



UNIVERSITÀ DEGLI STUDI DI BERGAMO

Dipartimento di Ingegneria Gestionale, dell'Informazione e della Produzione

Laurea Magistrale in Ingegneria Informatica

Documentazione progetto per il corso di PROGETTAZIONE, ALGORITMI E COMPUTABILITÀ

Prof.ssa:

Patrizia Scandurra

Studenti:

Giorgia BRESSANELLI

Matricola n. 1053903

Lorenzo FERRARI

Matricola n. 1053161

ANNO ACCADEMICO 2023 / 2024

Indice

3	Iterazione 3	3
3.1	Introduzione	3
3.2	UC1: Login	4
3.3	UC9: Logout	4
3.4	UC3: Accettazione chiamata emergenza	4
3.5	UC4 e UC5: Comunicazione con centrale	4
3.6	Analisi statica	5
3.7	Analisi dinamica	5

3 Iterazione 3

3.1 Introduzione

Per completare il funzionamento base dell'applicativo, per questa iterazione l'implementazione riguarda l'altro capo della comunicazione che avviene durante un'emergenza, ovvero l'attore **capo servizio**.

In particolare, i casi d'uso implementati sono i seguenti:

- UC1 Login
- UC3 Accettazione chiamata emergenza
- UC4 Comunicazione con centrale
- UC5 Comunicazione dati paziente
- UC9 Logout

Di seguito verranno analizzate nel dettaglio le fasi dello sviluppo e i test effettuati.

3.2 UC1: Login

Descrizione Classica fase di login riservata ai capi di servizio e agli operatori 118.

Dopo aver effettuato l'accesso, il capo servizio visualizza i dati forniti per le presenze del turno e in una sezione dedicata è possibile controllare se ci sono emergenze attivate di cui si attende la risposta per intervenire.

3.3 UC9: Logout

Descrizione Classica fase di logout.

3.4 UC3: Accettazione chiamata emergenza

Descrizione Il capo turno ha la possibilità di accettare o rifiutare la chiamata nel caso in cui venga contattato da un operatore 118 per rispondere ad un'emergenza nelle vicinanze.

3.5 UC4 e UC5: Comunicazione con centrale

Descrizione Comunicazione dei dati raccolti durante l'emergenza tramite form adibito per l'inserimento dei dati del/i paziente/i coinvolti nell'emergenza accettata.

3.6 Analisi statica

Anche nel corso di questa iterazione l'analisi statica, effettuata tramite plugin *Qodana*, ha prodotto gli stessi risultati ottenuti nell'iterazione precedente.

3.7 Analisi dinamica

Per quanto riguarda la sezione relativa al volontario è stata testata la seguente API:

```
### Logout dalla applicazione  
GET http://localhost:8080/logout  
Content-Type: application/json
```

Come per l'analisi dinamica relativa all'operatore 118, i test effettuati sul login, con l'ausilio di *JUnit* hanno avuto esito positivo (figura ??).

In particolare, per i test inerenti ai volontari la classe LoginControllerTest viene estesa con i seguenti metodi:

```
public class LoginControllerTest{
    //declarations
    @Test
    public void loginPageLoads() throws Exception {
        mockMvc.perform(MockMvcRequestBuilders.get("/login"))
                .andExpect(MockMvcResultMatchers.status().isOk());
    }
    @Test
    public void successfulLoginV() throws Exception {
        mockMvc.perform(MockMvcRequestBuilders.post("/login")
                .param("username", "V009")
                .param("password", "stellaalpina"))
                .andExpect(MockMvcResultMatchers.status()
                        .is3xxRedirection())
                .andExpect(MockMvcResultMatchers.redirectedUrl
                        ("/api/volontari/infoVolontarioSquadra"));
    }

    @Test
    public void failedLoginV() throws Exception {
        mockMvc.perform(MockMvcRequestBuilders.post("/login")
                .param("username", "V009")
                .param("password", "aaaaa"))
                .andExpect(MockMvcResultMatchers.status()
                        .is3xxRedirection())
                .andExpect(MockMvcResultMatchers.redirectedUrl
                        ("/login?error=true"));
    }
}
```