006350, 47860875	20,147
AID	(Action ID) A unique code identifying each parent action related to corresponding culture test action.	Integer	52028149, 52025079	53,635
SID	(Subaction ID) A unique code identifying each culture test action. SID is the primary key of this table.	Integer	2215362125, 2215093399	55,358
Name	Explicit name of the culture test type.	Character	KAN KULTURU (AEROP / ANAEROP), IDRAR	5
Aerobic Culture	Result of the aerobic culture test.	Character	ECHERICHIA COLI UREDI	-
Date	Collection date of the corresponding test sample.	Date	2018-02-04	-

Table 4.4: Culture Tests Table

this table, also, is linked to the inpatients table by “EID” field, and the primary key of this table is “SID”.

4.2 System Analysis

This section presents analyses, observations and recommendations by utilizing the proposed tool on the American Hospital data.

To start with, Figure 4.1 shows the long-term changes in the antibiotic consumption trends in American Hospital. This figure is captured from “System Tracking” tab. The date range is from January 1, 2005 to May 31, 2019. The values on the figure are monthly aggregations, in other words, points on the graph illustrates the rate of inappropriate antibiotic consumption on a monthly basis. No filtering is applied on the data. CRP threshold is set to 5 mg/L and PCT threshold is set to 1 ng/mL, meaning that the test results below these values will be regarded as normal. The dashed line on the 5 percent level indicates that American Hospital aims to keep inappropriate antibiotic consumption percentage below this threshold.

This figure shows that inappropriate antibiotic consumption (IAC) rates followed a significantly decreasing trend between 2005 and 2013, and then, fluctuated around 10 percent level. Although after 2013 there are a number of months whose IAC rates are close to 5 percent level, we do not observe a stable trend.

The following Figure 4.2 illustrates the changes in the monthly antibiotic consumption amount. This information is especially important when observing the cases with very low number of observations, because as the number of observations decrease, so as the confidence in statistical inferences. In this case, however, we see a considerably high number of observations, thus any hypotheses on this data can be tested with a high confidence.

From 2005 to January 2009, monthly antibiotic administration amount is around 3000 according to the figure. After January 2009, we observe a very sharp increase in the consumption amount and the values vary roughly between 5000 and 9000. A probable cause for this sharp increase could be a significant increase in the size of

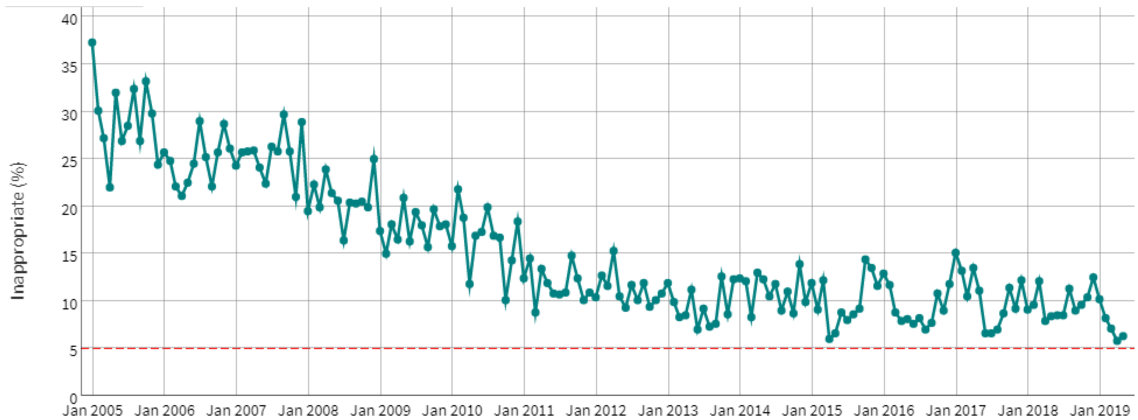


Figure 4.1: System Tracking Tab - IAC Rates

the hospital in 2009. Yet, it is not the case for American Hospital. Even though the hospital has been through various structural changes, its capacity has not changed significantly over time. See Table 4.5 for the number of hospitalization episodes per year. Note that there are a small number of episodes in 2019 since the data only covers the first half of this year.

The reason behind this immediate increase is the following. Until 2009, American Hospital had been tracking the use of selected few antibiotics. In other words, not the whole set of antibiotics had been indicated in the DMS of the hospital, thus until early 2009, the antibiotics dataset had not included those antibiotics which were not introduced to the system. After 2009, American Hospital have started to track the use of the whole set of antibiotics used in the hospital.

The following part of this chapter investigates the use of Carbapenems and Fluoroquinolones in the hospital.

Carbapenems belong to the beta-lactams class and they are used in the treatment of several gram-positive and gram-negative infections [Drugs.com, nd]. Mainly, they restrict the cell wall synthesis of the bacteria, leading bacteria to death. Carbapenems are the broadest spectrum and the strongest antibiotics, thus they are often used against more serious infections as a last resort. The four types of carbapenems are meropenems, imipenems, ertapenems and doripenems.

Year	Number of Episodes
2005	13546
2006	13191
2007	14201
2008	14852
2009	14287
2010	14527
2011	15010
2012	14683
2013	13967
2014	14118
2015	14531
2016	14347
2017	14426
2018	15365
2019	7458

