



C.A.R.E

Centro Accoglienza Regionale Ematica



TEAM

Magliacane Hermann

Team Leader

Ranauro Giuliano

Developer

Ricciuto Luigi

Tester & Developer

Crovella Alessio

Tester & Developer

Polvere Donato

Tester & Technical Writer

Signoriello Sara

Tester (usability & UX Tester)

Capriglione Vincenzo

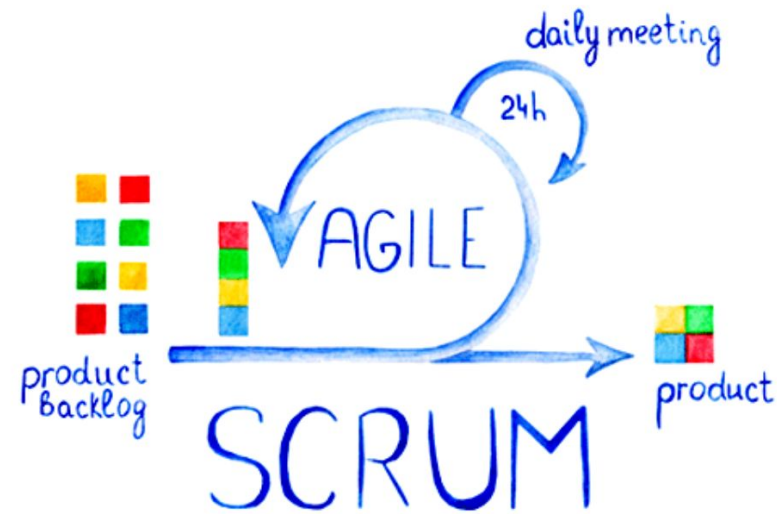
Technical Writer

Quatraro Raffaele

Developer



Team working



github.com/AccaEmme/CARE/commit/55997ee9cee4efc92a8e15941397e7789c4b3416#commitcommen

```
200 + if(serial == "") { alert("serial null"); k=1; }
201 + if(note == "") { alert("note null"); k=1; }
```



AccaEmme yesterday Owner

@Omni-star le note sono obbligatorie?



Omni-star 12 hours ago Author Collaborator

no, modificherò in seguito



Reply...



Problem Statement

CTT (Centro Trasfusionale Territoriale)

- Sacche di sangue locali CRUD
- Utenti locali CRUD
- Operazioni Offline
- Aggiunta e utilizzo di sacche
- Avviso sacche di sangue in scadenza

CCS (Centro Controllo e Smistamento)

- Aggiunta e rimozione di CTT
- Raccolta e gestione dati dei CTT
- Controllo prima e dopo la condivisione



I NOSTRI OBIETTIVI



Open Software

Il sistema C.A.R.E può essere utilizzato da qualsiasi ospedale senza costi perché open source



