ABX Ninja
Project Report

Group 16
Katie Hochberg
Allie Sanzi

Mentors
Dr. Jennifer Townsend
Michael Cohen
Andrew Hinton

Table of Contents

1. Technical Summary

1.1 Background

Centers for Disease Control and Prevention estimates that up to 50% of all the antibiotics prescribed for people are not needed or are not optimally effective as prescribed.[1] This overuse of antibiotics is a dangerous issue facing healthcare in the United States and other countries around the world. It leads to antibiotic resistance, meaning that the usual treatments for infections are no longer effective, and alternative treatments must be used. These alternative treatments are more expensive, less effective, and/or result in a greater cost to the patient.[1]

1.2 Problem

Antimicrobial stewardship programs have attempted to mitigate the overprescription of antibiotics through standard guidelines.[2] At Johns Hopkins Hospital, these guidelines are not convenient for use in the clinical setting, and there are multiple barriers to use. To be adopted into the clinical workflow, an electronic support system is needed that provides an assessment based on the patient's symptoms and a recommendation based on the antibiotic guidelines.

1.3 Approach

To improve the accuracy of antibiotic prescription, ABX Ninja assists healthcare providers by making an appropriate antibiotic recommendation for patients based on their clinical history, symptoms, vital signs, lab results, and image findings.

The main users of the application are healthcare providers and antimicrobial stewardship administrators. Healthcare providers can enter patient data to receive a patient assessment and antibiotic recommendation. In addition, they can report whether they agreed with the assessment and why and whether they used the recommendation. This feedback will be valuable to antimicrobial stewardship administrators when seeking FDA approval for the application. Antimicrobial stewardship administrators can edit recommendations based on antibiotic availability in their location and view usage data for their institution. Usage data is displayed based on the feedback given by healthcare providers using the application.

Currently, ABX Ninja supports three types of infections: Soft Skin and Tissue Infection, Urinary Tract Infection, and Respiratory Infection. Our advisor, Dr. Jenny Townsend, developed decision trees for these three infections that healthcare providers follow when diagnosing a suspected infection patient. The recommendations are derived from Johns Hopkins antibiotic guidelines and Infectious Diseases Society of America (IDSA) guidelines. These decision trees informed the underlying database structure and flow of our application.

Backend

The backend of the application was built using NodeJS and ExpressJS. We built out a RESTful API to interact with our PostgreSQL database and to authenticate users with JHED authentication. Initially, we planned to include integration with EPIC to pull certain patient information into the application to reduce healthcare provider input. However, this created a large dependency on the EPIC team which was not met prior to the end of this project. Instead, we designed the backend such that integrating with EPIC will not require any redesign.

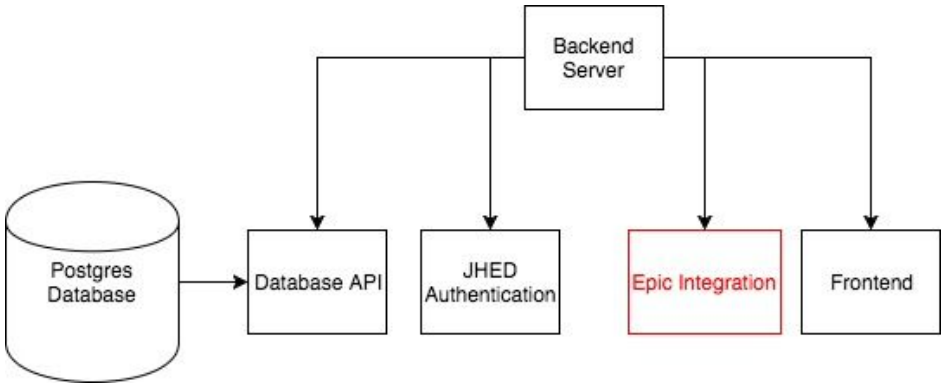

Figure 1: Component diagram to show the interaction of the various components used to create the ABX Ninja application.

The PostgreSQL database holds all of the data for the application, including the institutions (currently only supporting Johns Hopkins), users, encoded decision trees, and feedback data from the healthcare providers.

