

Scopo del progetto

Informatizzazione dell’A.V.I.S. di Montemarciano

Sede

Via San Pietro, 1 - 60018 Montemarciano (AN)

Corso

**Ingegneria del Software** • Corso di L. T. in Ingegneria Informatica e dell’Automazione A.A. 2021-2022

Data di pubblicazione

xx/xx/2022

Progettisti / Programmatori

Naja Omar, Piercecchi Gabriel, Rossi Alan, Staffolani Federico

Link di GitHub

<https://github.com/GabrielPiercecchi/pythonProjectAVIS.git>

“*We may have all come on different ships, but we’re in the same boat now.*”

(Martin Luther King, Jr.)

**INDICE**

* Descrizione ..……………………………………………………………………………………………………… pag. 1
* Glossario …………………………………………………………………………………………………………… pag. 3
* Analisi dei requisiti .…………………………………………………………………………………………… pag. 4
* Requisiti funzionali .……………………………………………………………………………….. pag. 4
* package Sistema …………………………………………………………………………. pag. 5
* package Amministratore …………………………………………………………….. pag. 6
* package GestioneUtenti ……………………………………………………………… pag. 6
* package GestionePersonale ………………………………………………………… pag. 7
* package GestioneOperazioni ………………………………………………………. pag. 7
* Requisiti non funzionali ..……………………………………………………………………….. pag. 8
* Diagrammi dei casi d’uso …………………………………………………………………………………. pag. 9
  + package GestioneDonazioni ………………………………………………………………….. pag. 10
  + package GestioneEmergenze ………………………………………………………………… pag. 15
  + package GestioneSistema ……………………………………………………………………… pag. 19
  + package GestioneUtenti ………………………………………………………………………… pag. 29
* Matrice di mapping dei requisiti ………………………………………………………………………. pag. 35
* Diagrammi di analisi …………………………………………………………………………………………. pag. 36
* Diagrammi delle classi di analisi …………………………………………………………….. pag. 36
  + package Attività ………………………………………………………………………….. pag. 36
  + package Servizio …………………………………………………………………………. pag. 37
* Diagrammi di sequenza…………………………………………………………………………… pag. 38
  + Backup ……………………………………………………………………………………….. pag. 38
  + CUDDipendente …………………………………………………………………………. pag. 39
  + InserisciPrenotazione …………………………………………………………………. pag. 40
  + RichiediIscrizione ……………………………………………………………………….. pag. 41
  + VisualizzaTesseraDonatore ………………………………………………………… pag. 42
* Diagrammi di attività ………………………………………………………………………………. pag. 43
  + EffettuaBackup ..………………………………………………………………………….. pag. 43
  + CUDDipendente …………………………………………………………………………… pag. 44
  + InserisciPrenotazione …………………………………………………………………… pag. 45
  + IscriviDonatore ….………………………………………………………………………… pag. 46
  + VisualizzaTesseraDonatore …………………………………………………………. pag. 47
* Diagrammi di progettazione ……………………………………………………………………………… pag. 48
  + Diagrammi delle classi di progettazione ………………………………………………… pag. 48
    - package Model ……………………………………………………………………………. pag. 48
    - package Service …………………………………………………………………………… pag. 49
    - package Controller ……………………………………………………………………… pag. 49
  + Diagrammi delle macchine a stati ………………………………………………………….. pag. 50
    - Dipendente …………………………………………………………………………………. pag. 50
    - Donatore …………………………………………………………………………………….. pag. 51
    - Prenotazione ………………………………………………………………………………. pag. 52
    - Rapportino ………………………………………………………………………………….. pag. 52
    - TesseraDonatore …………………………………………………………………………. pag. 53
  + Diagramma dei componenti ………………………………………………………………….. pag. 54
* Diagramma di deployment ……………………………………………………………………………….. pag. 54
* Mockup …………………………………………………………………………………………………………….. pag. 55

**DESCRIZIONE IN LINGUAGGIO NATURALE**

**Introduzione**

Il progetto proposto consiste nella realizzazione di un software gestionale per la sede dell’**A.V.I.S.** (**A**ssociazione **V**olontari **I**taliani **S**angue) di Montemarciano, ubicata in Via San Pietro, 1 - 60018 Montemarciano (AN).

La progettazione sarà basata interamente sulle informazioni reperite dalle interviste con il sig. Romano Giuseppe (**volontario** nel settore emergenze), il sig. Tonti Fabrizio (responsabile del settore donazioni) e il sig. Sbaffo Andrea (Presidente della sede A.V.I.S. di Montemarciano).

Il software dovrà pertanto supportare le operazioni correlate alle prenotazioni delle donazioni (inserimento, modifica ed eliminazione) effettuate dai **volontari** e quelle correlate al **servizio emergenza** di competenza del **personale** del 118. Il sistema prevede l’accesso mediante login dell’utente che, una volta registrato dall’**amministratore**, potrà effettuare ciò che è consentito dal suo ruolo, distinguendo tra: **donatore** e personale (di quest’ultimo fanno parte i **dipendenti** e i **volontari**).

**Servizio donazioni**

Il software gestisce automaticamente tutti i servizi di prenotazione delle donazioni del sangue. Il nuovo donatore viene registrato dall’**amministratore** che ne inserisce i dati nel sistema (nome, cognome, codice fiscale, cellulare, data di nascita, email, password, gruppo sanguigno, **idoneità** a donare e numero di tessera), una volta fatto ciò il donatore può procedere a scegliere il giorno e l’ora della prenotazione potendo visualizzare tutti gli orari disponibili in cui effettuarla e, qualora intendesse rinunciare all’appuntamento o cambiare orario e giorno, la prenotazione può essere modificata o eliminata dall’**amministratore**. Se una persona ritenuta non idonea a donare prova ad effettuare una prenotazione, il tentativo di prenotazione viene annullato dal sistema che riconosce la non **idoneità**.

Ogni donatore dispone inoltre di una tessera (dotata di codice di identificazione) che contiene informazioni sul numero totale di donazioni e le date in cui sono state effettuate. Il donatore e la rispettiva tessera possono essere eliminati dal sistema soltanto dall’**amministratore**, previa richiesta di rimozione.

Il calendario delle prenotazioni delle donazioni è gestito dal sistema che tiene conto di tutti gli orari disponibili e quindi prenotabili, esso è anche sottoposto a **backup** periodico (ogni giorno alle ore 23:00) e l’**amministratore** ha accesso ad esso con le funzioni di ricerca.

**Servizio emergenze**

L’altro aspetto di cui si occupa il software è l’organizzazione del servizio di pronto soccorso (effettuato dagli **autisti soccorritori**) che, in coordinamento con il sistema sanitario nazionale, provvede sia alla gestione delle emergenze sia ad altri servizi come l’assistenza e il trasporto di persone fragili che rientrano nella definizione di **servizi secondari**. Tutti questi servizi sono effettuati dal personale, di cui fanno parte i **dipendenti** e i **volontari**.

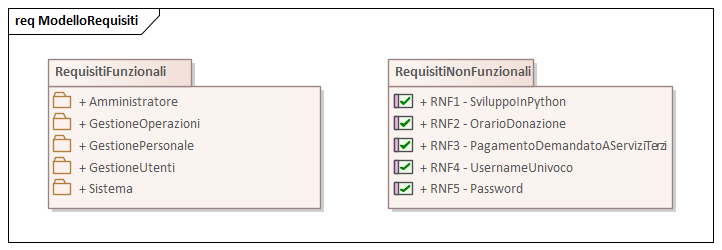
Il personale è registrato nel sistema dall’**amministratore** che inserisce i dati anagrafici come descritto precedentemente per i donatori, nel caso del **dipendente** vengono specificate anche le coordinate bancarie in quanto a differenza del volontario è a tutti gli effetti un lavoratore retribuito; dunque, nel momento stabilito il sistema procederà ad erogare lo stipendio nell’IBAN specificato. Il sistema, inoltre, per ogni utente registrato come parte del personale, contiene anche tutti gli orari lavorativi, i **turni** e lo stato che indica se il **dipendente** o il **volontario** è disponibile per essere messo in servizio e non è in congedo o in sospensione.

Tutta la cronistoria di un servizio di **emergenza** o **secondario** è contenuta nel **rapportino**, un documento ufficiale dove sono specificati il giorno e l’ora in cui il **dipendente** o il **volontario** ha effettuato il servizio e il numero di chilometri percorsi durante il tragitto, così da indicare al sistema la quantità di denaro spesa e quindi da rimborsare. Una volta inserito il **rapportino** non può essere modificato e soltanto l’**amministratore** può visualizzare il contenuto di tutti i rapportini.

L’**amministratore** può inoltre rimuovere dal sistema i **dipendenti** licenziati o i **volontari** in congedo permanente, come avveniva per i donatori. Inoltre, anche l’elenco dei rapportini e del personale registrato viene sottoposto periodicamente a un **backup** periodico, nello stesso giorno e orario di quello delle donazioni, così da non perdere i dati in caso di problemi tecnici.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| PAROLA | TIPO | DEFINIZIONE | SINONIMI | OMONIMI |
| amministra-tore | Business | professionista incaricato dell’am-ministrazione dei sistemi dell’A.V.I.S. | admin | nessuno |
| autista soccorritore | Business | esperto nella guida di veicoli per il trasporto d’urgenza e mezzi di soc-corso, tipicamente ambulanze e automediche | autista | nessuno |
| backup | Tecnico | procedura con la quale si realizza una copia di sicurezza di un certo insieme di dati | copia di b. | nessuno |
| dipendente | Business | colui che, nella gerarchia aziendale, è ed altri soggetto o subordinato | nessuno | nessuno |
| idoneità | Business | possesso dei requisiti necessari per effettuare una donazione | nessuno | nessuno |
| intervento | Business | impiego immediato in caso di ur-genza | nessuno | nessuno |
| personale | Business | il complesso dei dipendenti e volon-tari dell’A.V.I.S. | nessuno | nessuno |
| rapportino | Business | referto redatto alla fine di un servizio emergenza, secondario o civile | nessuno | nessuno |
| servizio emergenza | Business | attività legata ad interventi che coinvolgono il 118 | emergenza | nessuno |
| servizio secondario | Business | attività legata all’utilizzo di ambu-lanze che non prevedono un’emer-genza | secondario | nessuno |
| servizio civile | Business | il servizio civile universale è la scelta volontaria di dedicare alcuni mesi della propria vita al servizio di difesa, non armata e non violenta, della Patria | nessuno | nessuno |
| turno | Business | avvicendamento secondo un ordine prestabilito | disponibilità | nessuno |
| volontario | Business | persona che gratuitamente mette a disposizione la propria attività | nessuno | nessuno |

