

Technologie Aplikacji Internetowych

**Mini-projekt:
Kalendarz z organizerem wydarzeń**

Opis projektu i raport z testów penetracyjnych

Krzysztof Papciak

1. Opis projektu

Projekt obejmował stworzenie aplikacji webowej posiadającej funkcjonalności prostego kalendarza z możliwością dodawania wydarzeń (coś na wzór Google Calendar).

2. Wykorzystane technologie

Backend (Java):

- Jersey
- Spring Beans
- Spring Security
- MongoDB Driver
- Jackson

Frontend (Javascript + HTML + CSS):

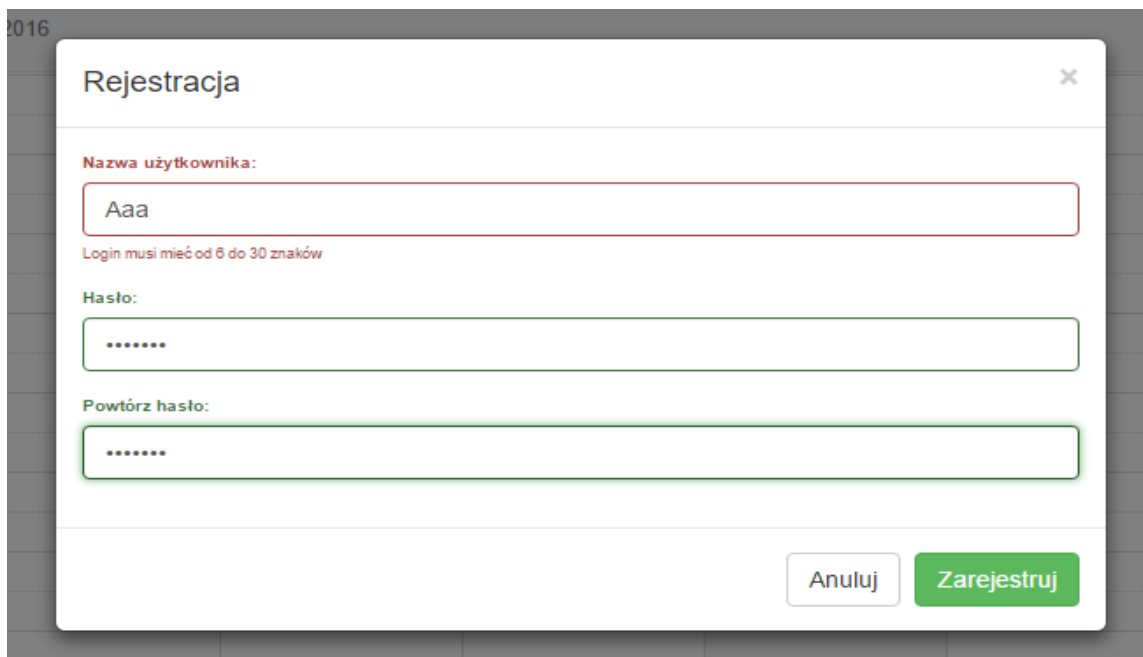
- jQuery Framework
- moment.js
- Bootstrap
- rozszerzenia Bootstrap do obsługi interfejsu użytkownika (kalendarze, wyświetlanie powiadomień)

Hosting: Heroku

3. Opis zaimplementowanych funkcjonalności

a) rejestracja użytkowników i logowanie

Do autentykacji użytkowników zastosowano API Spring Security. Dane kont użytkowników przechowywane są w bazie MongoDB.



The screenshot shows a web registration form titled "Rejestracja" (Registration) with a close button (X) in the top right corner. The form contains three input fields: "Nazwa użytkownika:" (Username) with the value "Aaa", "Hasło:" (Password), and "Powtórz hasło:" (Repeat password). Below the username field, there is a red error message: "Login musi mieć od 6 do 30 znaków" (Login must be 6 to 30 characters). At the bottom right, there are two buttons: "Anuluj" (Cancel) and "Zarejestruj" (Register).

Logowanie

Nazwa użytkownika:

testowy

Hasło:

.....

Anuluj

Zaloguj

b) kalendarze

Każdy użytkownik może stworzyć, edytować i usunąć własne kalendarze, które zawierają oddzielny zestaw dodanych wydarzeń. Po wybraniu kalendarza z listy, wyświetlone zostają wydarzenia do niego dodane.

