

ŚLĄSKA WYŻSZA SZKOŁA INFORMATYCZNO-MEDYCZNA

WYDZIAŁ GRAFIKI I INFORMATYKI

KIERUNEK: INFORMATYKA

TOMASZ DARŁOWSKI

**SYSTEM WSPOMAGAJĄCY PRACĘ DZIAŁU KONTROLI
JAKOŚCI**

Praca dyplomowa napisana pod kierunkiem

Dr inż. Paweł Kasprowski

.....

(podpis promotora)

CHORZÓW 2014

Spis treści

1. Wstęp.....	4
2. Cel i założenia pracy.....	5
2.1. Cel pracy.....	5
2.2. Analiza wymagań.....	5
2.3. Założenia pracy.....	7
2.4. Istniejące rozwiązania.....	7
2.5. Zakres pracy.....	8
3. Technologie.....	9
3.1. Java 7 (JEE6).....	9
3.2. Java ServerFaces (JSF).....	10
3.3. Mapowanie obiektowo-relacyjne (ORM).....	10
3.4. PostgreSQL.....	11
3.5. PrimeFaces 3.5.....	11
3.6. Glassfish 4.2.....	11
3.7. Wzorce projektowe.....	12
3.7.1. MVC.....	12
4. Projekt systemu.....	13
4.1. Opis działania systemu.....	13
4.2. Przypadki użycia.....	14
4.3. Diagramy przepływu wybranych funkcji programu.....	15
4.3.1. Diagram dodania zdarzenia.....	15
4.3.2. Diagram przepływu statusów sprawy.....	16
5. Projekt bazy danych.....	17
5.1. Encje – struktura i opis.....	17
5.1.1. BaseEntity.....	17
5.1.2. Item.....	18
5.1.3. AppUser.....	18
5.1.4. Supplier / Customer.....	19
5.1.5. Event.....	19

5.1.6. Issue.....	20
5.1.7. Component.....	21
5.1.8. FileEntity.....	21
5.1.9. Label.....	22
5.1.10. Comment.....	22
5.2. Schemat ERD.....	23
5.3. Diagram klas serwisów aplikacji.....	24
6. Specyfikacja zewnętrzna.....	25
6.1. Instalacja i przygotowanie bazy danych.....	25
6.2. Instalacja i konfiguracja serwera i aplikacji.....	26
6.3. Przeglądarki.....	26
6.4. Przepływ stron.....	27
7. Opis działania.....	28
7.1. Dialog z użytkownikiem.....	28
7.2. Logowanie.....	30
7.3. Okno główne programu.....	31
7.4. Defekty, zadania, ulepszenia.....	32
7.5. Terminarz.....	35
7.6. Komponenty.....	37
7.7. Przyrządy pomiarowe.....	40
7.8. Dokumenty.....	42
7.9. Kolekcje.....	43
7.10. Panel administracyjny.....	44
7.11. Komunikaty o błędach.....	46
7.12. Zmiana języka.....	46
8. Testy.....	47
9. Podsumowanie.....	48
10. Spis ilustracji.....	49
11. Bibliografia.....	51
12. Załącznik.....	52
12.1. Płyta CD.....	52
13. Oświadczenie.....	53

1. WSTĘP

Tematem niniejszej pracy inżynierskiej jest system wspomagający pracę działu kontroli jakości. Aplikacja ta powstała w celu maksymalnego usprawnienia pracy kontrolerom jakości, a także kierownictwu takiego działu. Będzie ją można w łatwy sposób rozszerzyć o kolejne elementy, takie jak integracja z działem technologicznym czy magazynem. Wynikiem niniejszej pracy będzie system, który jako uniwersalne narzędzie pozwoli na ułatwienie dostępu do informacji, pozwoli na łatwą kontrolę nad jakością produktu, a także w prosty sposób wyeliminuje potrzebę angażowania innych aplikacji w celu zapewnienia jakiejś funkcjonalności. System jest przeznaczony zarówno do sieci intranetowej oraz jako aplikacja internetowa.

2. CEL I ZAŁOŻENIA PRACY

2.1. CEL PRACY

Stworzenie systemu wspomagającego pracę działu jakości. System powinien wspomagać wykonywanie podstawowych czynności pracowników jak i pomagać w zarządzaniu takim działem kadrze kierowniczej.

2.2. ANALIZA WYMAGAŃ

Aby dobrze zaprojektować system musimy zrozumieć pracę działu kontroli jakości. Otóż codziennymi zadaniami ludzi w nich pracujących jest przede wszystkim kontrola produkcji. Każdy element czy to produkowany, czy dostarczany jako prefabrykat musi posiadać odpowiednią dokumentację oraz musi spełniać pewne normy. Program powinien pozwalać na tworzenie oraz aktualizację danych na temat każdego komponentu. Powinny tu znajdować się: kod referencyjny, nazwa zwyczajowa, opis, połączone produkty czy prefabrykaty, odpowiednie dokumenty takie jak plany kontroli czy indywidualne ustalenia z klientem, a także zdjęcia w celu szybkiej identyfikacji komponentu. Jest to też miejsce, w którym użytkownik będzie mógł zapisywać raporty pomiarowe, czy to z pomiarów manualnych, czy to ze stacji pomiarowych 3D. Może mieć także bardzo łatwy dostęp do listy wszystkich raportów, które dany komponent posiada.

Następnym ważnym elementem w centrum pomiarowym każdej firmy jest magazyn przyrządów pomiarowych. Każdy sprawdzian posiada trzy główne cechy. Są to: walidacja, numer referencyjny i okres walidacji. Każdy nowy sprawdzian musi przejść walidację, czy to robioną wewnątrz firmy macierzystej, czy też zewnętrznej. Na podstawie pozytywnego raportu pomiarowego, przyrząd taki jest dopuszczony do produkcji. Znajdziemy tu także opis dodatkowej dokumentacji oraz zdjęć każdego

sprawdzianu w celu łatwiejszego ich zidentyfikowania. Dodatkowo aplikacja powinna przypominać użytkownikom o zbliżającej się dacie końca walidacji.

Praca działu jakości jest silnie uzależniona od czasu. Pracownik takiego działu musi okresowo wykonywać pewne pomiary, analizy, czy też przygotowywać raporty. Z tego też powodu aplikacja powinna posiadać terminarz z możliwością tworzenia spotkań lub wydarzeń. Użytkownik mógłby dodawać kolejne terminy, ustalać ich priorytet, dodawać opis czy wysyłać zaproszenia do innych użytkowników poprzez adres email.

Nieodłącznym elementem w dziale jakości są sprawy. Użytkownik powinien mieć możliwość zarejestrowania nowej sprawy, dodania do niej odpowiedniej dokumentacji oraz zarządzania nią, aż do czasu rozwiązania problemu. Powinien mieć również możliwość nadania priorytetu, zmiany jego stanu, ustalenia jego typu, czy wreszcie przypisania do odpowiedniej osoby. Wyróżniamy trzy główne rodzaje spraw: są to defekt, zadanie oraz usprawnienie. Dodatkowo użytkownicy mogą grupować zadania za pomocą specjalnych etykiet, które mogą sami tworzyć. Bardzo pomocne i mogące stanowić swego rodzaju rejestr wydarzeń są komentarze, które każdy użytkownik może dopisywać do danej sprawy. Mogą one zawierać detale rozwiązania, czy też decyzje jakie zostały podjęte.

System powinien posiadać również możliwość rejestracji odbiorców i dostawców. Użytkownik mógłby w łatwy sposób podejrzeć ich dane, a także na ich podstawie wyszukiwać grupy produktów.

W systemie powinno znaleźć się też miejsce na podsumowanie aktualnych wydarzeń, zbliżających się spotkań, czy ostatnio dodanych spraw.

Ostatnim ważnym aspektem jest możliwość zmiany języka dla interfejsu. Umożliwiłoby to łatwiejszą obsługę dla firm zatrudniających obcokrajowców.

2.3. ZAŁOŻENIA PRACY

System powinien być zbudowany w oparciu o najnowsze trendy i wzorce projektowe. Interfejs użytkownika powinien zapewniać łatwy dostęp do funkcji systemu, być maksymalnie intuicyjny dla użytkownika oraz powinien poprawnie działać na wszystkich obecnych przeglądarkach internetowych takich jak Chrome, Firefox, Opera, Safari czy Internet Explorer.

Aplikacja powinna być napisana w sposób umożliwiający jej rozbudowanie o kolejne moduły w celu integracji z innymi działami jak np. dział technologiczny.

Stworzony system musi być również bezpieczny, w tym celu powinien on posiadać system logowania wraz z modułem autoryzacji. Dodatkowo użytkownicy oznaczeni jako administratorzy powinni mieć możliwość zarządzania kontami dostępu do systemu.

2.4. ISTNIEJĄCE ROZWIĄZANIA

Według mojej opinii największymi firmami na rynku są: HP (*QC – quality center*) oraz Atlassian (*Jira*). Oferują one potężne narzędzia, służące do prowadzenia projektów i śledzenia zarówno postępu prac jak i napotkanych defektów. Ich produkty są jednak drogie. JIRA powyżej 10 stanowisk to koszt około 3600zł, a produkt HP to kilku krotność ceny JIRY, w zależności od modułów. Aplikacje te jednak powstały z myślą o prowadzeniu projektów IT. Brak magazynów i konieczność adaptacji tych ostatnich w innym programie powoduje, że narzędzie to nie spełnia wszystkich wymagań w przypadku działu jakości firmy produkcyjnej.

Inną grupą systemów zarządzających jakością to wyspecjalizowane narzędzia właśnie pod firmy produkcyjne. Niestety nie udało mi się znaleźć żadnego kompleksowego narzędzia, które byłoby zupełnie darmowe. Do płatnych, komercyjnych aplikacji należą m.in. Isoft QMS i Galactica HYDRA.

Z darmowych narzędzi jakie udało mi się znaleźć mogę wymienić Bugzilla, fundacji Mozilla. Niestety aplikacja ta przystosowana została do projektów IT i również nie posiada pewnych istotnych funkcjonalności, które spełniłyby wymogi wynikające z analizy problemu.

2.5. ZAKRES PRACY

Zaprojektowanie i stworzenie aplikacji webowej do zarządzania i wspomagania pracy działu jakości.

3. TECHNOLOGIE

3.1. JAVA 7 (JEE6)

Do napisania aplikacji wykorzystany został język Java. Jest to język obiektowy stworzony przez grupę roboczą o nazwie Green Team pod kierownictwem James-a Gosling-a, a który swoją premierę miał w 1995.

Język ten cechuje:

- Wielo-platformowość – czyli raz napisany kod (skompilowany do kodu bajtowego) możemy uruchamiać na wszelkich systemach posiadających wirtualną maszynę Java-y (JVM).
- Silne typowanie – oznacza, że użytkownik musi zadeklarować rodzaj zmiennej nim jej użyje.
- Obiektość - wszelkie dane i akcje na nich podejmowane są pogrupowane w klasy obiektów.
- Dziedziczenie - w Javie wszystkie obiekty są pochodną obiektu nadrzędnego (jego klasa nazywa się po prostu Object), z którego dziedziczą podstawowe zachowania i właściwości.
- Sieciowość i obsługa programowania rozproszonego - dzięki wykorzystaniu reguł obiektości, Java nie widzi różnicy między danymi płynącymi z pliku lokalnego, a danymi z pliku dostępnego przez HTTP czy FTP. Biblioteki Javy udostępniają wyspecjalizowane funkcje umożliwiające programowanie rozproszone – zarówno między aplikacjami Javy (RMI), jak i między aplikacją Javy, a aplikacjami napisanymi w innych językach (CORBA, usługi sieciowe).
- Niezawodność i bezpieczeństwo - przede wszystkim Java ma system wyjątków, czyli sytuacji, kiedy kod programu natrafia na nieprzewidywane trudności, takie jak np.:

- operacje na elemencie poza zadeklarowaną granicą tablicy lub elemencie pustym
- czytanie z niedostępnego pliku lub nieprawidłowego adresu URL
- podanie nieprawidłowych danych przez użytkownika

3.2. JAVA SERVERFACES (JSF)

Framework webowy, który został wykorzystany przy pisaniu niniejszej aplikacji to Java Server Faces. Framework w dużym stopniu upraszcza tworzenie graficznego interfejsu aplikacji. Pierwsza wzmianka o niniejszej technologii pochodzi z konferencji JavaOne z roku 2002. Pierwsza implementacja pojawiła się jednak dopiero w 2004 (wersja 1.0). Pierwsze implementacje nie były zbyt przyjazne programistom, ale posiadały jedną bardzo ważną cechę: rozszerzalność. Dzięki programistom pojawiły się takie opensource-owe rozwiązania jak Facelets, Ajax4jsf, JSF Templates, Richfaces czy Primefaces. W 2009 pojawił się nowa specyfikacja oznaczona numerem 2.0. Druga wersja framework-u posiada lepszą integrację z specyfikacją Java Enterprise Edition oraz wspiera technologię Ajax oraz REST.

3.3. MAPOWANIE OBIEKTOWO-RELACYJNE (ORM)

Do zapisywania rekordów w bazie danych użyto wbudowanej w specyfikację JEE technologii Java Persistence API. Jest ona standardem ORM dla języka Java. Z punktu widzenia programisty jest to możliwość operowania na obiektach - zwanych encjami - oraz zapisywania wyników operacji do relacyjnej bazy danych za pomocą obiektu EntityManager. Sposób, w jaki obiekty i ich połączenia przekładają się na elementy bazy danych są definiowane za pomocą adnotacji lub dokumentów XML. Po standardowym zestawem operacji udostępnianych przez obiekt

EntityManager standard, JPA definiuje język zapytań JPA Query Language podobny do SQL.

