

UNIVERSITA' POLITECNICA DELLE MARCHE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA TRIENNALE IN INGEGNERIA INFORMATICA E
DELL'AUTOMAZIONE

CROCE VERDE PLUS

Progetto di Programmazione Mobile

Autori:
Scalzi Andrea
Ursu Leon

a.a 2022 / 2023



INTRODUZIONE.....	3
DESCRTIZIONE IN LINGUAGGIO NATURALE	3
SVILUPPO ANDROID	4
REQUISITI	4
Requisiti funzionali	4
Requisiti non funzionali.....	6
Casi d’Uso	7
ARCHITETTURA	13
DATABASE	13
UI (MOCKUP)	13
Mockup Amministratore	13
Mockup Centralinista	17
Mockup Volontario	20
Mockup Dipendente	21
Mockup Volontario/Dipendente	21
SVILUPPO.....	22
Amministratore	22
Centralinista.....	23
Volontario	23
Dipendente	24
TESTING.....	25
SVILUPPO FLUTTER	27
REQUISITI	27
Requisiti Funzionali.....	27
Requisiti non funzionali.....	28
Casi d’uso.....	29
ARCHITETTURA	31
DATABASE	31
UI (MOCKUP)	32
Mockup Amministratore	32
Mockup Milite.....	34
SVILUPPO.....	35
Milite	35
Amministratore	36
Login	36
CONCLUSIONI	36

INTRODUZIONE

DESCRIZIONE IN LINGUAGGIO NATURALE

L'associazione di volontariato Croce Verde opera nel settore pubblico e nel privato. Per la progettazione e realizzazione del sistema ci occuperemo soprattutto dell'operato dell'associazione nel settore pubblico.

Nel settore pubblico fornisce due tipi di servizi: 118 e H24:

118 - è il servizio di soccorso provinciale dove un mezzo di emergenza viene utilizzato per situazioni in cui viene richiesto l'intervento d'urgenza.

H24 - è un servizio dove viene utilizzata un'ambulanza per trasporto di persone allettate che non possiedono la capacità di muoversi autonomamente, il servizio prevede il trasporto del paziente dall'ospedale fino alla residenza o struttura di alloggio. L'H24 è un servizio che viene fornito a settimane alterne con un'altra associazione di volontariato che opera sul territorio.

Entrambi questi servizi operano su tre tipologie di turnazioni orarie:

1. 7-14 (turno mattutino)
2. 14-21 (turno pomeridiano)
3. 21-7 (turno notturno)

Durante i turni possono essere presenti sull'ambulanza un massimo di 3 persone e un minimo di 2 persone per equipaggio. L'equipaggio è diviso in: milite 1°, milite 2°, milite 3°, ordine di maggiore importanza decrescente. L'equipaggio del 118 necessita che uno dei militi prenda il ruolo di "Capo Equipaggio", sarà lui ad avere la responsabilità maggiore dell'operato della squadra.

Un milite può prenotarsi sui diversi servizi in base al proprio grado di milite, a seconda del grado ci si può prenotare su turni differenti.

Un utente dovrà essere in grado di effettuare il Login tramite NomeUtente/e-mail e password, una volta effettuato il login nel sistema, all'utente viene fornito il modo per effettuare la propria prenotazione al turno.

I dipendenti della Croce Verde sono coloro che hanno un contratto pagato con l'associazione e vengono inseriti dal centralinista dove c'è bisogno e quindi dove mancano i volontari per coprire un turno. Non sono in grado di prenotarsi al turno ma possono solo visionare il tabellone disponibile. Il sistema dovrebbe permettere di gestire un dipendente.

Il militi possono fornire la propria disponibilità e il centralinista li segna sul campo libero del tabellone.

Il sistema deve avere una sezione dove i volontari vedono i propri dati e statistiche sui turni fatti.

Durante l'inizio del turno del 118 si procede a verificare la presenza dei presidi all'interno dell'ambulanza tramite una checklist dell'ambulanza del 118. Colui che ha effettuato la checklist sarà registrato tramite id del proprio profilo. Il sistema dovrebbe permettere di effettuare tale checklist.

I tipi di iscrizioni secondo la gerarchia di mansioni che possono essere effettuare. Vedere schema sottostante:

RF3 - PrenotazioniGerarchieMiliti

Il sistema dovrà gestire le prenotazioni in base alle gerarchie di Militi

RF4 - GestionePrenotazioni

Il sistema dovrà permettere ai Militi di prenotarsi ai turni disponibili

RF5 - GestioneLogin

Il sistema dovrà permettere di gestire il Login

RF6 - VisualizzazioneTurni

Il sistema dovrà permettere di vedere un tabellone di turni disponibili

RF7 - GestisciMilite

Il sistema dovrà permettere all'amministratore di gestire un milite tramite le operazioni CRUD

RF8 – InserisciDisponibilitàMilite

Il sistema dovrà permettere ai militi di fornire la propria disponibilità

RF9 – VisualizzaProfiloPersonale

Il sistema dovrà permettere di vedere il proprio profilo

RF10 – VisualizzaStatistichePersonal

Il sistema dovrà permettere di vedere le proprie statistiche

RF11 – GestioneChecklist

Il sistema dovrà permettere di effettuare le checklist dell'ambulanza del 118

RF12 – VisualizzaStatisticheMilite

Il sistema dovrà permettere all'amministratore di visualizzare le statistiche sui turni svolti dai militi.

RF13 – AumentoGerarchiaMiliti

Il sistema dovrà permettere ai Militi di progredire con il grado grazie all'amministratore che li promuove

RF14 – GestisciCentralinista

Il sistema dovrà permettere all'amministratore di gestire un centralinista tramite le operazioni CRD