Facile Da Usare

Il sistema è facile da usare per chiunque nel reparto medico



Niente Sangue Spreco

Grazie alla comunicazione tra ospedali, le sacche non verranno più buttate

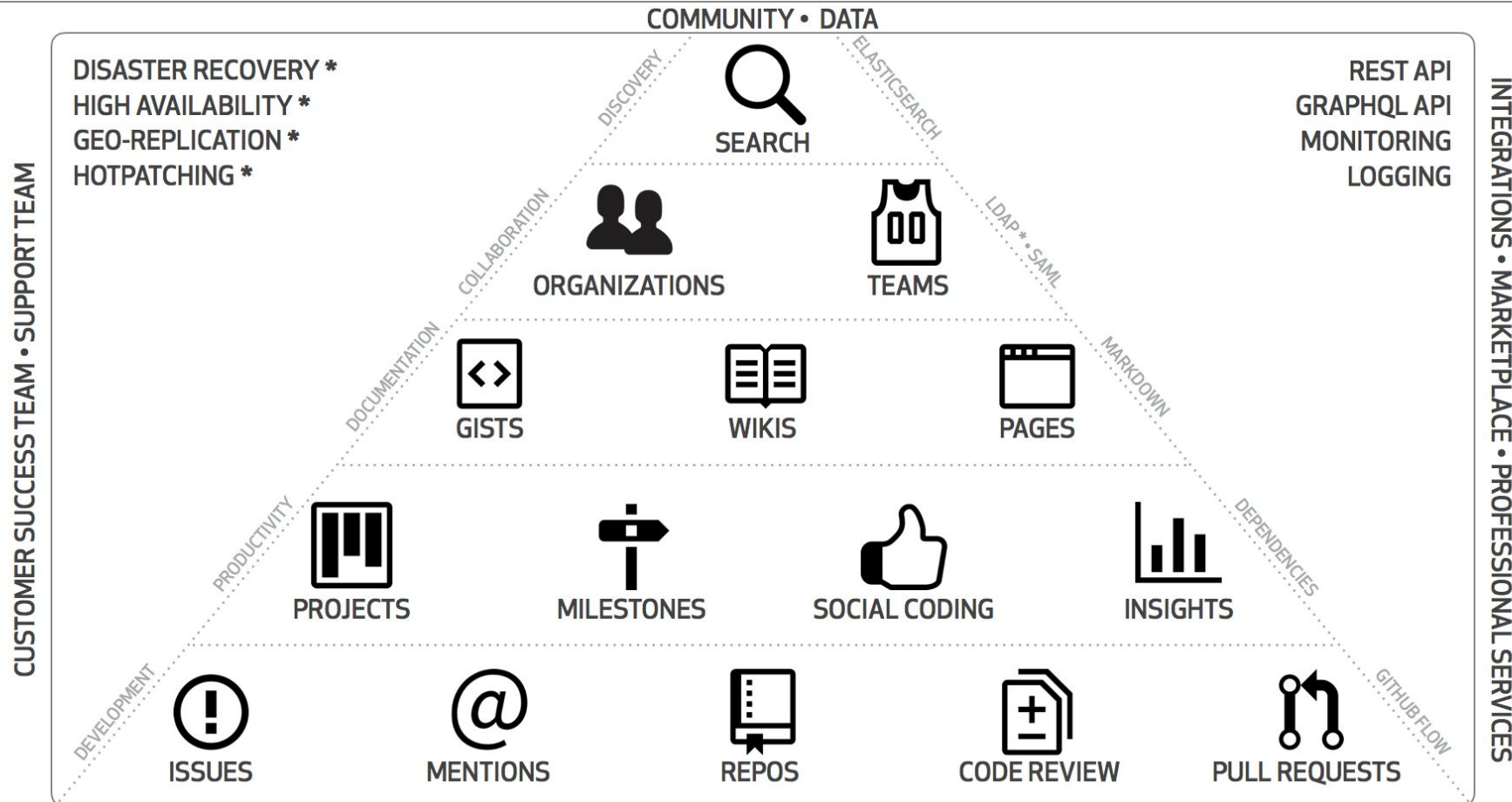


Work Offline

Anche se il CCS non risponde, il sistema funziona indipendentemente da ciò



Source Code Management



- Source - <https://github.com/AccaEmme/CARE>
- Wiki - <https://github.com/AccaEmme/CARE/wiki>
- Discussions - <https://github.com/AccaEmme/CARE/discussions>
- Insights - <https://github.com/AccaEmme/CARE/pulse>
- Bug Tracking - <https://github.com/AccaEmme/CARE/issues>



CONCEPT

CUSTOMER

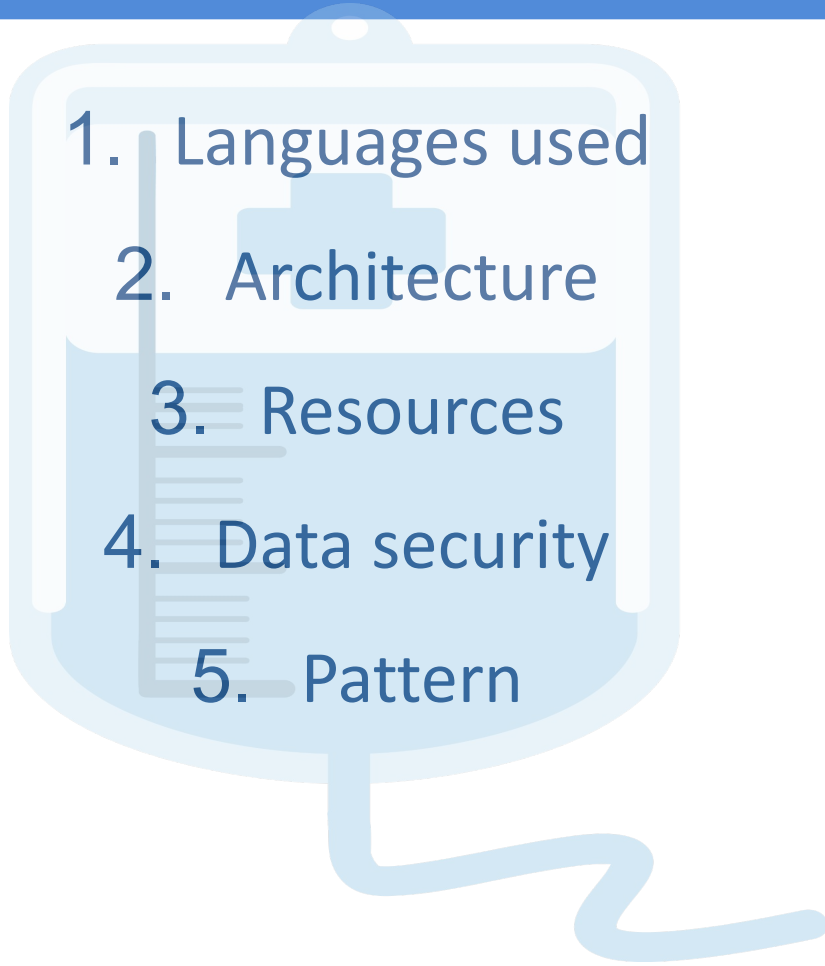


* ENTERPRISE ONLY





Overview Of The System

- 
- A light blue medical drip chamber graphic with a central vertical tube and horizontal slots. It contains a list of five items. A wavy line extends from the bottom of the chamber.
1. Languages used
 2. Architecture
 3. Resources
 4. Data security
 5. Pattern



Languages Used



Java

with IDEs: Eclipse, Visual Studio, IntelliJ



(Spring Framework)