3.4. POSTGRESQL

Do przechowywania danych użyto bazy danych PostgreSQL. PostgreSQL często nazywany także Postgres to, obok MySQL i Firebird, jeden z trzech najpopularniejszych wolno dostępnych systemów zarządzania relacyjnymi bazami danych. Początkowo opracowywany na Uniwersytecie Kalifornijskim w Berkeley i opublikowany pod nazwą Ingres. W miarę rozwoju i zwiększania funkcjonalności, baza danych otrzymała nazwy Postgres95 i ostatecznie PostgreSQL, aby upamiętnić pierwowzór oraz zaznaczyć zgodność ze standardem SQL. Aktualnie baza implementuje większość standardu SQL:2011.

3.5. PRIMEFACES 3.5

Biblioteka komponentów jaka została dodana to Primefaces. Są to w tej chwili najczęściej wykorzystywane komponenty dla framework-u JSF. Ich dużą zaletą jest ogrom gotowych komponentów, wbudowana obsługa technologii Ajax. Ich użycie jest bardzo łatwe i przyspiesza w znacznym stopniu czas budowy aplikacji od strony wizualnej.

3.6. GLASSFISH 4.2

Jako serwer aplikacji wybrano serwer Glassfish. Jest to otwarty serwer aplikacji dla platformy Java EE, rozwijany przez Oracle (dawniej Sun Microsystems).

3.7. WZORCE PROJEKTOWE

3.7.1. MVC

Model-View-Controller (pol. Model-Widok-Kontroler) to wzorec architektoniczny służący do organizowania struktury aplikacji posiadających graficzne interfejsy użytkownika. Wiele prac traktuje go jako pojedynczy wzorec, lecz może on być także traktowany jako złożony wzorec wykorzystujący idee wzorców prostych takich, jak Obserwator, Strategia czy Kompozyt. Oba te podejścia nie wykluczają się. MVC nie był traktowany jako samodzielny wzorec również w pracy „Design Patterns: Elements of Reusable Object-Oriented Software” autorstwa „Bandy Czworga”.

Model-View-Controller zakłada podział aplikacji na trzy główne części:

- Model – jest pewną reprezentacją problemu bądź logiki aplikacji.
- Widok – opisuje, jak wyświetlić pewną część modelu w ramach interfejsu użytkownika. Może składać się z podwidoków odpowiedzialnych za mniejsze części interfejsu.
- Kontroler – przyjmuje dane wejściowe od użytkownika i reaguje na jego poczynania, zarządzając aktualizacje modelu oraz odświeżenie widoków.

Wszystkie trzy części są ze sobą wzajemnie połączone.

4. PROJEKT SYSTEMU

4.1. OPIS DZIAŁANIA SYSTEMU

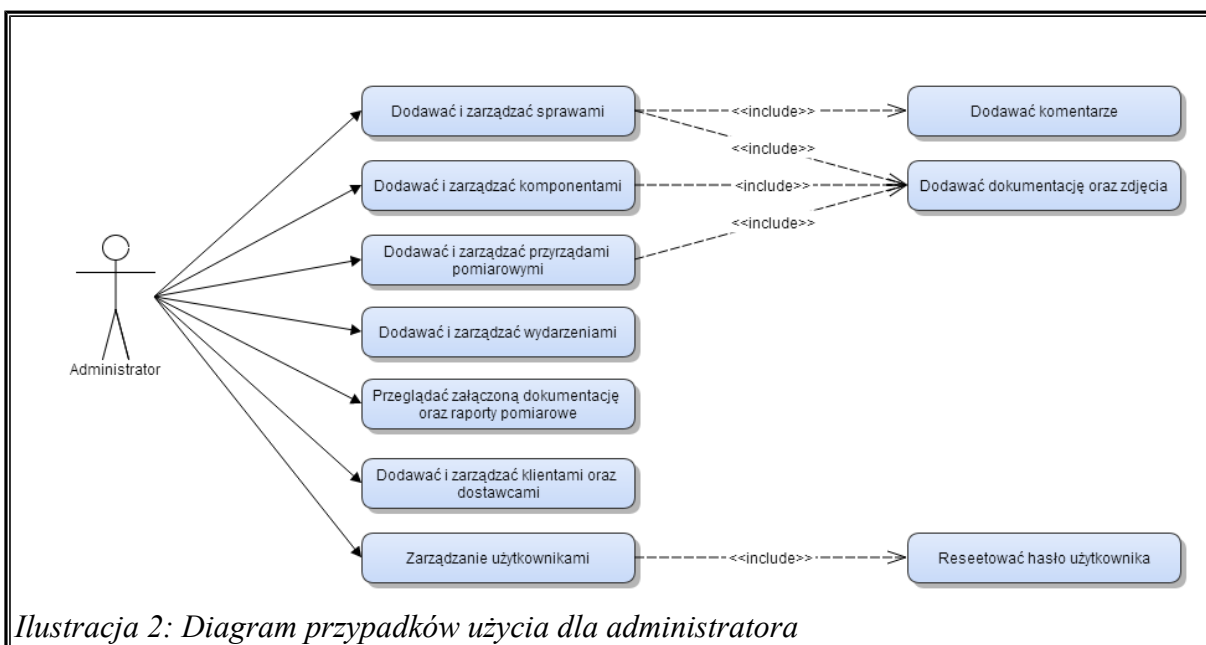
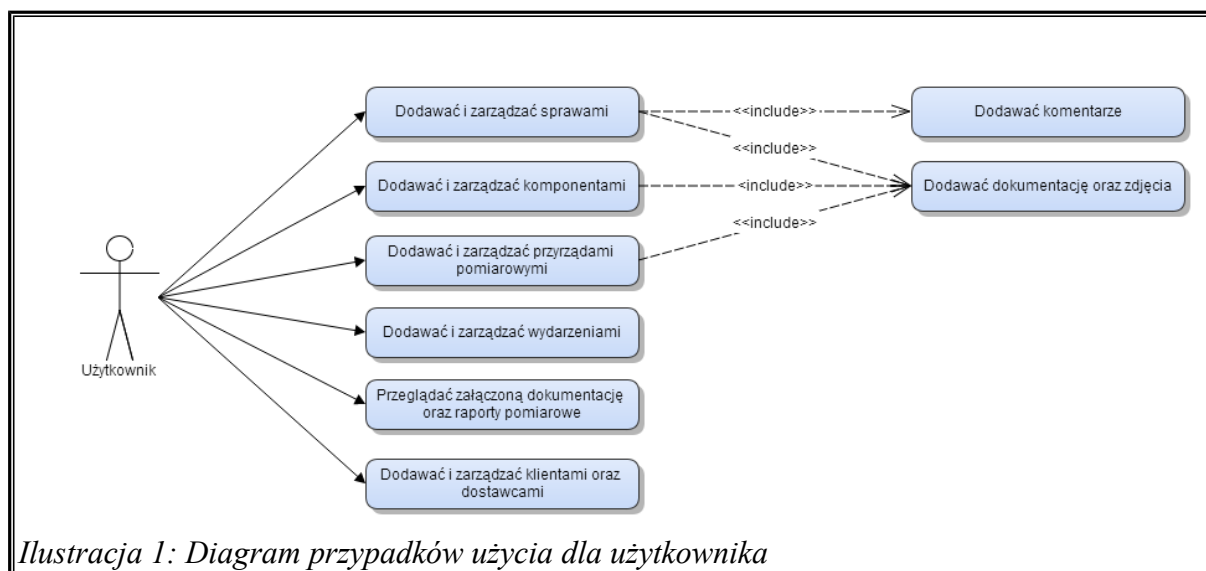
System powstał w celu zarządzania działem kontroli jakości. Dostęp do aplikacji powinni mieć tylko uprawnieni do tego użytkownicy, w związku z tym system został wyposażony w proces autoryzacji i identyfikacji.

Każdy użytkownik ma pełną swobodę w używaniu aplikacji. Do podstawowych czynności pracownika należy: zarządzanie produktami i półproduktami pod względem kontroli jakości, walidacja przyrządów pomiarowych, prowadzenie prac naprawczych w ramach ujawnionych defektów oraz wykonywanie okresowych pomiarów zgodnych z ustaleniami z klientem. Każdej z tych czynności przysługuje osobny widok w aplikacji. Dzięki temu w prosty sposób możemy dotrzeć do konkretnej funkcjonalności.

System zbudowany jest w oparciu o kontrolery widoku. Każdy z nich zarządza inną częścią aplikacji dlatego też rozbudowa aplikacji nie powinna stanowić większego problemu. Kontroler widoku odpowiedzialny jest za przetwarzanie danych pobranych z widoku (strony internetowej) i przekazywanie jej do odpowiednich serwisów.

4.2. PRZYPADKI UŻYCIA

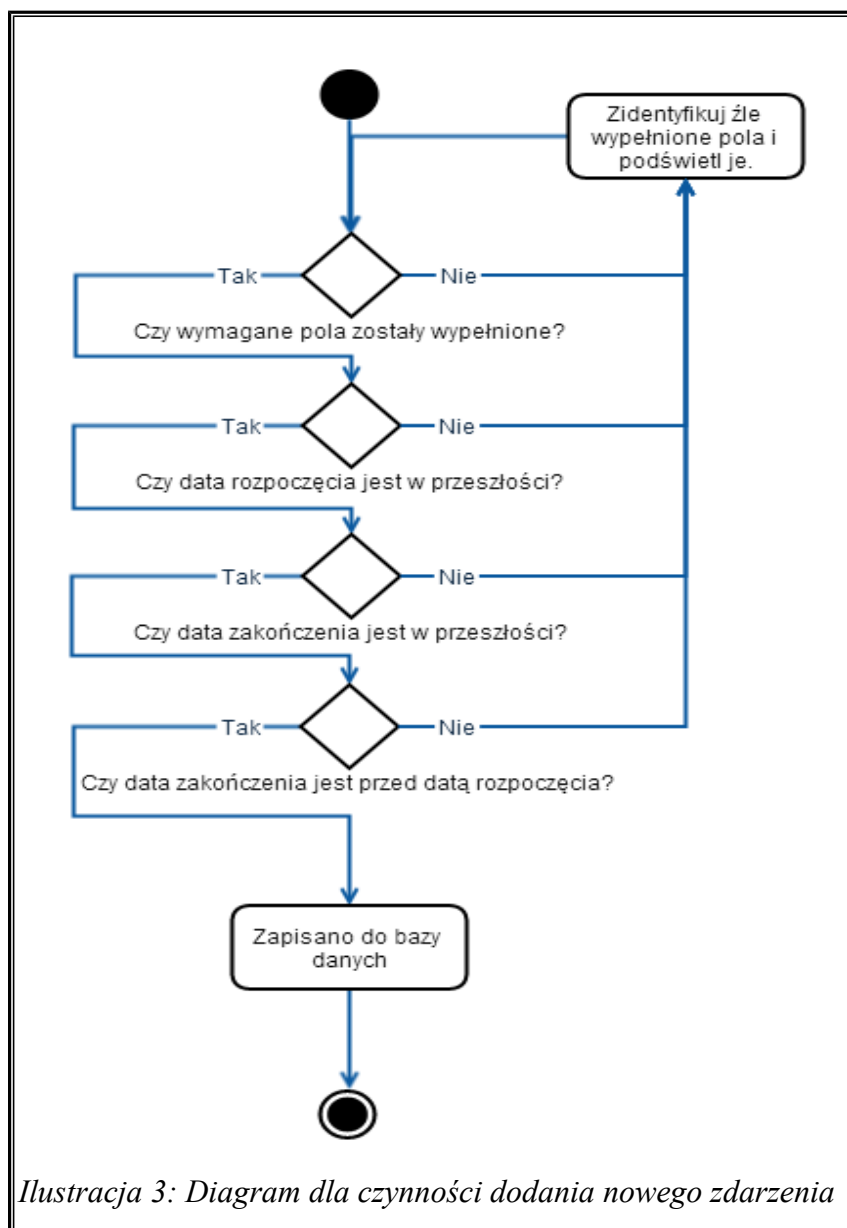
Poniżej przedstawione zostały diagramy przypadków użycia dla użytkownika oraz administratora.



4.3. DIAGRAMY PRZEPŁYWU WYBRANYCH FUNKCJI PROGRAMU

4.3.1. Diagram dodania zdarzenia

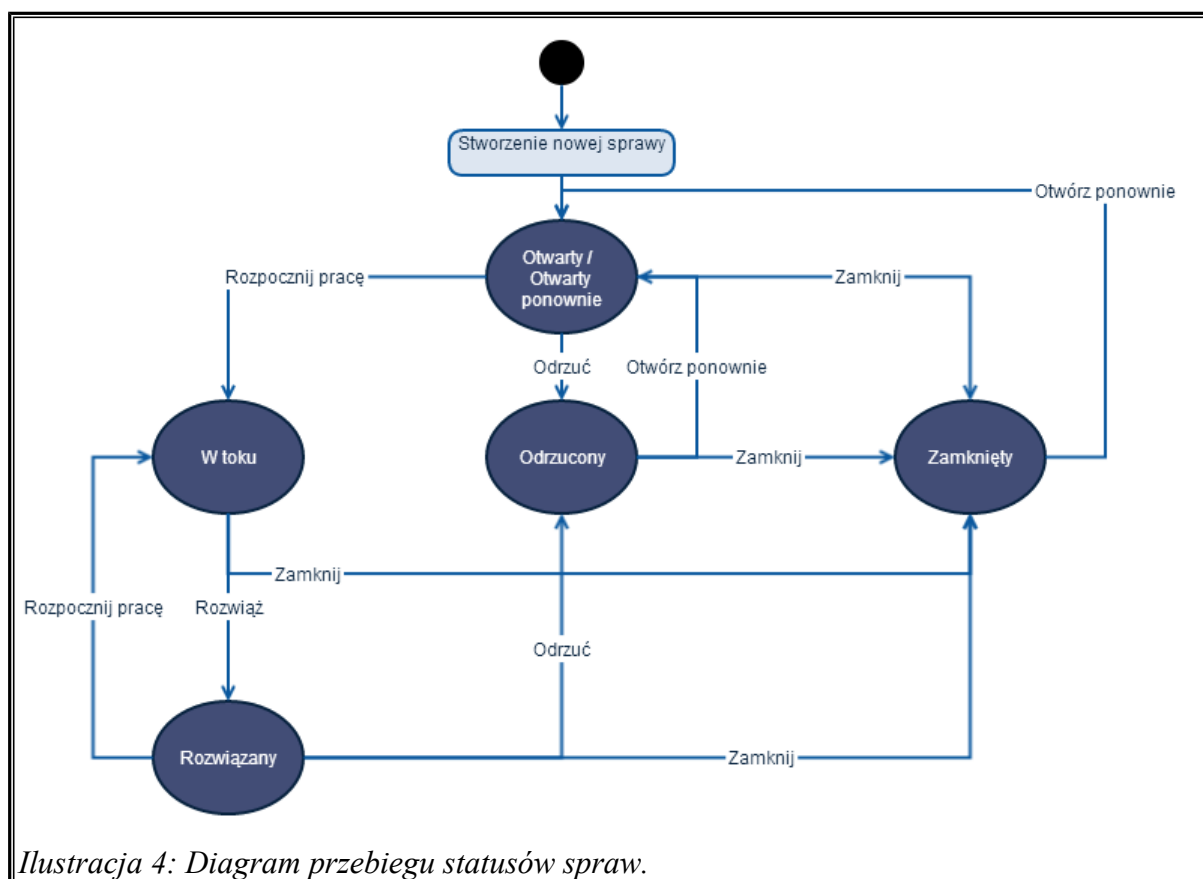
Na poniższej ilustracji przedstawiono diagram czynności jakie zachodzą podczas dodawania nowego zdarzenia do terminarza. Istnieją dwie możliwości dodania nowego terminu. Pierwszy to kliknięcie na obszarze kalendarza w wybrany przez nas termin, drugi to wybranie opcji „Dodaj zdarzenie”.