Table 4.5: Number of Episodes per Year

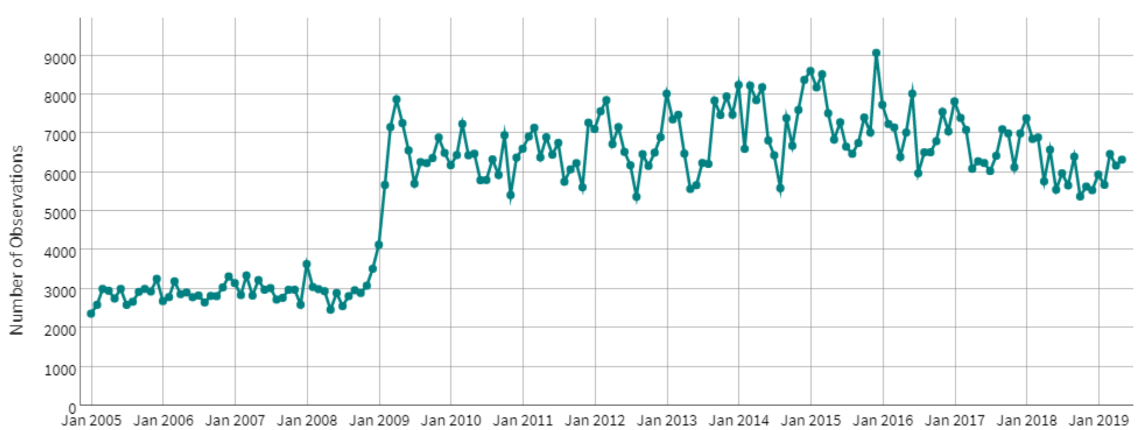


Figure 4.2: System Tracking Tab - Amount of Antibiotic Use

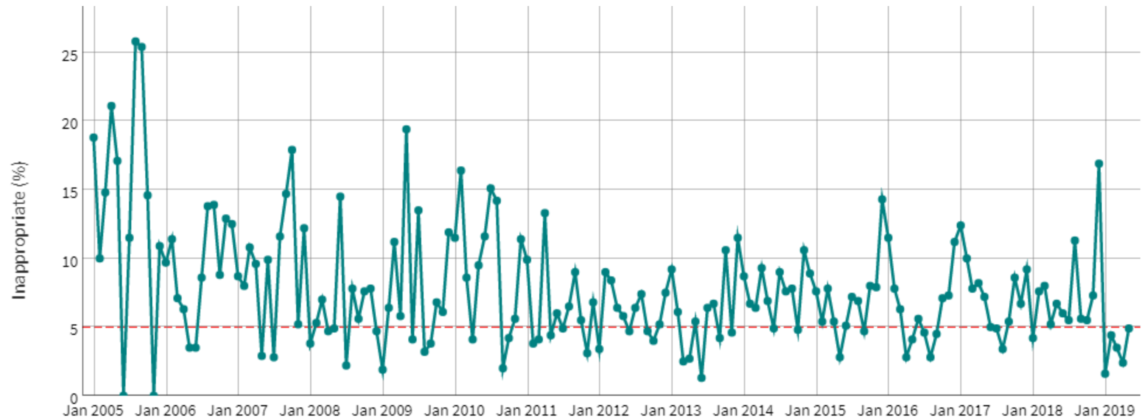


Figure 4.3: Inappropriate Meropenem Use Rates

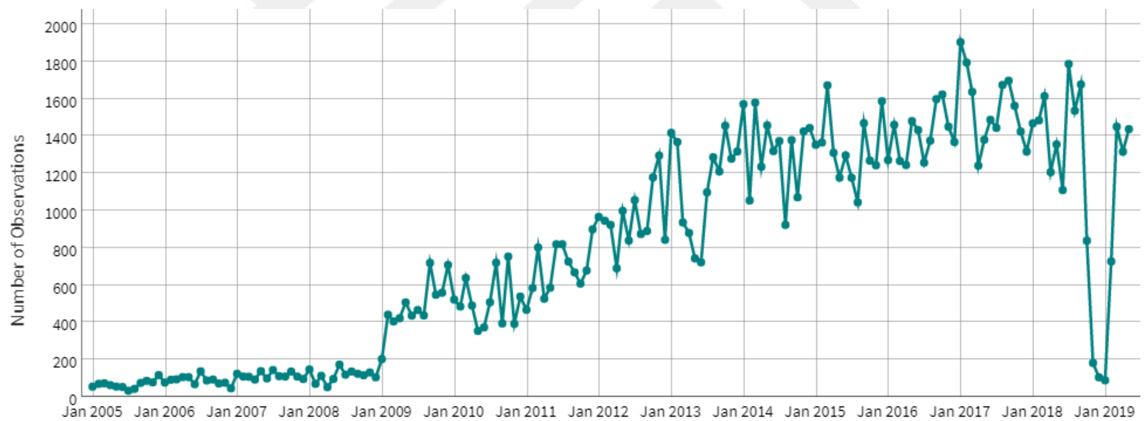


Figure 4.4: Amount of Meropenem Use

Fluoroquinolones are another class of broad-spectrum antibiotics. They inhibit the activities of DNA replication enzymes of bacteria [Fookes, 2018]. Fluoroquinolones occasionally cause serious side effects due to toxicity, and thus, they are not suggested to be used in common infections that other antibiotics can treat. Common drugs in this class are moxifloxacin, levofloxacin and ciprofloxacin.

Figure 4.3 and 4.4 show the change in the compliance and the usage amount of meropenems over time, respectively. The date range is from January 1, 2005 to May 31, 2019, and the values are monthly aggregations as in the previous example. CRP threshold is set to 5 mg/L and PCT threshold is set to 1 ng/mL. The data is filtered by setting active ingredient to be meropenem using the input panel as illustrated in Figure 3.3. Compliance rates show significant monthly fluctuations especially in the first years included in the figure. One of the reasons for this is the low number of observations in the data. As the number of observations increase, the magnitude of fluctuations decreases considerably, showing the system follows an arguably stable compliance trend.

As explained in the previous example case, American Hospital had not indicated all antibiotics in the DMS until 2009, and until then, the number of observations in figures does not reflect the real total usage amount. Therefore, the rest of the section will focus on the data starting from 2009.

Focusing on the data after 2009, we observe a flat long-term trend in the incompliance rates fluctuating around 8 percent.

Figure 4.4 shows an alarming increase in the use of meropenems between 2009 and 2017. After 2017, the increasing trend in the meropenem use seems to stop.

The figure shows an extreme decrease in the use of meropenems between late 2018 and early 2019. Such huge changes are usually the consequences of data-related problems. Discussing this issue with the DMS responsibles and ASP personnel in American Hospital, we investigated the root cause of this abnormality. First of all, we concluded that this extreme decrease is indeed a consequence of incorrect data in the DMS. The main driver of this incorrect data is the defective design of the antibiotic labeling system. This system works as follows. The pharmacy unit personnel manually label each drug according to whether it is an antibiotic or not. There are various brands offering drugs with the same active ingredient, and these drugs are used interchangeably in the hospital. Moreover, each brand may offer the same drug in multiple dose and forms. For example, Meronem, Merosid and Merozan are the three brand names of meropenem antibiotics used in the hospital and “Merozan 1 g flacon” and “Merozan 500 mg flacon” are the two dosages versions of the brand Merozan. The current system requires each of these drug names with distinct dosages to be specifically labeled as antibiotics by the pharmacy unit. Otherwise records of these drugs may not appear in the antibiotic consumption reports and other antibiotic-related studies, leading to deceptive inferences.

About the extreme decrease in the example case, ASP personnel reported that some meropenem antibiotics were not indicated as antibiotic drug in DMS, thus administration records of these antibiotics were not present in the antibiotic consumption data. However, in reality, use of meropenem antibiotics did not experience such a sharp decrease.

We recommend American Hospital to employ an automated system that detects and labels antibiotics by their active ingredients. This way, without manual intervention, any new drug containing these ingredients (e.g., meropenem) will be automatically registered to the DMS as an antibiotic. The accuracy of this data is particularly important for proper interventions and correct assessment of ASPs.

This example showed that data quality is very important and thus, it may lead to false conclusions. However, we will investigate a couple of more cases using this data, because one of the main objectives of this section is to exemplify the use cases of the proposed application.