HTML, CSS, Vanilla Javascript



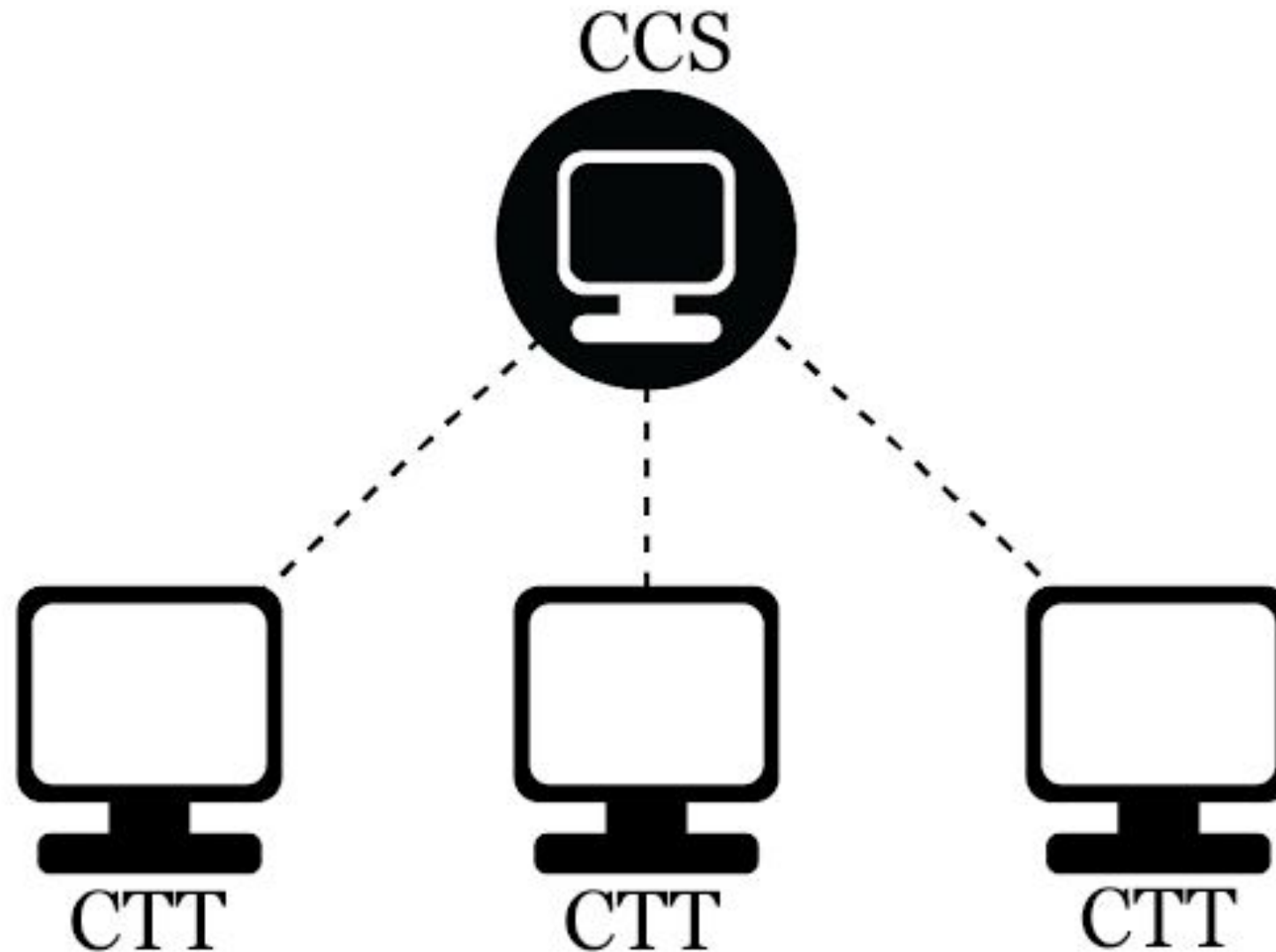
PHP



jQuery, RegExp



Architecture: client-server





Constraints & Definitions

- The bloodbag serial is an univoque protocol syntactically validated via RegExp (i.g. IT-NA206000-Apos-20210416-0001)
- The serial can't be set by users, set only by the system.
- Only administrator can create users
- "1900-01-01"(timestamp -2208996000) as null date
- Password complexity RegExp: "(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#&()-[{]}:; ',?/*~\$^+=<>])(?=.*{8,})";

Regex explanation of password secure requirements:

- * Password must contain at least one digit [0-9].
- * Password must contain at least one lowercase Latin character [a-z].
- * Password must contain at least one uppercase Latin character [A-Z].
- * Password must contain at least one special character like ! @ # & ().
- * Password must contain a length of at least 8 characters and a maximum of 20 characters.

^	# start of line
(?=.*[0-9])	# positive lookahead, digit [0-9]
(?=.*[a-z])	# positive lookahead, one lowercase character [a-z]
(?=.*[A-Z])	# positive lookahead, one uppercase character [A-Z]
(?=.*[!@#&()-[{]}:; ',?/*~\$^+=<>])	# positive lookahead, one of the special character in this [..]
.	# matches anything
{8,20}	# length at least 8 characters and maximum of 20 characters
\$	# end of line

- ...