**ANALISI DEI REQUISITI**

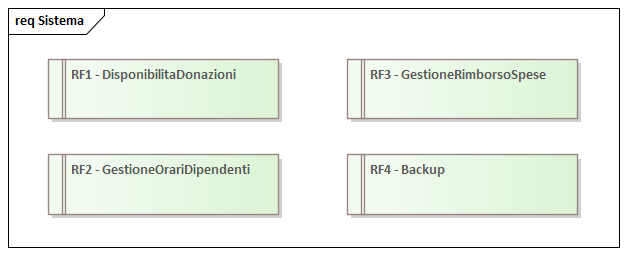


**REQUISITI FUNZIONALI**

Immagine che contiene testo

Descrizione generata automaticamente

**PACKAGE SISTEMA**



|  |  |  |  |
| --- | --- | --- | --- |
| **RF1 - DisponibilitaDonazioni** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire la disponibilità degli orari per le donazioni | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RF2 - GestioneOrariDipendenti** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 2.0 |  |
| Il sistema dovrà gestire l’assegnazione degli orari di servizio dei di-pendenti | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RF3 - GestioneRimborsoSpese** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire il rimborso delle spese | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RF4 - Backup** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà effettuare un backup periodico dei dati | | |

**PACKAGE AMMINISTRATORE**

Immagine che contiene testo

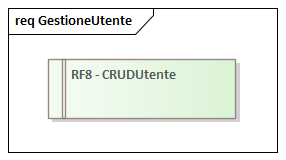
Descrizione generata automaticamente

|  |  |  |  |
| --- | --- | --- | --- |
| **RF5 - CRUDDonatore** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire le attività CRUD sul donatore | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RF6 - CRUDVolontario** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire le attività CRUD sul volontario | | |

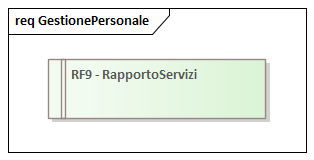
|  |  |  |  |
| --- | --- | --- | --- |
| **RF7 - CRUDDipendente** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire le attività CRUD sul dipendente | | |

**PACKAGE GESTIONEUTENTE**



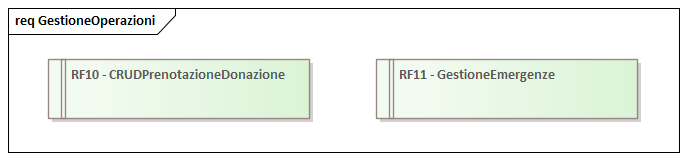
|  |  |  |  |
| --- | --- | --- | --- |
| **RF8 - CRUDUtente** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire le attività CRUD sull’utente | | |

**PACKAGE GESTIONEPERSONALE**



|  |  |  |  |
| --- | --- | --- | --- |
| **RF9 - RapportoServizi** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il personale dovrà compilare e caricare nel sistema i rapportini per le emergenze e i servizi secondari | | |

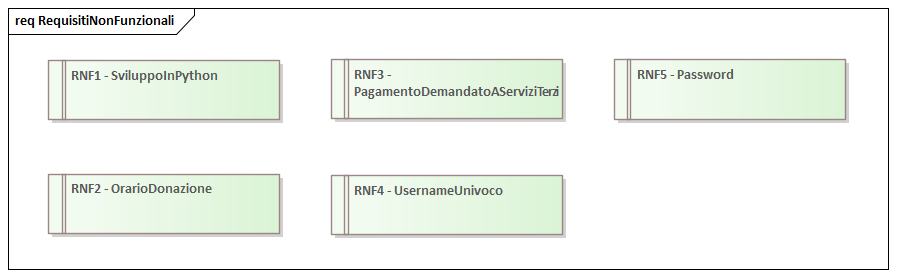
**PACKAGE GESTIONEOPERAZIONI**



|  |  |  |  |
| --- | --- | --- | --- |
| **RF10 - CRUDPrenotazioneDonazione** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire le attività CRUD sulla prenotazione delle donazioni di sangue | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RF11 – GestioneEmergenze** | | | |
| “Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il sistema dovrà gestire (visualizzare e confermare) le emergenze assegnate da enti esterni | | |

**REQUISITI NON FUNZIONALI**



|  |  |  |  |
| --- | --- | --- | --- |
| **RNF1 – SviluppoInPython** | | | |
| “Non-Functional” | Status: Validated | Priority: High | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il progetto deve essere implementato nel linguaggio di program-mazione Python | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RNF2 - OrarioDonazione** | | | |
| “Non-Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| L’orario in cui è possibile donare è stabilito dall’ospedale | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RNF3 - PagamentoDemandatoAServiziTerzi** | | | |
| “Non-Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| Il pagamento dello stipendio dei dipendenti è demandato a servizi terzi | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **RNF4 - UsernameUnivoco** | | | |
| “Non-Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| L’amministratore e ogni altro utente devono essere identificati tramite un username univoco | | |

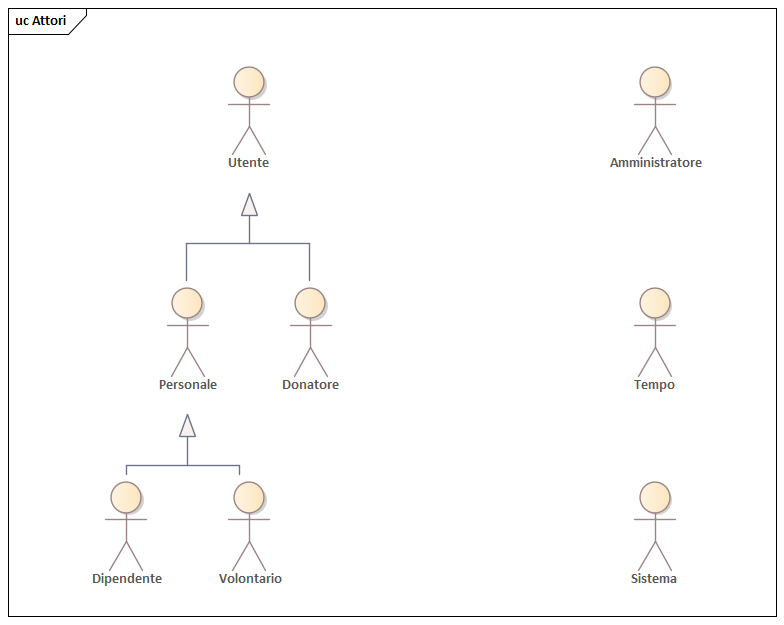
|  |  |  |  |
| --- | --- | --- | --- |
| **RNF5 - Password** | | | |
| “Non-Functional” | Status: Validated | Priority: Medium | Difficulty: Medium |
| Phase: 1.0 | Version: 1.0 |  |
| L’amministratore e ogni altro utente devono essere in possesso di una password univocamente associata al relativo username | | |