Requisiti non funzionali

RNF1 – Linguaggio di implementazione

Il sistema dovrà essere implementato utilizzando il linguaggio Kotlin

RNF2 – Database

Il sistema dovrà avere un DB condiviso online

RNF3 – Usabilità

Il sistema deve essere intuitivo e facile da utilizzare per chiunque

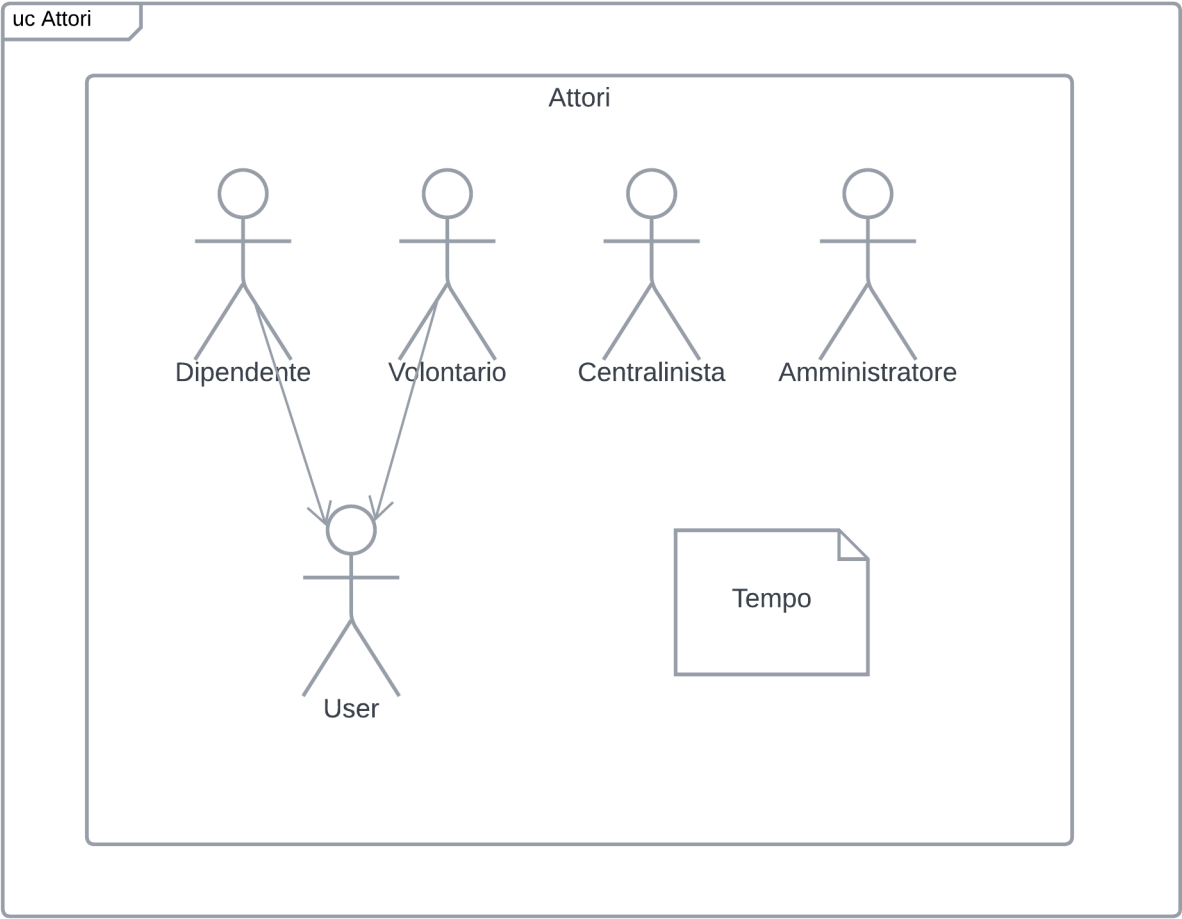


Figura 2 - Attori

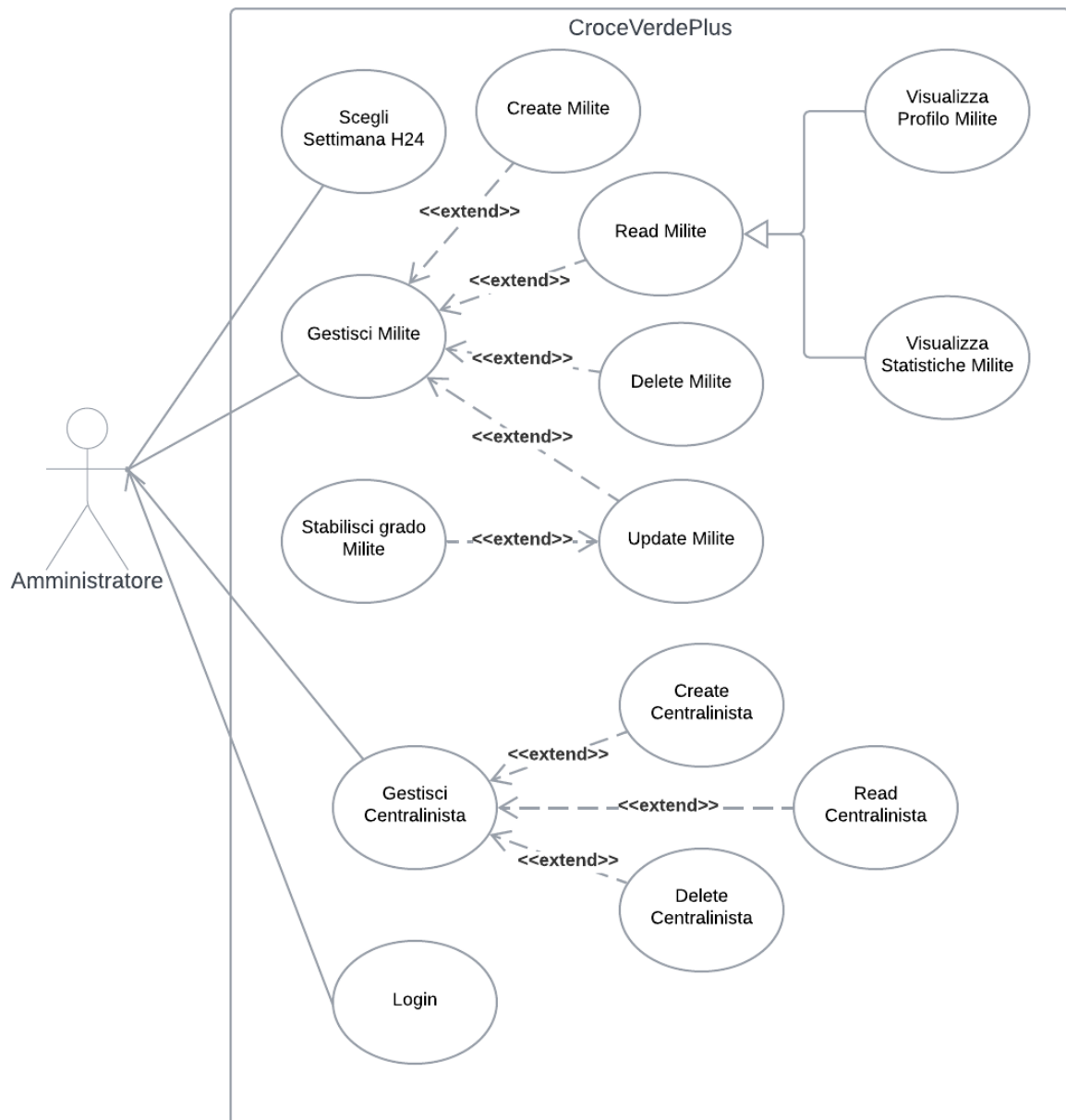


Figura 3 - UC - Utilizzo Amministratore

Scegli Settimana H24: Questo caso d'uso inizia quando l'amministratore decide di voler stabilire in quale settimana avere il servizio H24.

Gestisci Milite: Questo caso d'uso inizia quando l'amministratore vuole effettuare operazioni di lettura, modifica, creazione ed eliminazione del milite.

Gestisci Centralinista: Questo caso d'uso inizia quando l'amministratore vuole effettuare operazioni di lettura, creazione ed eliminazione del centralinista.

Effettua Login : Questo caso d'uso inizia quando l'amministratore decide di effettuare il login nel proprio account personale grazie alle credenziali di accesso.

UC - Centralinista

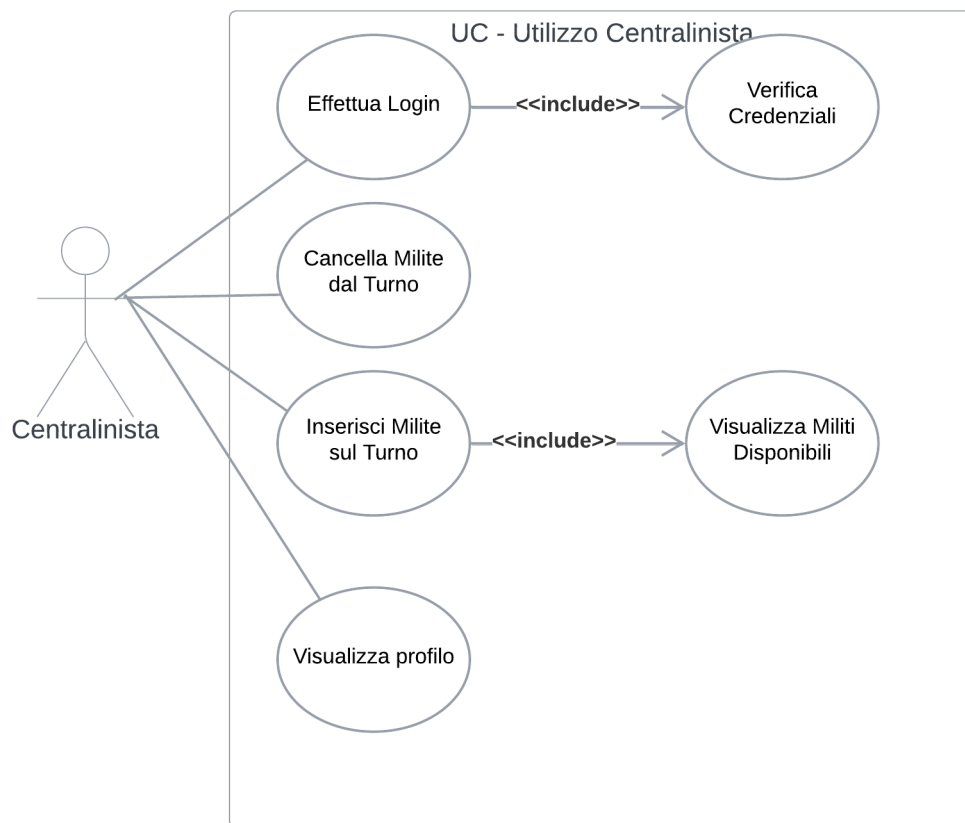


Figura 4 - UC - Utilizzo Centralinista

Effettua Login : Questo caso d'uso inizia quando il centralinista decide di effettuare il login nel proprio account personale grazie alle credenziali di accesso.

Cancella Milite dal Turno : Questo caso d'uso inizia quando il centralinista decide di effettuare una cancellazione dal turno di un determinato milite selezionato.

Inserisci Milite sul Turno: Questo caso d'uso inizia quando il centralinista decide di inserire un milite in un determinato turno, per questo fine è necessario disporre della lista dei militi disponibili

Visualizza Profilo Personale: Questo caso d'uso inizia quando il centralinista decide di visualizzare il proprio profilo personale attraverso il sistema.