Ilustracja 3: Diagram dla czynności dodania nowego zdarzenia

4.3.2. Diagram przepływu statusów sprawy.

Kolejny diagram przedstawia kolejne etapy dodawania nowej sprawy i jej przepływ wewnątrz aplikacji. Każda sprawa po stworzeniu ma status „Otwarty”. Następnie z biegiem postępów prac użytkownik może zmieniać status sprawy.



Ilustracja 4: Diagram przebiegu statusów spraw.

5. PROJEKT BAZY DANYCH

5.1. ENCJE – STRUKTURA I OPIS

Wszystkie encje rozszerzają jedną wspólną encję „BaseEntity”. Dzięki takiemu zabiegowi w późniejszym czasie można było stworzyć generyczny serwis, który potrafił wykonywać wszystkie podstawowe operacje na stworzonych encjach. Dodatkowo niektóre encje za pomocą JPA i adnotacji @Inheritance oraz @DiscriminatorColumn dziedziczyły po klasie Item (*klasa Item dziedziczy po BaseEntity*). Te wszystkie zabiegi pozwoliły na łatwe zarządzanie stworzonymi serwisami. Poniżej przedstawiono strukturę wszystkich encji programu.

5.1.1. BaseEntity

BaseEntity jest klasą oznaczoną adnotacją @MappedSuperClass. Dzięki temu jest ona abstrakcją dla wszystkich pozostałych klas. Posiada ona tylko trzy pola, są to:

- id – opowiadające za przechowywanie identyfikatora
- uid – dodatkowe pole identyfikujące
- active – pole służące do oznaczania rekordów jako usunięte, domyślnie oznaczane jako „true”.

Wartości dwóch pierwszych pól są generowane automatycznie i służą do odpowiedniej identyfikacji encji.

5.1.2. Item

Item jest klasą, która posiada wspólne pola dla 3 klas głównych programu. Są to: Component, Gauge, Issue. Posiada ona następujące pola:

- referenceNumber – odpowiada ono za unikatowy numer wewnętrzny. Jest traktowane jako dodatkowy identyfikator dla potrzeb biznesowych.
- Description – przechowuje opis elementu.
- CreateDate – przechowuje datę stworzenia.
- ModifyDate – przechowuje ostatnią datę modyfikacji.

Jako klasa, przechowuje ona także relację do plików, komponentów, osoby, która stworzyła dany element oraz do osoby, która go zmodyfikowała.

5.1.3. AppUser

Tabela ta przechowuje wszystkich użytkowników systemu. Pola *name*, *surname* odpowiadają imieniu i nazwisku użytkownika. Pole *email* służy do przechowywania adresu email, który jest potrzebny do wysyłania powiadomień do użytkownika, a *login* i *password* za przechowywanie nazwy użytkownika i hasła. Hasło jest zaszyfrowane algorytmem MD5. Dodatkowo pole *locale* służy do określenia języka interfejsu.

appuser	
# id	int8 (19)
○ active	bool
○ email	varchar (255)
○ locale	varchar (255)
○ login	varchar (255)
○ name	varchar (255)
○ password	varchar (255)
○ surname	varchar (255)
○ uid	varchar (255)

Ilustracja 5: Encja AppUser

5.1.4. Supplier / Customer

Encje te odpowiadają za przechowywanie danych dostawcy i klienta. Obie posiadają te same pola. Pola *name*, *email*, *telephone*, *address* są polami wymaganymi.

supplier		customer	
# id	int8 (19)	# id	int8 (19)
o active	bool	o active	bool
o address	varchar (255)	o address	varchar (255)
o contactperson	varchar (255)	o contactperson	varchar (255)
o email	varchar (255)	o email	varchar (255)
o name	varchar (255)	o name	varchar (255)
o telephone	varchar (255)	o telephone	varchar (255)
o uid	varchar (255)	o uid	varchar (255)

Ilustracja 6: Encje Supplier i Customer

5.1.5. Event

Encja ta odpowiada za przechowywanie informacji o zdarzeniach w terminarzu. Pole *editable* determinuje czy zdarzenie jest edytowalne. Edycja zdarzenia jest dostępna tylko wtedy, jeżeli początek zdarzenia nadal jest w przyszłości. Pole *styleclass* przechowuje nazwę klasy CSS, która określa kolor zdarzenia na kalendarzu. Pola wymagane to *datefrom*, *dateto*, *title*.

event	
# id	int8 (19)
o active	bool
o allday	bool
o datefrom	timestamp
o dateto	timestamp
o description	text (2147483647)
o editable	bool
o styleclass	varchar (255)
o title	varchar (255)
o uid	varchar (255)

Ilustracja 7: Encja Event

5.1.6. Issue

Encja ta przechowuje wszystkie typy spraw w aplikacji. Znajdują się tutaj: pole tytułu sprawy (*title*), pole opisu (*description*), a także pola opisujące stan (*issuestatus*), priorytet (*issuepriority*) i typ sprawy (*issuetype*). Każda sprawa posiada, zupełnie tak samo jak komponent czy przyrząd pomiarowy, użytkownika, który ją stworzył oraz zmodyfikował, a także datę rozwiązania sprawy.

issue	
# id	int8 (19)
o active	bool
o createdate	timestamp
o description	text (2147483647)
o issuenumber	int8 (19)
o issuepriority	varchar (255)
o issuestatus	varchar (255)
o issuetype	varchar (255)
o modifydate	timestamp
o referencenumber	varchar (255)
o resolvedate	timestamp
o title	varchar (255)
o uid	varchar (255)
o assigneto_id	int8 (19)
o createby_id	int8 (19)
o modifyby_id	int8 (19)

Ilustracja 8: Encja Issue

5.1.7. Component

Ta tabela przechowuje wszystkie produkty i półprodukty z aplikacji.

component	
# id	int8 (19)
o active	bool
o createdate	timestamp
o description	text (2147483647)
o finalproduct	bool
o modifydate	timestamp
o name	varchar (255)
o referencenumber	varchar (255)
o uid	varchar (255)
o createby_id	int8 (19)
o customer_id	int8 (19)
o modifyby_id	int8 (19)
o supplier_id	int8 (19)

Ilustracja 9: Encja Component

5.1.8. FileEntity

Jest to tabela przechowująca serializowane pliki. Maksymalny rozmiar pliku wynosi 2MB. FileEntity przechowuje zarówno zdjęcia, dokumenty jakościowe, a także raporty pomiarowe. Kolumna determinująca rodzaj pliku to *doctype*.

fileentity	
# id	int8 (19)
o active	bool
o content	bytea (2147483647,0)
o createdate	timestamp
o description	varchar (255)
o doctype	varchar (255)
o filename	varchar (255)
o mimetype	varchar (255)
o modifydate	timestamp
o relateduid	varchar (255)
o status	bool
o uid	varchar (255)
o createby_id	int8 (19)
o item_id	int8 (19)
o modifyby_id	int8 (19)

Ilustracja 10: Encja FileEntity

5.1.9. Label

Encja Label służy do przechowywania etykiet, które mogą identyfikować grupę spraw takich jak: defekty jednego klienta, grupę zadań lub innowacji.

comment	
# id	int8 (19)
o active	bool
o content	text (2147483647)
o createdate	timestamp
o uid	varchar (255)
o creator_id	int8 (19)
o item_id	int8 (19)

Ilustracja 11: Encja Label

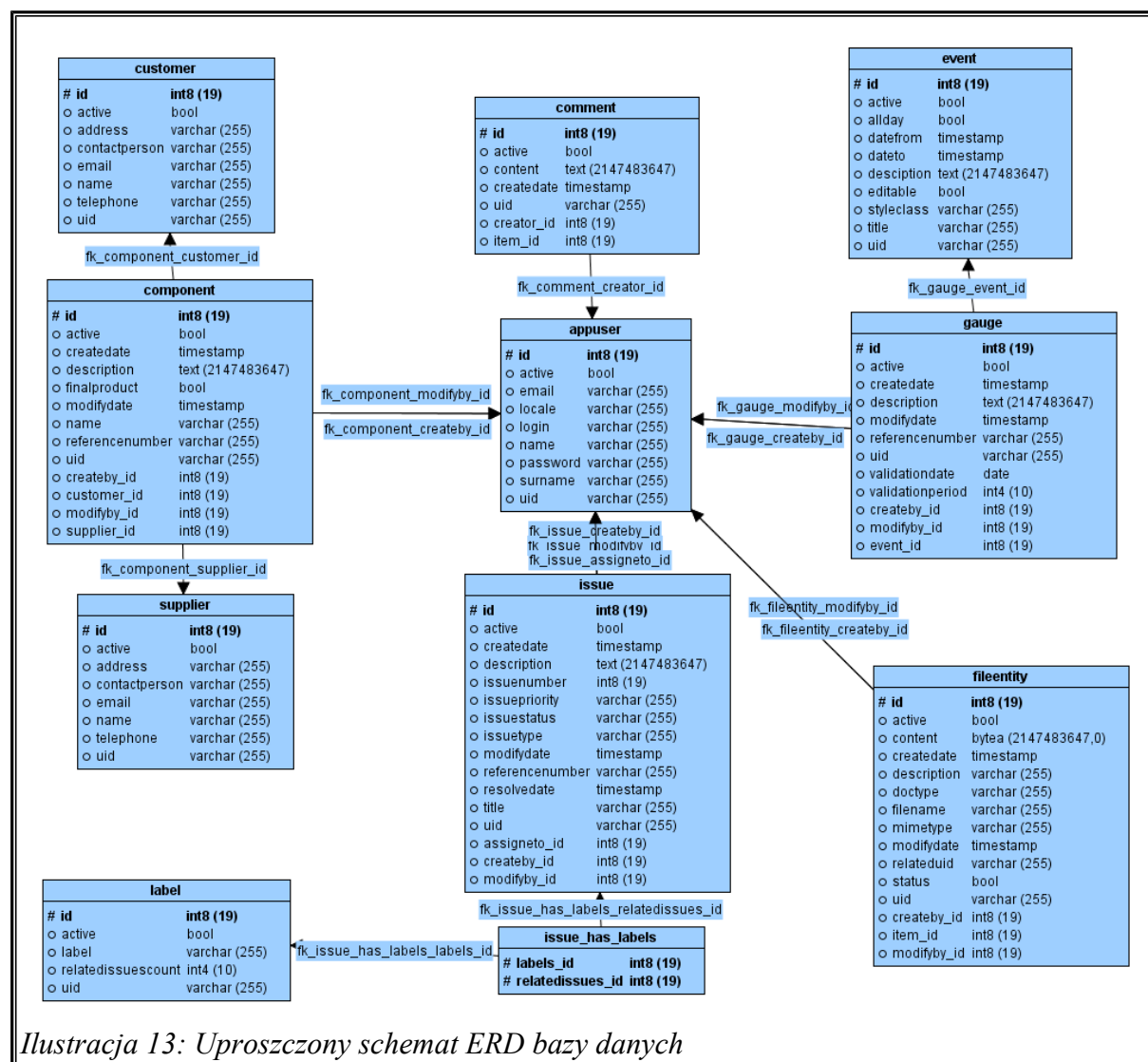
5.1.10. Comment

Ostatnia tabela służy do przechowywania komentarzy. Jej pola poza samym tekstem komentarza identyfikują autora i sprawę do której został dodany.

comment	
# id	int8 (19)
o active	bool
o content	text (2147483647)
o createdate	timestamp
o uid	varchar (255)
o creator_id	int8 (19)
o item_id	int8 (19)