**DIAGRAMMI DEI CASI D’USO**

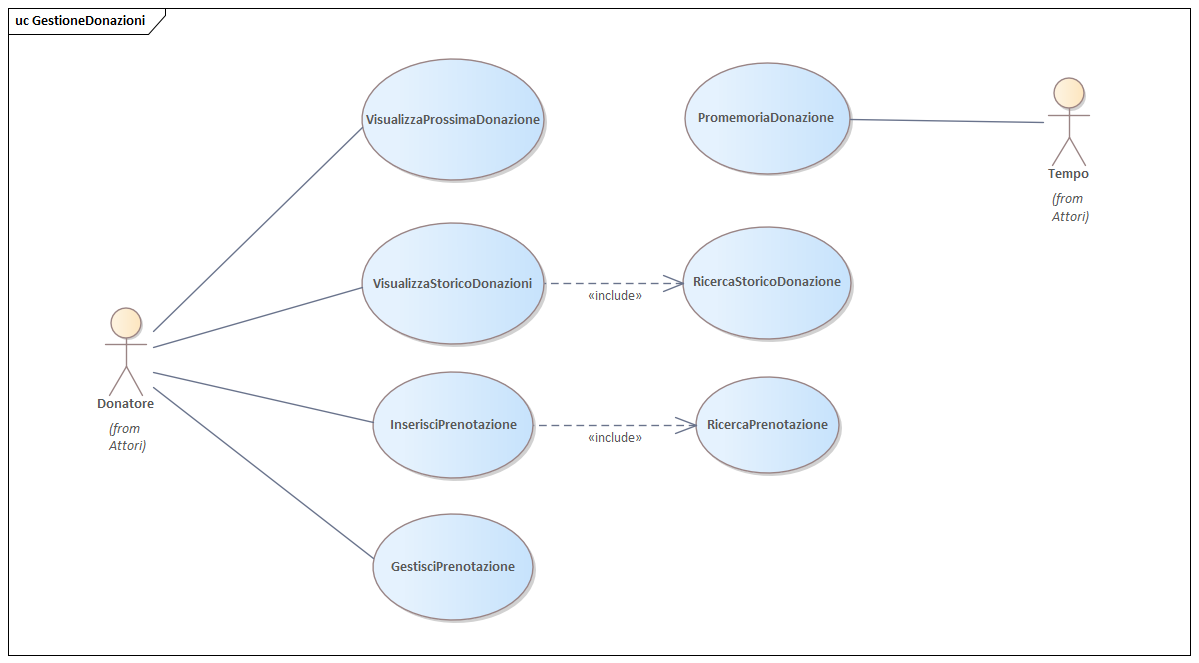
Immagine che contiene tavolo

Descrizione generata automaticamente

**ATTORI**



**PACKAGE GESTIONEDONAZIONI**



|  |
| --- |
| Caso d’uso: InserisciPrenotazione |
| ID: 1 |
| Breve descrizione:  Il Donatore prenota una donazione |
| Attori primari:  Donatore |
| Attori secondari:  Nessuno |
| Precondizioni:   1. Il Donatore deve essere idoneo a donare per poter prenotare una donazione 2. Tra una donazione e la successiva donazione - da parte del Donatore - devono essere rispettati i periodi di tempo previsti dalla normativa vigente    1. Il lasso di tempo intercorso tra una donazione di sangue e una successiva donazione di sangue – da parte del Donatore, maschio o femmina in età non fertile - deve essere di almeno tre mesi    2. Il lasso di tempo intercorso tra una donazione di sangue e una successiva donazione di sangue – da parte del Donatore, femmina in età fertile - deve essere di almeno tre mesi e il Donatore deve aver effettuato al più un’altra donazione nel corso dell’ultimo anno    3. Il lasso di tempo intercorso tra una donazione di plasma e una successiva donazione di sangue deve essere di almeno 14 giorni    4. Il lasso di tempo intercorso tra una donazione di plasma e una successiva donazione di plasma deve essere di almeno 14 giorni    5. Il lasso di tempo intercorso tra una donazione di sangue e una successiva donazione di plasma deve essere di almeno un mese |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Donatore seleziona “*prenota donazione*” 2. if Una prenotazione per una donazione futura è già presente nel sistema    1. Viene visualizzato un messaggio di errore ad hoc 3. else Il sistema consente al Donatore di procedere con la fase di prenotazione    1. Il Donatore seleziona una data - dal novero delle date disponibili - in cui vuole effettuare la donazione    2. Il Donatore seleziona un orario - dal novero degli orari disponibili - in cui vuole effettuare la donazione    3. Il sistema registra la prenotazione    4. Il sistema aggiorna gli orari disponibili per le prenotazioni |
| Postcondizioni:   1. L’orario della prenotazione selezionato dal Donatore - relativo al giorno della donazione - non sarà disponibile per altre prenotazioni, salvo annullamento della suddetta |
| Sequenze degli eventi alternative:  Nessuna |

|  |
| --- |
| Caso d’uso: GestisciPrenotazione |
| ID: 2 |
| Breve descrizione:  Il Donatore gestisce la prenotazione per una donazione prefissata |
| Attori primari:  Donatore |
| Attori secondari:  Nessuno |
| Precondizioni:   1. Il Donatore deve aver effettuato una prenotazione per una donazione futura |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Donatore seleziona una prenotazione già presente nel sistema 2. If Il Cliente seleziona “*conferma prenotazione*”    1. Il sistema conferma la prenotazione 3. If Il Cliente seleziona “*annulla prenotazione*”    1. Il sistema elimina la prenotazione    2. Il sistema aggiorna gli orari disponibili per le prenotazioni |
| Postcondizioni:  Nessuna |
| Sequenze degli eventi alternative:  Nessuna |

|  |
| --- |
| Caso d’uso: PromemoriaDonazione |
| ID: 3 |
| Breve descrizione:  Il sistema notifica al Donatore una prenotazione imminente |
| Attori primari:  Sistema |
| Attori secondari:  Tempo |
| Precondizioni:   1. Il Donatore deve aver effettuato una prenotazione per una donazione futura |
| Sequenza degli eventi principale:   1. Il caso d’uso ha inizio quando il Sistema rileva un intervallo di tempo uguale a una quantità prestabilita tra il Tempo e la data - ora della prenotazione 2. Il Sistema invia al Donatore un promemoria contenente le data e l’orario della prenotazione 3. Il sistema seleziona automaticamente la prenotazione per la suddetta donazione    1. include(GestisciPrenotazione) |
| Postcondizioni:  Nessuna |
| Sequenze degli eventi alternative:  Nessuna |

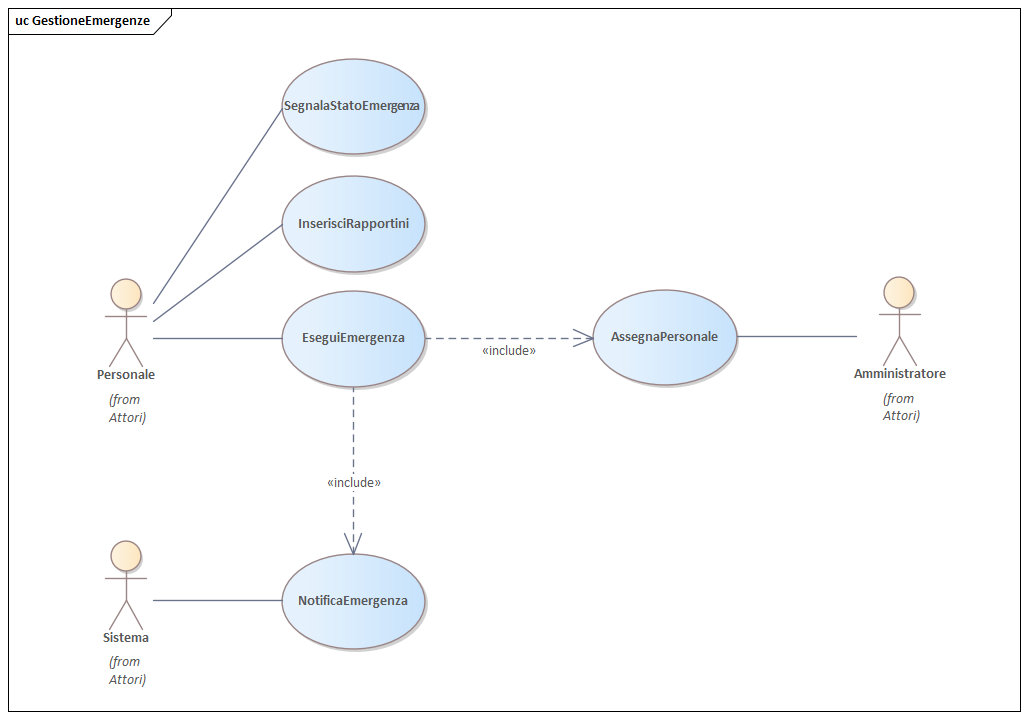
|  |
| --- |
| Caso d’uso: VisualizzaStoricoDonazioni |
| ID: 4 |
| Breve descrizione:  Il sistema individua tutte le prenotazioni effettuate dal Donatore e le mostra al Donatore |
| Attori primari:  Donatore |
| Attori secondari:  Nessuno |
| Precondizioni:  Nessuna |
| Sequenza degli eventi principale:   1. Il caso d’uso ha inizio quando il Donatore seleziona “*visualizza donazioni passate*” 2. if Il Donatore ha effettuato almeno una donazione    1. Il sistema mostra al Donatore tutte le donazioni pregresse 3. else Viene visualizzato un messaggio di errore ad hoc |
| Postcondizioni:  Nessuna |
| Sequenze degli eventi alternative:  Nessuna |

|  |
| --- |
| Caso d’uso: RicercaStoricoDonazione |
| ID: 5 |
| Breve descrizione:  Il sistema individua alcune prenotazioni in base ai criteri di ricerca specificati dal Donatore e le mostra al Donatore |
| Attori primari:  Donatore |
| Attori secondari:  Nessuno |
| Precondizioni:   1. Il Donatore deve aver selezionato “*visualizza donazioni passate*” 2. Il Donatore deve aver effettuato almeno una donazione |
| Sequenza degli eventi principale:   1. Il caso d’uso ha inizio quando il Donatore seleziona “*ricerca donazioni passate*” 2. if Il Donatore seleziona un range di date    1. While(esiste una donazione effettuata dal Donatore)       1. if La donazione è stata effettuata nel periodo selezionato          1. Il sistema mostra al Donatore la donazione 3. else if Il Donatore seleziona una categoria di donazione (“*donazione di sangue*” o “*plasmaferesi*”)    1. While(esiste una donazione effettuata dal Donatore)       1. if La donazione appartiene alla categoria selezionata          1. Il sistema mostra al Donatore la donazione 4. else    1. Il sistema mostra al Donatore tutte le donazioni pregresse |
| Postcondizioni:  Nessuna |
| Sequenze degli eventi alternative:  Nessuna |

|  |
| --- |
| Caso d’uso: VisualizzaProssimaDonazione |
| ID: 6 |
| Breve descrizione:  Il Donatore visualizza la prenotazione per una donazione in programma |
| Attori primari:  Donatore |
| Attori secondari:  Nessuno |
| Precondizioni:   1. Il Donatore deve aver effettuato una prenotazione per una donazione |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Donatore seleziona “*visualizza prossima donazione*” 2. Il sistema seleziona la prenotazione per la donazione in programma e la mostra al Donatore    1. include(GestisciPrenotazione) |
| Postcondizioni:  Nessuna |
| Sequenze degli eventi alternative:  Nessuna |

|  |
| --- |
| Caso d’uso: RicercaPrenotazioni |
| ID: 7 |
| Breve descrizione:  Questo caso d’uso permette la ricerca di donazioni disponibili e prenotabili da parte del Donatore |
| Attori primari:  Donatore |
| Attori secondari:  Sistema |
| Precondizioni:  Nessuna |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Donatore cerca delle donazioni disponibili e prenotabili 2. Il Sistema mostra le date con le rispettive funzioni disponibili e prenotabili |
| Postcondizioni:  Le date con le donazioni disponibili e prenotabili sono state mostrate |
| Sequenza degli eventi alternativa:  Nessuna |

**PACKAGE GESTIONEERMERGENZE**



|  |
| --- |
| Caso d’uso: AssegnaPersonale |
| ID: 8 |
| Breve descrizione:  Questo caso d’uso permette l’assegnamento del Personale in modo da eseguire un'emergenza |
| Attori primari:  Amministratore |
| Attori secondari:  Personale, Sistema |
| Precondizioni:  L’Amministratore deve disporre di un account di utilizzo con appropriati diritti  d’accesso e  il sistema deve segnalare un’emergenza. |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’Amministratore vuole effettuare un assegnamento del Personale per gestire un'emergenza 2. L’Amministratore visualizza lo stato di disponibilità del Personale    1. include(RicercaPersonale)    2. if membri del Personale sono disponibili       1. L’Amministratore assegna il Personale per l’emergenza    3. else       1. viene visualizzato il messaggio di errore opportuno |
| Postcondizioni:  Il Personale è stato assegnato |
| Sequenza degli eventi alternativa:  nessuna |