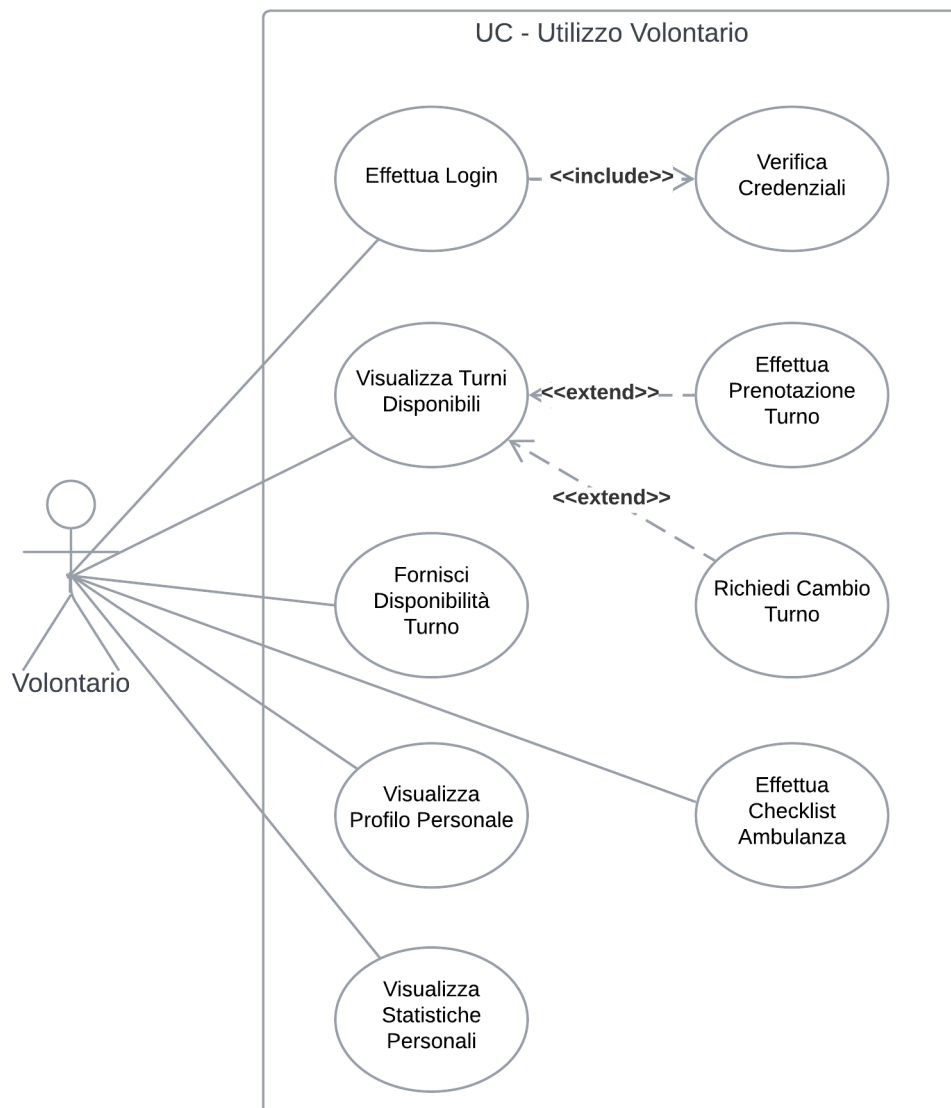


Figura 5 - UC - Utilizzo Volontario

Effettua Login: Questo caso d'uso inizia quando il volontario decide di effettuare il login nel proprio account personale grazie alle credenziali di accesso.

Visualizza profilo personale: Questo caso d'uso inizia quando il volontario decide di visualizzare il proprio profilo personale attraverso il sistema.

Visualizza turni disponibili: Questo caso d'uso inizia quando il volontario decide di visualizzare i turni disponibili.

Fornisci disponibilità turno: Questo caso d'uso inizia quando il volontario decide di fornire la propria disponibilità per il turno.

Visualizza statistiche personali: Questo caso d'uso inizia quando il volontario decide di visualizzare le proprie statistiche personali attraverso il sistema

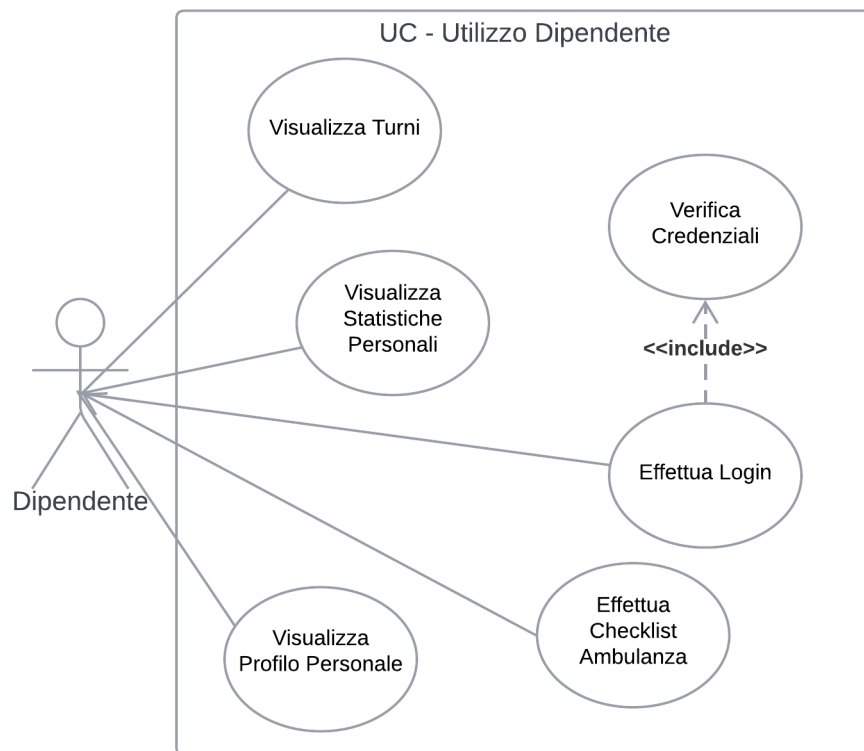


Figura 6 - UC - Utilizzo Dipendente

Visualizza Profilo Personale: Questo caso d'uso inizia quando il dipendente decide di visualizzare il proprio profilo personale attraverso il sistema.

Effettua Login: Questo caso d'uso inizia quando il dipendente decide di effettuare il login nel proprio account personale grazie alle credenziali di accesso.

Effettua CheckList Ambulanza: Questo caso d'uso inizia quando Il dipendente decide di effettuare la checklist dell'ambulanza.

Visualizza turni disponibili: Questo caso d'uso inizia quando il dipendente decide di visualizzare i turni disponibili.

UC – Sistema

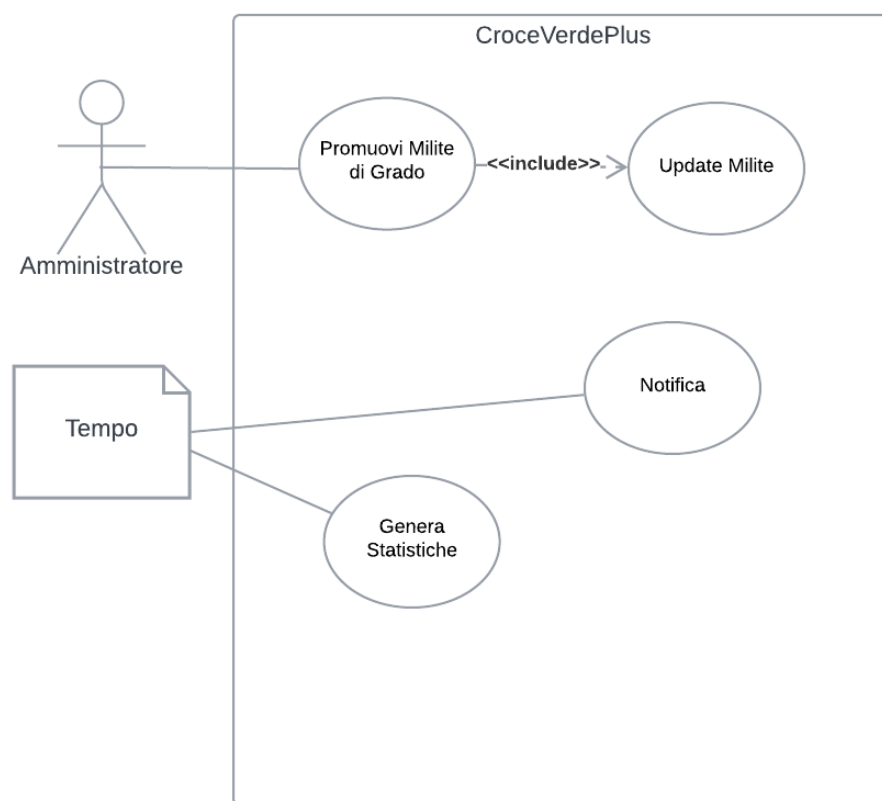


Figura 7 - UC – Gestione Sistema

Promuovi milite di grado: Questo caso d’uso inizia quando l’amministratore decide di promuovere un milite modificandone i gradi

Notifica: Questo caso d’uso inizia quando il tempo mostra all’utente la notifica della regolarità dei turni fatti nello stesso mese.