Best Practices & Design Pattern

Some “Best Practice” used:

- **Indentation:** tabbed (not spaced) indentation to make code human readable easy
- **var names:** variables with significative name
- **Documentation:** every filesource is well commented, without useless info. In Java we provide javadoc too.
- **Code optimization:** reused properly functions(in javascript) and implemented simple and short classes in Java (no “God Classes” has been made”).
- **Date timestamp** let’s use timestamp instead of strings or object.
- **Enumerator**
- **SALT**
- **login anti-bruteforce** if attempts more than 3, user is blocked and should be managed by admin.

Some Pattern used:

- **MVC:** clear and easy file organization in Model-View-Control approach.
- **DAO PATTERN:** we defined Bean through “Data Access Object pattern” that represents the entity as in RDBMS. It will be used from “JPA Entity Manager”
- **Repository Pattern** provides an abstraction of data, so that your application can work with a simple abstraction that has an interface approximating that of a collection.
- **Iterator Pattern** In most of list we returned an iterator.
- **Composite Pattern** in report class we manage “report” for users and for bloodbag as composition of the same element
- **Decorator Pattern** CORS filter
- **Singleton Pattern** in SecurityContext we need to be ensure that we create only one instance of this class.
- **Builder Pattern:** our Password class that manages an object “Password”. To avoid the overload of constructor User.





Data Security: Pattern & Best Practices

- Interfaces and patterns
- SALT
- Password cifrata nel database
- Password Complexity pattern constraint
- GitGuardian check uploaded code
- JWT token per i metodi esposti
- protocollo HTTPS per cifrare la comunicazione
- Attempts per evitare bruteforce
- Different DB users



1 secret detected!

MongoDB URI

2021-06-30 05:54:11 pm (UTC)

AccaEmme/CARE (commit 45af576)

See on GitGuardian

SEE ON GITHUB

	id_user	active_user	creation_date	email	last_access	login_attempts	password	temppass	user_role	userr
▶	1	0	1625071544336	NULL	-2208988800000	0	\$2a\$10\$MsUNMafydo7uZ3nRzsPJjOzFjxkEhZQs...		ROLE_ADMINISTRATOR	giulian
	2	0	1625073465660	NULL	-2208988800000	0	\$2a\$10\$8aFKj1HEog2BNov.PN2mHu.N.bmPs5h...		ROLE_ADMINISTRATOR	giulian

Password.java

Constants.java

```
34
35 // ##### User Strings #####
36 public static final String PASSWORD_SALT = "CanforaMarkUs30L";
37 // public static final String USER_DEFAULT_TEMP_PASS = "CARE:Changemenow";
38 public static final int USER_TEMPPASS_LENGTH = 10;
```

Password.java

Constants.java

```
87- public static String getBCrypt(String input) {
88
89     PasswordEncoder passwordEncoder = new BCryptPasswordEncoder();
90     input += Constants.PASSWORD_SALT;
91     return passwordEncoder.encode(input);
}
```





Data Security: Pattern & Best Practices

- Interfaces and patterns
- SALT
- Password cifrata nel database
- Password Complexity pattern constraint
- GitGuardian check uploaded code
- JWT token per i metodi esposti
- protocollo HTTPS per cifrare la comunicazione
- Attempts per evitare bruteforce
- Different DB users

Encoded PASTE A TOKEN HERE

```
eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzdG9yZT  
k5IiwidXNlclJvbGUiOiJST0xFOX1NUT1JFTUF0Q  
UdFUiIsImV4cCI6MTYyNTg0NDUxNSwiaWF0Ijox  
NjI1ODM5NTE1fQ.KX5fFzdP2qYvoXo6wiOWxeyW  
ePq50hu0yyGjLAGX98t5wzFh219iLISzatZHHCs  
a54fMAYMto9yDc00rWCT-ug
```

Decoded EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS512"  
}
```

PAYLOAD: DATA

```
{  
  "sub": "store99",  
  "userRole": "ROLE_STOREMANAGER",  
  "exp": 1625844515,  
  "iat": 1625839515  
}
```

VERIFY SIGNATURE

```
HMACSHA512(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) ☐ secret base64 encoded
```



Data Security: Pattern & Best Practices

- Interfaces and patterns
- SALT
- Password cifrata nel database
- Password Complexity pattern constraint
- GitGuardian check uploaded code
- JWT token per i metodi esposti
- protocollo HTTPS per cifrare la comunicazione
- Attempts per evitare bruteforce
- Different DB users

```
Body  Cookies  Headers (14)  Test Results
Pretty  Raw  Preview  Visualize  Text
1 User inactive: Too many login attempts.
```