Ilustracja 12: Encja Comment

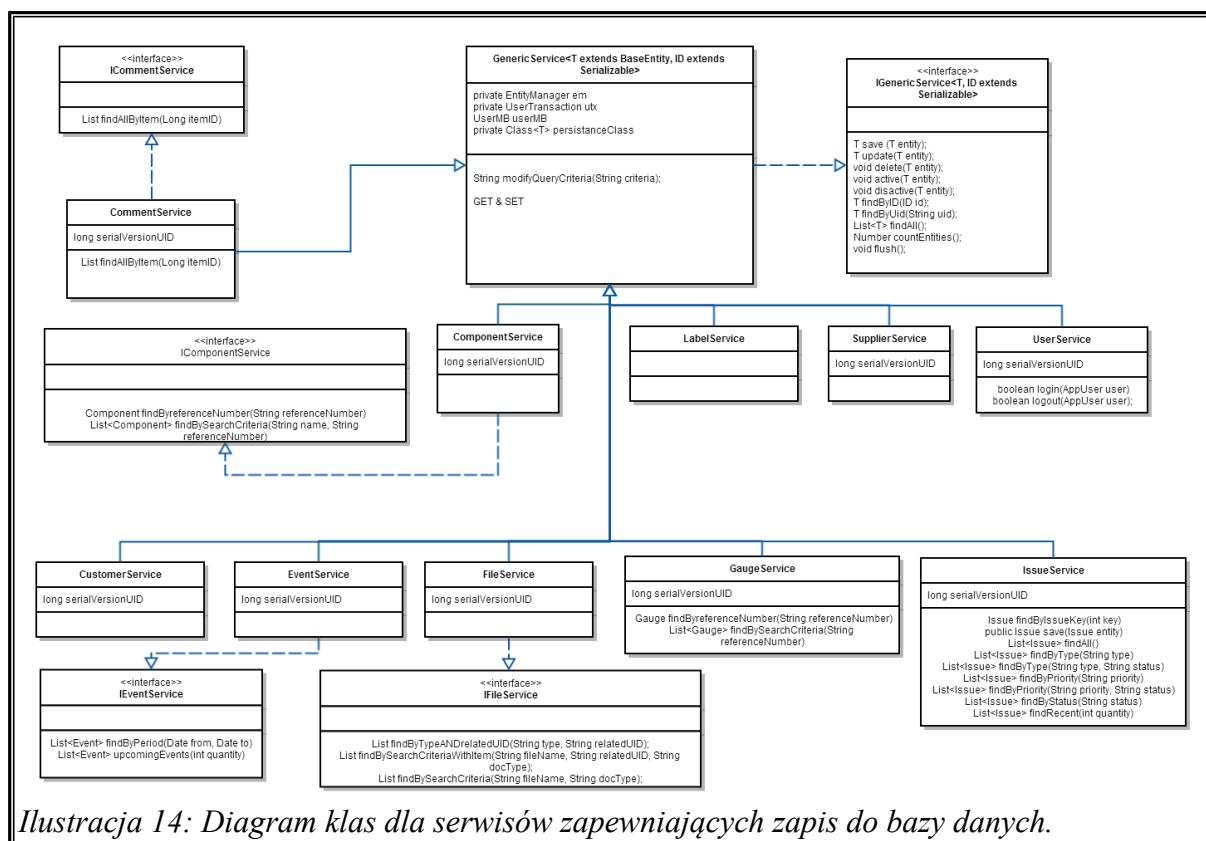
5.2. SCHEMAT ERD



Ilustracja 13: Uproszczony schemat ERD bazy danych

5.3. DIAGRAM KLAS SERWISÓW APLIKACJI

Aplikacja zbudowana jest w oparciu o serwisy. Każdy serwis dziedziczy z serwisu generycznego, a ten implementuje interfejs generyczny. Dzięki takiemu zabiegowi za pomocą samego generycznego serwisu możemy wykonywać praktycznie wszystkie podstawowe operacje. Serwisy pochodne z serwisu generycznego przeciążają niektóre ich metody w celu zwiększenia ich precyzyjności ewentualnie rozszerzenia podstawowego pakietu operacji.



6. SPECYFIKACJA ZEWNĘTRZNA

W ramach niniejszej pracy inżynierskiej zaprojektowana i utworzona została baza danych, aplikacja internetowa dostępna z poziomu przeglądarki internetowej oraz uruchomiony został serwer aplikacyjny.

6.1. INSTALACJA I PRZYGOTOWANIE BAZY DANYCH

W celu utworzenia bazy należy zainstalować bazę danych PostgreSQL. Uruchamiamy plik *postgresql-9.3.2-1-windows-x64.exe* zamieszczony na płycie dołączonej do tej pracy. Po zakończonej instalacji uruchamiamy aplikację do zarządzania bazą danych (pgAdmin III). Przechodzimy do miejsca instalacji i uruchamiamy skrypt:

```
<ścieżka instalacji>\bin\psql -U postgres -a -f <ścieżka do skryptu>\createQMSDBscript.sql
```

następnie możemy dodatkowo, tylko w celach testowych, uruchomić skrypt dodającego jednego użytkownika o danych autoryzacyjnych: login - „user1”, hasło - „pwd”.

```
<ścieżka instalacji>\bin\psql -U postgres -a -f <ścieżka do skryptu>\createUser.sql
```

Jeżeli nie uruchomimy drugiego skryptu musimy ręcznie dodać jednego użytkownika w bazie danych, aby można było się zalogować do systemu.

6.2. INSTALACJA I KONFIGURACJA SERWERA I APLIKACJI

Instalacja serwera polega na uruchomieniu pliku *java_ee_sdk-7-jdk7-windows-x64-m1.exe* z dołączonej płyty. Po instalacji podmieniamy jeszcze bibliotekę *javax.faces.jar* w katalogu:

<ścieżka_instalacji>\glassfish\modules

Po instalacji uruchamiamy:

<ścieżka_instalacji>\glassfish\bin\startserv.bat

Następnie:

<ścieżka_instalacji>\glassfish\bin\asadmin.bat deploy <plik WAR>

Po tej operacji możemy uruchomić aplikację wpisując w adres przeglądarki:

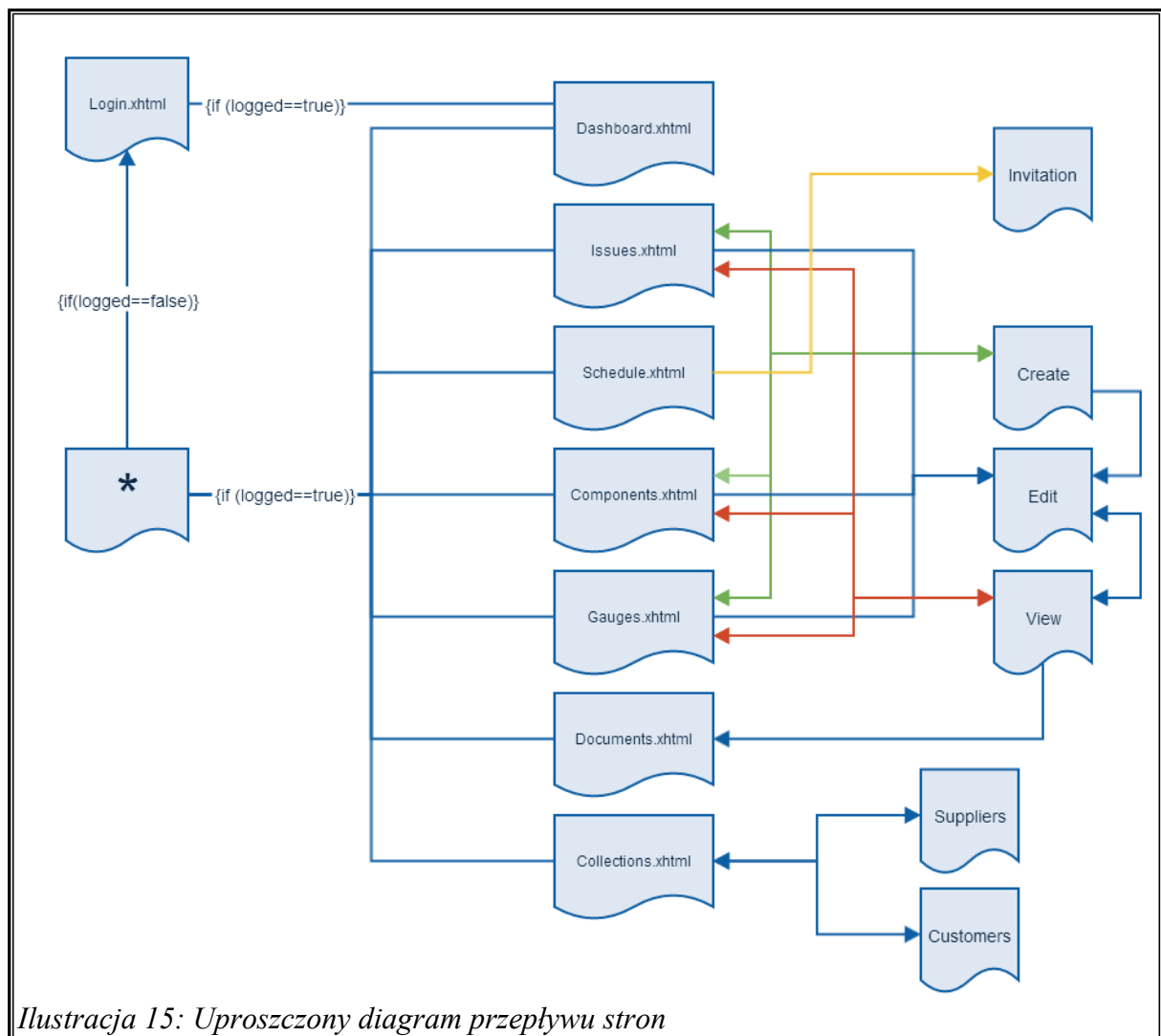
localhost:8080/QMSystem/

6.3. PRZEGLĄDARKI

Aplikacja testowana była głównie w przeglądarce Chrome (*Wersja 32.0.1700.76 m*). Testy również wykazały, że aplikacja działa także poprawnie w przeglądarce Firefox, Opera, Internet Explorer w aktualnych wersjach na dzień 27.01.2014.

6.4. PRZEPŁYW STRON

Poniżej przedstawiony został diagram przepływu stron w aplikacji. Pomimo iż, „Create”, „Edit” oraz „View” nie są wspólne, a każda ze stron „Issues.xhtml”, „Components.xhtml” oraz „Gauges.xhtml” posiada własny ich zestaw to mają one wszystkie podobny styl graficzny, co też pokazane będzie na kolejnych stronach niniejszej pracy.



Ilustracja 15: Uproszczony diagram przepływu stron

7. OPIS DZIAŁANIA

Aplikacja QMS (*Quality Management System*) stworzona została pod kątem prowadzenia w szerokim spektrum działu kontroli jakości. Jego pracownicy zobligowani są do kolekcjonowania wszelkiej dokumentacji pomiarowej, nadzorowania walidacji sprawdzianów pomiarowych, przygotowywaniu audytów jakości wewnętrznych jak i zewnętrznych oraz ustalanie reguł kontroli dla każdego komponentu. Wszystkie te prace muszą być rejestrowane, a ogrom danych z jakimi ma się do czynienia powoduje, że nie jest się w stanie zapamiętać tylu informacji. System pomaga w szybki sposób odnaleźć informacje, które są potrzebne. W kolejnych podrozdziałach zaprezentuję wszystkie funkcje programu.

7.1. DIALOG Z UŻYTKOWNIKIEM

System w celu dialogu z użytkownikiem wykorzystuje szereg narzędzi. Podstawowym z nich jest graficzne przedstawienie statusu dla wykonanej operacji. Rozróżniamy tu trzy różne typy:

- INFORMACJA – informuje użytkownika o poprawnie wykonanej operacji dodania lub aktualizacji.



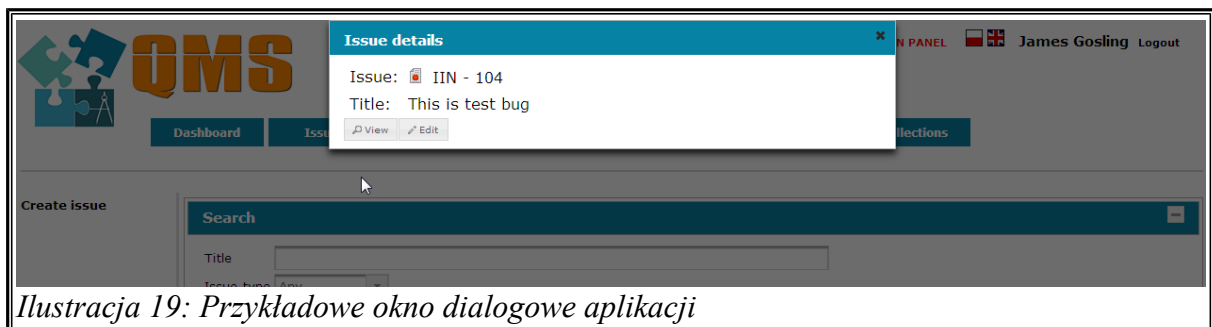
- OSTRZEŻENIE – służy do informowania użytkownika o błędzie dotyczącym restrykcji w ładowaniu pliku do systemu.



- BŁĄD – służy do informowania użytkownika o błędzie dotyczącym zapisu lub braku wypełnienia pola wymaganego.



Kolejnym elementem za pomocą której system nawiązuje kontakt z użytkownikiem jest okno dialogowe. Wymusza ono skupienie uwagi użytkownika na dokonaniu wyboru. Poniżej przedstawiono przykładowy widok okna dialogowego w aplikacji.