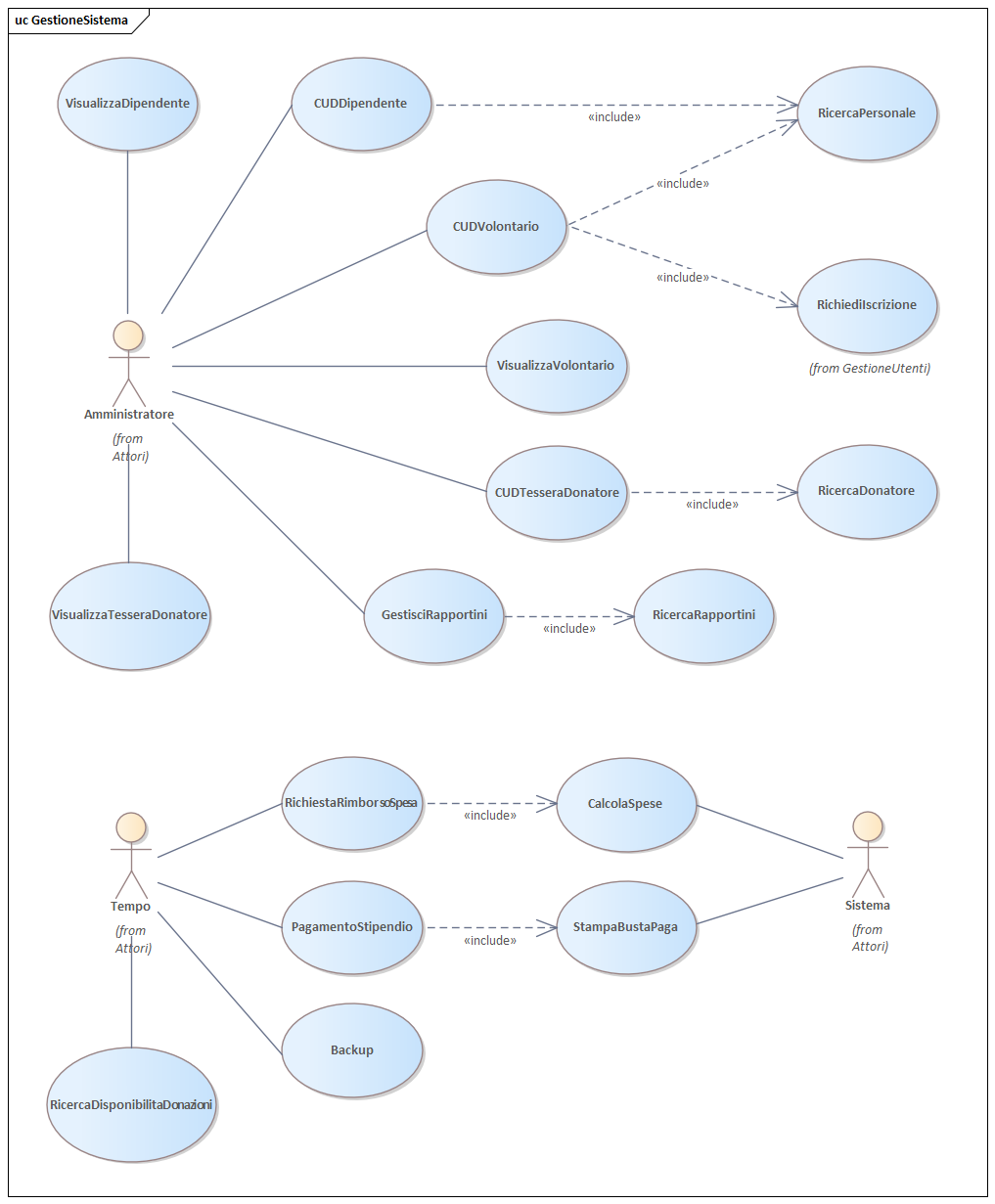
|  |
| --- |
| Caso d’uso: EseguiEmergenza |
| ID: 9 |
| Breve descrizione:  Questo caso d’uso permette l’esecuzione di un'emergenza |
| Attori primari:  Personale |
| Attori secondari:  Amministratore, Sistema |
| Precondizioni:  Il Personale deve disporre di un account di utilizzo con appropriati diritti  d’accesso e la disponibilità ad eseguire l’emergenza. |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Personale deve eseguire un’emergenza 2. if Il Personale è disponibile    1. include(RicercaDonatore)    2. ifè presente un’emergenza       1. include(NotificaEmergenza)       2. Il Personale esegue l’emergenza;       3. include*(*AssegnaPersonale*)* 3. else    * 1. viene visualizzato il messaggio di errore opportuno; |
| Postcondizioni:  Il Personale ha eseguito l’emergenza |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’uso: InserisciRapportini |
| ID: 10 |
| Breve descrizione:  Questo caso d’uso permette l’inserimento nel Sistema dei rapportini |
| Attori primari:  Personale |
| Attori secondari:  Sistema |
| Precondizioni:  Il Personale deve disporre di un account di utilizzo con appropriati diritti  d’accesso. |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Personale deve inserire un rapportino nel Sistema 2. if Il Personale deve inserire un rapportino    1. Il Personale carica il rapportino nell’attore secondario    2. Il Sistema salva i dati contenuti nel rapportino 3. else    1. il caso d’uso termina |
| Postcondizioni:  Il rapportino è stato inserito nel sistema |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’uso: NotitficaEmergenza |
| ID: 11 |
| Breve descrizione:  Questo caso d’uso permette la segnalazione di un'emergenza da parte di enti esterni |
| Attori primari:  Sistema |
| Attori secondari:  Amministratore |
| Precondizioni:  Il Sistema deve ricevere un’emergenza da un ente esterno |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Sistema riceve la segnalazione di un’emergenza da un ente esterno 2. Il Sistema segnala la notifica dell’emergenza all’Amministratore |
| Postcondizioni:  L’emergenza è stata notificata |
| Sequenza degli eventi alternativa:  Nessuno |

|  |
| --- |
| Caso d’uso: SegnalaStatoEmergenza |
| ID: 12 |
| Breve descrizione:  Questo caso d’uso permette di segnalare lo stato in cui si trova l’emergenza presa in carico |
| Attori primari:  Personale |
| Attori secondari:  Sistema |
| Precondizioni:  Il Personale deve disporre di un account di utilizzo con appropriati diritti  d’accesso. |
| Sequenza degli eventi principale:   1. il caso d’uso inizia quando il Personale segnala lo stato dell’emergenza presa in carico 2. if Il Personale segnala che sta eseguendo l’emergenza    1. Il Sistema modifica la disponibilità dell’attore primario in “occupato” 3. else ifIl Personale segnala che ha terminato l’emergenza    1. Il Sistema modifica la sua disponibilità in "disponibile" 4. else    1. viene visualizzato di errore opportuno |
| Postcondizioni:  Lo stato dell’emergenza è stato segnalato |
| Sequenza degli eventi alternativa:  Nessuna |

**PACKAGE GESTIONESISTEMA**



|  |
| --- |
| Caso d’Uso: Backup |
| ID: 13 |
| Breve descrizione:  Questo caso d’uso permette di salvare tutti i dati nel sistema ogni giorno alle ore 23:00 |
| Attori primari:  Tempo |
| Attori secondari:  Sistema |
| Precondizioni:  Nessuna |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando sono le ore 23:00 2. il Sistema preleva i dati dei rapportini 3. il Sistema copia i dati dei rapportini sul disco 4. il Sistema preleva i dati del Personale 5. il Sistema copia i dati del Personale sul disco 6. il Sistema preleva i dati dei Donatori 7. il Sistema copia i dati dei Donatori sul disco |
| Postcondizioni:  I dati sono stati salvati |
| Sequenza degli eventi alternativa:  Nessuno |