```
localhost
localhost dice
===Login failed===
User inactive: Too many login attempts.
OK
```

CARE Access Manager Billing

Care Atlas Realm Charts

CARE > CARE

Database Access

Database Users Custom Roles

User Name	Authentication Method	MongoDB Roles
Avellino	SCRAM	readWriteAnyDatabase@admin
Benevento	SCRAM	readWriteAnyDatabase@admin
Caserta	SCRAM	readWriteAnyDatabase@admin
Napoli	SCRAM	readWriteAnyDatabase@admin
Salerno	SCRAM	readWriteAnyDatabase@admin

Centro Accoglienza Regionale Ematica

Hermann80

.....

[Recupera Password](#)

LOGIN

All Resources EDIT DELETE

All Resources EDIT DELETE



JSON-WEB-TOKEN

Encoded

PASTE A TOKEN HERE

eyJhbGciOiJIUzUxMiJ9.eyJzdWIiOiJzdG9yZW
M5OSIsInVzZXJSb2x1IjoieUk9MRV9DRU5UkFMX
1NUT1JFTUF0QUdFUlIsImV4cCI6MTYyNjIwMTI5
NSwiaWF0IjoxNjI2MTk2Mjk1fQ.vifiTBYV8h3
mFy_1nuAaq6XNy9SdBQHgJ_FlEtc1FUpX0bHjqn
4AnVD9r2kkZkixD6SYVr2mY4uJBSwi7E8LQ

Decoded

EDIT THE PAYLOAD AND SECRET

HEADER: ALGORITHM & TOKEN TYPE

```
{  
  "alg": "HS512"  
}
```

PAYLOAD: DATA

```
{  
  "sub": "storec99",  
  "userRole": "ROLE_CENTRAL_STOREMANAGER",  
  "exp": 1626201295,  
  "iat": 1626196295  
}
```

VERIFY SIGNATURE

```
HMACSHA512(  
  base64UrlEncode(header) + "." +  
  base64UrlEncode(payload),  
  your-256-bit-secret  
) ☐ secret base64 encoded
```

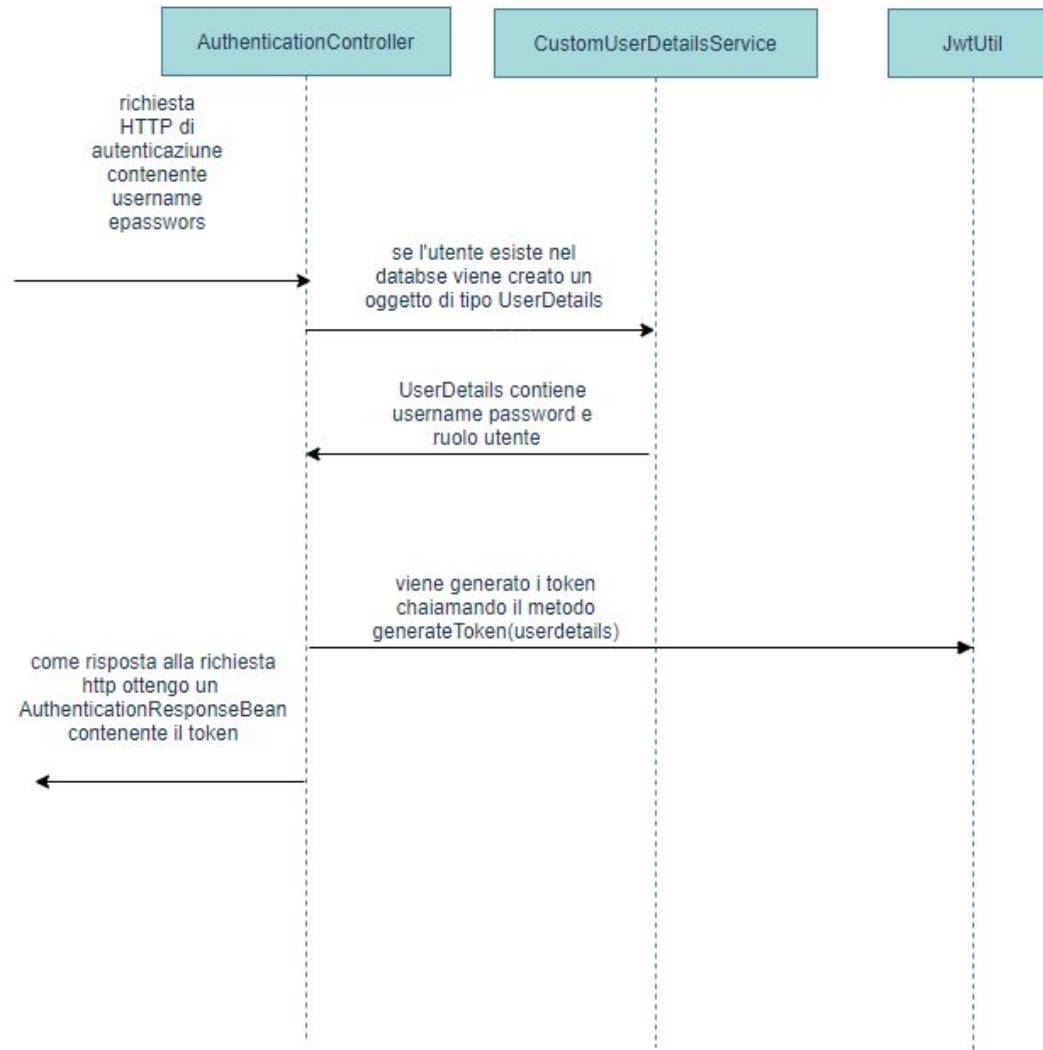
- Il JWT è uno standard utilizzato per l'autenticazione delle richieste http.
- Ha 3 campi Header, Payload, Signature
- Il token può essere crittografato e firmato utilizzando una chiave disponibile solo per il sistema che lo ha generato.
- Nel payload troviamo la durata del token e altre informazioni come il ruolo
- Il contenuto del payload non è crittografato, è importante che contenga dati sensibili.