Genera statistiche: Questo caso d’uso inizia quando il sistema aggiunge automaticamente le ore di lavoro effettuate ai militi.

ARCHITETTURA

In questo progetto non è stata utilizzata nessuna architettura particolare. Questa decisione è dovuta principalmente per motivi di necessità, ovvero i requisiti specifici richiesti dall'app non si prestavano all'utilizzo di un modello strutturale predefinito, e per motivi di efficienza; infatti, la scelta di non adottare un'architettura rigidamente definita in precedenza, ha permesso uno sviluppo più semplificato e facilmente comprensibile dell'app.

DATABASE

Il Database utilizzato per la creazione dell'app è Cloud Firestore, un database NoSQL fornito da Firebase. La scelta di impiegare Cloud Firestore è dovuta a diverse motivazioni: innanzitutto per la sua flessibilità e struttura gerarchica dei dati; infatti, questi ultimi sono organizzati in raccolte, le quali sono suddivise in documenti contenenti campi associati a valori. Inoltre, Cloud Firestore permette di effettuare query complesse, includendo filtri multipli, e fornisce supporto offline, in modo tale che l'app possa scrivere, leggere ed eseguire query sui dati anche se il dispositivo è offline, salvando in locale i dati utilizzati dall'app e sincronizzandoli successivamente quando il dispositivo tornerà online.

Il Database utilizzato per la nostra app permette di effettuare operazioni CRUD e di salvare dati nelle raccolte relative a:

- **Gli Amministratori:** *username, password*
- **I Centralinisti:** *nome, cognome, data di nascita, residenza, username, password*
- **I Militi:** *nome, cognome, data di nascita, residenza, username, password, gradi di 118 e H24, volontario/dipendente, ore svolte nei vari turni di 118 e H24*
- **Le Checklist dei presidi eseguite dai militi:** *nome e cognome del milite che ha effettuato la checklist, data dell'ultima modifica apportata alla checklist, lista dei presidi*
- **Le disponibilità dei militi per i turni:** *nome e cognome, data, turno*
- **Le prenotazioni dei militi per i turni:** *nome e cognome, data, turno*
- **Le tabelle del "118" e del "118 e H24":** *date dei giorni della settimana, turni*

I campi dei documenti relativi alle rispettive raccolte sono stati definiti lato codice attraverso la creazione di Data Class, mentre le operazioni principali di creazione, modifica, lettura ed eliminazione dei documenti e di campi specifici collegati ad essi, sono state realizzate tramite semplici e complesse query definite nella classe Database del progetto.

UI (MOCKUP)

Mockup Amministratore



Figura 8 - Amministratore

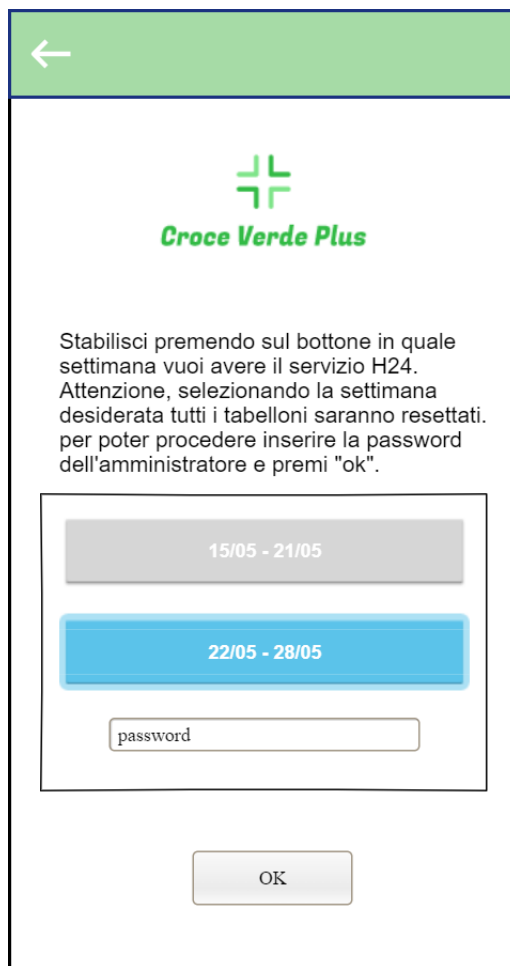


Figura 9 – Gestione settimana H24

←

Elliot Rossi 12/05/1998 via Elliot 28 Volontario GradoH24prima GradoH24seconda
Tarrel Rossi 29/09/1994 via Tarrel 27 Dipendente Grado118prima GradoH24terza
Luca Rossi 02/04/2001 via Luca 34 Volontario Grado118prima GradoH24prima GradoH24seconda
+

Figura 10 – Gestione Tutti Militi

←



Nome
 Cognome
 Data di nascita
 Residenza
 Grado

Aggiorna Milite

Cancella Milite

Figura 11 – Gestione Milite

118 Prima

H24 Prima

118 Seconda

H24 Seconda

118 Terza

H24 Terza

Salva

Figura 12 – Gestione Milite Modifica Crea

Elliot
Rossi
12/05/1998
via Elliot 28

Tarrel
Rossi
29/09/1994
via Tarrel 27

Luca
Rossi
02/04/2001
via Luca 34

Figura 13 – Gestione Tutti Centralinisti


A mobile app mockup for 'Gestione Centralinista'. It features a green header bar with a white back arrow. Below the header is a large grey circular placeholder for a profile picture. Underneath the picture, the fields 'Nome', 'Cognome', 'Data di nascita', and 'Residenza' are listed vertically. At the bottom of the screen is a grey button with the text 'Cancella Centralinista'.

Figura 14 – Gestione Centralinista

A mobile app mockup for 'Gestione Centralinista Crea'. It features a green header bar with a white back arrow. Below the header are four text input fields labeled 'Nome', 'Cognome', 'Data di nascita', and 'Residenza'. Below these fields is a grey button with the text 'Salva'.

Figura 15 – Gestione Centralinista Crea

Mockup Centralinista



Name

Surname

Data di nascita

118 1a - 500 ore

118 2a - 56 ore


118 3a - 100 ore

H24 1a - 98 ore

H24 2a - 110 ore

H24 3a - 56 ore

Esci






Figura 18 - Profilo Centralinista

Mockup Volontario

settimana h24

settimana 118/h24

	H24			118		
	dalle 7 alle 14	dalle 14 alle 21	dalle 21 alle 7	dalle 7 alle 14	dalle 14 alle 21	dalle 21 alle 7
Lunedì 27/03						
Martedì 28/03						
Mercoledì 29/03						
Giovedì 30/03						
Venerdì 31/03						
Sabato 01/04						
Domenica 02/04						

118▼

Milite 1a▼

Lunedì▼

14 - 21▼

Disponibilità

Segnami

Figura 19 - Tabellone Turni Volontario

Mockup Dipendente

settimana h24

settimana 118/h24

	H24			118		
	dalle 7 alle 14	dalle 14 alle 21	dalle 21 alle 7	dalle 7 alle 14	dalle 14 alle 21	dalle 21 alle 7
Lunedì 27/03						
Martedì 28/03						
Mercoledì 29/03						
Giovedì 30/03						
Venerdì 31/03						
Sabato 01/04						
Domenica 02/04						

Figura 20 - Tabellone Turni Dipendente

Mockup Volontario/Dipendente

Figura 21 - Profilo Volontario Dipendente

Figura 22 - Checklist Volontario Dipendente

SVILUPPO

Di seguito si troveranno le spiegazioni di tutte i componenti individuati durante l'analisi, principalmente sono individuati come le activity per ogni singolo tipo di utente del servizio. Possiamo individuare i seguenti macro-componenti: Amministratore, Centralinista, Volontario, Dipendente

Amministratore

L'activity dell'amministratore viene attivata dopo il login, ha il compito di gestire il cambio fra i vari Fragment appartenenti a questo utente grazie al nagivation graph. La navigazione avviene da una schermata all'altra utilizzando pulsanti appositi a seconda dell'area di interesse. L'amministratore può effettuare diverse operazioni:

Scegliere la data della settimana H24/118.