In order to encode the decision trees, we used four database tables (see Figure 2 for schema diagram):

1. Tree - currently has only three entries which correspond to the three infections we support (Soft Skin and Tissue, Urinary Tract, and Respiratory).
2. Node - contains the information for each decision point in the tree. Each node has a threshold score that must be met to take the “yes” path. If that threshold is not met, then the “no” path will be taken. The threshold score is calculated using a formula encoded in each node that uses the information entered by the healthcare provider.
3. Factors - holds the information for each of the inputs entered by the provider. These factors have a corresponding type to determine which input screen they are displayed on (clinical history, symptoms, vital signs and labs, or image findings), and an input type (multiple choice or text box).
4. Result Node - holds the assessments and recommendations for each outcome of each infection. These result nodes are stored separately from regular nodes because they can be modified for each institution. Storing these separately reduces the amount of space required by the database to store decision tree information.

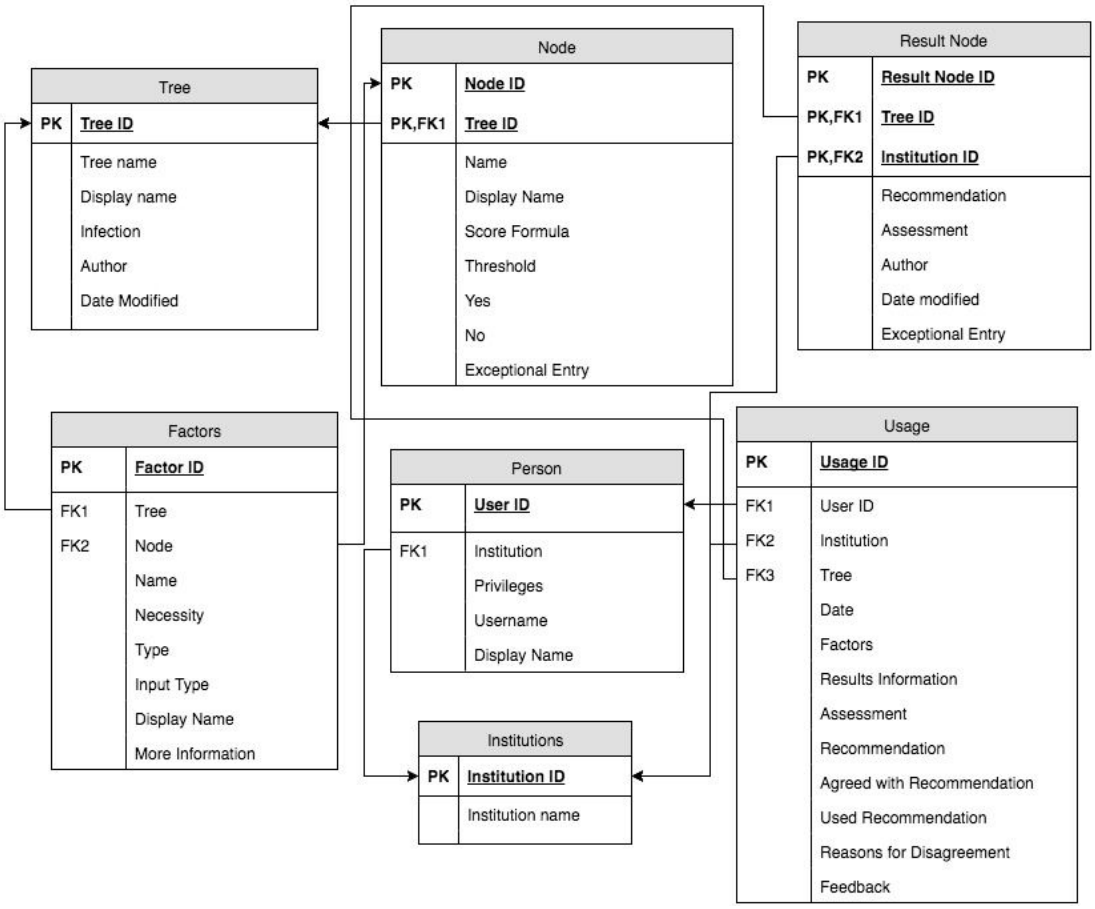

Figure 2: Database schema diagram which shows the contents of each table in the database and the relations between the tables.

Frontend

The frontend of the web application was built using AngularJS and Bootstrap. AngularJS is responsible for making the frontend reactive, while Bootstrap was used to enhance the user interface. We used wireframes provided by our team for guidance on the structure of the frontend. However, we did modify some aspects of the frontend design due to modified features and feedback received throughout the development process. After modifications were made, we compiled a frontend design document including screenshots of the application. This was sent to our mentors and shown to a physician. Feedback from this document was used to enhance the user interface.