|  |
| --- |
| Caso d’Uso: CalcolaSpese |
| ID: 14 |
| Breve descrizione:  Questo caso d’uso permette di calcolare le spese sostenute nell’arco di un lasso di tempo specifico |
| Attori primari:  Tempo |
| Attori secondari:  Sistema |
| Precondizioni:  Nessuno |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’attore Tempo individua come data odierna il primo giorno del mese 2. While (il giorno del mese precedente è compreso tra 1 e 31)    1. if viene trovato un servizio       1. il Sistema calcola le spese dovute ai trasporti, distinguendo tra servizi secondari e servizi emergenze    2. il Sistema calcola le spese dovute a rifornimenti di materiali (mascherine, guanti monouso etc.)    3. Il Sistema calcola tutte le spese effettuate |
| Postcondizioni:  Le spese sono state calcolate |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: CUDDipendente |
| ID: 15 |
| Breve descrizione:  Questo caso d'uso permette l'inserimento, la rimozione o la modifica di un Dipendente |
| Attori primari:  Amministratore |
| Attori secondari:  Dipendente, Sistema |
| Precondizioni:  L’Amministratore deve disporre di un account di utilizzo con appropriati diritti di accesso |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’Amministratore vuole effettuare un’operazione CUD su un Dipendente 2. if l’Amministratore vuole inserire un nuovo Dipendente    1. L’Amministratore inserisce i dati del nuovo Dipendente    2. include(RicercaPersonale)    3. *if* il Dipendente è già presente       1. viene visualizzato il messaggio di errore opportuno    4. else       1. il Sistema memorizza i dati 3. 3. else if L’Amministratore vuole aggiornare i dati del Dipendente    1. L’Amministratore inserisce la chiave del Dipendente    2. include(RicercaPersonale)    3. if Il Dipendente viene trovato       1. L’Amministratore scrive i nuovi dati da inserire       2. Il Sistema aggiorna i dati 4. 4. else if L’Amministratore vuole cancellare un Dipendente    1. L’Amministratore inserisce la chiave del Dipendente    2. include(RicercaPersonale)    3. if Il Dipendente viene trovato       1. il Sistema procede con la sua rimozione    4. else       1. viene visualizzato il messaggio di errore opportuno |
| Postcondizioni:  I dati di uno o più Dipendenti sono stati aggiornati/aggiunti/eliminati |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: CUDTesseraDonatore |
| ID: 16 |
| Breve descrizione:  Questo caso d'uso permette l'inserimento, la rimozione o la modifica di una tessera di Donatore |
| Attori primari:  Amministratore |
| Attori secondari:  Sistema |
| Precondizioni:  L’Amministratore deve disporre di un account di utilizzo con appropriati diritti  d’accesso. |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’Amministratore vuole effettuare un operazione CUD su una tessera 2. if L’Amministratore vuole inserire una nuova tessera    1. L’Amministratore inserisce i dati della nuova tessera    2. include(RicercaDonatore)    3. ifla tessera del Donatore è già presente       1. viene visualizzato il messaggio di errore opportuno    4. else       1. il Sistema memorizza i dati 3. else if L’Amministratore vuole aggiornare i dati della tessera    1. L’Amministratore inserisce la chiave della tessera    2. include(RicercaDonatore)    3. if la tessera viene trovata       1. L’Amministratore scrive i nuovi dati da inserire       2. Il Sistema aggiorna i dati 4. else if L’Amministratore vuole cancellare una tessera    1. L’Amministratore inserisce la chiave della tessera    2. include(RicercaDonatore)    3. ifla tessera viene trovata       1. Il Sistema procede con la sua rimozione    4. else       1. viene visualizzato il messaggio di errore opportuno |
| Postcondizioni:  Nessuna |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: CUDVolontario |
| ID: 17 |
| Breve descrizione:  Questo caso d'uso permette l'inserimento, la rimozione o la modifica di un Volontario |
| Attori primari:  Amministratore |
| Attori secondari:  Volontario, Sistema |
| Precondizioni:  L’Amministratore deve disporre di un account di utilizzo con appropriati diritti  d’accesso. |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’Amministratore vuole effettuare un operazione CUD su un Volontario; 2. if L’Amministratore vuole inserire un nuovo Volontario    1. L’Amministratore inserisce i dati del nuovo Volontario;    2. include(RicercaPersonale)    3. ifIl Volontario è già presente       1. viene visualizzato il messaggio di errore opportuno;    4. else       1. include(RichiediIscrizione);       2. if la richiesta è stata inviata correttamente          1. il sistema memorizza i dati;       3. else          1. viene visualizzato il messaggio di errore opportuno; 3. else if L’Amministratore vuole aggiornare i dati del volontario    1. L’Amministratore inserisce la chiave del volontario;    2. include(RicercaPersonale);    3. if Il Volontario viene trovato       1. L’Amministratore scrive i nuovi dati da inserire;       2. Il Sistema aggiorna i dati; 4. else if L’Amministratore vuole eliminare un Volontario    1. L’Amministratore inserisce la chiave del Volontario;    2. include(RicercaPersonale);    3. if il Volontario viene trovato       1. il Sistema procede con la rimozione;    4. else       1. viene visualizzato un opportuno messaggio di errore. |
| Postcondizioni:  Nessuna |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: GestisciRapportini |
| ID: 18 |
| Breve descrizione:  Questo caso d’uso permette di visualizzare e/o modificare un rapportino |
| Attori primari:  Amministratore |
| Attori secondari:  Sistema |
| Precondizioni:  L’attore primario deve disporre di un account di utilizzo con appropriati diritti  d’accesso |
| Sequenza degli eventi principale:   1. il caso d’uso inizia quando L’Amministratore vuole visualizzare un rapportino 2. l’attore primario inserisce la chiave del rapportino 3. include(RicercaRapportini) 4. if il rapportino viene trovato    1. L’Amministratore visualizza il contenuto del rapportino    2. if L’Amministratore modifica il rapportino       1. il Sistema aggiorna le modifiche 5. else    1. viene visualizzato un opportuno messaggio di errore |
| Postcondizioni:  Il rapportino è stato visualizzato e/o modificato |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: PagamentoStipendio |
| ID: 19 |
| Breve descrizione:  Questo caso d’uso permette di effettuare i pagamenti dovuti |
| Attori primari:  Tempo |
| Attori secondari:  Sistema, Dipendente |
| Precondizioni:  il Dipendente deve aver inserito nel sistema il proprio IBAN |
| Sequenza degli eventi principale:   1. il caso d’uso inizia quando viene effettuato un controllo temporale giornaliero; 2. if sono trascorsi più di 30 giorni dall’ultimo pagamento    1. include(StampaBustaPaga);    2. il sistema emette i pagamenti ai Dipendenti; 3. else    1. il Sistema aumenta di 1 il contatore dei giorni. |
| Postcondizioni:  I pagamenti dovuti  sono stati effettuati |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: RicercaDisponibiltàDonazione |
| ID: 20 |
| Breve descrizione:  Questo caso d’uso permette al Sistema di cercare le disponibilità per le donazioni ogni lunedì alle ore 08:30 per la settimana stessa |
| Attori primari:  Tempo |
| Attori secondari:  Sistema |
| Precondizioni:  La data odierna corrisponde alle ore 08:30 di ogni lunedì |
| Sequenza degli eventi principale:   1. il caso d’uso inizia quando il Tempo effettua il controllo sulle disponibilità 2. if ci sono disponibilità    1. il Sistema aggiorna le disponibilità correnti 3. else    1. il Sistema comunica la mancanza di disponibilità |
| Postcondizioni:  Le disponibilità delle donazioni sono aggiornate |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: RicercaDonatore |
| ID: 21 |
| Breve descrizione:  questo caso d’uso permette la ricerca dei dati di un donatore |
| Attori primari:  Amministratore |
| Attori secondari:  Donatore, Sistema |
| Precondizioni:  L’Amministratore deve disporre di un account di utilizzo con appropriati diritti d’accesso |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando L’Amministratore vuole cercare i dati di un Donatore 2. L’Amministratore specifica il codice fiscale del Donatore 3. il Sistema cerca utilizzando il dato inserito in input 4. if Il Donatore cercato è presente nel Sistema    1. il Sistema fa visualizzare tutte le informazioni relative al Donatore 5. else    1. il Sistema visualizza un opportuno messaggio di errore |
| Postcondizioni:  La ricerca del donatore è stata effettuata o si è verificata l’inesistenza di esso nel sistema |
| Sequenza degli eventi alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: RicercaPersonale |
| ID: 22 |
| Breve descrizione:  L’Amministratore ricerca i dati di un membro generico dell’AVIS |
| Attori primari:  Amministratore |
| Attori secondari:  Dipendente, Volontario |
| Precondizioni:  Nessuna |
| Sequenza degli eventi principale  1.  Il caso d’uso inizia quando l’Amministratore seleziona “*ricerca personale*”  2.  L’Amministratore inserisce il nome di un Dipendente o un Volontario  2.1.   While (esiste un Dipendente o un Volontario avente il nome inserito)  2.1.1. if il nome del Dipendente o del Volontario corrisponde al nome inserito  2.1.1.1.            Il sistema mostra all’Amministratore il Dipendente o il Volontario |
| Postcondizioni:  Il sistema restituisce all’Amministratore i dati dell’attore secondario ricercato |
| Sequenza degli eventi alternativa:  Nessuna |

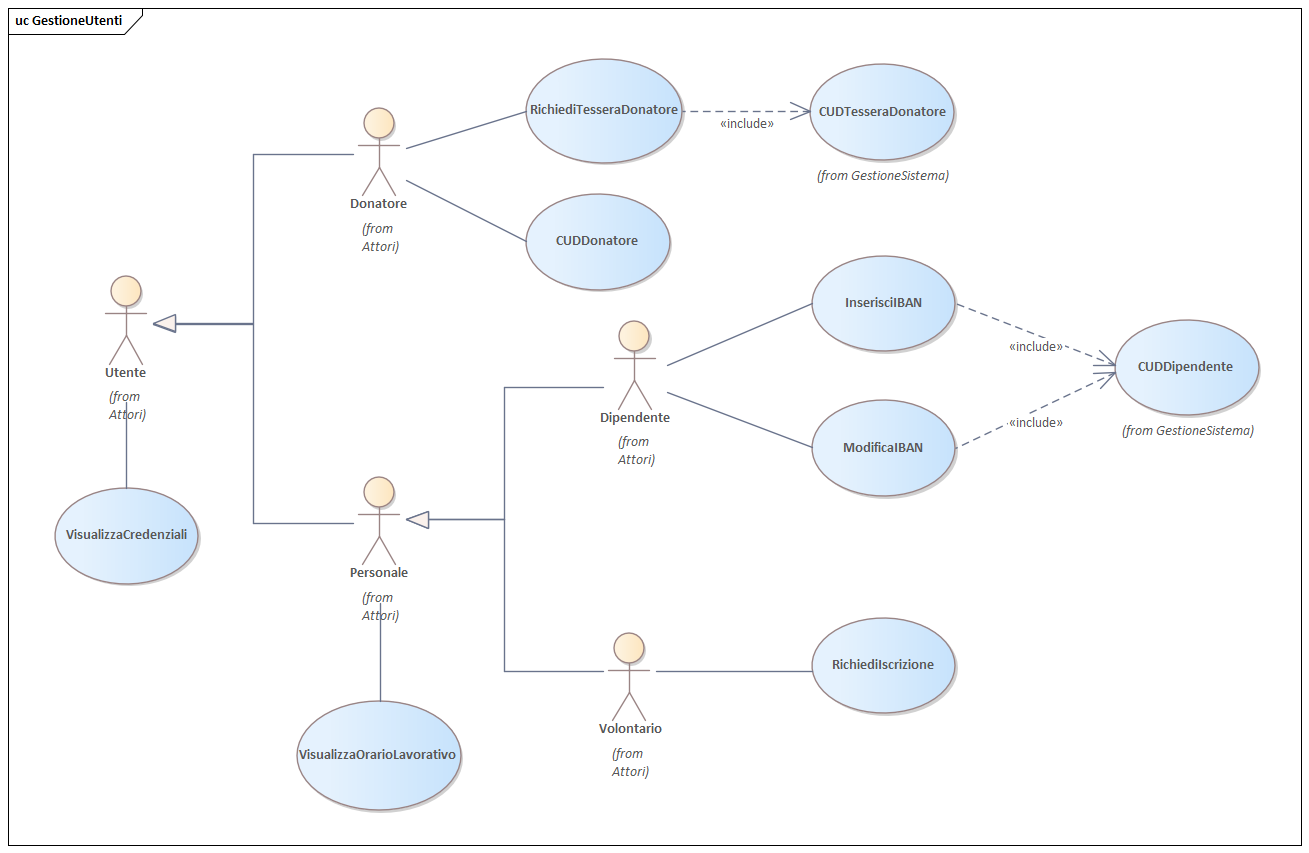
|  |
| --- |
| Caso d’Uso: RicercaRapportini |
| ID: 23 |
| Breve descrizione:  L’Amministratore ricerca le informazioni contenute nei rapportini |
| Attori primari:  Amministratore |
| Attori secondari:  Sistema |
| Precondizioni:  Deve esistere almeno un rapportino nel database |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’amministratore vuole ricercare uno specifico rapportino 2. L’Amministratore selezione data e ora del servizio di interesse    1. While (esiste un rapportino)       1. if  Esiste un rapportino con data e ora selezionate          1. Il sistema mostra all’Amministratore il rapportino. |
| Postcondizioni:  il sistema restituisce il rapportino ricercato |
| Sequenza degli eventi alternativi:  Nessuno |