```
# jwt Authentication  
jwt.secret = CARE@Unisannio  
jwt.expirationDateInMs=5000000
```






Autenticazione



SCHERMATA DI LOGIN

C.A.R.E



Centro Accoglienza Regionale Ematica

Username

Password

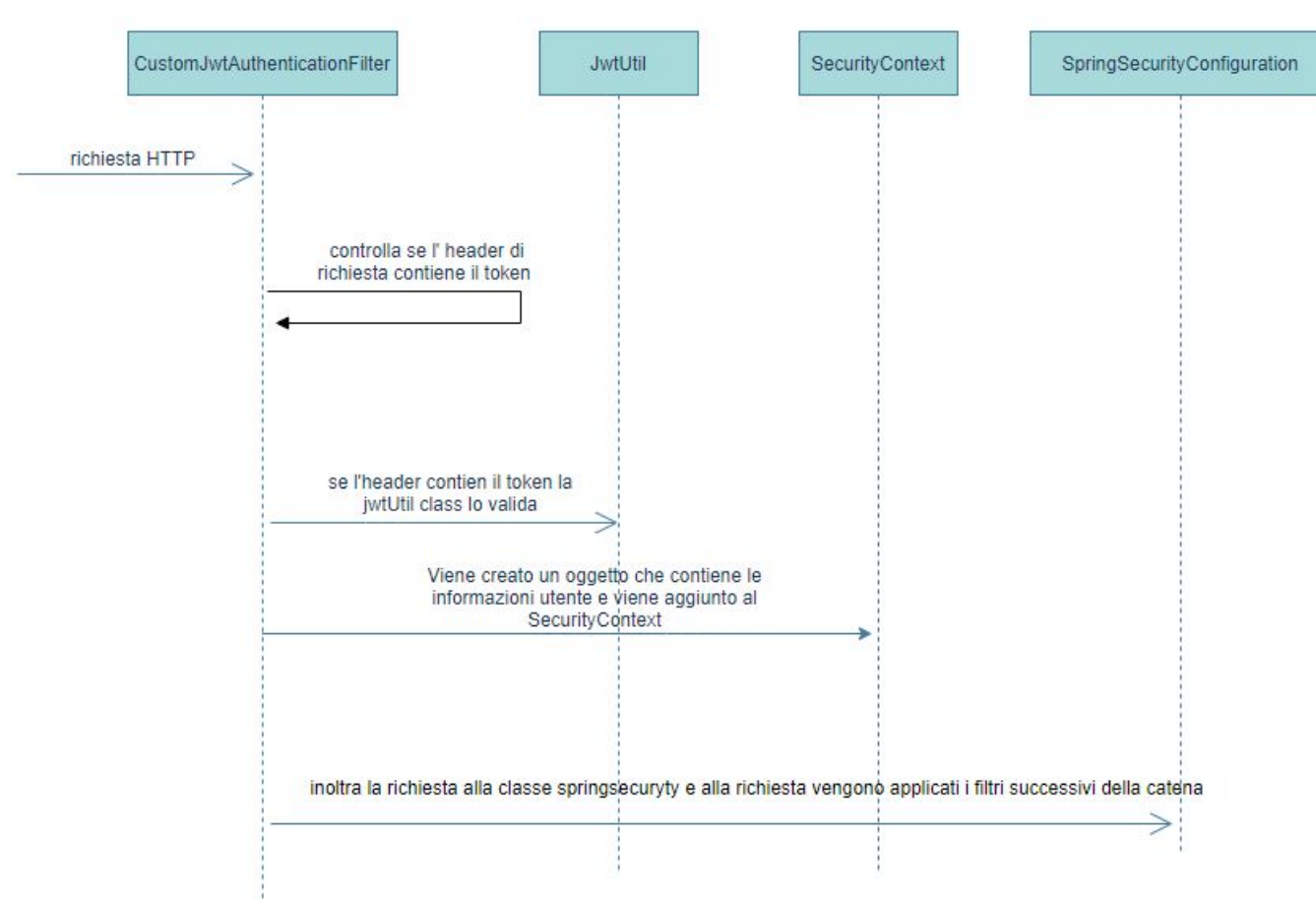
[Recupera Password](#)

- l'endpoint di autenticazione è accessibile a tutti l'header di richiesta HTTP non ha bisogno del JWT.





Autenticazione basata su ruolo



TOKEN SCADUTO

```
{
  "error": "JWT expired at 2021-07-14T15:31:28Z. Current time: 2021-07-14T22:39:35Z, a difference of 25687271 milliseconds. Allowed clock skew: 0 milliseconds."
}
```


ACCESSO VIETATO