Questo fragment permette all'amministratore di poter stabilire in quale settimana sarà attivo il servizio di H24, scegliendo tra la settimana corrente o quella prossima. Questa decisione viene effettuata premendo uno dei due bottoni contenenti le date di inizio e fine di ognuna delle rispettive settimane; il bottone scelto rimarrà evidenziato graficamente. Successivamente per confermare la scelta l'amministratore dovrà inserire

la propria password e selezionare il bottone "ok". In questo modo entrambe le tabelle verranno resettate e le date modificate.

Gestione militi

Questo fragment mostra la lista completa di tutti i militi e un bottone circolare tramite il quale è possibile navigare alla schermata di creazione milite.

Selezionando un elemento della lista si apre una nuova schermata relativa alla gestione del milite, che permette all'amministratore di leggere i dati anagrafici del milite e di effettuare due operazioni: la prima di eliminazione del milite, dopo successiva conferma richiesta tramite popup, e la seconda di modifica del milite, tramite schermata apposita contenente già i dati anagrafici del milite scelto.

Dopo aver selezionato il bottone circolare invece, è possibile creare un nuovo milite inserendo i dati anagrafici, selezionando i gradi e scegliendo volontario o dipendente. Dopo aver confermato, il sistema crea lo username e genera la password, necessari al milite per compiere il login.

Gestione centralinisti

Questo fragment mostra la lista completa di tutti i centralinisti, permettendo, in seguito alla selezione di uno degli elementi della lista, di visualizzare il profilo del centralinista scelto e di poterlo eliminare. Infine, Gestione Centralinisti offre la possibilità di poter creare un nuovo centralinista inserendo i suoi dati anagrafici, dopo la conferma il sistema crea lo username e genera la password relativi al seguente centralinista.

Centralinista

L'activity del centralinista viene attivata dopo il login con un utente che è centralinista, ha il compito di gestire il cambio fra i vari Fragment appartenenti a questo utente grazie alla Bottom bar. I fragment gestiti da questa activity sono: TabelloneTurniCentralinista, ProfiloCentralinista, Disponibilità. I nomi appena elencati sono i titoli delle classi che gestiscono i fragment associate ad esse.

TabelloneTurniCentralinista

Questo fragment rappresenta il tabellone dei turni modificato per permettere di eseguire il compito del Centralinista, il centralinista può assegnare i volontari ai turni in base a varie opzioni: servizio, giorno, orario e grado, quando le opzioni sono settate con un determinato stato nel menù a tendina verranno visualizzati solamente i militi che hanno un grado sufficiente per svolgere quel determinato turno, se un opzione viene mutata vi è una call-back al database per recuperare i militi in base ad un diverso filtro o meglio, un diverso stato delle opzioni. Se si verifica che nel turno è presente il milite selezionato, il sistema elimina il milite dal turno e rende il turno vuoto per altri eventuali militi. Se invece il milite è diverso dal milite selezionato, i dati del turno verranno aggiornati con il nuovo milite che verrà segnato nel turno.

ProfiloCentralinista

Questo fragment rappresenta il profilo del Centralinista, mostra solamente qualche informazione personale e permette di effettuare il logout dal sistema.

Disponibilità

Questo fragment rappresenta la disponibilità dei militi, è una parte del programma che permette di vedere le disponibilità fornite dai militi, vengono mostrate come una lista con il nome del milite la data di prenotazione e altre info del turno.

Volontario

L'activity del centralinista viene attivata dopo il login con un utente che è volontario, ha il compito di gestire il cambio fra i vari fragment appartenenti a questo utente grazie alla Bottom bar. I fragment gestiti da questa activity sono: TabelloneTurniVolontario, ProfiloVolontario, Checklist. I nomi appena elencati sono i titoli delle classi che gestiscono i fragment associate ad esse.

TabelloneTurniVolontario

Questo fragment rappresenta il tabellone dei turni dalla visuale del Volontario, permette di prenotarsi sul tabellone dei turni e di fornire la propria disponibilità settando un determinato stato delle opzioni.

La prenotazione viene permessa scegliendo tramite il menu a tendina il turno, vengono effettuate delle verifiche e il volontario viene informato tramite un messaggio a comparsa dell'esito dell'operazione, se l'esito è affermativo il volontario vedrà comparire il proprio nome nel tabellone e le ore verranno aggiunte alle sue statistiche

Se un milite decide di cancellarsi da un turno basta che selezioni il turno di interesse e se è presente il quel turno verrà cancellato altrimenti riceverà un messaggio d'errore. Questo fragment permette di fornire la propria disponibilità selezionando il turno d'interesse e selezionando il pulsante "Disponibilità".

ProfiloVolontario

Il profilo del volontario mostra diverse opzioni:

Dati personali: mostra i dati personali del milite.

Statistiche personali: mostra le statistiche personali del milite in base alle ore che ha effettuato.

Stato di attività: Nella Croce Verde si parte dal presupposto che un milite faccia almeno due turni al mese di volontariato per mantenere il proprio stato di attivo. Lo stato viene mostrato di colore "verde" e un numero $(n/2)$ con $n \geq 2$ se il milite ha effettuato oppure si è prenotato su almeno due turni nell'ultimo mese. Viene mostrato rosso e un numero $(n/2)$ con $0 < n < 2$ se il milite non ha effettuato oppure si è prenotato su almeno due turni nell'ultimo mese.

Checklist

Questo Fragment permette di effettuare la checklist dell'ambulanza, il milite loggato avrà a sua disposizione una lista di presidi con una casella spuntabile se quel determinato presidio è presente nell'ambulanza. Una volta spuntate tutte le caselle di interesse, il milite dovrà selezionare il pulsante "Salva" per effettuare il salvataggio della checklist. Il nome del milite che salva la checklist verrà registrato automaticamente.

Dipendente

L'activity del volontario viene attivata dopo il login, ha il compito di gestire il cambio fra i vari Fragment appartenenti a questo utente grazie alla Bottom bar. I fragment gestiti da questa activity sono: TabelloneTurniDipendente, ProfiloVolontario, Checklist. I nomi appena elencati sono i titoli delle classi che gestiscono i fragment associati ad esse.

TabelloneTurniDipendente

Questo Fragment permette ai dipendenti di visualizzare lo stato dei turni senza permettergli di prenotarsi o cancellarsi da essi.

ProfiloDipendente

Il profilo del dipendente mostra diverse opzioni:

Dati personali: mostra i dati personali del milite.

Statistiche personali: mostra le statistiche personali del milite in base alle ore che ha effettuato.

Stato di attività: Questa opzione non è necessaria per un dipendente ma è stata lasciata per utilizzare lo stesso fragment per volontario e dipendente.

TESTING

Il Trattandosi di una app che effettua poche operazioni in locale, sono stati testati con i Junit test solamente alcuni dei metodi disponibili. Di seguito sono riportati i test effettuati

Test: `id_builder_test()`

Questo test è stato effettuato per verificare la corretta formazione del id del turno che servirà per individuare in quale punto della tabella andrà ad essere inserito il milite.