|  |
| --- |
| Caso d’Uso: RichiestaRimborsoSpesa |
| ID: 24 |
| Breve descrizione:  Il Sistema richiede il rimborso spese dei servizi del mese precedente |
| Attori primari:  Tempo |
| Attori secondari:  Sistema |
| Precondizioni:  La data odierna è il primo giorno del mese successivo |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’amministratore effettua la richiesta del rimborso per il mese precedente 2. include(CalcolaSpese) 3. L’Amministratore richiede il rimborso a \*\*\* |
| Postcondizioni:  Il Sistema ha stabilito la cifra esatta di cui chiedere il rimborso |
| Sequenza degli eventi alternativi:  Nessuno |

|  |
| --- |
| Caso d’Uso: StampaBustaPaga |
| ID: 25 |
| Breve descrizione:  Questo caso d’uso permette al Sistema di restituire a Dipendente ed Amministratore la busta paga di uno specifico Dipendente |
| Attori primari:  Amministratore |
| Attori secondari:  Dipendente, Sistema |
| Precondizioni:  il Sistema riceve dall’INAIL le buste paga dei dipendenti per le attività del mese precedente |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’Amministratore deve inviare al Dipendente la busta paga 2. L’Amministratore inserisce il codice fiscale del Dipendente riportato nella busta paga    1. While (esiste un Dipendente)       1. if Il codice fiscale del Dipendente e quello riportato nella busta paga coincidono          1. Il Sistema invia al Dipendente la sua rispettiva busta paga |
| Postcondizioni:  Tutti i Dipendenti hanno ricevuto la loro busta paga |
| Sequenza degli eventi alternativa:  Nessuno |

|  |
| --- |
| Caso d’Uso: VisualizzaDipendente |
| ID: 26 |
| Breve descrizione:  Questo caso d’uso permette all’Amministratore di visualizzare dati anagrafici e turni lavorativi di un dipendente |
| Attori primari:  Amministratore |
| Attori secondari:  Sistema, Dipendente |
| Precondizioni:  Nessuno |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’amministratore deve visionare le informazioni dettagliate di uno specifico dipendente 2. L’Amministratore inserisce il codice fiscale    1. While (esiste un Dipendente)       1. if viene trovato un Dipendente con i dati inseriti          1. il Sistema visualizza dati anagrafici, turni lavorativi e contratto del Dipendente ricercato    2. else       1. viene visualizzato un messaggio di errore |
| Postcondizioni:  Il sistema restituisce le informazioni del dipendente ricercato |
| Sequenza degli eventi alternativi:  Nessuna |

**PACKAGE GESTIONEUTENTE**



|  |
| --- |
| Caso d’Uso: VisualizzaTesseraDonatore |
| ID: 27 |
| Breve descrizione:  Questo caso d’uso permette la visualizzazione della tessera di un donatore |
| Attori primari:  Donatore |
| Attori secondari:  Sistema |
| Precondizioni:  La tessera deve esistere |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Donatore vuole visionare la sua tessera 2. While (esiste una tessera)    1. if  esiste una tessera con il codice fiscale del Donatore interessato       1. il sistema mostra al Donatore la sua tessera. |
| Postcondizioni:  Nessuno |
| Sequenza degli eventi alternativi:   1. La sequenza degli eventi alternativi inizia se il Donatore non ha ancora una tessera personale 2. Attraverso il caso d’uso “CUDTesseraDonatore”, viene creata dall’Amministratore la nuova tessera |

|  |
| --- |
| Caso d’Uso: VisualizzaVolontario |
| ID: 28 |
| Breve descrizione:  Questo caso d’uso permette la visualizzazione dati anagrafici e turni assegnati ad un volontario |
| Attori primari:  Amministratore |
| Attori secondari:  Sistema, Volontario |
| Precondizioni:  Nessuno |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando l’amministratore deve visionare le informazioni dettagliate di uno specifico Volontario 2. 2. l’Amministratore inserisce il codice fiscale 3. if esiste Volontario    1. While (esiste un Volontario)       1. if viene trovato un Volontario con i dati inseriti          1. il Sistema visualizza dati anagrafici, turni lavorativi e contratto del Volontario ricercato       2. else          1. viene visualizzato un errore opportuno 4. else viene visualizzato un messaggio di errore opportuno |
| Postcondizioni:  Il sistema restituisce le informazioni del volontario ricercato |
| Sequenza degli eventi alternativi:  Nessuno |

|  |
| --- |
| Caso d’Uso: CUDDonatore |
| ID: 29 |
| Breve descrizione:  Questo caso d’uso permetta all’amministratore di creare, modificare o eliminare un Donatore |
| Attori primari:  Amministratore |
| Attori secondari:  Donatore |
| Precondizioni:  Nessuno |
| Sequenza degli eventi principali:   1. Il caso d’uso inizia quando l’amministratore deve inserire o aggiornare le informazioni 2. *If* l’amministratore vuole inserire un nuovo Donatore    1. L’amministratore inserisce i dati del nuovo Donatore;    2. *If* il donatore già esiste       1. il sistema restituisce un messaggio di errore opportuno    3. *else*       1. Il sistema memorizza i dati; 3. *else if* L’Amministratore vuole aggiornare i dati del donatore    1. L’Amministratore inserisce il codice fiscale del donatore;    2. *if* il donatore esiste       1. L’Amministratore scrive i nuovi dati da inserire;       2. il sistema aggiorna i dati;    3. *else*       1. il sistema restituisce un messaggio di errore opportuno 4. *else if* L’Amministratore vuole rimuovere un Donatore    1. L’Amministratore inserisce il codice fiscale del Donatore    2. *if* il Donatore esiste       1. Il sistema procede con la rimozione del Donatore       2. il sistema aggiorna la lista dei Donatori    3. *else*       1. il sistema restituisce un messaggio di errore opportuno |
| Postcondizioni:  L’Amministratore inserisce un nuovo donatore e se è già iscritto modifica le sue informazioni o lo elimina |
| Sequenza degli eventi alternativa:  Nessuno |

|  |
| --- |
| Caso d’Uso: InserisciIBAN |
| ID: 30 |
| Breve descrizione:  Questo caso d’uso permette al Dipendente di inserire l’IBAN |
| Attori primari:  Dipendente |
| Attori secondari:  Nessuno |
| Precondizioni:  Non è stato inserito precedentemente l’IBAN |
| Sequenza degli eventi principali:   1. Il caso d’uso inizia quando l’utente inserisce il proprio IBAN 2. Il Dipendente inserisce l’IBAN    1. If L’IBAN del dipendente esiste già       1. Viene mostrato un messaggio di errore opportuno    2. else       1. il sistema memorizza l’IBAN |
| Postcondizioni:  L’IBAN viene inserito dal Dipendente |
| Sequenza alternativa degli eventi:  Nessuno |

|  |
| --- |
| Caso d’Uso: ModificaIBAN |
| ID: 31 |
| Breve descrizione:  Questo caso d’uso permette la modifica del l’IBAN precedentemente inserito |
| Attori primari:  Dipendente |
| Attori secondari:  Nessuno |
| Precondizioni:  l’IBAN è stato inserito precedentemente |
| Sequenza degli eventi:   1. Il caso d’uso inizia con il Dipendente che vuole modificare il proprio IBAN che ha precedentemente inserito 2. If l’IBAN è già presente    1. Viene cancellato l’IBAN precedente    2. Viene inserito un nuovo IBAN 3. else    1. Si inserisce un nuovo IBAN    2. il sistema memorizza l’IBAN |
| Postcondizioni:  il Dipendente modifica il proprio IBAN |
| Sequenza alternativa:  Nessuno |