Wybierz kalendarz:

aaaa ▼

Dodaj kalendarz

Edytuj kalendarz

c) wydarzenia

Do kalendarza mogą zostać dodane wydarzenia przypisane do konkretnego dnia i godziny. Zostają one wyświetlone na osi czasu zgodnie z wybranym zakresem dat.

Wydarzenie

Wydarzenie:

Oddanie projektu

Data:

2016-06-30

Godzina rozpoczęcia:

18:30

Godzina zakończenia:

20:00

Dodatkowe informacje:

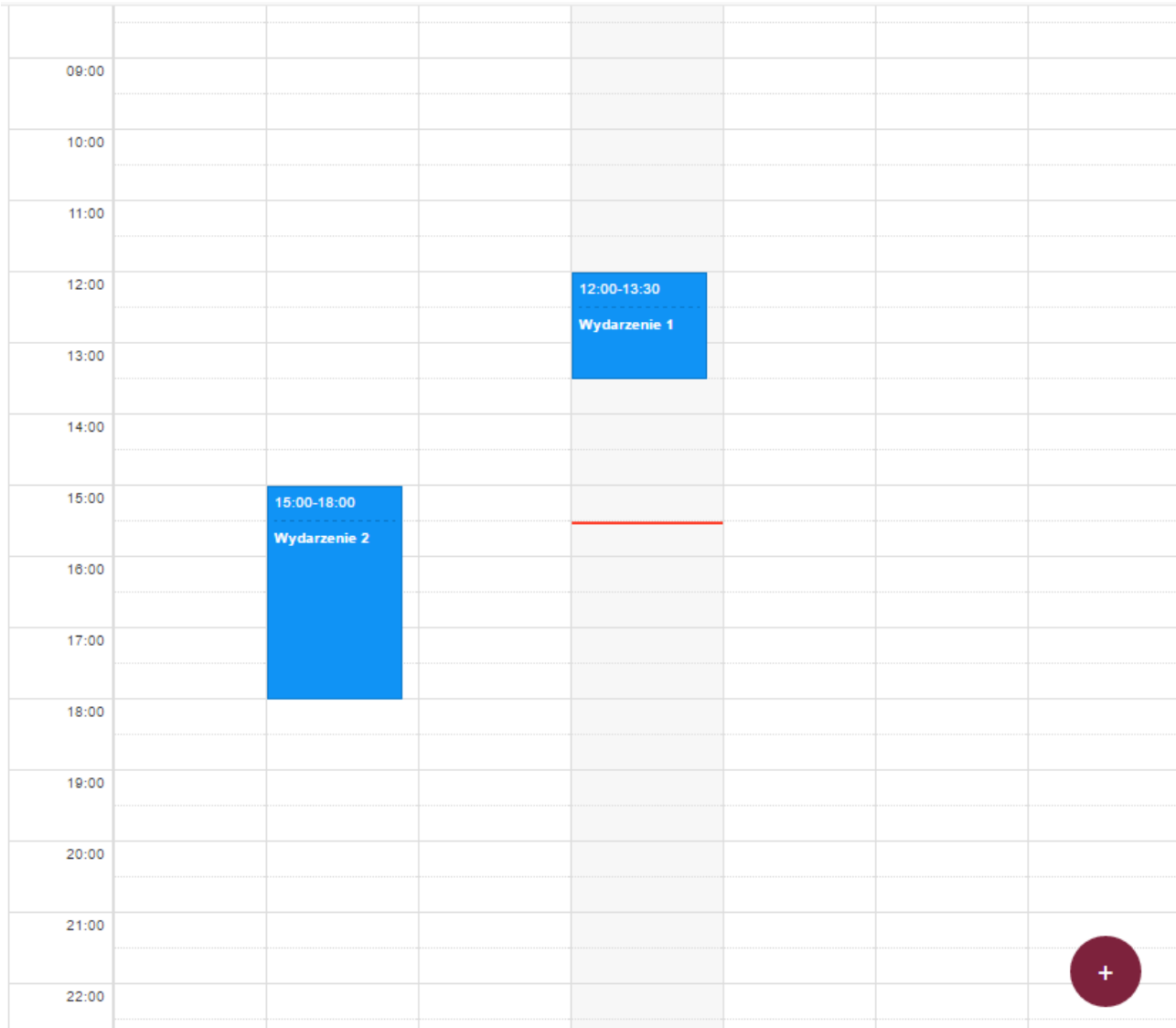
Budynek D17

Zapisz

Anuluj


d) oś czasu

Prezentuje dodane wydarzenia z danego przedziału czasowego. Umożliwia także edycję wydarzeń po kliknięciu w nie i dodanie nowego po wybraniu czerwonego przycisku z plusem.



e) opcje wyboru przedziału czasu

Pozwalają w łatwy sposób wybrać tydzień, z którego wydarzenia mają być wyświetlone. Dostępny jest mały kalendarz, który umożliwia łatwe ustawienie daty, a także przyciski do przewijania tygodni.