```
{
  "timestamp": "2021-07-15T12:36:18.909+00:00",
  "status": 403,
  "error": "Forbidden",
  "path": "/bloodbag/add"
}
```

- Alcuni servizi per essere utilizzati hanno bisogno di una autenticazione jwt basata su ruolo.
- Ogni richiesta deve avere nell'header un token.
- SecurityContextHolder usa utilizza una variabile ThreadLocal per archiviare il securityContext, il che significa che il SecurityContext è sempre disponibile per i metodi nello stesso thread di esecuzione.



Graphic examples C.A.R.E



Richieste

Profilo

Logout (officer)

Next Version:

- Manage Patients
- Manage Donators

Sacche disponibili

Ho bisogno di:


none

Filtra

Richiedi	Priorità Bassa	Priorità Media	Priorità Alta	Seriale	Data creazione	Donatore	Data di scadenza	Gruppo	Note	Stato
<input type="checkbox"/>	<div></div>	<div></div>	<div></div>	IT-NA206000-APOS-2021C	2021-07-13	CRVBXB52C07L287P	2021-08-13	Apos	grazie	Available
<input type="checkbox"/>	<div></div>	<div></div>	<div></div>	IT-AV201000-ZEROPOS-2C	2021-07-13	RNRGLN92A1AA504Q	2021-08-13	ZEROpos	ottimo donatore	Available
<input type="checkbox"/>	<div></div>	<div></div>	<div></div>	IT-AV201000-BPOS-20210	2021-07-14	PLVGN796P22A783B	2021-08-14	Bpos	sacca nuova	Available
<input type="checkbox"/>	<div></div>	<div></div>	<div></div>	IT-AV201000-APOS-20210	2021-07-14	MDDMRZ97C700783Q	2021-08-14	Apos	Sacca buona 2	Available
<input type="checkbox"/>	<div></div>	<div></div>	<div></div>	IT-AV201000-APOS-20210	2021-07-14	PLVDNT96P33A783A	2021-08-14	Apos	NuovaSacca Ricevuta	Available
<input type="checkbox"/>	<div></div>	<div></div>	<div></div>	IT-AV201000-APOS-20210	2021-07-14	PLVDNT96P21A786A	2021-08-14	Apos	Sacca donatore buono	Available

Richieste:

Istantanea schermo



In Magazzino

Richiedi/spedisci

Profilo

Logout (store)

Next Version:

- Manage Patients
- Manage Donators

Ciao store

Il tuo ruolo è: ROLE: STOREMANAGER

Hai creato il token: Thursday 15th of July 2021 05:30:17 PM

Scadenza token: Thursday 15th of July 2021 07:13:37 PM


STOREMANAGER

Token countdown:
0d 1h 22m 40s

Benvenuto in CARE

centro accoglienza regionale ematica

C.A.R.E



Centro Accoglienza Regionale Ematica

Username

Password

[Recupera Password](#)

LOGIN





Graphic examples C.A.R.E

In Magazzino

Richiedi/spedisci

Profilo

Logout (store)

Next Version:

• Manage Patients

• Manage Donators

MAGAZZINO

(Assegna seriale alla sacca e la aggiunge al magazzino CIT)

Apos

cod.fiscale CF_DONATORE

Note

CREA

IMPORT SACCHE

(Aggiungi sacche al magazzino CIT)

scannerizza il QR code della sacca premi il tasto "import sacca" per confermare

seriale sacca

IMPORT SACCA

Code Scanner

IDLE

Request Camera Permissions

Instantanea schermo

Scan an Image File

Gestione utenti

Report

Profilo

Logout (admin)

Next Version:

• Manage Patients

• Manage Donators

ADMINISTRATOR

Token countdown:
0d 0h 13m 55s

Profile:

idUser

6

Username

admin

temppass

New Password(*)

Enter a valid new password

Confirm Password

Retype new password again

E-Mail