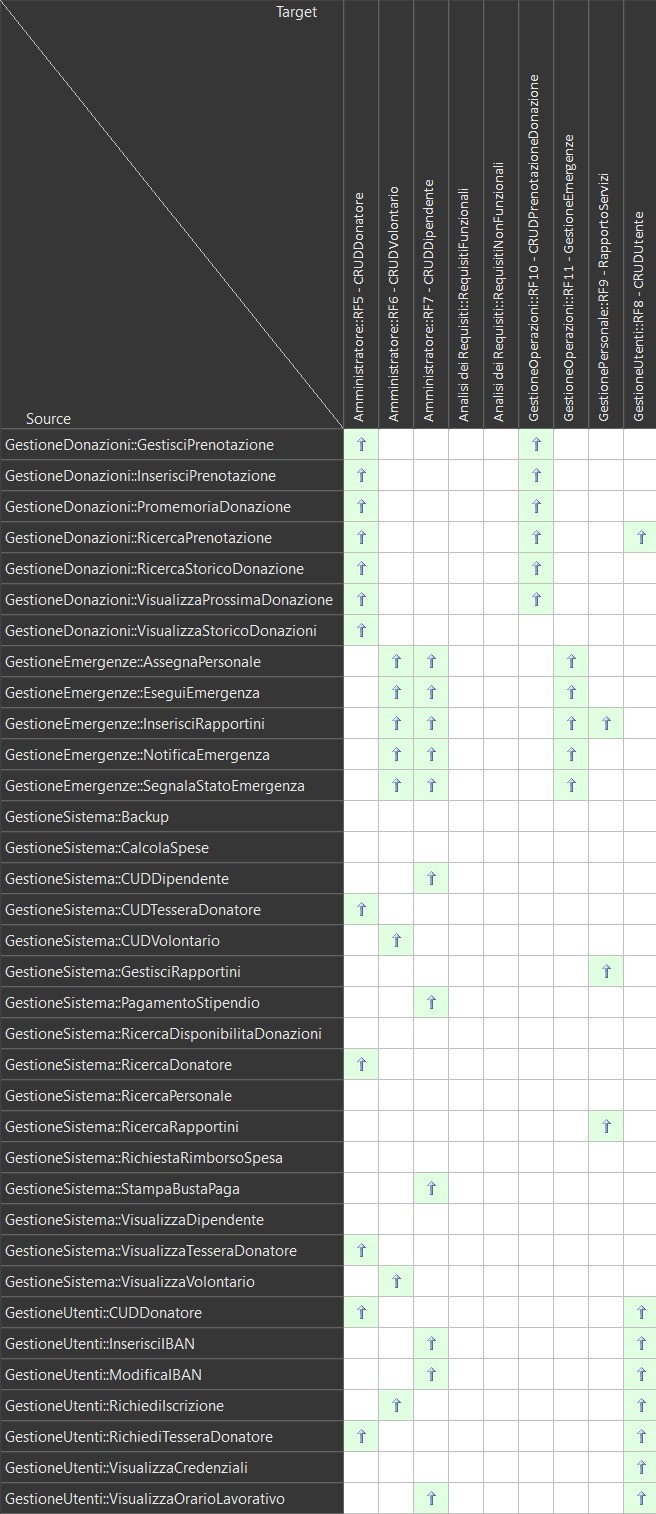
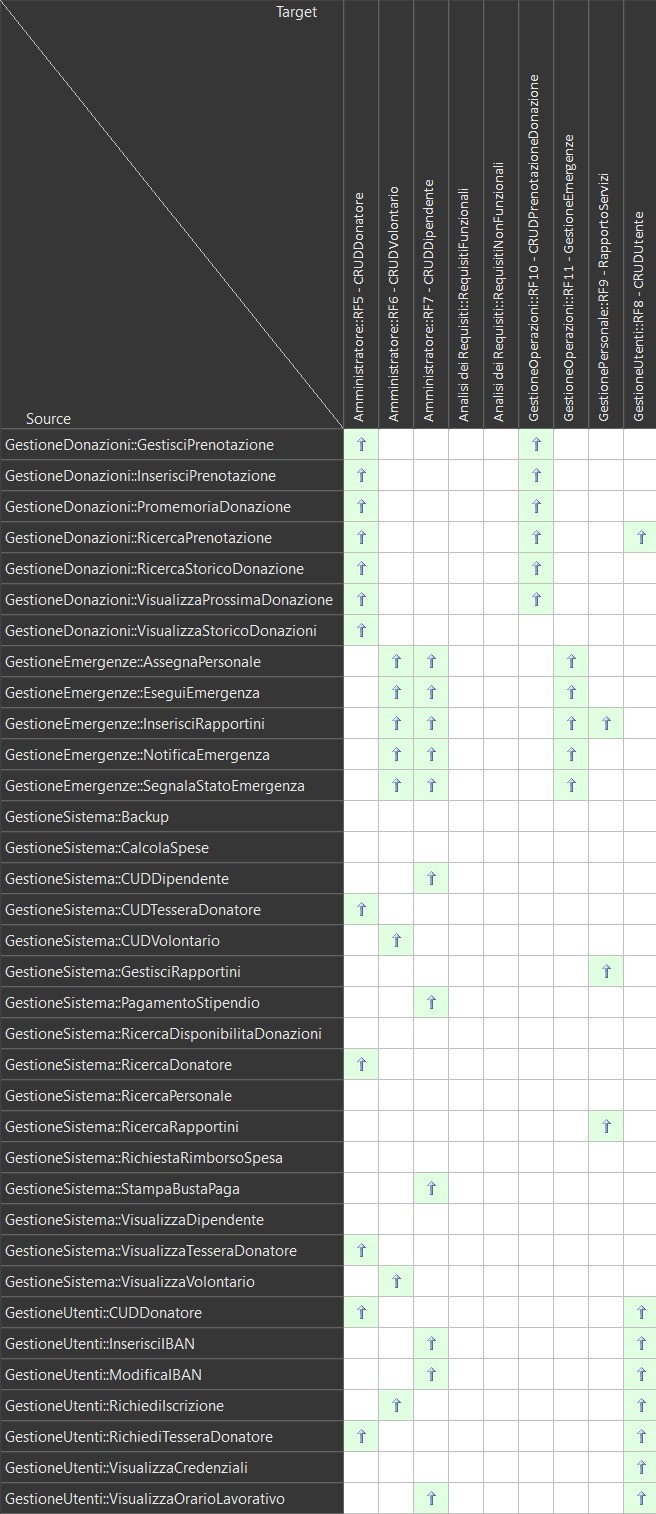
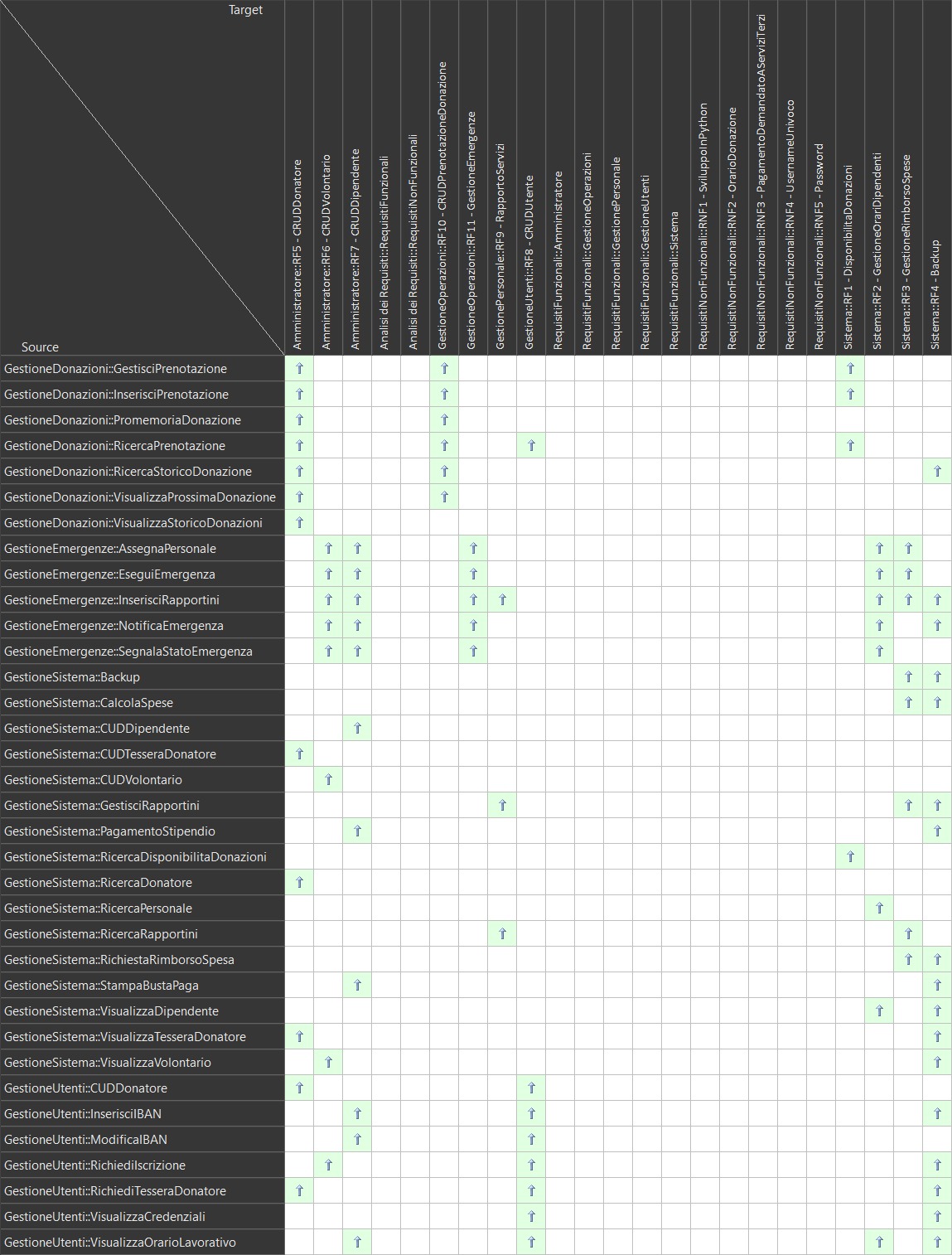
|  |
| --- |
| Caso d’Uso: RichiediIscrizione |
| ID: 32 |
| Breve descrizione:  Questo caso d’uso permette la richiesta di iscrizione del Donatore al sito dell’AVIS |
| Attori primari:  Amministratore |
| Attori secondari:  Donatore |
| Precondizioni:  Il Donatore non possiede l’iscrizione |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando il Donatore richiede l’iscrizione 2. While (esiste un Donatore)    1. if il Donatore è già iscritto       1. Viene visualizzato un messaggio di errore    2. else       1. L’Amministratore iscrive il Donatore |
| Postcondizioni:  Il Donatore è iscritto |
| Sequenza alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: RichiediTesseraDonatore |
| ID: 33 |
| Breve descrizione:  Queste caso d’uso permette al donatore di richiedere la tessera del donatore |
| Attori primari:  Amministratore |
| Attori secondari:  Donatore |
| Precondizioni:  Il Donatore non aveva la tessera |
| Sequenza degli eventi:   1. Il caso d’uso inizia quando il Donatore richiede la tessera 2. If il Donatore ha già la tessera    1. viene mostrato un messaggio di errore opportuno 3. else    1. l’Amministratore consegna la tessera al Donatore |
| Postcondizioni:  Il Donatore ha fatto la richiesta della tessera |
| Sequenza alternativa:  Nessuna |

|  |
| --- |
| Caso d’Uso: VisualizzaCredenziali |
| ID: 34 |
| Breve descrizione:  Questo caso d’uso permette la visualizzazione delle credenziali del donatore |
| Attori primari:  Sistema |
| Attori secondari:  Nessuno |
| Precondizioni:  Nessuno |
| Sequenza degli eventi principale:   1. Il caso d’uso inizia quando si vogliono visualizzare le credenziali del Donatore 2. ifDonatore non esiste    1. Viene mostrato un messaggio di errore 3. else    1. Vengono mostrate le credenziali del Donatore |
| Postcondizioni:  Vengono mostrate le credenziali del Donatore |
| Sequenza alternativa:  Nessuna |