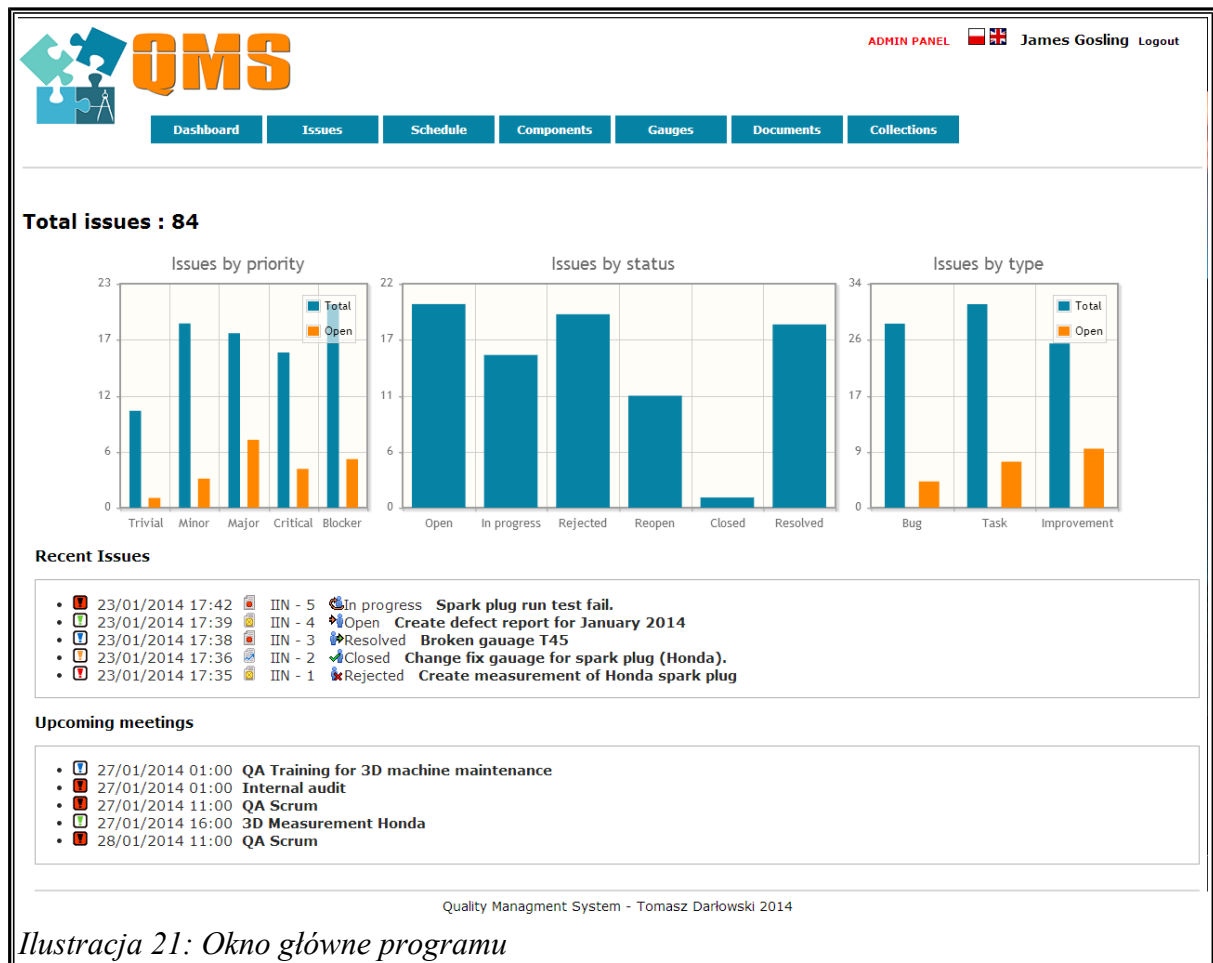
7.2. LOGOWANIE



Podczas uruchamiania aplikacji system sprawdza czy użytkownik jest zalogowany. Jeżeli nie, aplikacja poprosi użytkownika o wprowadzenie loginu i hasła. Kiedy system rozpozna użytkownika zostanie on przeniesiony na kolejną stronę.

Jeżeli użytkownik nie wyloguje się poprawnie jego sesja wygasa po określonym czasie (15minut). Przy próbie wykonania jakiegokolwiek operacji aplikacja przeniesie niezalogowanego użytkownika z powrotem do okna logowania. Wyróżniamy dwie grupy użytkowników: użytkownik i administrator. Jedyna różnica pomiędzy nimi to możliwość przejścia do panelu administracyjnego, w którym drugi z nich może zarządzać innymi kontami.

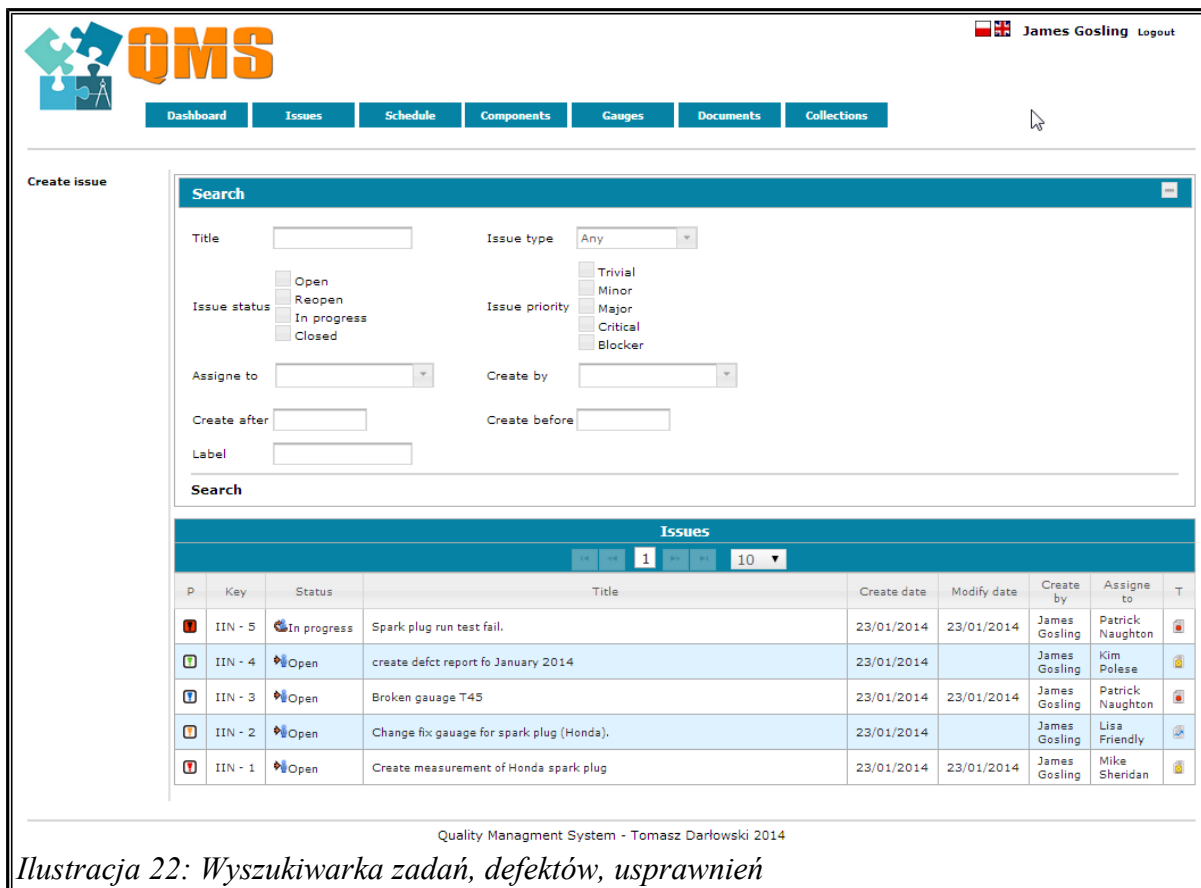
7.3. OKNO GŁÓWNE PROGRAMU



Ilustracja 21: Okno główne programu

Po zalogowaniu użytkownik zostaje przeniesiony do okna głównego. Użytkownik może tutaj zobaczyć ostatnio stworzone sprawy, zbliżające się spotkania czy zobaczyć statusy i statystyki defektów lub zadań.

7.4. DEFEKTY, ZADANIA, ULEPSZENIA



Create issue

Search

Title: Issue type:

Issue status: ☐ Open ☐ Reopen ☐ In progress ☐ Closed

Issue priority: ☐ Trivial ☐ Minor ☐ Major ☐ Critical ☐ Blocker

Assign to: Create by:

Create after: Create before:

Label:

Search

Issues

P	Key	Status	Title	Create date	Modify date	Create by	Assign to	T
IIN - 5	In progress	Spark plug run test fail.	23/01/2014	23/01/2014	James Gosling	Patrick Naughton		
IIN - 4	Open	create defct report fo January 2014	23/01/2014		James Gosling	Kim Polese		
IIN - 3	Open	Broken gauage T45	23/01/2014	23/01/2014	James Gosling	Patrick Naughton		
IIN - 2	Open	Change fix gauage for spark plug (Honda).	23/01/2014		James Gosling	Lisa Friendly		
IIN - 1	Open	Create measurement of Honda spark plug	23/01/2014	23/01/2014	James Gosling	Mike Sheridan		

Quality Managment System - Tomasz Darlowski 2014

Ilustracja 22: Wyszukiwarka zadań, defektów, usprawnień

Główną część aplikacji stanowi funkcjonalność związana z zarządzaniem zadaniami, defektami czy ulepszeniami w centrum jakościowym zakładu. Po przejściu w to miejsce, użytkownik ma możliwość ustalenia parametrów wyszukiwania zadań pod kątem takich parametrów jak: tytuł, typ, status, priorytet, do kogo dana sprawa została przydzielona lub kto jest jej twórcą. Dodatkowo możemy wyszukać zadanie na podstawie daty powstania lub etykiety.

Za pomocą funkcji „Dodaj sprawę” użytkownik może dodać kolejny element. Aplikacja przenosi użytkownika do formularza, gdzie z kolei wypełnia on pola i może przejść do edycji nowo stworzonej sprawy lub przejść z powrotem do strony z wyszukiwarką.

Okno edycji poza modyfikacją danych wprowadzonych podczas tworzenia, pozwala dodać dokumenty oraz pliki graficzne. Dodawanie plików graficznych posiada walidację na rozmiar pliku i ich typ. Dokumenty natomiast posiadają tylko walidację na rozmiar pliku.

Po wyszukaniu żądanej sprawy i kliknięciu na nią użytkownik może wybrać pomiędzy dwoma akcjami. Może iść kolejny raz do edycji lub do podglądu który dysponuje szeregiem funkcjonalności.

QMS James Gosling Logout

Dashboard Issues Schedule Components Gauges Documents Collections

Edit Issue Workflow Back

Spark plug run test fail.
IIN - 5

STATUS: In progress (assigned to Patrick Naughton)
PRIORITY: blocker
COMPONENTS: HON-157523-L - Honda spark plug
LABELS: defect, honda

DESCRIPTION: After run test QA worker indicate several spark plugs with heavy deposits. This can be caused by poor fuel quality, high oil consumption from a mechanically-worn engine or burning of coolant from damaged cylinder head seals and can promote glow ignitions (the deposits glow after).
Customer internal claim: 000157/2014/LoC/S

LINKED ISSUES:

- IIN - 2: Change fix gavage for spark plug (Honda).
- IIN - 3: Broken gavage T45

CREATE DATE: 23/01/2014 17:42
CREATE BY: James Gosling
MODIFY DATE: 23/01/2014 18:08
MODIFY BY: James Gosling
RESOLVE DATE:

Documents

1. Honda defect 000157.pdf - 23/01/2014

Images

Comments

James Gosling add a comment - 23/01/2014 17:54

Created measurements 3D - results send to customer.

Add comment

Ilustracja 23: Podgląd przykładowej sprawy

W oknie widoku mamy możliwość przejścia bezpośrednio do komponentu, który został wskazany jako referencja do tego defektu.

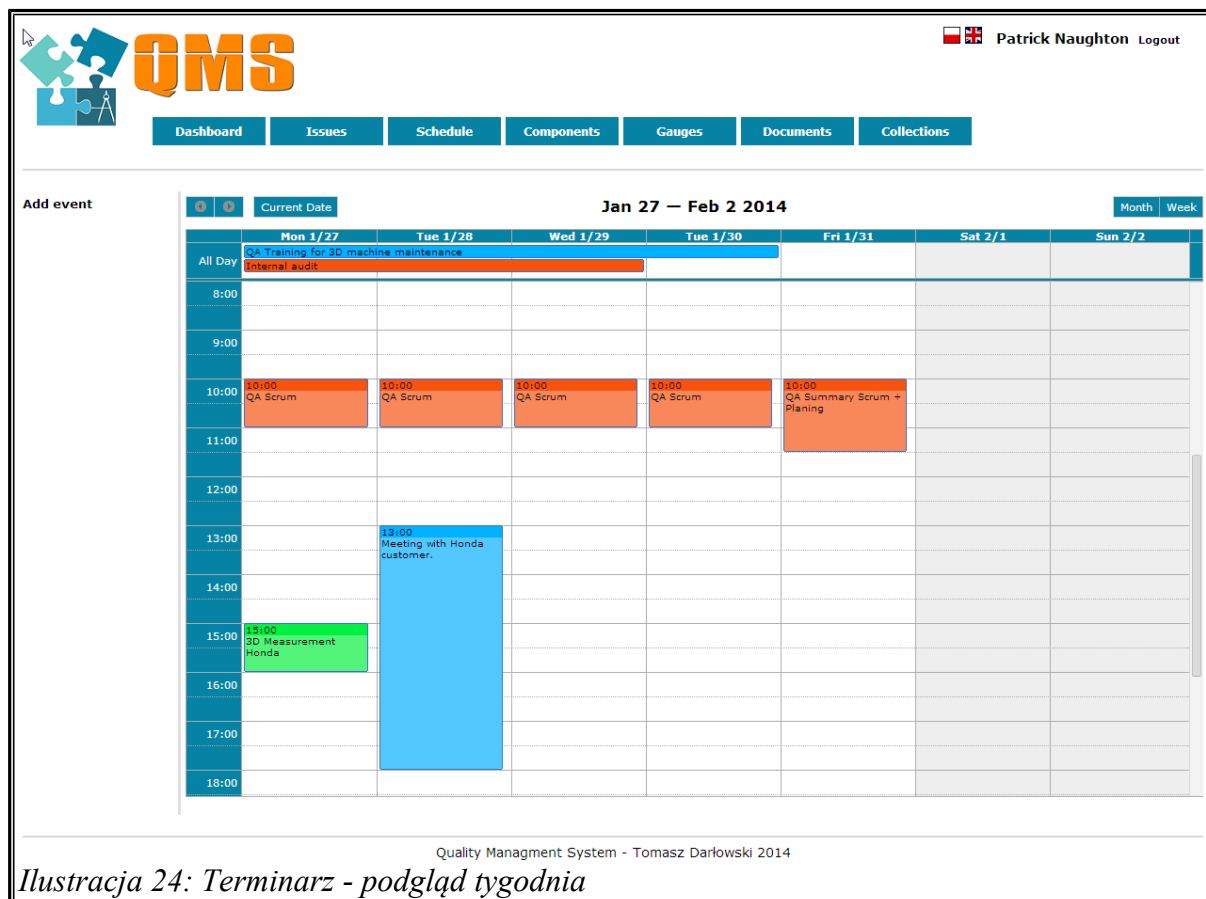
Możemy przeczytać opis oraz zobaczyć dołączone inne sprawy, a także pobrać dokumentację defektu oraz zobaczyć załączone zdjęcia.