A screenshot of a code editor with a dark background. The code is written in Kotlin and represents a JUnit test. It starts with an '@Test' annotation, followed by a function definition 'fun id_builder_test()'. Inside the function, several variables are declared: 'servizio_val' with value 'H24', 'giorno_val' with value 'Domenica', 'orario_val' with value '7 - 14', and 'grado_val' with value '2a'. Then, 'idCostruito' is assigned the result of 'TabelloneTurni().id_builder(servizio_val, giorno_val, orario_val, grado_val)'. Finally, 'Assert.assertEquals' is used to compare the 'idCostruito' variable with the expected string 'turno_h24_dom_mat_2'.

```
@Test
fun id_builder_test(){
    val servizio_val = "H24"
    val giorno_val = "Domenica"
    val orario_val = "7 - 14"
    val grado_val = "2a"
    val idCostruito = TabelloneTurni().id_builder(servizio_val,giorno_val,orario_val,grado_val)
    Assert.assertEquals( expected: "turno_h24_dom_mat_2", idCostruito)
}
```

Figura 23 – Id Builder Test

Test: `filtra_militi_test()`

Questo test è stato selezionato per testare il filtraggio corretto dei militi in base al grado del milite che è stato selezionato in base ai parametri passati. Il metodo filtra i militi disponibili e ritorna sotto forma di lista i militi che sono compatibili.

```

militi.add(
    Milite(
        username: "username$_n",
        password: "pass",
        nome: "nome_test$_n",
        cognome: "cognome_test$_n",
        dataDiNascita: "10/04/1998",
        residenza: "residenza$_n",
        volontario: true,
        dipendente: false,
        cognomeNomeSpinner: "cognomeNomeSpinner$_n",
        grado118prima: false,
        grado118seconda: false,
        grado118terza: false,
        gradoh24prima: false,
        gradoh24seconda: false,
        gradoh24terza: true,
        oreTurno118prima: 0,
        oreTurno118seconda: 0,
        oreTurno118terza: 0,
        oreTurnoh24prima: 0,
        oreTurnoh24seconda: 0,
        oreTurnoh24terza: 0
    )
)
var lista_filtrata = TabelloneTurni().filtra_militi(militi, servizio_val, grado_val)
Assert.assertEquals(
    expected: 2,
    lista_filtrata.size
) // si verifica se i militi sono stati filtrati in base al parametro passato
Assert.assertEquals( expected: 12, militi.size)

```

Figura 24 - Filtra Militi Test

Test: testGetWeekDates()

Questo test è stato effettuato per verificare il corretto funzionamento del metodo `getWeekDates()`, contenuto all'interno della classe `GestioneSettimanaH24`, il quale restituisce una coppia di date che rappresentano l'inizio (lunedì) e la fine (domenica) di una settimana a partire da una data di riferimento.

```

@Test
fun testGetWeekDates() {
    val weekDates = getWeekDates(calendar.time)

    //Lunedì
    val expectedWeekStart =
        SimpleDateFormat(pattern: "yyyy-MM-dd", Locale.getDefault()).parse(source: "2023-06-26")
    //Domenica
    val expectedWeekEnd =
        SimpleDateFormat(pattern: "yyyy-MM-dd", Locale.getDefault()).parse(source: "2023-07-02")

    assertEquals(expectedWeekStart, weekDates.first)
    assertEquals(expectedWeekEnd, weekDates.second)
}

```

Figura 25 - getWeekDates Test

SVILUPPO FLUTTER

REQUISITI

Requisiti Funzionali

RF1 - GestioneH24

Il sistema dovrà permettere di gestire l'H24 a settimane alterne

RF2 - GestioneTurni

Il sistema dovrà gestire diverse tipologie di turnazioni orarie

RF3 - GestioneLogin

Il sistema dovrà permettere di gestire il Login

RF4 - VisualizzazioneTurni

Il sistema dovrà permettere di vedere un tabellone di turni disponibili

RF5 - GestisciMilite

Il sistema dovrà permettere all'amministratore di gestire un milite tramite le operazioni CRUD

RF6 – VisualizzaProfiloPersonale

Il sistema dovrà permettere di vedere il proprio profilo (solo militi)

RF7 – VisualizzaStatistichePersonali

Il sistema dovrà permettere di vedere le proprie statistiche (solo militi)

RF8 – AumentoGerarchiaMiliti

Il sistema dovrà permettere ai Militi di progredire con il grado grazie all'amministratore che li promuove

Requisiti non funzionali

RNF1 – Framework di implementazione

Il sistema dovrà essere implementato utilizzando Flutter

RNF2 – Linguaggio di implementazione

Il sistema dovrà essere implementato utilizzando Dart

RNF3 – Database

Il sistema dovrà avere un DB condiviso online

RNF4 – Usabilità

Il sistema deve essere intuitivo e facile da utilizzare per chiunque

Casi d'uso
Attori Flutter

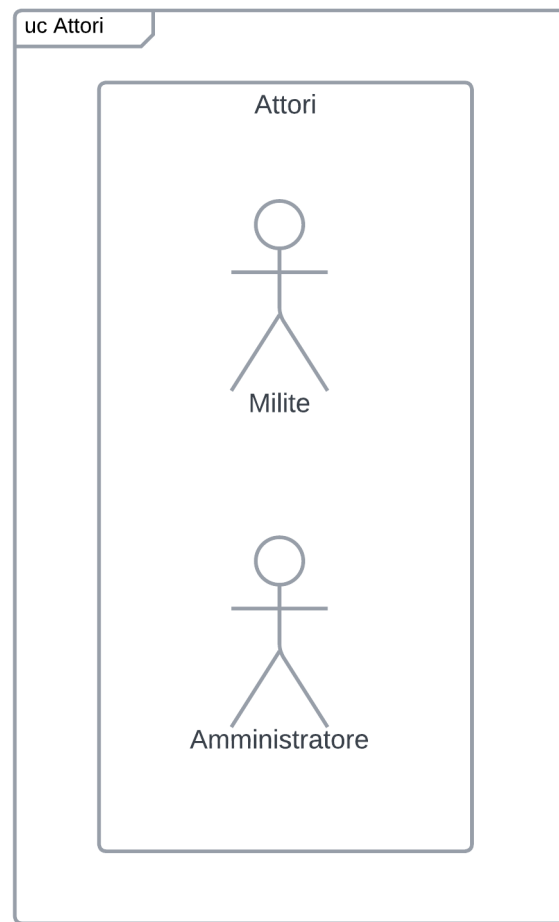


Figura 26 - Attori Flutter

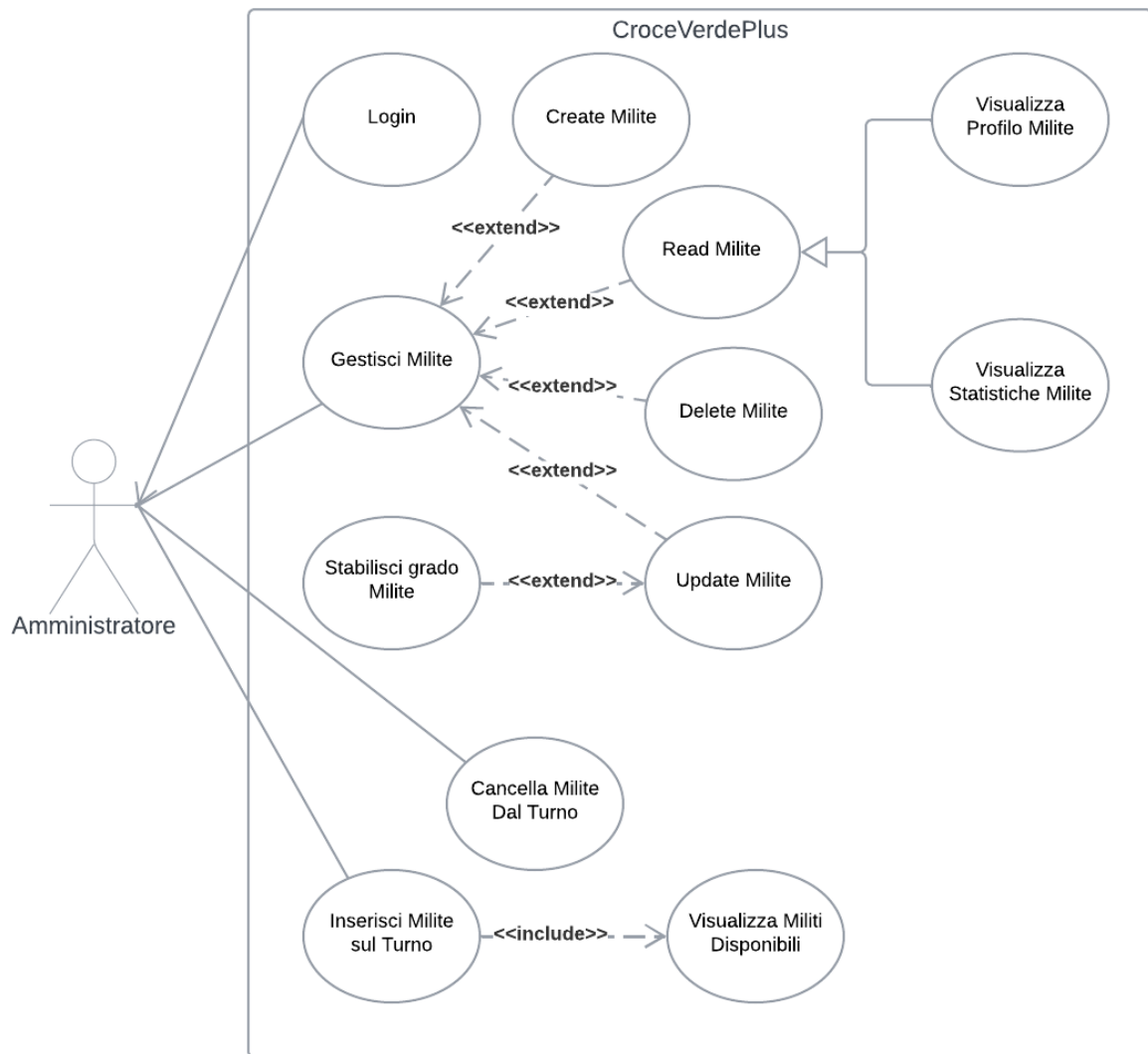


Figura 27 - UC - Utilizzo Amministratore

Gestisci Milite: Questo caso d'uso inizia quando l'amministratore vuole effettuare operazioni di lettura, modifica, creazione ed eliminazione del milite.