In the next example scenario, we investigate the change in the consumption amount of selected fluoroquinolones, namely moxifloxacin, ciprofloxacin, and levofloxacin. In order to observe the consumption trends of these three drugs simultaneously, we utilize "filter" and "investigate" features in system tracking tab together. We filter the data as antibiotics containing maxifloxacin, ciprofloxacin, and levofloxacin, and set the investigation parameter as active ingredient as shown in Figure 4.5. The date range is from January 1, 2009 to May 31, 2019, and the values are monthly aggregations as in the previous examples. CRP threshold is set to 5 mg/L and PCT threshold is set to 1 ng/mL.

Figure 4.6 shows the consumption trends of selected fluoroquinolones. The figure indicates ciprofloxacins, moxifloxacin and levofloxacin in blue, purple and green colors, respectively. Comparing the total monthly fluoroquinolone consumption between early 2010s and late 2010s, we observe a significant decrease. The average monthly

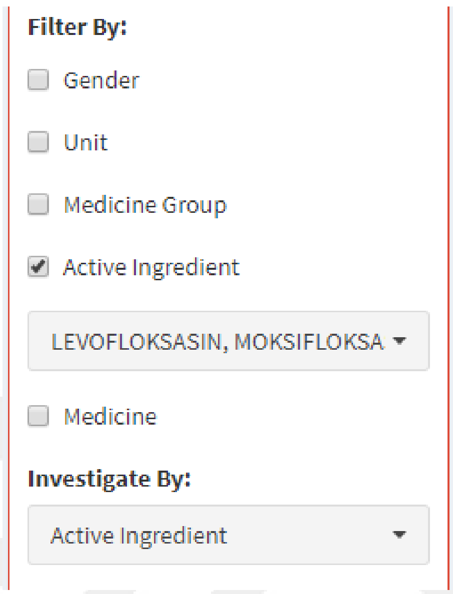


Figure 4.5: Input Panel in Fluoroquinolones Example

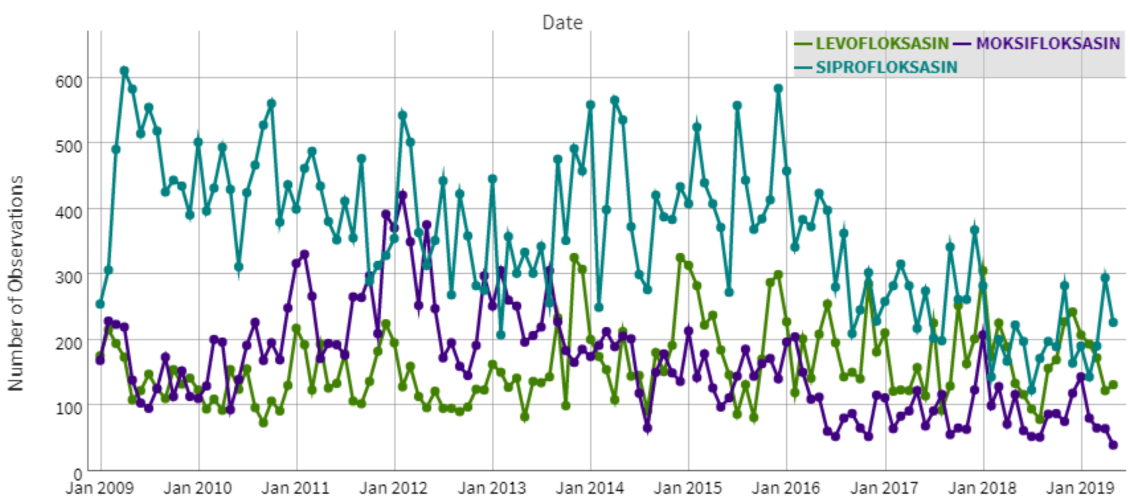


Figure 4.6: Amount of Fluoroquinolone Use

fluoroquinolon consumption in 2010–2012 period was 767.19 whereas it decreased to 488.31 in 2017–2019 period.

Also, as is evident from the figure, the use of ciprofloxacins has decreased significantly after 2016. Figure 4.7 shows the corresponding interrupted time-series graph, and the red lines on the figure indicate the best fit lines of pre– and post–intervention periods i.e., before and after the turning point in 2016.

Use of moxifloxacins, also, follows a steady decrease after 2012. The corresponding time-series is illustrated in Figure 4.8. The same technique with ciprofloxacins is employed here, as well. The figure shows that the increasing trend in the number of moxifloxacin administration is reversed after 2012.

On the other hand, the use of levofloxacins shows a slightly increasing trend in the long term (see Figure 4.9).

The next example demonstrates the use of alerts tab of the application. In this tab, the application displays the inappropriate antibiotic use cases on a selected date. The main purpose of this tab is to list the potential misuse cases, to provide information support to the medical experts. Physicians check this list on a daily basis, reassess the cases in the list, and discontinue the ones which they specify as, in fact, inappropriate.

Figure 4.10 shows the potentially inappropriate antibiotic use cases on a selected date. The date is chosen as March 2nd, 2019, and no filter is applied on the data. CRP threshold is set to 5 mg/L and PCT threshold is set to 1 ng/mL. The application indicates five of the antibiotic uses as inappropriate on the selected date. Users can filter these results using either the input panel or the filter boxes on top of the result table. For example, users can filter the results by the corresponding unit, and check the cases which they are responsible of. Figure 4.11 shows the results filtered by the unit “KAT 6 B”.

The results show three potential misuse cases in the selected unit. By chance, all of these records belong to the same patient. Utilizing the patient tracking tab, users can examine the personal health records of this patient, as described below.

Figure 4.12 shows the episode information of this patient. Previous hospitalization