Dodane w oknie edycji pliki pojawiają się w oknie podglądu i system pozwala na ich podgląd (*pliki graficzne*) lub pobranie (*pozostałe dokumenty*).

Dodatkowym elementem poprawiającym przepływ informacji o sprawie, jest możliwość dodawania komentarzy. Stanowią one swego rodzaju rejestr wykonanych postępów, a także mogą zawierać dodatkowe informacje na jej temat. Za pomocą menu w górnej części ekranu również możemy wykonać szereg funkcji. Możemy przejść do edycji, dołączyć kolejną sprawę jako odnośnik, przydzielić sprawę innemu pracownikowi lub zmienić jej status.

Dodawanie dokumentacji do sprawy możliwe jest tylko i wyłącznie za pomocą widoku edycji.

7.5. TERMINARZ



Ilustracja 24: Terminarz - podgląd tygodnia

Terminarz jest miejscem gdzie użytkownik może w łatwy sposób planować wykonywanie prac, tworzyć nowe spotkania i zapraszać na nie słuchaczy. Główny ekran udostępnia nam podgląd tygodniowy jak i miesięczny. Za pomocą odpowiednich kolorów w łatwy sposób identyfikujemy priorytet danego zdarzenia. Możemy także tworzyć ewent całodniowy lub kilkudniowy. Za pomocą przycisku „Dodaj zdarzenie” użytkownik zostaje przeniesiony do formularza, gdzie może sprecyzować czas rozpoczęcia i zakończenia meetingu, jego tytuł, opis oraz priorytet.

Kolejnym ważnym elementem po stworzeniu takiego spotkania jest możliwość wysyłania zaproszeń za pomocą funkcji „Wyślij zaproszenie”.

Możemy dodać innych pracowników jako słuchaczy i wysłać na ich skrzynki pocztowe zaproszenia z naszego programu.

Jeżeli wydarzenie zostanie zmienione (*np. zmieniony czas, tytuł*) system automatycznie rozsyła zaktualizowane zaproszenia dla wszystkich wcześniej zaproszonych użytkowników.



7.6. KOMPONENTY

Components

Search

Name:

Reference number:

Supplier:

Customer:

Final component: ☐ Final part ☐ Sub part ☒ Both

Active: ☒ Active ☐ In active ☐ Both

Search **Reset**

Name	Reference number	Customer	Supplier
A Foundation	31284	Mitsubishi Motor J...	Magnetti Marelli Bl...
Ac Ipsum Phasellus Consulting	4900TX	Ford GMT Co. Ame...	Garett TurboBoost ...
Ad Incorporated	30901	Honda JDM Co.	Magnetti Marelli Bl...
Adipiscing Ligula Aenean PC	44972	Honda JDM Co.	Magnetti Marelli Bl...
Adipiscing Mauris Molestie PC	8028	Ford GMT Co. Ame...	Koni Sport America
Aenean Limited	24649	Ford GMT Co. Ame...	Sparco Interior Cu...
Aliquam Adipiscing Institute	38721	Honda JDM Co.	Magnetti Marelli Bl...
Arcu LLP	02542	BMW Sauber Co.	Nitrous Express Ja...
Arcu PC	P2U 5C7	Daewoo Tico Korea...	Nitrous Express Ja...
Arcu Sed Corp.	V7T 016	Mitsubishi Motor J...	Sparco Interior Cu...

Quality Managment System - Tomasz Darlowski 2014

Ilustracja 26: Okno wyszukiwania komponentu

Ta strona pozwala użytkownikowi na szybkie wyszukanie produktu lub półproduktu w celu zdobycia najbardziej istotnych informacji. Wyszukiwarka pozwala na sprecyzowanie nazwy, numeru referencyjnego, klienta i odbiorcę, a także czy komponent jest produktem finalnym czy surowcem. W związku z tym, że system nie zezwala na usuwanie komponentów, użytkownik może tylko oznaczyć komponent jako nieaktywny. To również jest parametrem wyszukiwania.

Po wyszukaniużądanego komponentu użytkownik może przejść do podglądu danego elementu. Możemy stąd bezpośrednio przejść do półproduktów i produktów do których relacje zostały wskazane. Mamy również możliwość przejścia do spraw, w których dany komponent został użyty, a także do sprawdzianów pomiarowych, które ten komponent

mierzą. Panel „Dokumenty” pozwala na pobranie każdej dokumentacji, jaka została dołączona, a z kolei panel „Obrazy” pokazuje miniatury zdjęć komponentu. Miniatury można powiększyć klikając na nie.

Zupełnie tak jak w sprawach, użytkownik ma możliwość dodania plików graficznych i dokumentów. Proces ten odbywa się w oknie edycji do którego mamy dostęp z poziomu widoku lub po wybraniu komponentu z listy.

The screenshot displays the QMS web application interface. At the top, there is a header with the QMS logo, an 'ADMIN PANEL' link, a user profile for 'James Gosling', and a 'Logout' button. Below the header is a navigation bar with tabs: 'Dashboard', 'Issues', 'Schedule', 'Components', 'Gauges', 'Documents', and 'Collections'. The main content area is titled 'Edit Operation Back' and shows the details for a component named 'Spark plug for BMW' with ID '281247/BMW/14-1'. The details include fields for 'SUPPLIER' (Magnetti Marelli Bielsko - Biala), 'CUSTOMER' (BMW Sauber Co.), 'COMPONENT PARTS' (148/22547/VM/10-4), 'GAUGES' (CC-78), and 'DESCRIPTION' (Description of the component). To the right of these details are fields for 'CREATE DATE' (15/02/2014 15:39), 'CREATE BY' (James Gosling), 'MODIFY DATE' (15/02/2014 23:43), and 'MODIFY BY' (James Gosling). Below the details are three sections: 'Issues' with two items (1. test (IIN - 1), 2. Internal audit of BMW components (IIN - 2)), 'Documents' with one item (1. Control Plan rev.1.3.pdf - 15/02/2014 15:40), and 'Images' with one image (a spark plug). At the bottom of the interface, there is a footer with the text 'Quality Managment System - Tomasz Darłowski 2014'.

Ilustracja 27: Przykładowy podgląd komponentu

Ważnym elementem każdego produktu są raporty pomiarowe. Za pomocą menu „Operacje” użytkownik może załadować do systemu nowy raport określając jego status oraz może przejść do listy wszystkich raportów pomiarowych. Lista raportów w czytelny sposób identyfikuje, które raporty są ważne, a które nie, za pomocą kolumny Status.

Documents

Search

File name

Item reference

Type

From

To

Report status ☒ Both ☐ NOK ☐ OK

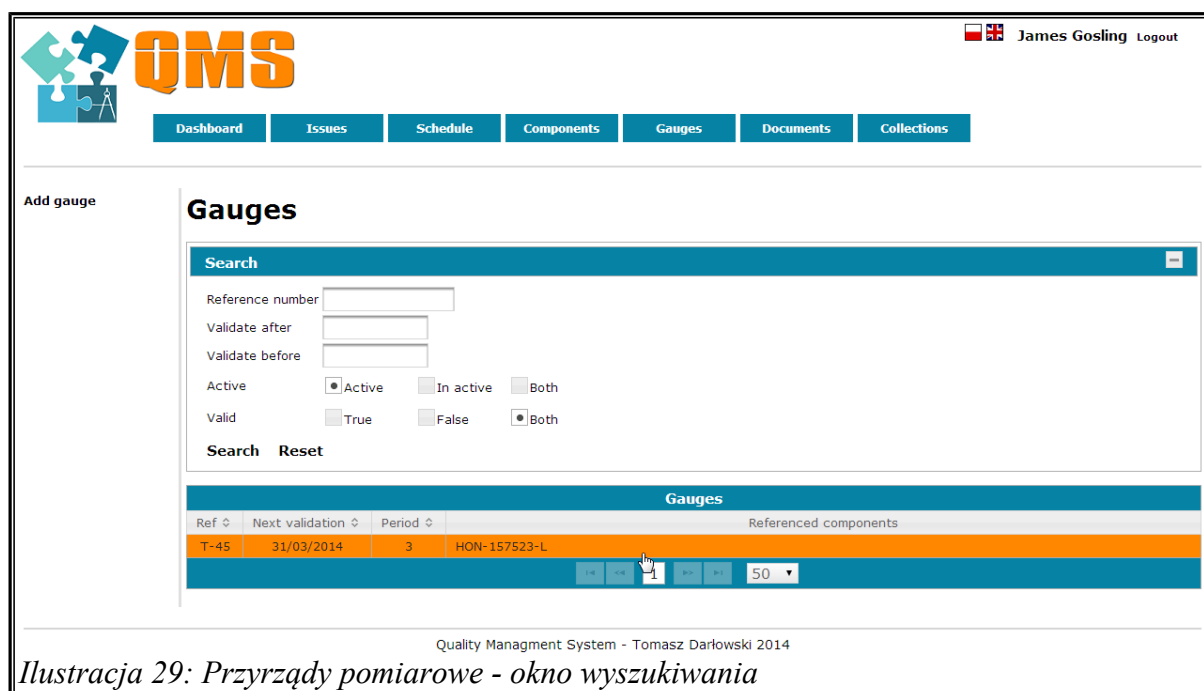
Search Reset

File name	Status	Date	Created by
vaski_net_component_diagram.pdf	NOK	06/02/2014 14:11	James Gosling
vaski_net_layers.pdf	NOK	06/02/2014 14:11	James Gosling
VASKI_NET_Architecture.docx	NOK	06/02/2014 14:12	James Gosling
VASKI_Staff.xlsx	OK	06/02/2014 14:12	James Gosling

Quality Managment System - Tomasz Darłowski 2014

Ilustracja 28: Przykładowy podgląd raportów jednego z komponentu

7.7. PRZYZRZĄDY POMIAROWE

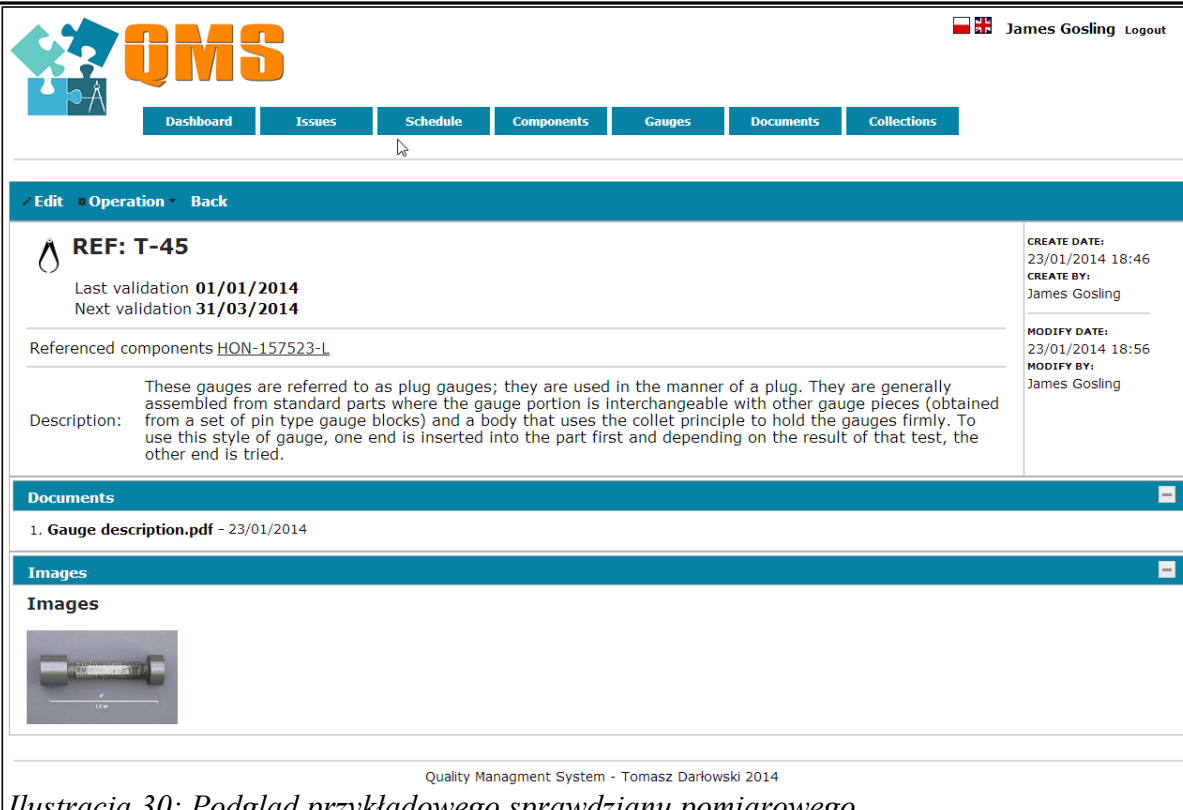


Ilustracja 29: Przyrządy pomiarowe - okno wyszukiwania

Magazyn przyrządów pomiarowych został zaimplementowany, podobnie jak komponenty. Tutaj również możemy przechowywać istotne dla działu jakości informacje, takie jak: data walidacji, okres ważności walidacji, referencje do komponentów, które dany sprawdzian mierzy, opis, zdjęcia oraz inną dokumentację. Kolejny raz system pozwala na wyszukiwanie sprawdzianu pomiarowego na podstawie szeregu najbardziej istotnych informacji.