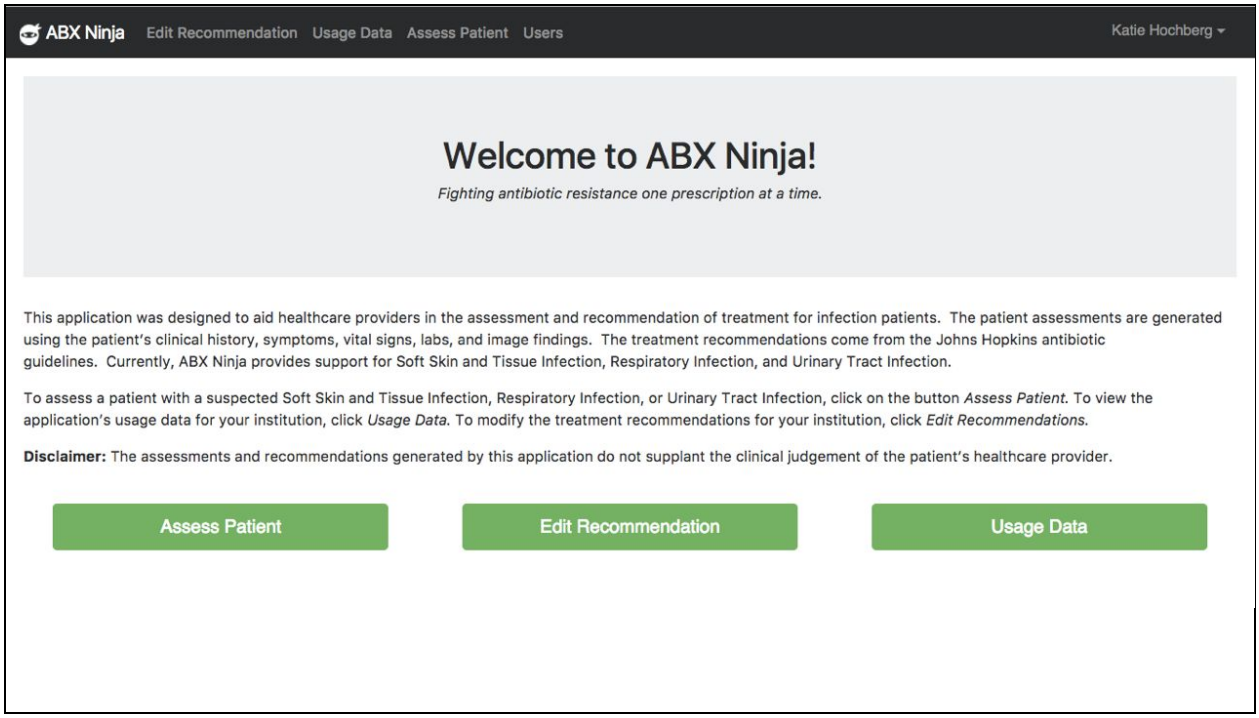

Figure 3: The ABX Ninja homepage displays all available actions for the current user.