 **Kalendarz**

<

>

27 - 3 lipiec 2016

<

czerwiec 2016

>

Pn	Wt	Śr	Cz	Pt	So	Nd
30	31	1	2	3	4	5
6	7	8	9	10	11	12
13	14	15	16	17	18	19
20	21	22	23	24	25	26
27	28	29	30	1	2	3
4	5	6	7	8	9	10

07:00		
08:00		
09:00		
10:00		
11:00		

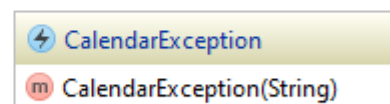
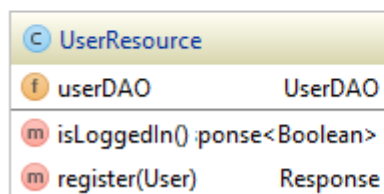
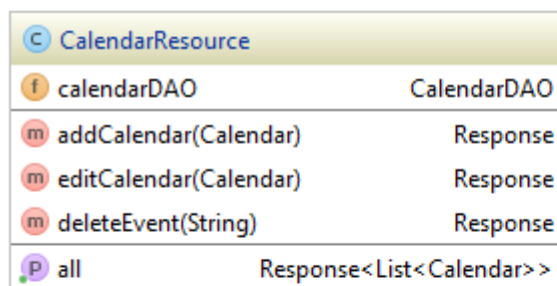
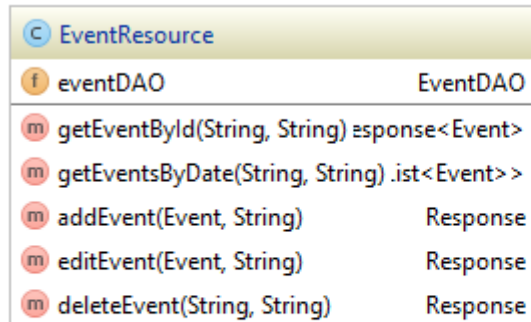
4. Porównanie produktu końcowego z wizją projektu

Ze względu na pewne skomplikowanie projektu, zrezygnowano z niektórych funkcjonalności względem opisanych w wizji. Zrezygnowano z możliwości dodawania wydarzeń cyklicznych oraz logowania przy użyciu konta na portalu Facebook

s5. Diagramy klas (backend)

a) pakiet pl.edu.agh.tai.main

Klasy zasobów odpowiadające na zapytania restowe.



b) pakiet pl.edu.agh.tai.data

Klasy wykorzystywane do mapowania wyników na obiekty JSON i odwrotnie przy komunikacji przez HTTP.

C Event	
m toString()	String
m genId()	void
<hr/>	
P title	String
P startTime	String
P info	String
P date	String
P color	int
P id	String
P endTime	String

C Response	
P errorCode	int
P message	String
P object	T
P success	boolean

C Calendar	
m toString()	String
m genId()	void
<hr/>	
P name	String
P id	String

C User	
m genId()	void
<hr/>	
P password	String
P username	String
P id	String

Powered by yFiles

c) pakiet pl.edu.agh.tai.security

Klasy wykorzystywane przez Spring Security i do pobierania aktualnego użytkownika

C RestAuthenticationSuccessHandler		
m	onAuthenticationSuccess(HttpServletRequest, HttpServletResponse, Authentication)	void
P	requestCache	RequestCache

C MongoUserDetailsService		
f	userDAO	UserDAO
m	loadUserByUsername(String	

C RestAuthenticationEntryPoint		
m	commence(HttpServletRequest, HttpServletResponse, AuthenticationException)	void

C CurrentUser		
m	getName()	String

Powered by yFiles

d) pakiet pl.edu.agh.tai.dao

Klasy zapewniające dostęp do danych przechowywanych w bazie MongoDB.