Zupełnie tak jak i w sprawach użytkownik ma możliwość dodania plików graficznych i dokumentów. Proces ten odbywa się w oknie edycji do którego mamy dostęp z poziomu widoku lub po wybraniu komponentu z listy.

Podgląd sprawdzianu pozwala załadować raporty pomiarowe, z tymże załadowanie raportu powoduje automatyczną walidację przyrządu i odnowienie jego daty przydatności.



QMS

James Gosling Logout

Dashboard Issues Schedule Components Gauges Documents Collections

Edit Operation Back

REF: T-45

Last validation **01/01/2014**
Next validation **31/03/2014**

Referenced components **HON-157523-L**

Description: These gauges are referred to as plug gauges; they are used in the manner of a plug. They are generally assembled from standard parts where the gauge portion is interchangeable with other gauge pieces (obtained from a set of pin type gauge blocks) and a body that uses the collet principle to hold the gauges firmly. To use this style of gauge, one end is inserted into the part first and depending on the result of that test, the other end is tried.

CREATE DATE: 23/01/2014 18:46
CREATE BY: James Gosling

MODIFY DATE: 23/01/2014 18:56
MODIFY BY: James Gosling

Documents

1. Gauge description.pdf - 23/01/2014

Images

Images

Quality Management System - Tomasz Darlowski 2014

Ilustracja 30: Podgląd przykładowego sprawdzianu pomiarowego

Zarówno podczas tworzenia jak i ładowania raportu, tworzone jest lub modyfikowane zdarzenie w terminarzu przedstawiające datę końca walidacji. Na terminarzu jest ono oznaczone w zupełnie odmienny sposób niż reszta wydarzeń.



Current Date

Mon 3/31 Tue 4/1

All Day Gauge T-45

8:00

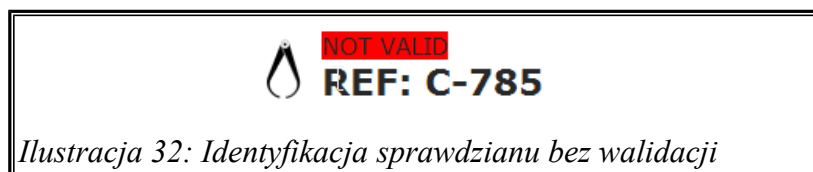
9:00

10:00

11:00

Ilustracja 31: Identyfikacja końca daty ważności sprawdzianu

Jeżeli natomiast sprawdzian nie posiada walidacji to jest to oznaczone na jego podglądzie odpowiednią etykietą, a wydarzenie znika z terminarza.



Ilustracja 32: Identyfikacja sprawdzianu bez walidacji

7.8. DOKUMENTY

Documents

Search

File name

Item reference

Type

From

To

Report status ☒ Both ☐ NOK ☐ OK

Search Reset

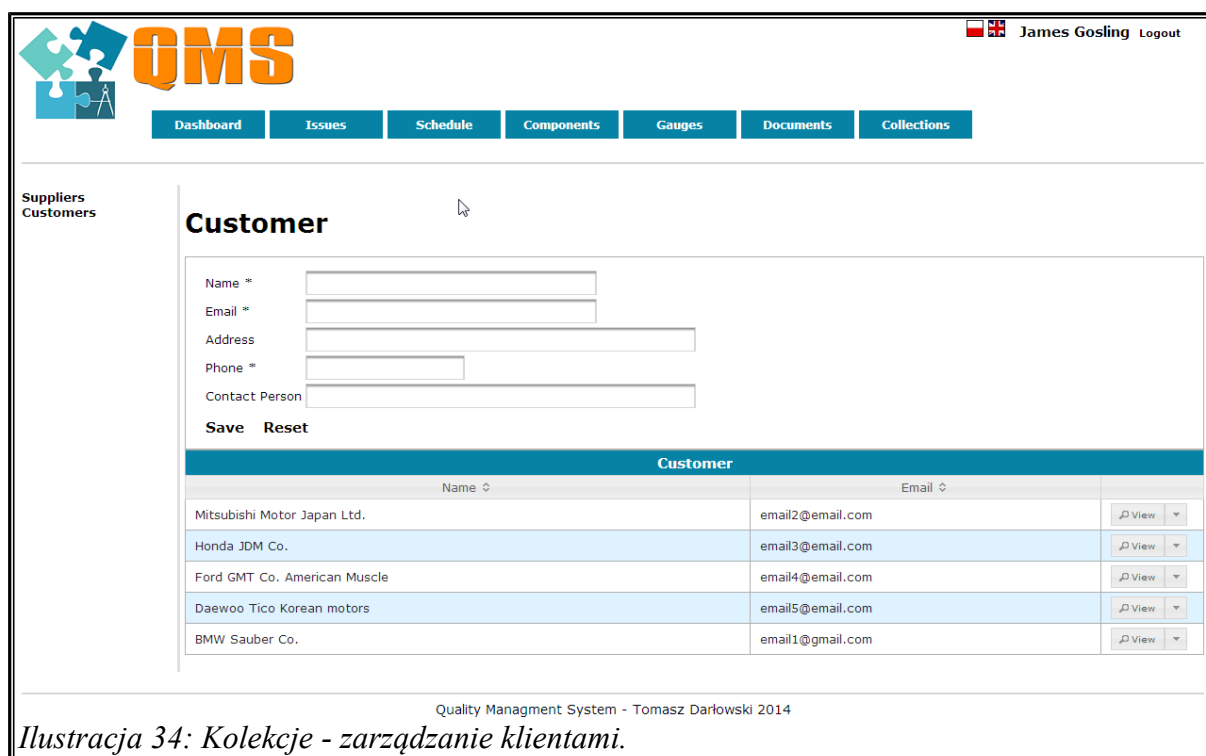
File name	Status	Date	Created by
vaski_net_component_diagram.pdf	NOK	06/02/2014 14:11	James Gosling
vaski_net_layers.pdf	NOK	06/02/2014 14:11	James Gosling
VASKI_NET_Architecture.docx	NOK	06/02/2014 14:12	James Gosling
VASKI_Staff.xlsx	OK	06/02/2014 14:12	James Gosling

Quality Managment System - Tomasz Darłowski 2014

Ilustracja 33: Dokumenty - okno wyszukiwania

Do wyszukiwania wszystkich dokumentów służy zakładka „Dokumenty”. Użytkownik może przeglądać tu zarówno raporty pomiarowe, jak i wszelkie inne dokumenty, które zostały załadowane do systemu. Po wyszukaniużądanego dokumentu użytkownik może pobrać dokument na dysk lokalny. Po wskazaniu kursorem nazwy pliku, w chmurce podpowiedzi wyświetla się adnotacja dodana do pliku.

7.9. KOLEKCJE



Customer

Name *

Email *

Address

Phone *

Contact Person

Save **Reset**

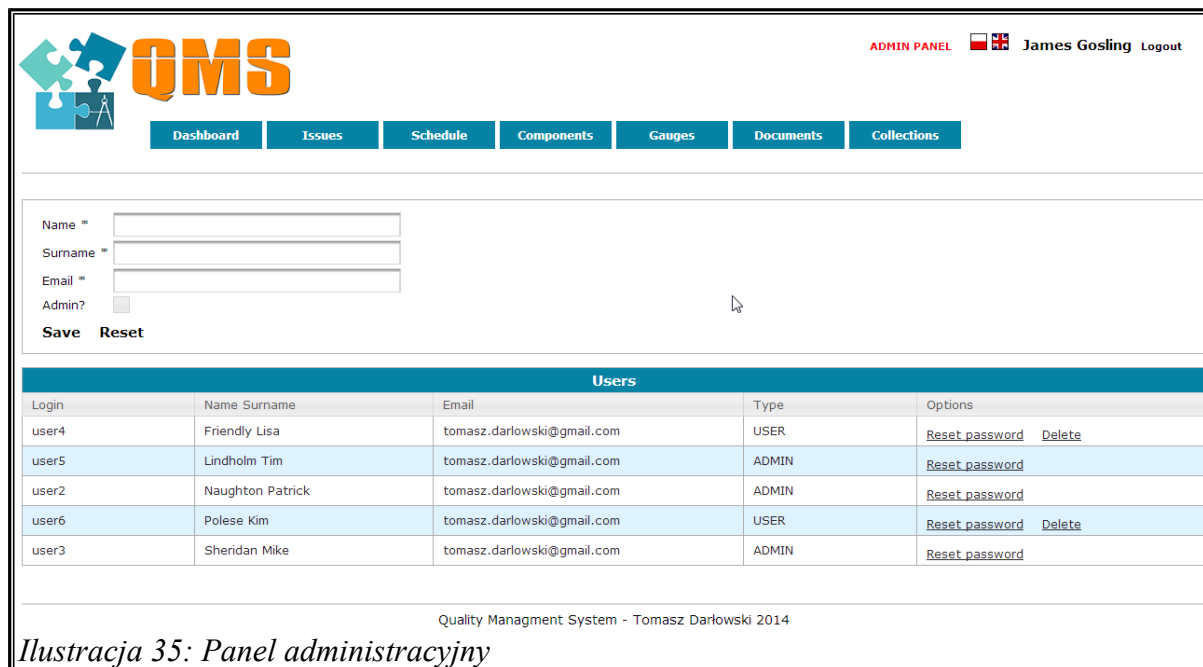
Customer		
Name ↕	Email ↕	
Mitsubishi Motor Japan Ltd.	email2@email.com	<input type="button" value="View"/>
Honda JDM Co.	email3@email.com	<input type="button" value="View"/>
Ford GMT Co. American Muscle	email4@email.com	<input type="button" value="View"/>
Daewoo Tico Korean motors	email5@email.com	<input type="button" value="View"/>
BMW Sauber Co.	email1@gmail.com	<input type="button" value="View"/>


Quality Managment System - Tomasz Darłowski 2014

Ilustracja 34: Kolekcje - zarządzanie klientami.

Ostatnia zakładka pozwala na zarządzanie bazą klientów i dostawców. Możemy dodawać tutaj nowych jak i edytować dane aktualnie już istniejących klientów oraz dostawców.

7.10. PANEL ADMINISTRACYJNY



ADMIN PANEL  James Gosling Logout

Dashboard Issues Schedule Components Gauges Documents Collections

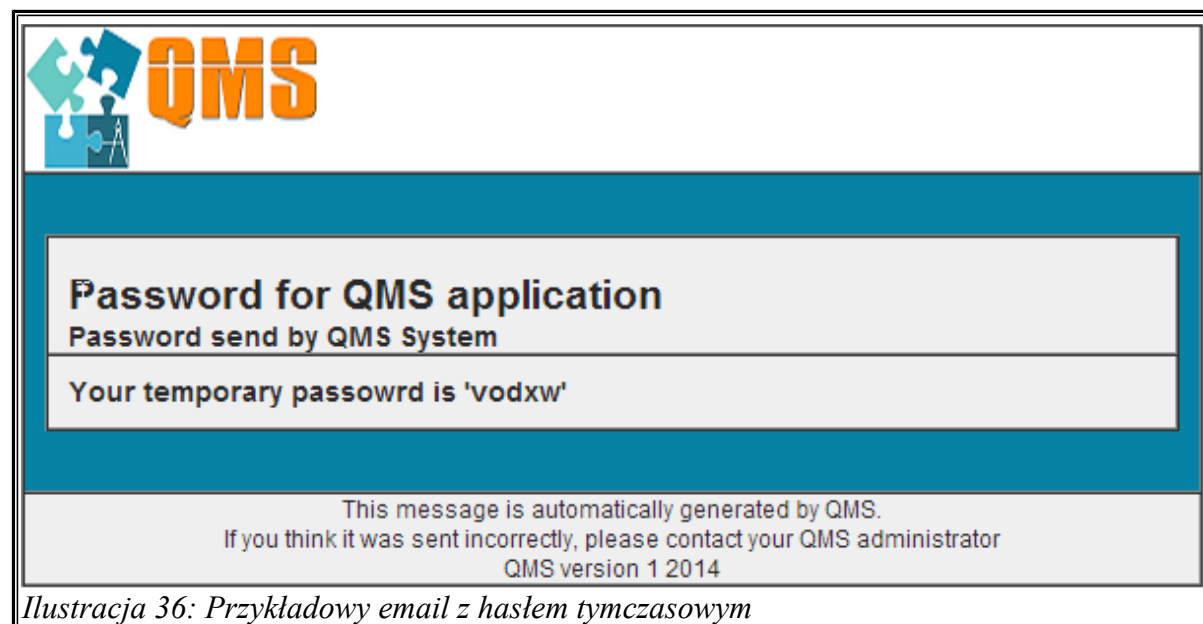
Name *
Surname *
Email *
Admin? ☐
Save Reset


Users				
Login	Name Surname	Email	Type	Options
user4	Friendly Lisa	tomasz.darlowski@gmail.com	USER	Reset password Delete
user5	Lindholm Tim	tomasz.darlowski@gmail.com	ADMIN	Reset password
user2	Naughton Patrick	tomasz.darlowski@gmail.com	ADMIN	Reset password
user6	Polese Kim	tomasz.darlowski@gmail.com	USER	Reset password Delete
user3	Sheridan Mike	tomasz.darlowski@gmail.com	ADMIN	Reset password

Quality Managment System - Tomasz Darlowski 2014

Ilustracja 35: Panel administracyjny

Panel administracyjny służy do zarządzania użytkownikami systemu, a także w razie konieczności resetowania ich haseł dostępu. Po zresetowaniu hasła system wysyła wiadomość email z wygenerowanym hasłem tymczasowym.