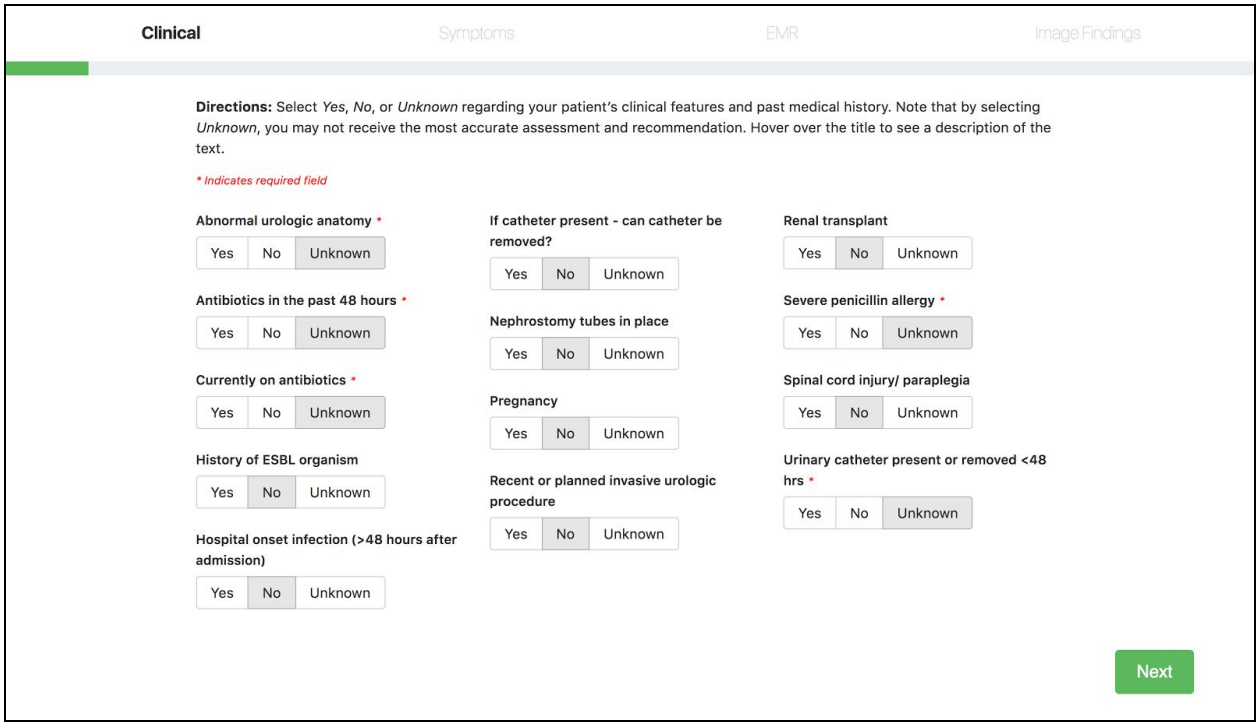

Figure 4: When the user chooses to assess a patient, they will be brought to the data collection pages in the following order: clinical, symptoms, EMR, and image findings.

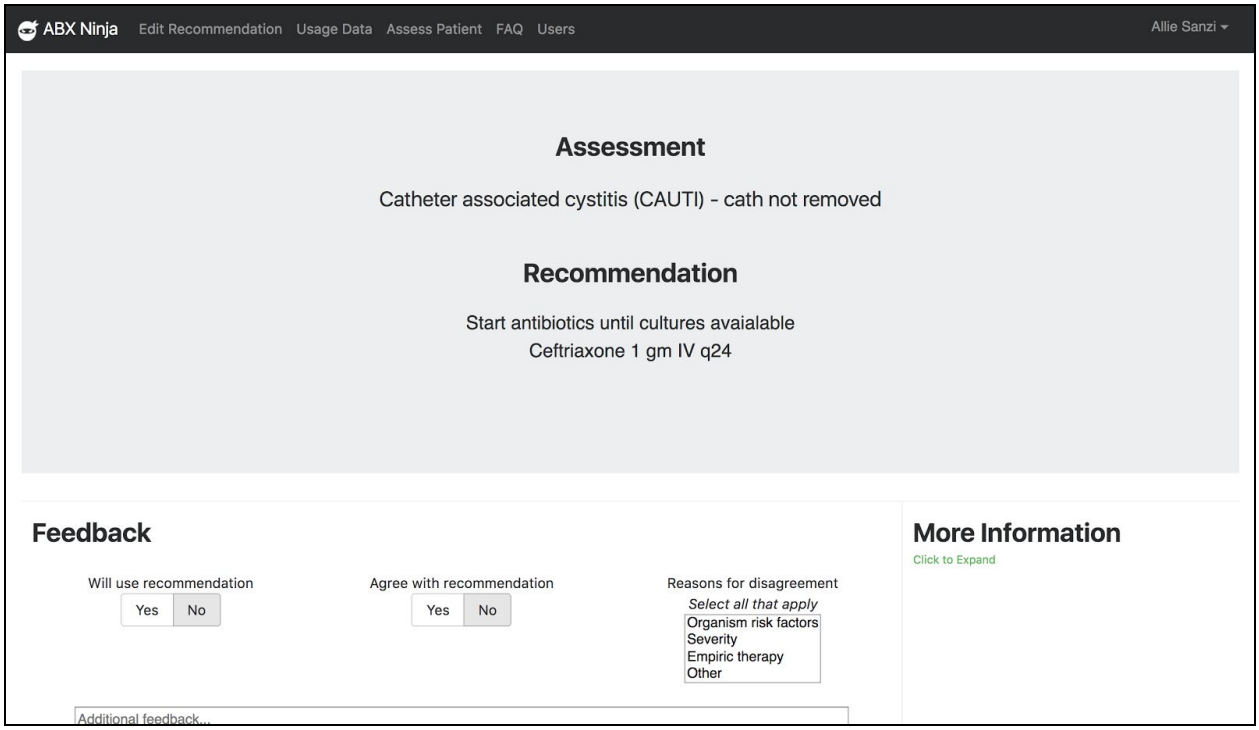

Figure 5: Once the user has finished entering all patient information, they will be brought to the "Assessment" page which shows the patient assessment and antibiotic recommendation and allows the user to enter feedback and see more information about the assessment process.