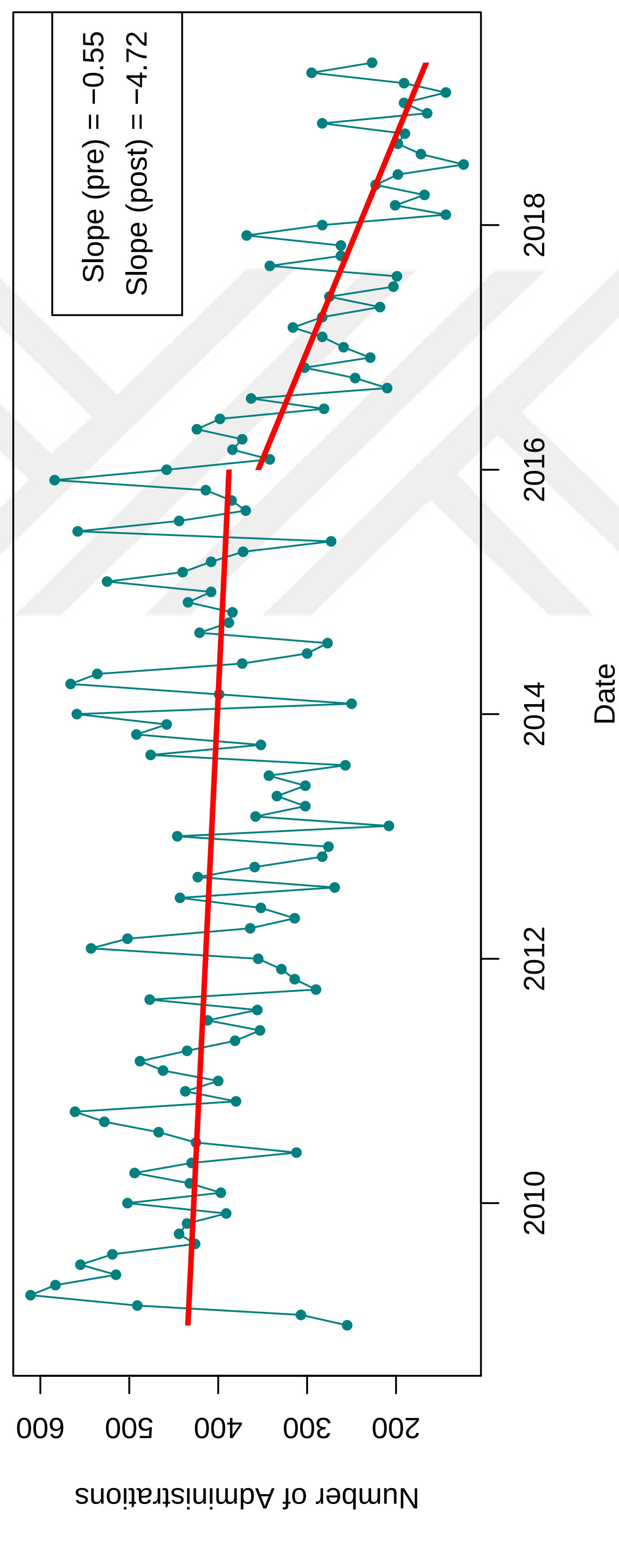


Figure 4.7: Ciprofloxacin Use

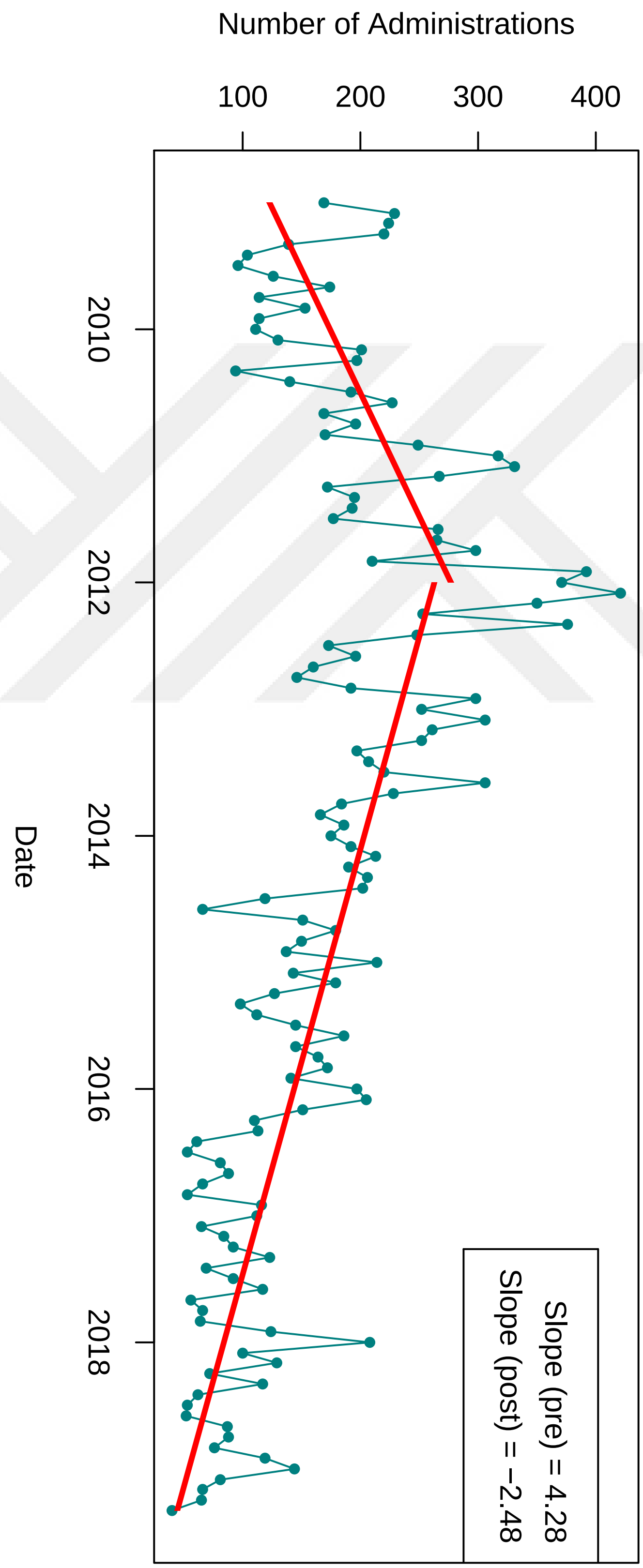


Figure 4.8: Moxifloxacin Use

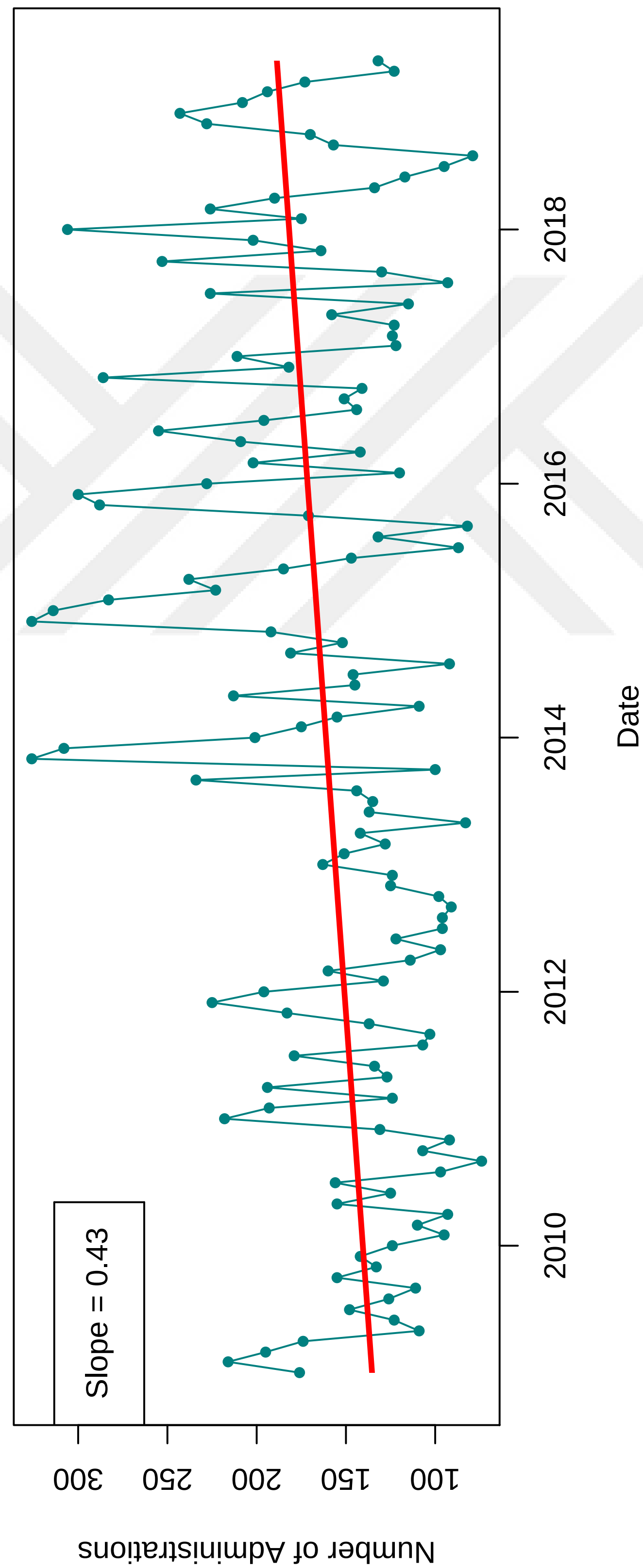


Figure 4.9: Levofloxacin Use

Show All entries

Search:

PID	EID	Date	Time	Gender	Floor	GRID	Active Ingredient	Name	Proper
All	All	All	All	All	All	All	All	All	All
73733	52143426	2019-03-02	08:59:06	FEMALE	KAT 4 B	1062	OSELTAMIVIR FOSFAT	ENFLUVIR 75 MG '' 10 KAPSUL ''	0
182420	52071716	2019-03-02	20:12:45	MALE	KAT 6 B	1025	MEROPENEM	MEROSID (I.V) 1000 MG FLAKON	0
182420	52071716	2019-03-02	12:07:19	MALE	KAT 6 B	1025	MEROPENEM	MEROSID (I.V) 1000 MG FLAKON	0
182420	52071716	2019-03-02	01:56:57	MALE	KAT 6 B	1025	MEROPENEM	MEROSID (I.V) 1000 MG FLAKON	0
933156	52228849	2019-03-02	05:46:34	FEMALE	KAT 4 A	1038	AMPISILIN + SULBAKTAM	SULCID IM/IV 1 GR 1 FLAKON	0

Showing 1 to 5 of 5 entries

Previous 1 Next

Download Table

Figure 4.10: Alerts Tab

Show All entries

Search:

PID	EID	Date	Time	Gender	Floor	GRID	Active Ingredient	Name	Proper
All	All	All	All	All	["KAT 6 B"]	All	All	All	All
182420	52071716	2019-03-02	20:12:45	MALE	KAT 6 B	1025	MEROPENEM	MEROSID (I.V) 1000 MG FLAKON	0
182420	52071716	2019-03-02	12:07:19	MALE	KAT 6 B	1025	MEROPENEM	MEROSID (I.V) 1000 MG FLAKON	0
182420	52071716	2019-03-02	01:56:57	MALE	KAT 6 B	1025	MEROPENEM	MEROSID (I.V) 1000 MG FLAKON	0

Showing 1 to 3 of 3 entries (filtered from 5 total entries)

Previous 1 Next

Download Table

Figure 4.11: Alerts Tab - Unit Filter

Episode Information

PID	EID	Floor ID	Floor	Gender	Hospitalization Date	Discharge Date
182420	51635910	140009967	KAT 6 B	MALE	2019-01-06	2019-01-13
182420	52071716	140009967	KAT 6 B	MALE	2019-02-11	2019-03-03

Figure 4.12: Patient Tracking Tab - Episode Information