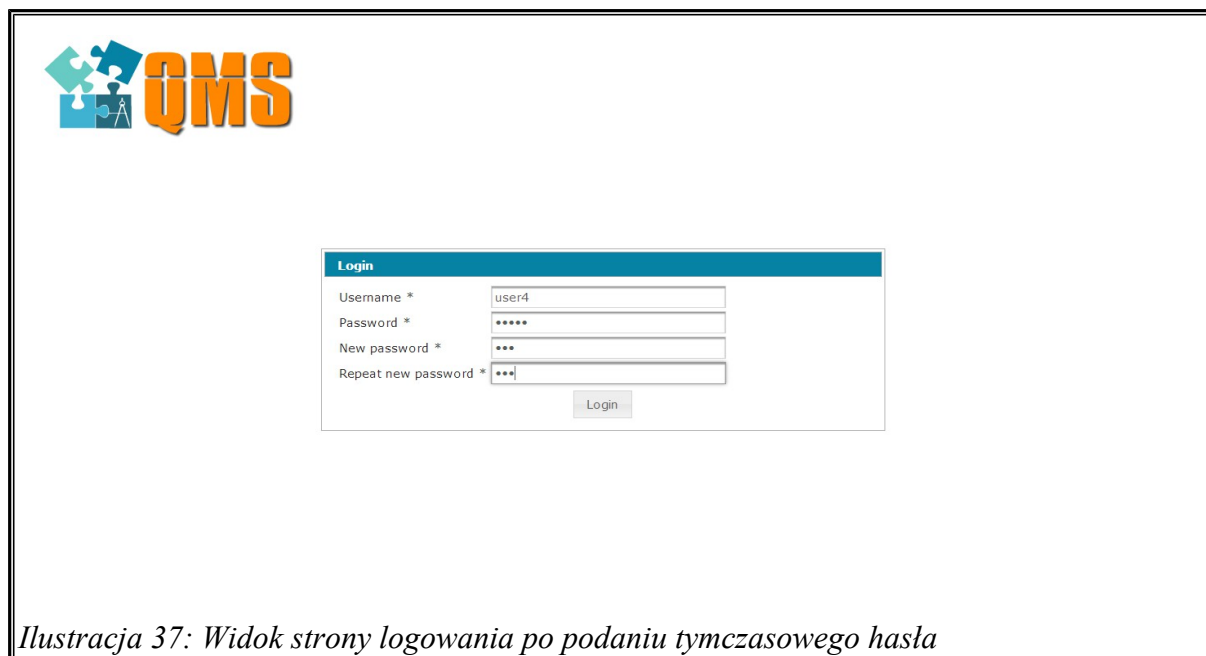
Password for QMS application
Password send by QMS System

Your temporary passowrd is 'vodxw'

This message is automatically generated by QMS.
If you think it was sent incorrectly, please contact your QMS administrator
QMS version 1 2014

Ilustracja 36: Przykładowy email z hasłem tymczasowym

Po ponownej próbie zalogowania użytkownik z zresetowanym hasłem musi ustawić nowe hasło zamiast tymczasowego. W tym celu w oknie logowania pojawiają się dodatkowe pola.

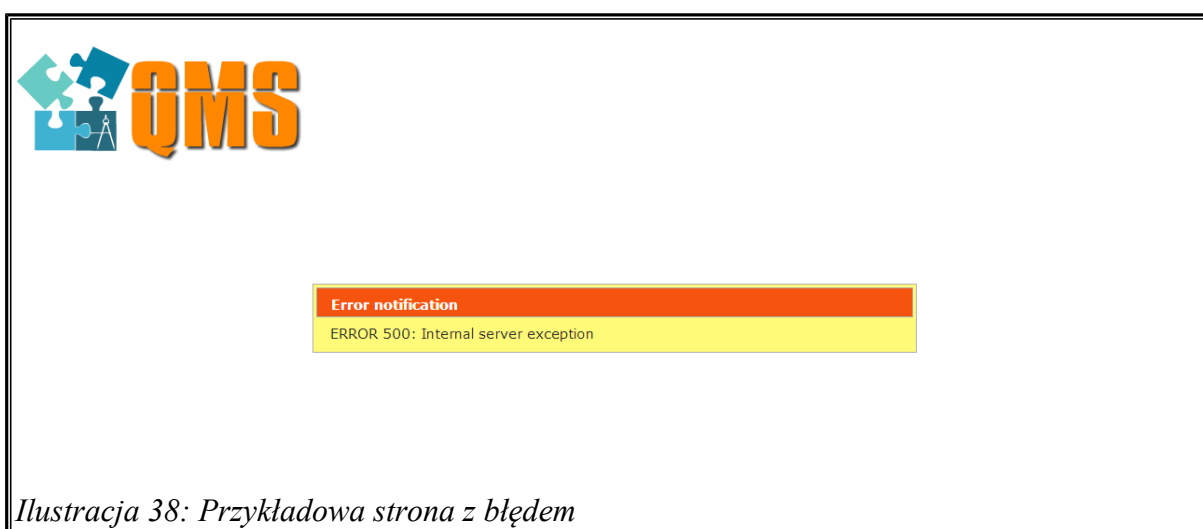


The screenshot displays the QMS login interface. At the top left is the QMS logo, which consists of a blue puzzle piece icon and the letters 'QMS' in orange. Below the logo is a login form with a blue header bar labeled 'Login'. The form contains four input fields: 'Username *' with the value 'user4', 'Password *' with masked characters '*****', 'New password *' with masked characters '***', and 'Repeat new password *' with masked characters '***'. A 'Login' button is positioned at the bottom right of the form.

Ilustracja 37: Widok strony logowania po podaniu tymczasowego hasła

7.11. KOMUNIKATY O BŁĘDACH

Błędy są nieodzowną częścią każdej aplikacji. Podstawowa walidacja danych i prezentacja o jej błędach została opisana w rozdziale poświęconym dialogowi pomiędzy systemem, a użytkownikiem. Nie mniej jednak system narażony jest na wiele nieoczekiwanych błędów, których twórca nie jest w stanie przewidzieć. Jeżeli wystąpi nieoczekiwany błąd system poinformuje użytkownika o tym w przedstawiony poniżej sposób.



Ilustracja 38: Przykładowa strona z błędem

7.12. ZMIANA JĘZYKA

Użytkownik ma możliwość zmiany języka podczas użytkowania aplikacji. W tym celu w górnym prawym rogu zostały umieszczone ikony flag odpowiadające odpowiedniemu językowi. Za ich pomocą użytkownik zmienia język interfejsu na wybrany. Informacja ta jest automatycznie przypisywana do konta użytkownika.

8. TESTY

Przypadki testowe dla testów systemowych pokrywały wszystkie funkcjonalności i były przeprowadzane też jako testy regresyjne po każdej kolejnej wersji systemu. W końcowej fazie wytwarzania systemu, przeprowadziłem testy akceptacyjne, aby sprawdzić pokrycie wymagań systemu oraz obsługi błędów walidacji. Wszystkie testy przyniosły wynik pozytywny. Jednakże warto nadmienić, iż zgodnie z certyfikatem ISTQB i standardami testowania aplikacji, testy systemowe aplikacji powinny być przeprowadzone przez osobę trzecią, a już na pewno nie przez twórcę systemu. W związku z tym testy mogą okazać się niewiarygodne dla potencjalnego odbiorcy i aby zagwarantować mu wysoką jakość produktu należałoby przekazać aplikację grupie testerów.

9. PODSUMOWANIE

Celem pracy było stworzenie systemu wspomagającego pracę działu kontroli jakości. Cel ten został osiągnięty. System pozwala na realizację podstawowych zadań działu kontroli jakości, czyli zarządzanie produktami i ich dokumentacją, defektami i zadaniami, przyrządami pomiarowymi oraz czasem. Architektura systemu pozwala na jego rozbudowę lub dostosowanie do indywidualnych wymagań odbiorcy w prosty i łatwy sposób.

Uważam, że system bardzo dobrze sprawdzi się w małych i średnich firmach. Nie ma on szans konkurować z takimi systemami jak Jira czy QC, ale jako darmowa alternatywa wprowadza wiele ciekawych i także potrzebnych rozwiązań.

Podczas wytworzenia systemu pojawiło się jeszcze wiele pomysłów, które być może w przyszłości zostaną jeszcze zaimplementowane. Chciałbym w przyszłości dodać system lokowania czasu pracy poszczególnym pracownikom, a także dodać obsługę działu technologicznego oraz działu zajmującego się prototypami.

10. SPIS ILUSTRACJI

Ilustracja 1: Diagram przypadków użycia dla użytkownika.....	14
Ilustracja 2: Diagram przypadków użycia dla administratora.....	14
Ilustracja 3: Diagram dla czynności dodania nowego zdarzenia.....	15
Ilustracja 4: Diagram przebiegu statusów spraw.....	16
Ilustracja 5: Encja AppUser.....	18
Ilustracja 6: Encje Supplier i Customer.....	19
Ilustracja 7: Encja Event.....	19
Ilustracja 8: Encja Issue.....	20
Ilustracja 9: Encja Component.....	21
Ilustracja 10: Encja FileEntity.....	21
Ilustracja 11: Encja Label.....	22
Ilustracja 12: Encja Comment.....	22
Ilustracja 13: Uproszczony schemat ERD bazy danych.....	23
Ilustracja 14: Diagram klas dla serwisów zapewniających zapis do bazy danych.....	24
Ilustracja 15: Uproszczony diagram przepływu stron.....	27
Ilustracja 16: Dialog z użytkownikiem - informacja.....	28
Ilustracja 17: Dialog z użytkownikiem - ostrzeżenie.....	28
Ilustracja 18: Dialog z użytkownikiem - błąd.....	29
Ilustracja 19: Przykładowe okno dialogowe aplikacji.....	29
Ilustracja 20: Okno logowania.....	30
Ilustracja 21: Okno główne programu.....	31
Ilustracja 22: Wyszukiwarka zadań, defektów, usprawnień.....	32
Ilustracja 23: Podgląd przykładowej sprawy.....	33
Ilustracja 24: Terminarz - podgląd tygodnia.....	35
Ilustracja 25: Przykładowe zaproszenie na spotkanie.....	36
Ilustracja 26: Okno wyszukiwania komponentu.....	37
Ilustracja 27: Przykładowy podgląd komponentu.....	38
Ilustracja 28: Przykładowy podgląd raportów jednego z komponentu.....	39
Ilustracja 29: Przyrządy pomiarowe - okno wyszukiwania.....	40
Ilustracja 30: Podgląd przykładowego sprawdzianu pomiarowego.....	41
Ilustracja 31: Identyfikacja końca daty ważności sprawdzianu.....	41

Ilustracja 32: Identyfikacja sprawdzianu bez walidacji.....	42
Ilustracja 33: Dokumenty - okno wyszukiwania.....	42
Ilustracja 34: Kolekcje - zarządzanie klientami.....	43
Ilustracja 35: Panel administracyjny.....	44
Ilustracja 36: Przykładowy email z hasłem tymczasowym.....	44
Ilustracja 37: Widok strony logowania po podaniu tymczasowego hasła.....	45
Ilustracja 38: Przykładowa strona z błędem.....	46

11. BIBLIOGRAFIA

1. „Thinking in Java” - Bruce Eckel
(ISBN 9780131872486)
2. „JavaServer Faces” - David Geary, Cay S. Horstmann
(ISBN 9878324629046)
3. „Wzorce projektowe : elementy oprogramowania obiektowego wielokrotnego użytku” -
Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides
(ISBN: ISBN 9788320434729)
4. „Core Java Techniki zaawansowane” - David Geary, Cay S. Horstmann
(ISBN: 9788324614837)
5. <http://stackoverflow.com/>
6. www.wikipedia.pl

12. ZAŁĄCZNIK

12.1. PŁYTA CD

Zawartość:

1. Katalog „Narzędzia i pliki”:

- *java_ee_sdk-7-jdk7-windows-x64-m1.exe* – instalator serwera aplikacji.
- *postgresql-9.3.2-1-windows-x64.exe* – instalator bazy danych.
- *jre-7u51-windows-x64.exe* – Wirtualna maszyna Java-y.
- *javax-faces.jar* – plik z implementacją JSF 2.2.5.
- *createQMSDBscript.sql* – skrypt tworzący bazę danych dla aplikacji.
- *CreateUser.sql* – skrypt opcjonalny do tworzenia użytkownika.

2. Katalog główny:

- *INSTRUKCJA.txt* – instrukcja instalacji
- *source-project.7z* – skompresowany plik źródłowy.
- *QMSystem.war* – plik instalacyjny aplikacji dla serwera Glassfish.

3. Katalog „Dokumenty”

- *Tomasz Darłowski - „System wspomagający pracę działu kontroli jakości”.odt* – Plik pakietu LibreOffice zawierający treść tego dokumentu.
- *Tomasz Darłowski - „System wspomagający pracę działu kontroli jakości”.pdf* – Plik w formacie PDF zawierający treść tego dokumentu.

13. OŚWIADCZENIE

Imię i nazwisko

Tomasz Darłowski

Nr albumu

2343

Kierunek, tryb

Informatyka, zaocznie

O Ś W I A D C Z E N I E

Oświadczam, że przedkładana praca dyplomowa pt.: „*System wspomagający pracę działu jakości*” została napisana przeze mnie samodzielnie, tzn. nie zlecałem/am opracowania pracy dyplomowej, ani jej części osobom trzecim, jak również nie odpisywałem/am pracy dyplomowej, ani jej części od innych osób.

Jednocześnie przyjmuję do wiadomości, że w przypadku stwierdzenia popełnienia przeze mnie czynu polegającego na przypisaniu sobie autorstwa istotnego fragmentu lub innych elementów cudzej pracy, właściwy organ stwierdzi nieważność postępowania w sprawie nadania mi tytułu zawodowego (art. 193 ustawy z dnia 27 lipca 2005 r. – Prawo o szkolnictwie wyższym, Dz. U. 164, poz. 1365, z późn. zm.).

Oświadczam, że wersja pracy znajdująca się na przedłożonej przeze mnie płycie CD jest zgodna z wydrukiem komputerowym pracy.

Wyrażam zgodę na udostępnianie przez Uczelnię części książkowej pracy do wglądu osobom trzecim, w szczególności innym studentom SWSIM.

Chorzów, dnia

Podpis