Effettua Login : Questo caso d'uso inizia quando l'amministratore decide di effettuare il login nel proprio account personale grazie alle credenziali di accesso.

Cancella Milite dal Turno : Questo caso d'uso inizia quando l'amministratore decide di effettuare una cancellazione dal turno di un determinato milite selezionato.

Inserisci Milite sul Turno: Questo caso d'uso inizia quando l'amministratore decide di inserire un milite in un determinato turno, per questo fine è necessario disporre della lista dei militi disponibili

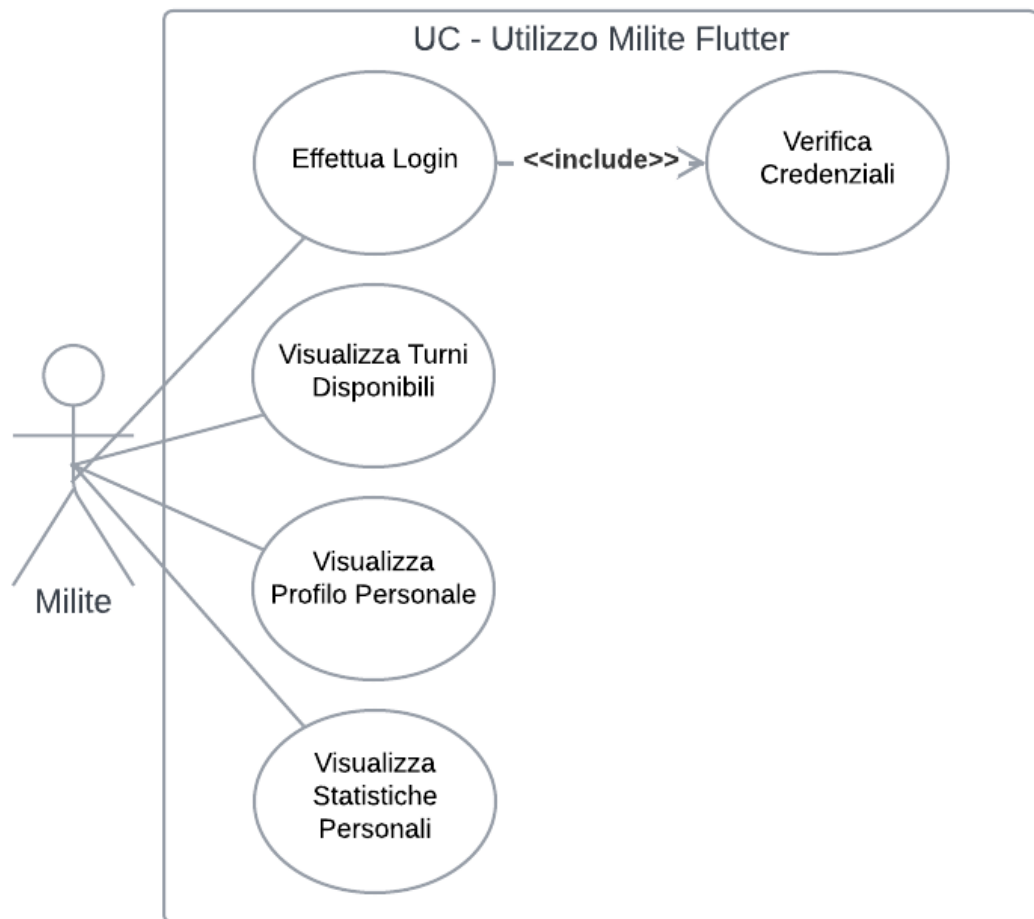


Figura 28 - UC - Utilizzo Milite

Effettua Login: Questo caso d'uso inizia quando il milite decide di effettuare il login nel proprio account personale grazie alle credenziali di accesso.

Visualizza profilo personale: Questo caso d'uso inizia quando il milite decide di visualizzare il proprio profilo personale attraverso il sistema.

Visualizza turni disponibili: Questo caso d'uso inizia quando il milite decide di visualizzare i turni disponibili.

Visualizza statistiche personali: Questo caso d'uso inizia quando il milite decide di visualizzare le proprie statistiche personali attraverso il sistema

ARCHITETTURA

In questo progetto non è stata utilizzata nessuna architettura particolare.

DATABASE

Il Database utilizzato per lo sviluppo in Flutter è lo stesso adoperato per il progetto Android. In questo modo i dati sono coerenti e accessibili indipendentemente dalla piattaforma su cui viene eseguita l'app,

semplificando notevolmente la gestione dei dati e consentendo un'esperienza uniforme per gli utenti su entrambe le piattaforme.

UI (MOCKUP)

Mockup Amministratore



Figura 29 - Amministratore

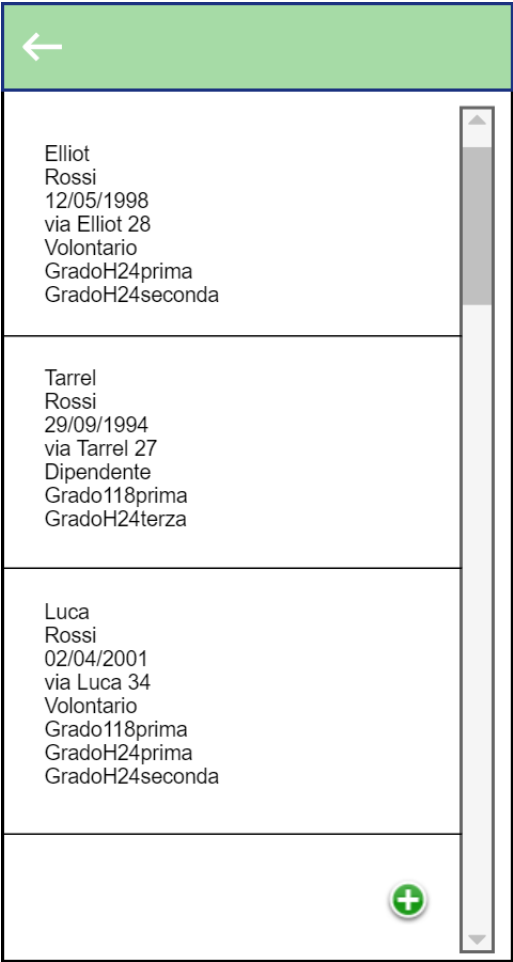



Figura 30 – Gestione Tutti Militi



Nome

Cognome

Data di nascita

Residenza

Volontario/Dipendente

118 Prima

OFF

H24 Prima

OFF

118 Seconda

OFF

H24 Seconda

OFF

118 Terza

OFF

H24 Terza

OFF

Salva

Figura 31 – Gestione Milite Modifica Crea





Nome

Cognome

Data di nascita

Residenza

Grado

Aggiorna Milite

Cancella Milite

Figura 32 – Gestione Milite

←

settimana 118

settimana 118/H24

	H24			118		
	dalle 7 alle 14	dalle 14 alle 21	dalle 21 alle 7	dalle 7 alle 14	dalle 14 alle 21	dalle 21 alle 7
Lunedì 27/03						
Martedì 28/03						
Mercoledì 29/03						
Giovedì 30/03						
Venerdì 31/03						
Sabato 01/04						
Domenica 02/04						