admin@care.it

Role-Based Access Control

ROLE_ADMINISTRATOR

creationDate

2021-07-12

lastAccess

2021-07-16

loginAttempts

0

activeUser

Attivo

Annula

Salva

* Password must contain at least one digit [0-9].

* Password must contain at least one lowercase Latin character [a-z].

* Password must contain at least one uppercase Latin character [A-Z].

* Password must contain at least one special character like ! @ # & ().

* Password must contain a length of at least 8 characters and a maximum of 20 characters.

Gestione utenti

Report

Profilo

Logout (admin)

Next Version:

• Manage Patients

• Manage Donators

Ciao admin

Il tuo ruolo è: ROLE_ADMINISTRATOR

Hai creato il token: Thursday 15th of July 2021 05:30:29 PM

Scadenza token: Thursday 15th of July 2021 07:13:49 PM

ADMINISTRATOR

Token countdown:
0d 0h 14m 22s

Report

Download CSV report

Download JSON report

Download TXT report



JUnit Test

Al fine di garantire maggiore qualità al committente, il sistema CARE è stato testato sia mediante l'approccio **glass-box** sia in **black-box**.

In **White-box** (o "glass-box") analizziamo la percentuale di **coverage** nonché i rami potenzialmente non coperti.

In **black-box**, nello specifico **JUnit 5**, si garantisce il testing dei casi d'uso non conoscendo le istruzioni che verranno eseguite, ma garantendo mediante una ipotesi che la premessa di utilizzo dei metodi ritorni quanto previsto.

Nell'esempio in questione i JUnit Test delle classi di modellazione base. I test non garantiscono l'impeccabilità del codice, ma in modo ragionevole

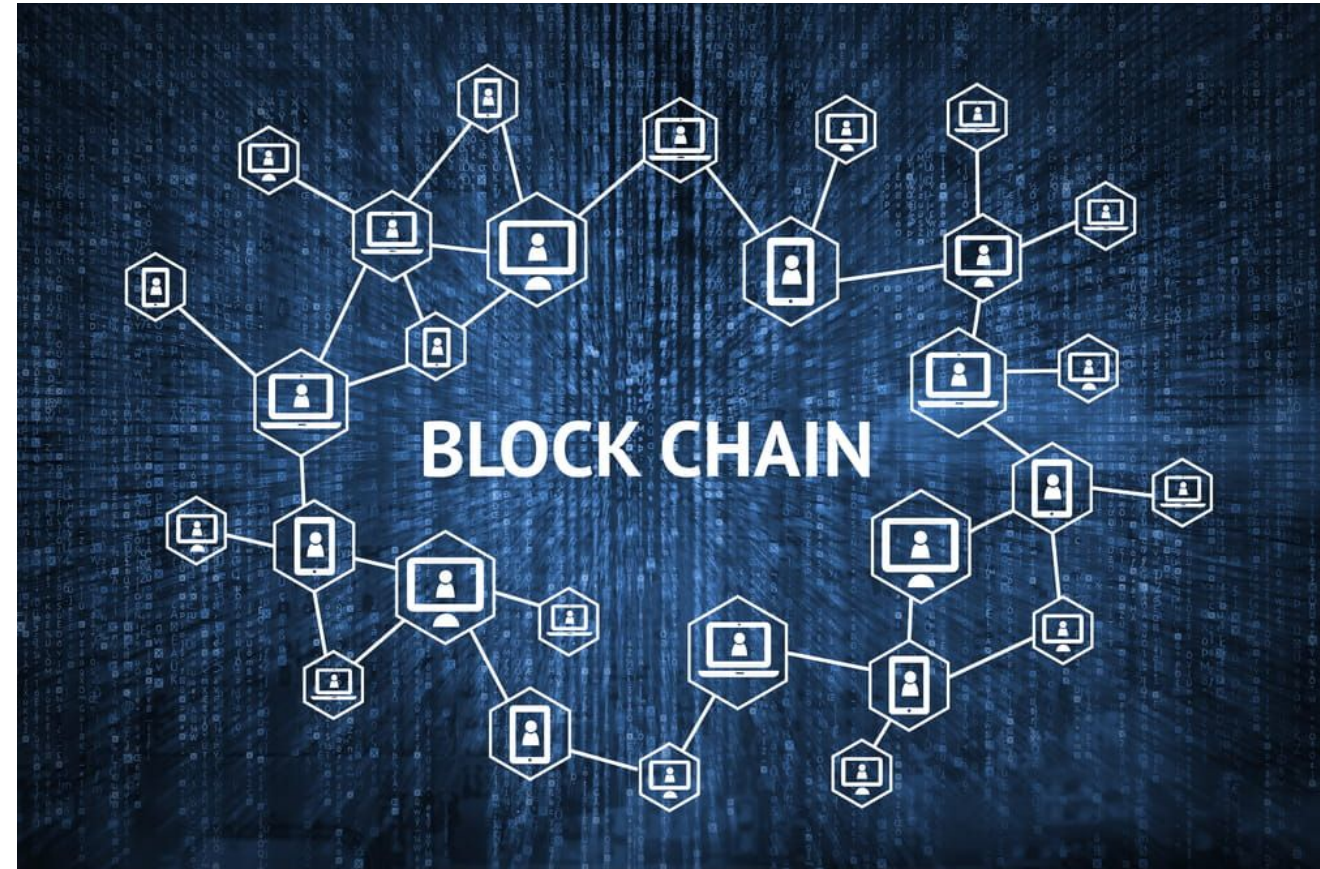
```
/**
 * Junit creation for verifying the insertion of an incorrect password
 * combo through the static streams
 * @param password the password to be tested
 */
@ParameterizedTest(name =
    "#{index} - Run test with valid password complexity pattern = {0}")
@MethodSource("validPasswordsProvider")
public void test_password_regex_valid(String password) {
    Assert.assertTrue( Password.validatePlaintextPasswordPattern
(password) );
}

static Stream<String> validPasswordsProvider() {
    return Stream.of(
        "AAAbbbccc@123",
        "Hello world$123",
        "A!@#&()-a1",           // valid: punctuation part 1
        "A[{}];:','/*a1",       // valid: punctuation part 2
        "A~$^+=<>a1",           // valid: symbols
        "0123456789$abcdefgAB", // valid: 20 chars
        "123Aa$Aa"               // valid: 8 chars
    );
}
```

>	RequestState.java	0,0 %	0	74	74
>	BloodBag.java	88,1 %	496	67	563
>	RequestPriority.java	0,0 %	0	34	34
>	Serial.java	93,1 %	231	17	248
>	BloodGroup.java	97,6 %	483	12	495
>	BloodBagState.java	100,0 %	64	0	64
>	it.unisannio.CARE.model.util.Logger	23,3 %	91	299	390
✓	it.unisannio.CARE.model.util	81,6 %	1.053	237	1.290
>	XMLHelper.java	29,2 %	73	177	250
>	QRCode.java	88,0 %	300	41	341
>	Password.java	90,9 %	159	16	175
>	Constants.java	98,7 %	229	3	232
>	LabelGenerator.java	100,0 %	292	0	292



Future Implementations



One more thing...