episodes are also included in this table since we entered the patient ID from the input panel and left episode ID input intentionally blank as shown in Figure 4.13. In this way, the application returns previous health records of the specified patient, as it is usually valuable to consider the health history of patients. However, users can also choose to display only the current episode by entering the corresponding episode ID from the input panel.

Figure 4.14 shows the laboratory test results of the specified patient. CRP results are shown as blue dots and PCT results are shown as dark red dots. Left y axis of the graph indicates CRP values and right y axis indicates PCT values. The group of points on the left-hand side (i.e., between January 6th and January 13th) belong to the first hospitalization episode, and the group of points on the right hand side (i.e., between February 11th and March 3rd) belong to the second hospitalization episode. These test results are also shown in a separate table in detail (see Figure 4.15). In this table, the test results exceeding the specified thresholds are highlighted with red color.

Also, there are red, green and gray ribbons at the bottom of the graph. The red ribbons indicate the dates with inappropriate antibiotic use while green ribbons indicate the dates with appropriate use. The gray ribbons indicate that there is not enough evidence for any conclusion for the corresponding dates. These ribbons are in sync with the highlighting colors in the antibiotic use information table shown in Figure 4.16. This table includes all the antibiotic use records for the specified patient in detail.

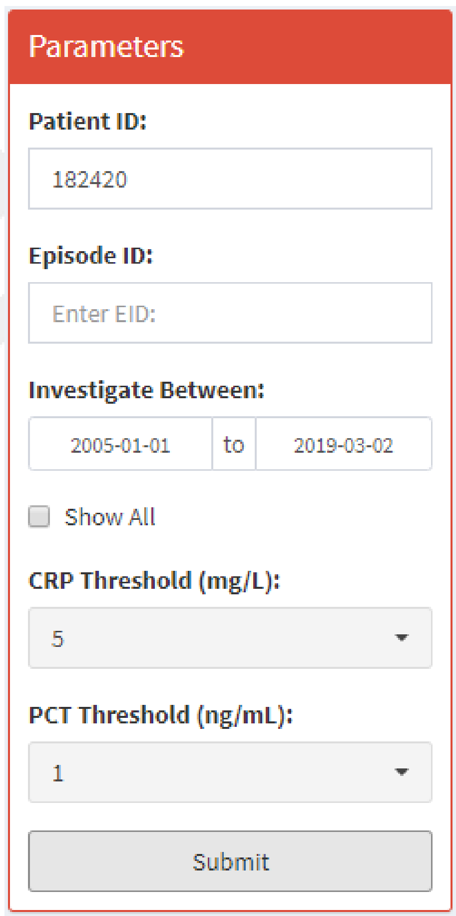


Figure 4.13: Patient Tracking Tab - Input Panel

Figure 4.17 shows the culture test results of the specified patient. Both test result belong to the previous hospitalization episode of this patient. Thus, the application will not be able to employ culture test information in antibiotic use assessment.