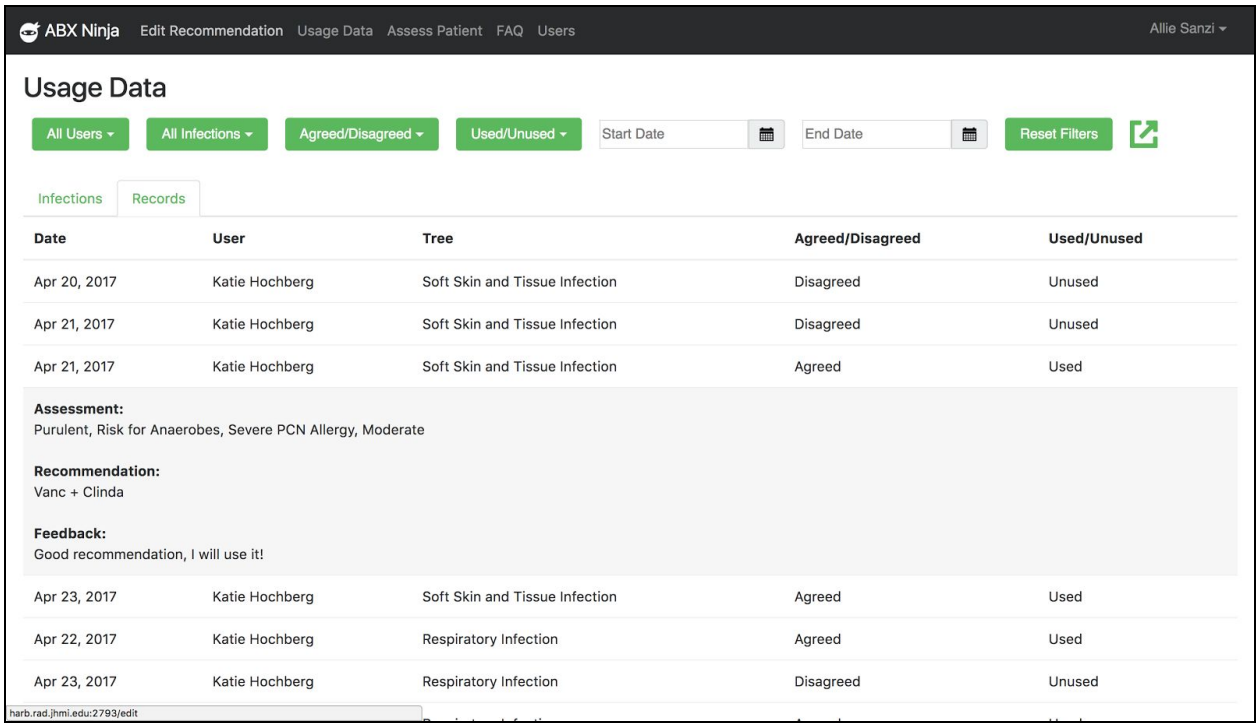

Figure 6: The usage data page summarizes the feedback provided by providers on the Assessment page.

1.4 Results

We created a minimum viable product (MVP) that is ready for testing at the Bayview Medical center. The core features that we planned to implement, including assessing a patient, editing a recommendation, and viewing usage data, are complete and tested. We have successfully encoded three decision trees provided by our mentor: Soft Skin and Tissue Infection, Urinary Tract Infection, and Respiratory Infection. Throughout the project, we independently researched features for success in the acceptance and adoption of a clinical decision support system. These features were taken into consideration when building the application to give ABX Ninja the best chance for acceptance and adoption into the clinical workflow.

1.5 Significance

Antibiotic overuse is a significant issue in the United States. Antibiotic stewardship programs across the country have developed guidelines to standardize antibiotic prescription, but in their current form, there are many barriers to use. ABX Ninja uses these guidelines in addition to the patient's clinical history, symptoms, lab results, vital signs, and image findings to make an antibiotic recommendation. Each patient that presents with the same clinical signs and symptoms will be treated with the same antibiotics, and patients that do not require antibiotics will receive a corresponding recommendation. By helping clinicians make standard and safe prescriptions, ABX Ninja will be able to reduce the amount of antibiotics prescribed, decrease antibiotic resistance across the country, and save lives.