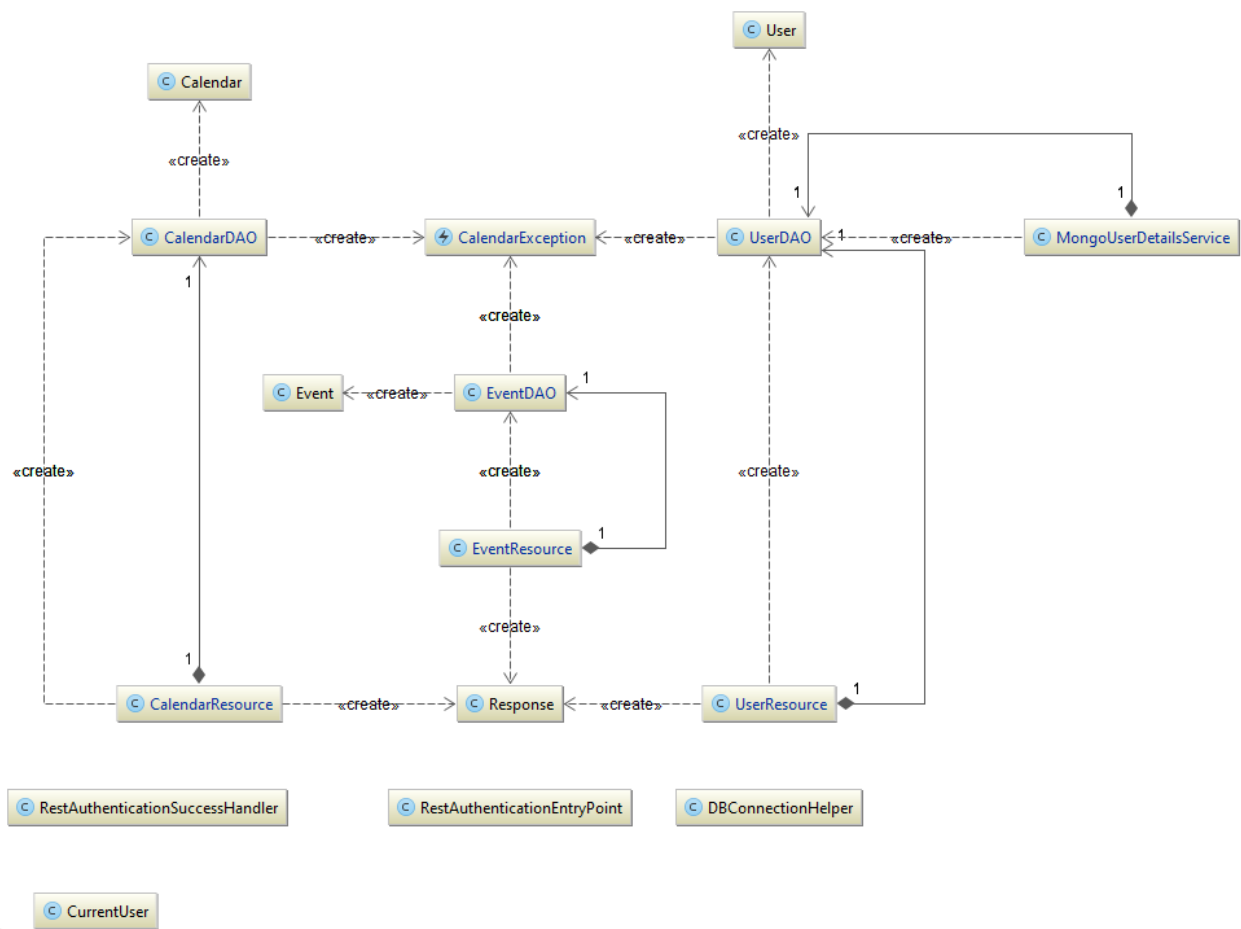
C EventDAO		
m	EventDAO()	
m	docToEvent(Document)	Event
m	eventToDoc(Event, String)	Document
m	getCalendarIdByName(String)	String
m	getEventById(String, String)	Event
m	getEventsByDate(String, String)	List<Event>
m	addEvent(String, Event)	void
m	editEvent(String, Event)	void
m	deleteEvent(String, String)	void

C DBConnectionHelper		
f	db	MongoDatabase
m	DBConnectionHelper()	
m	init()	void
m	getCalendarsCol()	MongoCollection<Document>
m	getUsersCol()	MongoCollection<Document>
m	getEventsCol()	MongoCollection<Document>
m	getDb()	MongoDatabase

C CalendarDAO		
m	CalendarDAO()	
m	checkCalendarOwner(String, MongoCollect	
m	getAll()	List<Calendar>
m	add(Calendar)	void
m	edit(Calendar)	void
m	delete(String)	void
m	deleteEventsByCalendarId(String)	void