Utilizing the information provided by these tables and graph, users can reassess the antibiotic use cases specified in the alert tab and discontinue the use when necessary. In our example case (see Figure 4.11), the application marks these records as inappropriate since the test results are in normal levels (i.e., CRP < 5 mg/L and PCT < 1 ng/mL) on March 2nd, 2019. Also, there is no culture result information to imply otherwise. Therefore, the application interprets these antibiotic uses as potentially unnecessary.

Since American Hospital data does not include health records other than test results, the application merely considers the test results to determine the appropriateness of antibiotic uses. However, in fact, laboratory test results alone do not provide enough evidence to draw conclusions about the appropriateness of antibiotic use. Ideally, other medical information such as fever and antibiotic resistance must be considered in antibiotic use assessment, as well. Otherwise, the application is likely to draw conclusions with high sensitivity and imperfect specificity. In other words, the application is likely to give false alarms.

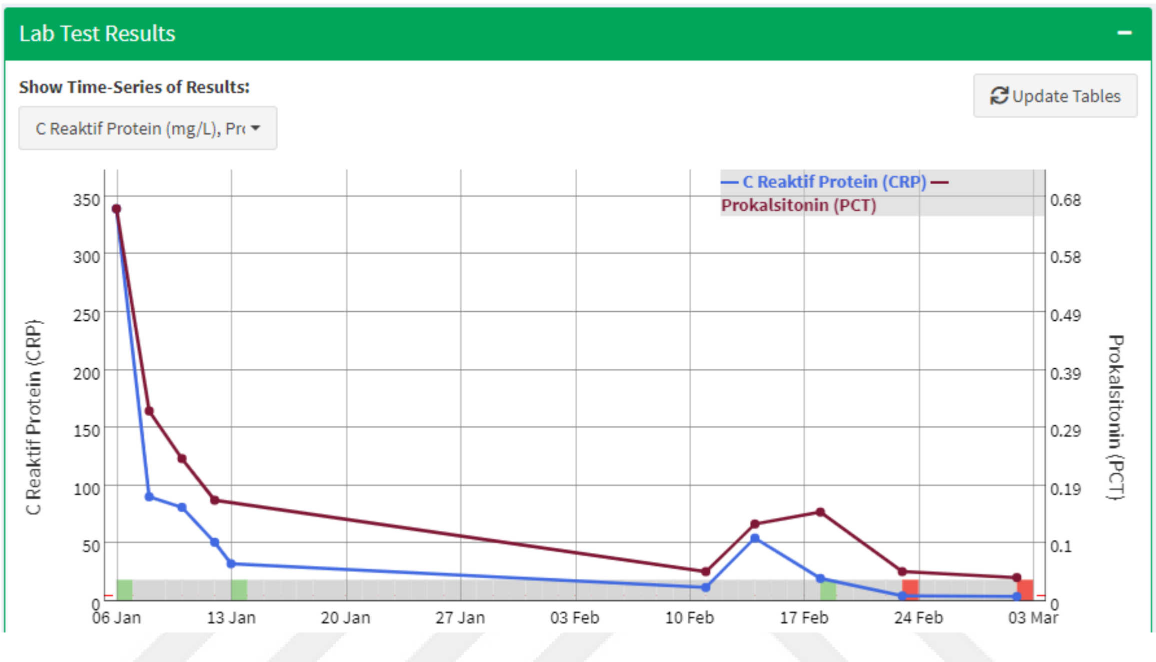
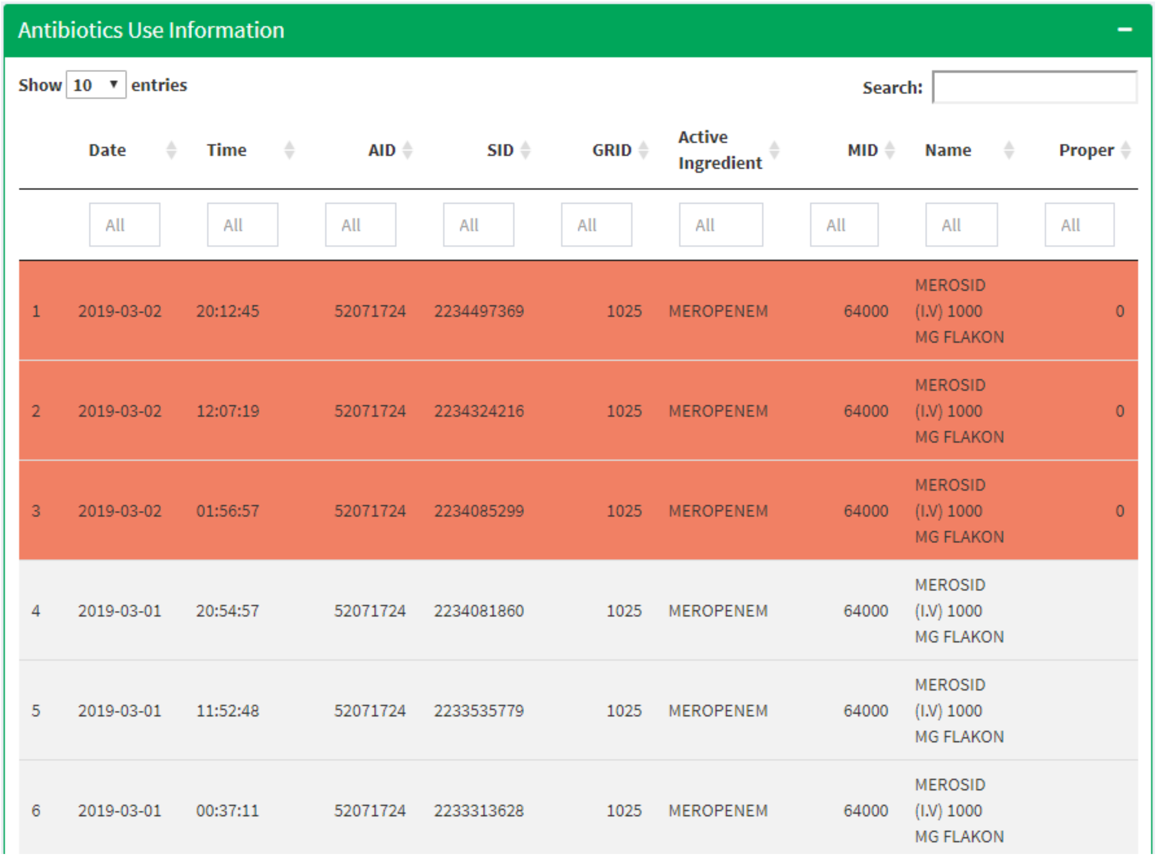


Figure 4.14: Patient Tracking Tab - Lab Test Results Graph

Show 10 entries Search:

	Date	Time	AID	SID	Name	Result	Unit	Normal	Need AB
	All	All	All	All	All	All	All	All	All
1	2019-03-02	05:42:19	52299120	2234090094	C Reaktif Protein (CRP)	4.1	mg/L	5	0
2	2019-03-02	05:42:19	52299120	2234090100	Prokalsitonin (PCT)	0.0	ng/mL	1	0
3	2019-02-23	21:08:15	52226832	2229358925	C Reaktif Protein (CRP)	4.7	mg/L	5	0
4	2019-02-23	21:08:15	52226832	2229358931	Prokalsitonin (PCT)	0.1	ng/mL	1	0
5	2019-02-18	08:30:56	52152895	2224032895	C Reaktif Protein (CRP)	19.8	mg/L	5	1
6	2019-02-18	08:30:56	52152895	2224033219	Prokalsitonin (PCT)	0.1	ng/mL	1	0

Figure 4.15: Patient Tracking Tab - Lab Test Results Table

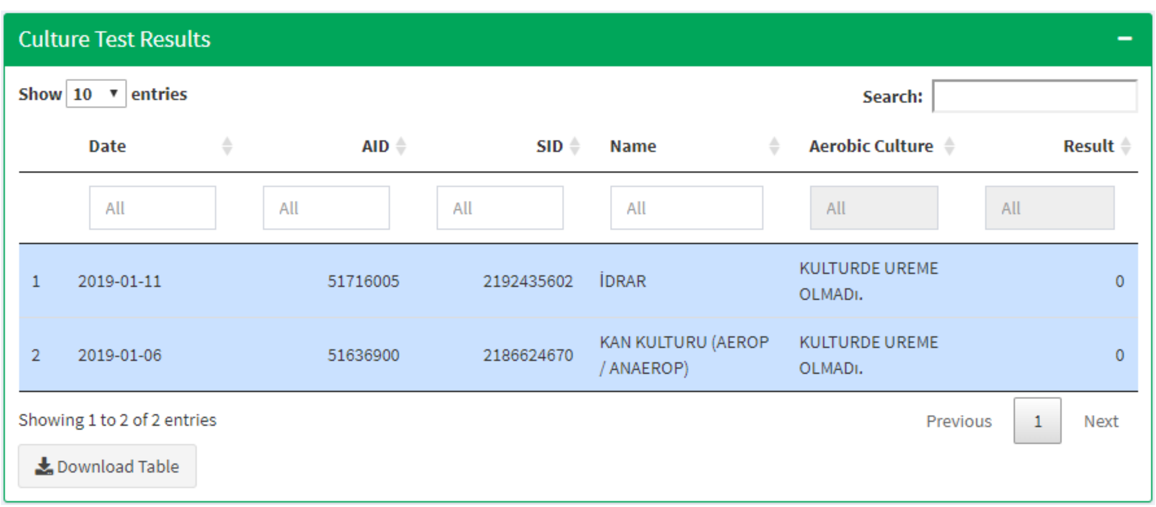


Antibiotics Use Information

Show 10 entries Search:

	Date	Time	AID	SID	GRID	Active Ingredient	MID	Name	Proper
	All	All	All	All	All	All	All	All	All
1	2019-03-02	20:12:45	52071724	2234497369	1025	MEROPENEM	64000	MEROSID (I.V) 1000 MG FLAKON	0
2	2019-03-02	12:07:19	52071724	2234324216	1025	MEROPENEM	64000	MEROSID (I.V) 1000 MG FLAKON	0
3	2019-03-02	01:56:57	52071724	2234085299	1025	MEROPENEM	64000	MEROSID (I.V) 1000 MG FLAKON	0
4	2019-03-01	20:54:57	52071724	2234081860	1025	MEROPENEM	64000	MEROSID (I.V) 1000 MG FLAKON	
5	2019-03-01	11:52:48	52071724	2233535779	1025	MEROPENEM	64000	MEROSID (I.V) 1000 MG FLAKON	
6	2019-03-01	00:37:11	52071724	2233313628	1025	MEROPENEM	64000	MEROSID (I.V) 1000 MG FLAKON	

Figure 4.16: Patient Tracking Tab - Antibiotic Use



Culture Test Results

Show 10 entries Search:

	Date	AID	SID	Name	Aerobic Culture	Result
	All	All	All	All	All	All
1	2019-01-11	51716005	2192435602	İDRAR	KULTURDE UREME OLMADı.	0
2	2019-01-06	51636900	2186624670	KAN KULTURU (AEROP / ANAEROP)	KULTURDE UREME OLMADı.	0

Showing 1 to 2 of 2 entries Previous 1 Next

Download Table

Figure 4.17: Patient Tracking Tab - Culture Test Results

Chapter 5

CONCLUSION

In this thesis, we introduce a new application for antibiotic consumption monitoring. This application allows easy, fast, and in-depth investigation of antibiotic consumption trends in health centers without the need for technical knowledge. Users access to the system from their web browsers and set the specifications they are interested in through the user interface of the application. The application displays the information of interest embedded in interactive graphs and tables, providing ease of interpretation.

The main contribution of this thesis study is providing a real-time antibiotic consumption tracking tool which allows in depth investigations. Health centers running antibiotic stewardship programs, as the common practice, keep track of daily (or monthly) numbers of essential antibiotic use statistics in the aggregate level. These numbers mainly contribute to long-term investigations in the hospital and national level while they have little contribution to short-term and more practical stewardship activities in health centers. The proposed application allows stewardship personnel to study antibiotic consumption trends in various detail levels. This is especially valuable since it helps users to conduct root cause analyses and perform timely and accurate interventions. On the other hand, the tool is designed to provide patient level information (e.g., laboratory test results and information about other medical actions) to the prescribers, as well. The way it conveys this information is aimed to be as easy and as quick to interpret as possible. This design choice makes the proposed tool more useful than typical data management systems already employed in health centers.

Another contribution with practical importance is providing an antibiotic consumption tracking tool which is free to use yet advanced at the same time. In Turkey, health centers in general cannot afford investing in already available commercial antibiotic stewardship products due to significant licensing costs. Thus, one of the design criteria of the proposed application was utilizing open-source technologies. Removing the financial barriers, the tool is intended to be desirable by health centers.

The proposed application is implemented using data from American Hospital of İstanbul to show its capabilities in a real-life scenario. During the analyses, the application helped us easily reveal that the data is incomplete. We contacted with the hospital personnel and verified that there are indeed missing antibiotic administration records in the data. The personnel reported that the proposed tool would be helpful in early detection and correction of such problems with the data. More importantly, the tool allowed us to easily detect both favorable and dangerous long-term trends in antibiotic consumption which are verified by the professionals, as well. In addition, the tool showed its potential in providing information support to prescribers/physicians with their prescribing decisions. Since it provides a list of patients with potentially inappropriate antibiotic use in real-time, it can contribute to direct antibiotic management in the patient level, as well.

The main natural limitation of the application is that it is highly dependent on the information provided by the health center. In other words, as the variety and accuracy of the information that flows into the tool decrease, the tool will be more likely to produce misleading outputs. Thus, a comprehensive and accurate information supply is arguably the most important requirement of the tool.