118 ▼

Milite 1a ▼

Lunedì ▼

14 - 21 ▼

Name Surname ▼

Segna/Cancela

Figura 33 – Tabellone Turni Amministratore

Mockup Milite

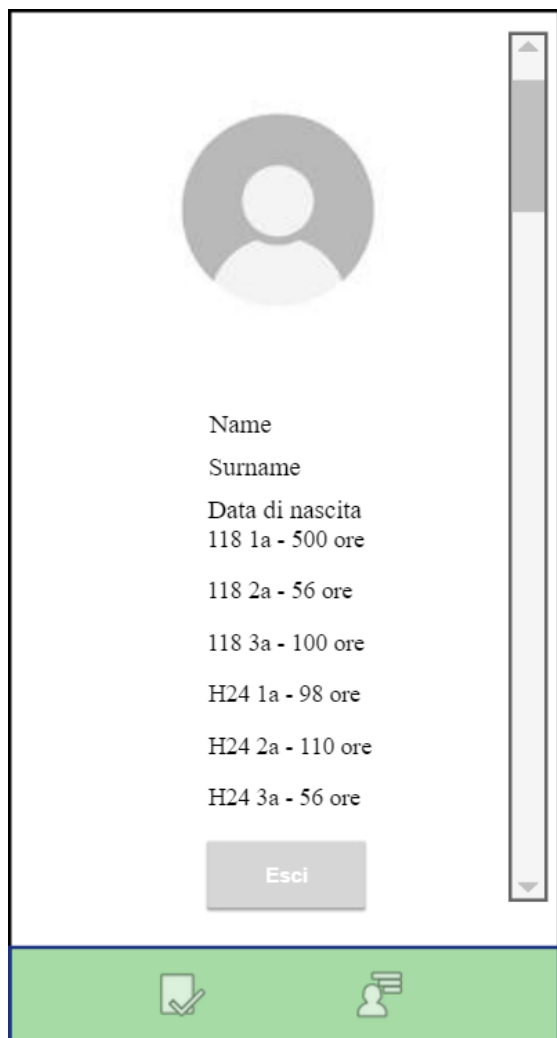


Figura 34 - Profilo Milite Flutter

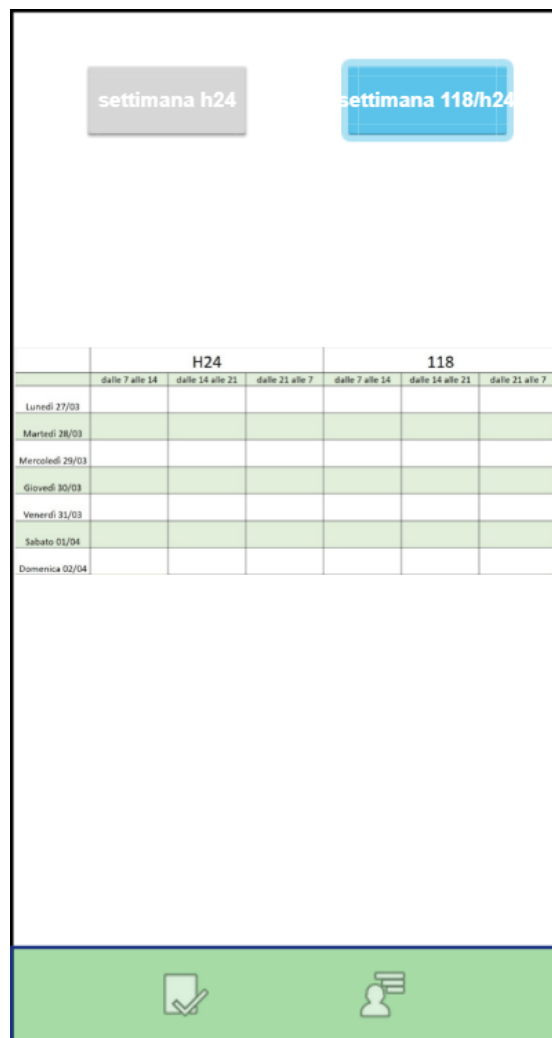


Figura 35 - Tabellone Turni Milite Flutter

SVILUPPO

Di seguito si troveranno le spiegazioni di tutte i componenti individuati durante l'analisi, principalmente sono individuate le schermate di ogni utente del servizio. Possiamo individuare i seguenti macro-componenti: Amministratore, Milite, Login

Milite

Per lo sviluppo in Flutter è stato scelto di non fare distinzione tra l'attore Volontario e l'attore Dipendente.

Questa scelta ha permesso di semplificare la realizzazione dell'applicazione. Il Milite può navigare attraverso due schermate grazie alla bottomNavigationBar, dove sono disponibili due pulsanti: TabelloneTurni e ProfiloMilite.

TabelloneTurni

Grazie a questa schermata il Milite può visualizzare lo stato dei turni con l'aggiornamento in tempo reale. È disponibile la suddivisione con due pulsanti: Settimana 118 e Settimana 118/H24. Questi due pulsanti modificano la tabella a seconda della tabella che si vuole visualizzare.

ProfiloMilite

Grazie a questa schermata il Milite può visualizzare il proprio profilo personale con i propri dati e a seconda del grado del turno che un milite svolge vengono accumulate nelle proprie statistiche, all'interno dell'associazione. All'interno di questa schermata è presente un pulsante per eseguire l'uscita dal profilo.

Amministratore

Per lo sviluppo in Flutter, l'amministratore è stato semplificato mantenendo solo la parte relativa alla gestione dei militi, e aggiungendo la possibilità, come del centralista su android, di poter inserire ed eliminare i militi scelti dalle tabelle.

L'amministratore può navigare tra le schermate attraverso l'uso di due bottoni: Gestione militi e TabelloneTurniAmministratore

Gestione militi

Attraverso questa schermata viene visualizzata la lista completa dei militi, tramite la quale, selezionandone un elemento, è possibile aprire la schermata contenente le informazioni del milite scelto e la possibilità di compiere operazioni di modifica sui dati anagrafici e sui gradi, e di eliminazione del milite. Inoltre, Gestione militi offre la possibilità di poter navigare alla schermata di creazione milite; dopo aver inserito tutti i dati richiesti e aver confermato, viene creato lo username e generata la password relativa al nuovo milite.

TabelloneTurniAmministratore

In questa schermata l'amministratore può visualizzare, attraverso l'uso di due bottoni, entrambe le tabelle e può assegnare un milite ad un turno scelto oppure cancellarlo da uno già presente.

Login

Il login gestisce la navigazione tra la schermata del milite e quella dell'amministratore, controllando se i dati inseriti sono validi per l'uno o per l'altro. In caso negativo, l'utente visualizza a schermo un messaggio che lo avvisa che i dati da lui inseriti non sono validi.

CONCLUSIONI

Durante l'analisi del business case sono state individuate molti requisiti, i requisiti ritenuti "should" sono stati implementati nell'applicazione mentre quelli che sono stati ritenuti "could" non necessari per il seguente sistema, alcuni di questi ritenuti particolarmente laboriosi, sono stati riportati nelle conclusioni per tenere traccia e poter implementarli in un possibile aggiornamento dell'applicazione. Di seguito verranno riportati questi miglioramenti e verranno salvati tramite la tag "Migliorie".

Miglioria: si può colorare una sezione sulla schermata, ad esempio "lunedì" se il menu a tendina è stato settato su lunedì

Miglioria: nell'ambulanza di solito non manca nessuno dei presidi dopo un intero turno di lavoro, per questo motivo è più semplice rendere le caselle spuntate di default in questo modo il milite toglie la spunta dai presidi mancanti.

Miglioria: le ore alle statistiche dei militi vengono aggiunte ogni volta che si prenota, questo per la mancanza di una online function

Miglioria: L'Amministratore deve poter gestire gli orari dei turni e quindi di personalizzarli.

Miglioria: Ci sono dei militi che sono disponibili in un determinato giorno della settimana e quindi vengono inseriti nel turno automaticamente senza essere contattati ogni settimana.

Miglioria: Un milite che vuole cambiare il proprio turno perché impossibilitato a poterlo svolgere può chiedere un "cambio", il proprio nome nel tabellone si colorerà di blu. Se un milite possiede il grado adatto potrà visualizzare questa colorazione blu mentre il milite di grado non sufficiente non vedrà nessuna colorazione.

Miglioria: Ci sono dei militi che possono fare turni parziali (di alcune ore) ad esempio 14/20, nei turni incompleti ci potrebbe essere un pulsante di informazione da premere per vedere le informazioni dei militi che formano il turno su una schermata di informazione che si apre. In alternativa si potrebbe lasciare la possibilità di annotare qualcosa per ogni turno.

Miglioria: I presidi disponibili nelle checklist verranno inseriti dall'amministratore nel sistema.

Miglioria: Si può effettuare le checklist di più ambulanze.

Miglioria: Ci sono più amministratori abilitati ad effettuare le modifiche del sistema.

Miglioria: Alcuni militi possono essere abilitati ad effettuare alcuni turni con le implicazioni sovrastanti in giallo e inoltra bisogna comunque avere la possibilità di assegnare ad un milite dei privilegi di prenotazione su turni dove non è abilitato.

Miglioria: Le settimane non si alternano automaticamente ma manualmente per la mancanza delle online function che eseguono le modifiche in modo temporizzato e automatico.