2. Management Summary

2.1 Responsibilities

We used paired programming to complete much of the backend and frontend implementation. Towards the end of the project, Katie was primarily responsible for completing the technical implementation, while Allie was primarily responsible for encoding the decision trees.

2.2 Accomplished vs. Planned

We were able to successfully complete a minimum viable product for ABX Ninja, despite two major setbacks.

First, we were not able to include EPIC integration with our minimum viable product. However, we met with the EPIC team at FastForward in April and May. We were able to show them what data we need to pull from EPIC, and they showed us how to use the APIs that are currently available to prepare the application for EPIC integration.

Second, in our initial project plan document, we included working towards FDA approval. However, due to our inability to fully integrate with EPIC, we realized before the project checkpoint that this was not a realistic maximum deliverable. Instead, we spoke with our mentors about expanding the functionality of the application to include a graphical user interface where users could build decision trees that would be automatically encoded in the database. Since adding this functionality was not feasible in the time frame, we created an implementation plan for the next developers to follow.

2.3 Future Work

We will both be graduating at the end of this semester, however, we are planning to hand off the project to another undergraduate student, Naina Rao. Before graduation, we will meet with her to discuss our current progress and the features that she should focus on implementing. We will also go over our implementation plan for the decision tree GUI.

Features for ABX Ninja's second iteration: a decision tree GUI, full EPIC integration, expansion to other hospitals, and usage statistics by medication.

2.4 Lessons Learned

Throughout this experience, we learned a great deal about web development. Importantly, we learned how to balance functionality and features for a minimum viable product so that it can serve its purpose, but have the ability to expand. In addition, we learned that communication is hugely important between team members to ensure that everyone understands their individual responsibilities and role in the project.

3. Technical appendices

Private Repository
https://code.jh.edu/projects/ABX/repos/ninja/browse
Access to this repository must be granted by the FastForward team

Deployment instructions

1. Clone repository from bitbucket
2. Create a PostgreSQL database named abxninja with owner abxninja on port 5434
3. Seed the database with file “dump.sql” from the database folder in the git repository
    a. If necessary, also grant privileges to the user for all tables and sequences in the schema using the file “privileges.sql” located in the database folder
4. Change the following files:
    a. server.js
        i. Change port number
        ii. Update `swaggerDefinition` with updated host/port number
    b. ./server/_config.js
        i. Update database connection string
        ii. Ex:
            - ‘postgres://abxninja:abxninja@localhost:2793’
    c. ./server/auth/jhed.js
        i. Update callback URL for passport (line 20)
            - ‘http://hostname:port/auth/jhed/callback’
    d. ./src/documentation/index.html
        i. Change line 76 to: ‘http://hostname:port/swagger.json’
5. Run test suite: `npm test`
6. Start the application: `npm start`
7. Visit the application using: http://hostname:port/

Launching Application
While on the Hopkins network, visit http://harb.rad.jhmi.edu:2793/

API Documentation
To view API documentation, visit: http://harb.rad.jhmi.edu:2793/documentation

4. References

1. CDC. Antibiotic Resistance Threats in the United States, 2013. Vol CS239559-B. Atlanta, GA 2013:114.
2. Pollack LA, Srinivasan A. Core elements of hospital antibiotic stewardship programs from the Centers for Disease Control and Prevention. Clin Infect Dis. 2014;59 Suppl 3:S97-100.
3. Magill SS, Edwards JR, Beldavs ZG, et al. Prevalence of antimicrobial use in US acute care hospitals, May-September 2011. JAMA. 2014;312(14):1438-1446.
4. Magill SS, Edwards JR, Bamberg W, et al. Multistate point-prevalence survey of healthcare-associated infections. N. Engl J Med. 2014;370(13):1198-1208.
5. Hecker MT, Aron DC, Patel NP, Lehmann MK, Donskey CJ. Unnecessary use of antimicrobials in hospitalized patients: current patterns of misuse with an emphasis on the antianaerobic spectrum of activity. Arch Intern Med. 2003;163(8):972-978.
6. Braykov NP, Morgan DJ, Schweizer ML, et al. Assessment of empirical antibiotic therapy optimisation in six hospitals: an observational cohort study. Lancet Infect Dis. 2014;14(12):1220-1227.
7. MacDougall C, Polk RE. Variability in rates of use of antibacterials among 130 US hospitals and risk- adjustment models for interhospital comparison. Infect Control Hosp Epidemiol. 2008;29(3):203-211.