5.1 Future Work

Although the proposed tool is equipped with many functionalities and it fulfills the scope of this thesis study, there is still margin for improvement as listed below.

The proposed tool was not designed to be utilized in infection control studies, so it has no modules to investigate the trends in hospital infections. However, typically, antibiotic stewardship and infection control activities are linked together. Thus, an infection control module can be added to the system or the tool can be integrated with already available infection control tools.

Alerts can be emailed to responsible users, which further puts real-time utilization of the tool into practice. Also, an automated periodical report generation module can be valuable for the health centers which intend to track a list of predefined subgroups of the data repeatedly.

BIBLIOGRAPHY

[Agwu et al., 2008] Agwu, A., Lee, C., Jain, S., Murray, K., Topolski, J., Miller, R., Townsend, T., and Lehmann, C. (2008). A World Wide WebBased Antimicrobial Stewardship Program Improves Efficiency, Communication, and User Satisfaction and Reduces Cost in a Tertiary Care Pediatric Medical Center. *Clinical Infectious Diseases*.

[Allen, 2014] Allen, J. (2014). Shiny Server - Introduction. `https://shiny.rstudio.com/articles/shiny-server.html`.

[American Hospital, nd] American Hospital (n.d.). History. `https://www.amerikanhastanesi.org/history`.

[CDC, 2019] CDC (2019). 2019 AR Threats Report. `https://www.cdc.gov/DrugResistance/Biggest-Threats.html`.

[Davey et al., 2017] Davey, P., Marwick, C. A., Scott, C. L., Charani, E., Mcneil, K., Brown, E., Gould, I. M., Ramsay, C. R., and Michie, S. (2017). Interventions to improve antibiotic prescribing practices for hospital inpatients.

[Doron and Davidson, 2011] Doron, S. and Davidson, L. E. (2011). Antimicrobial stewardship. *Mayo Clinic proceedings*, 86(11):1113–1123.

[Drugs.com, nd] Drugs.com (n.d.). Carbapenems. `https://www.drugs.com/drug-class/carbapenems.html`.

[Elastic, 2019] Elastic (2019). Jdbc input plugin - Logstash Reference [7.5]. `https://www.elastic.co/guide/en/logstash/current/plugins-inputs-jdbc.html`.

[Elastic, nd] Elastic (n.d.). What is Elasticsearch. `https://www.elastic.co/what-is/elasticsearch`.

[FAO et al., 2017] FAO, OIE, and WHO (2017). Global framework for development & stewardship to combat antimicrobial resistance - draft roadmap. *OIE World Organization for Animal Health*.

[Fookes, 2018] Fookes, C. (2018). Quinolones. `http://www.drugs.com/drug-class/quinolones.html`.

[Glowacki et al., 2003] Glowacki, R. C., Schwartz, D. N., Itokazu, G. S., Wisniewski, M. F., Kieszkowski, P., and Weinstein, R. A. (2003). Antibiotic Combinations with Redundant Antimicrobial Spectra: Clinical Epidemiology and Pilot Intervention of Computer-Assisted Surveillance. *Clinical Infectious Diseases*.

[Hassouna and Deshpande, 2018] Hassouna, H. and Deshpande, A. (2018). Antimicrobial Stewardship, Volume 2. *Clinical Infectious Diseases*, 66(11):1822.

[HL7 International, nd] HL7 International (n.d.). Introduction to HL7 Standards. `https://www.hl7.org/implement/standards/`.

[Imperial College London, nd] Imperial College London (n.d.). Imperial Antibiotic Prescribing Policy (IAPP). `https://www.imperial.ac.uk/medicine/hpru-amr/applications-and-tools/imperial-antibiotic-prescribing-policy-iapp/`.

[Kose et al., 2008] Kose, I., Akpinar, N., Gurel, M., Arslan, Y., Ozer, H., Yurt, N., Ozcam, A., Kirici, S., Sabur, E., Kabak, Y., Yuksel, M., and Dogac, A. (2008). Turkey's National Health Information System (NHIS). In *e-Challenges*.

[Marquardt, 2019] Marquardt, A. (2019). How to keep Elasticsearch synced with a RDBMS using Logstash and JDBC. `https://www.elastic.co/blog/how-to-keep-elasticsearch-synchronized-with-a-relational-database-using-logstash`.

[McGregor et al., 2006] McGregor, J. C., Weekes, E., Forrest, G. N., Standiford, H. C., Perencevich, E. N., Furuno, J. P., and Harris, A. D. (2006). Impact of a Computerized Clinical Decision Support System on Reducing Inappropriate Antimicrobial Use: A Randomized Controlled Trial. *Journal of the American Medical Informatics Association*.

[MedMined, nd] MedMined (n.d.). Infection prevention surveillance. `https://www.bd.com/en-us/offerings/brands/medmined`.

[MySQL, nd] MySQL (n.d.). Date and Time Functions - MySQL 5.5 Reference Manual. `https://dev.mysql.com/doc/refman/5.5/en/date-and-time-functions.html`.

[O'Neill, 2016] O'Neill, J. (2016). Review on Antimicrobial Resistance. Tackling drug-resistant infections globally. *Who*.

[Sanford Guide, nd] Sanford Guide (n.d.). Antimicrobial Therapy App. `https://www.sanfordguide.com/products/digital-subscriptions/sanford-guide-to-antimicrobial-therapy-mobile/`.

[Schiff et al., 2001] Schiff, G. D., Wisniewski, M., Bult, J., Parada, J. P., Aggarwal, H., and Schwartz, D. N. (2001). Improving inpatient antibiotic prescribing: insights from participation in a national collaborative. *The Joint Commission journal on quality improvement*.

[Sentri7, nd] Sentri7 (n.d.). Real-Time Clinical Surveillance. `https://www.pharmacyonesource.com/products/sentri7/`.

[TheraDoc, nd] TheraDoc (n.d.). Clinical Surveillance. `https://www.theradoc.com/`.

[Van Den Bosch et al., 2015] Van Den Bosch, C. M., Geerlings, S. E., Natsch, S., Prins, J. M., and Hulscher, M. E. (2015). Quality indicators to measure appropriate antibiotic use in hospitalized adults. *Clinical Infectious Diseases*.

[Vecna Healthcare, nd] Vecna Healthcare (n.d.). Clinical Surveillance. `https://vecnahealthcare.com/platform/clinical-surveillance/`.

[WHO, 2018] WHO (2018). Antibiotic Resistance. `https://www.who.int/news-room/fact-sheets/detail/antibiotic-resistance`.

[WHO, 2019] WHO (2019). *Antimicrobial stewardship programmes in health-care facilities in low-and middle-income countries: A practical toolkit*. World Health Organization.

[Zhang et al., 2013] Zhang, L., Huang, Y., Zhou, Y., Buckley, T., and Wang, H. H. (2013). Antibiotic administration routes significantly influence the levels of antibiotic resistance in gut microbiota. *Antimicrobial Agents and Chemotherapy*.