C UserDAO		
m	UserDAO()	
m	getByName(String)	User
m	add(User)	void
m	userToDoc(User)	Document
m	edit(User)	void

e) zależności między klasami



6. Raport z testów penetracyjnych

a) wykorzystywane narzędzia

- Nessus
- Wapiti

b) Ops testów

Testy przeprowadzono na aplikacji zdeployowanej na Heroku. Głównie skupiono się na testach typu Vulnerability.

c) wyniki testów

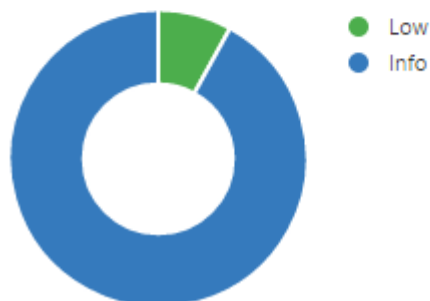
I. Nessus

Przeprowadzono testy typu Vulnerability z ustawieniem: *scan for all web vulnerabilities(quick)*.

Scan Details

Name:	http://basiccalendar.herokuapp.com/
Status:	Completed
Policy:	Web Application Tests
Scanner:	Local Scanner
Folder:	My Scans
Start:	Today at 2:54 PM
End:	Today at 3:20 PM
Elapsed:	26 minutes
Targets:	http://basiccalendar.herokuapp.com/

Vulnerabilities



Wyniki testów:

http://basiccalendar.herokuapp.com/		Configure	Audit Trail	Launch ▼	Export ▼
CURRENT RESULTS: TODAY AT 4:26 PM					
Scans	>	Hosts 1	Vulnerabilities 13	History 2	
<input type="checkbox"/>	Severity ▲	Plugin Name	Plugin Family	Count	
<input type="checkbox"/>	LOW	Web Server Transmits Cleartext Credentials	Web Servers	1	
<input type="checkbox"/>	INFO	Nessus SYN scanner	Port scanners	3	
<input type="checkbox"/>	INFO	Apache Tomcat Default Error Page Version Detection	Web Servers	2	
<input type="checkbox"/>	INFO	CGI Generic Tests Load Estimation (all tests)	CGI abuses	2	
<input type="checkbox"/>	INFO	HTTP Methods Allowed (per directory)	Web Servers	2	
<input type="checkbox"/>	INFO	HTTP Reverse Proxy Detection	Web Servers	2	
<input type="checkbox"/>	INFO	HTTP Server Type and Version	Web Servers	2	
<input type="checkbox"/>	INFO	HyperText Transfer Protocol (HTTP) Information	Web Servers	2	
<input type="checkbox"/>	INFO	Missing or Permissive Content-Security-Policy HTTP Response H...	CGI abuses	2	
<input type="checkbox"/>	INFO	Web Application Sitemap	Web Servers	2	
<input type="checkbox"/>	INFO	Web mirroring	Web Servers	2	
<input type="checkbox"/>	INFO	Web Server Allows Password Auto-Completion	Web Servers	2	
<input type="checkbox"/>	INFO	HSTS Missing From HTTPS Server	Web Servers	1	

Interpretacja wyników:

Najgroźniejszym błędem jaki udało się znaleźć jest *Web Server Transmits Cleartext Credentials*.

Jest on spowodowany faktem, że hasło użytkownika przesyłane jest w niezabezpieczonej postaci do serwera. Rozwiązaniem mogłoby być użycie protokołu HTTPS.

Kolejnym błędem, który może się przydać przy próbie shackowania strony jest *Apache Tomcat Default Error Page Version Detection*, który spowodowany jest wyświetlaniem domyślnych stron błędów serwera Tomcat. Na tych stronach podana jest wersja serwera. Informacja ta może ułatwić atak na stronę.

Inne wykryte problemy, które potencjalnie zmniejszają bezpieczeństwo strony to:

- Missing or Permissive Content-Security-Policy HTTP Response Header
- Web Server Allows Password Auto-Completion
- HSTS Missing From HTTPS Server
-

II. Wapiti

Testy przeprowadzone przy użyciu Wapiti nie wykazały żadnych luk

Wapiti vulnerability report for <http://localhost:8080/index.html>

Date of the scan: Thu, 30 Jun 2016 11:55:39 +0000. Scope of the web scanner : folder

Summary

Category	Number of vulnerabilities found
Cross Site Scripting	0
Htaccess Bypass	0
Backup file	0
SQL Injection	0
Blind SQL Injection	0
File Handling	0
Potentially dangerous file	0
CRLF Injection	0
Commands execution	0
Resource consumption	0
Internal Server Error	0