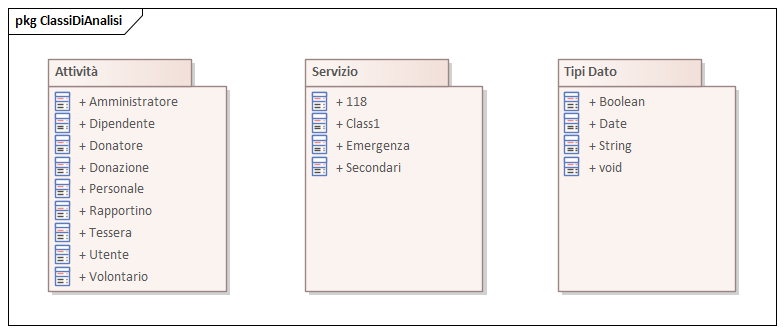
|  |
| --- |
| Caso d’Uso: VisualizzaOrarioLavorativo |
| ID: 35 |
| Breve descrizione:   Questo caso d’uso permette la visualizzazione dell'orario lavorativo |
| Attori primari:  Personale |
| Attori secondari:  Utente |
| Precondizioni:  Nessuno |
| Sequenza degli eventi:   1. Il caso d’uso inizia quando l’Utente vuole visualizzare gli orari di lavoro dei vari dipendenti 2. Vengono mostrati gli orari di lavoro dei vari dipendenti    1. if si cerca un Dipendente       1. if dipendente non esiste          1. Viene mostrato un messaggio di errore opportuno       2. else          1. Vengono mostrati gli orari di lavoro del Dipendente    2. else       1. Il caso d’uso termina |
| Postcondizioni:  Sono mostrati gli orari di lavoro |
| Sequenza alternativa:  Nessuno |

**MATRICE DI MAPPING DEI REQUISITI**

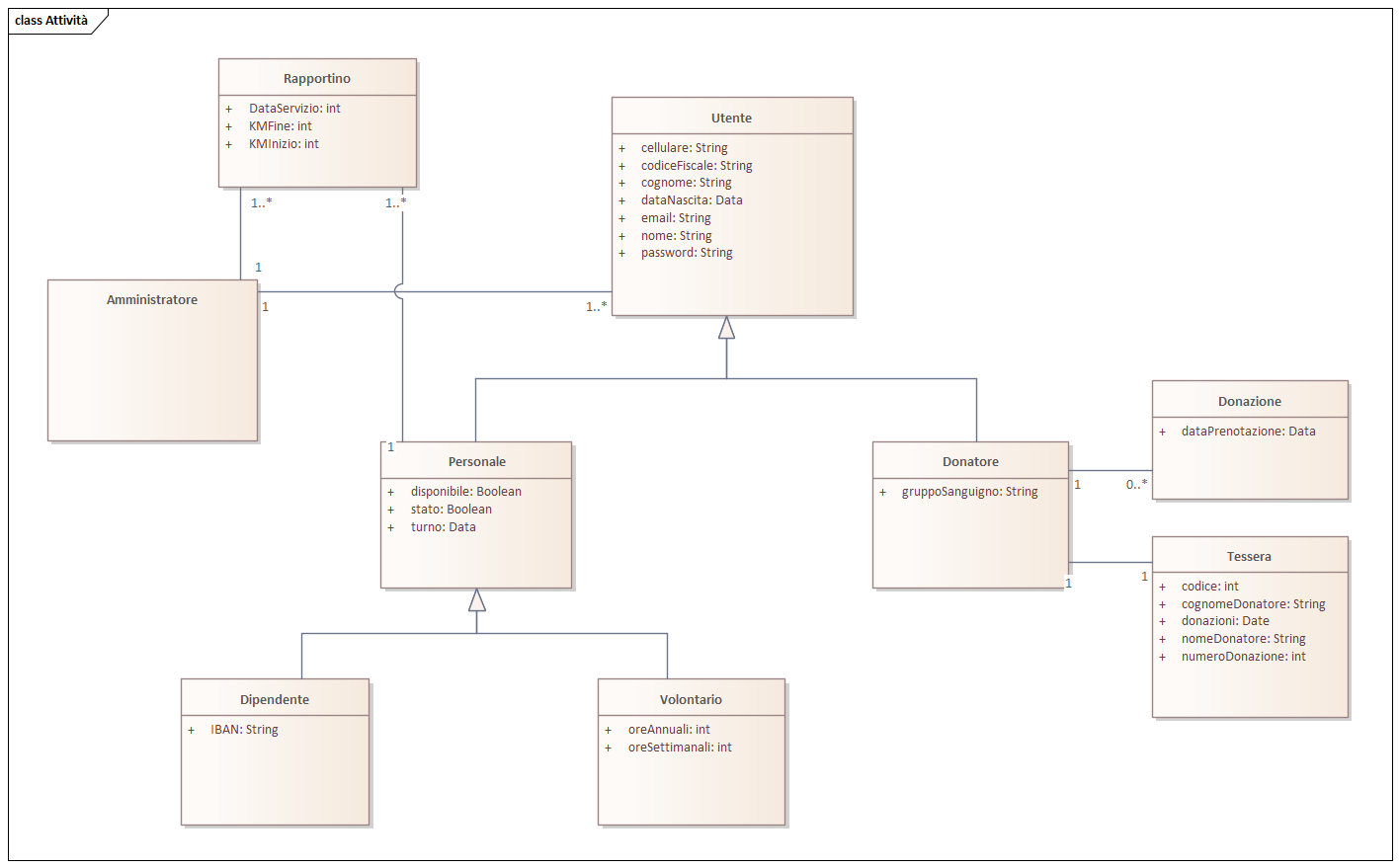


**DIAGRAMMI DI ANALISI**

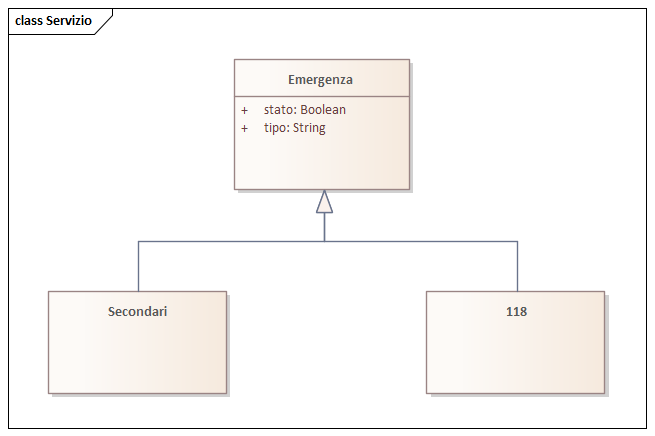
**DIAGRAMMI DELLE CLASSI DI ANALISI**

****

**PACKAGE ATTIVITÀ**

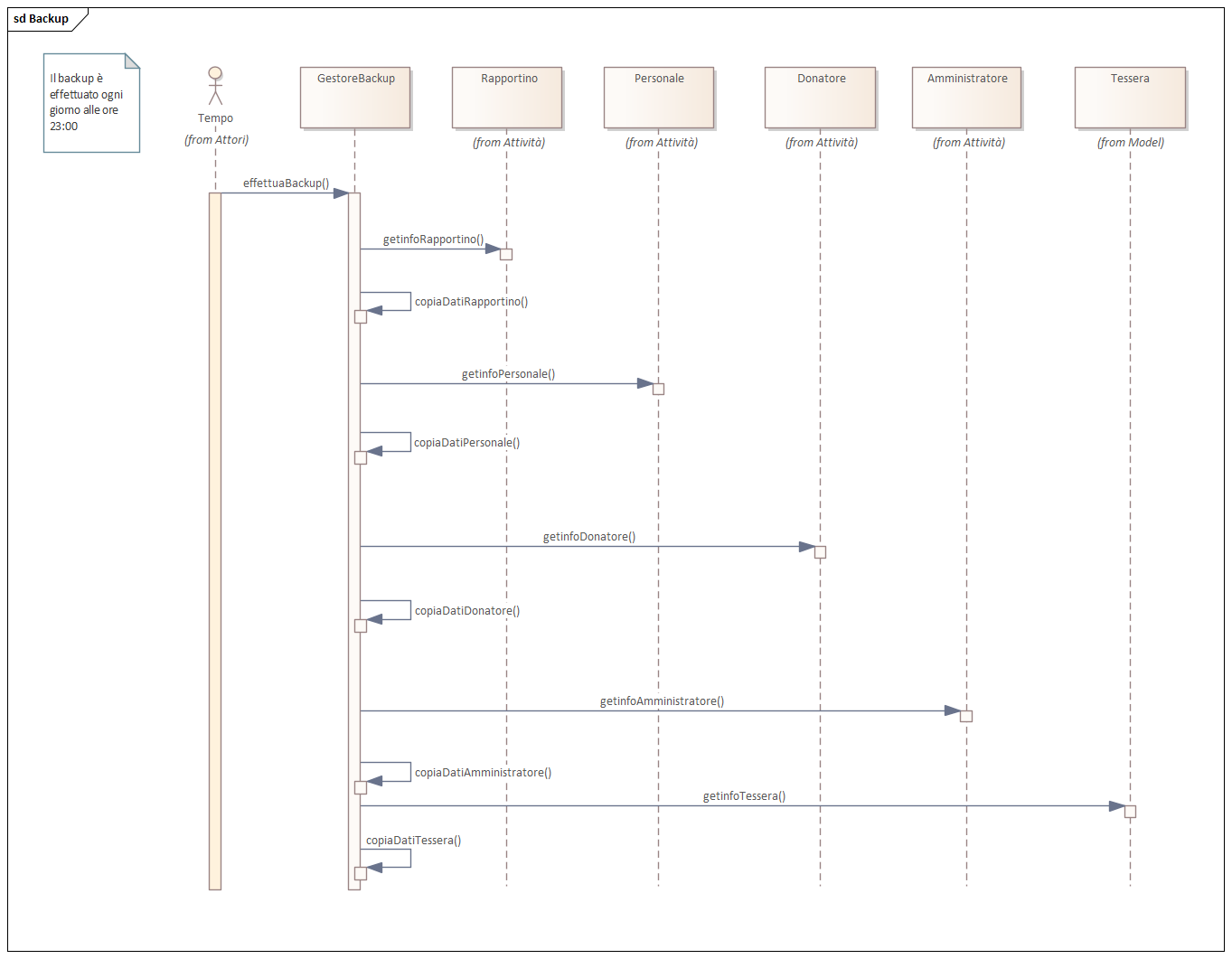
****

**PACKAGE SERVIZIO**

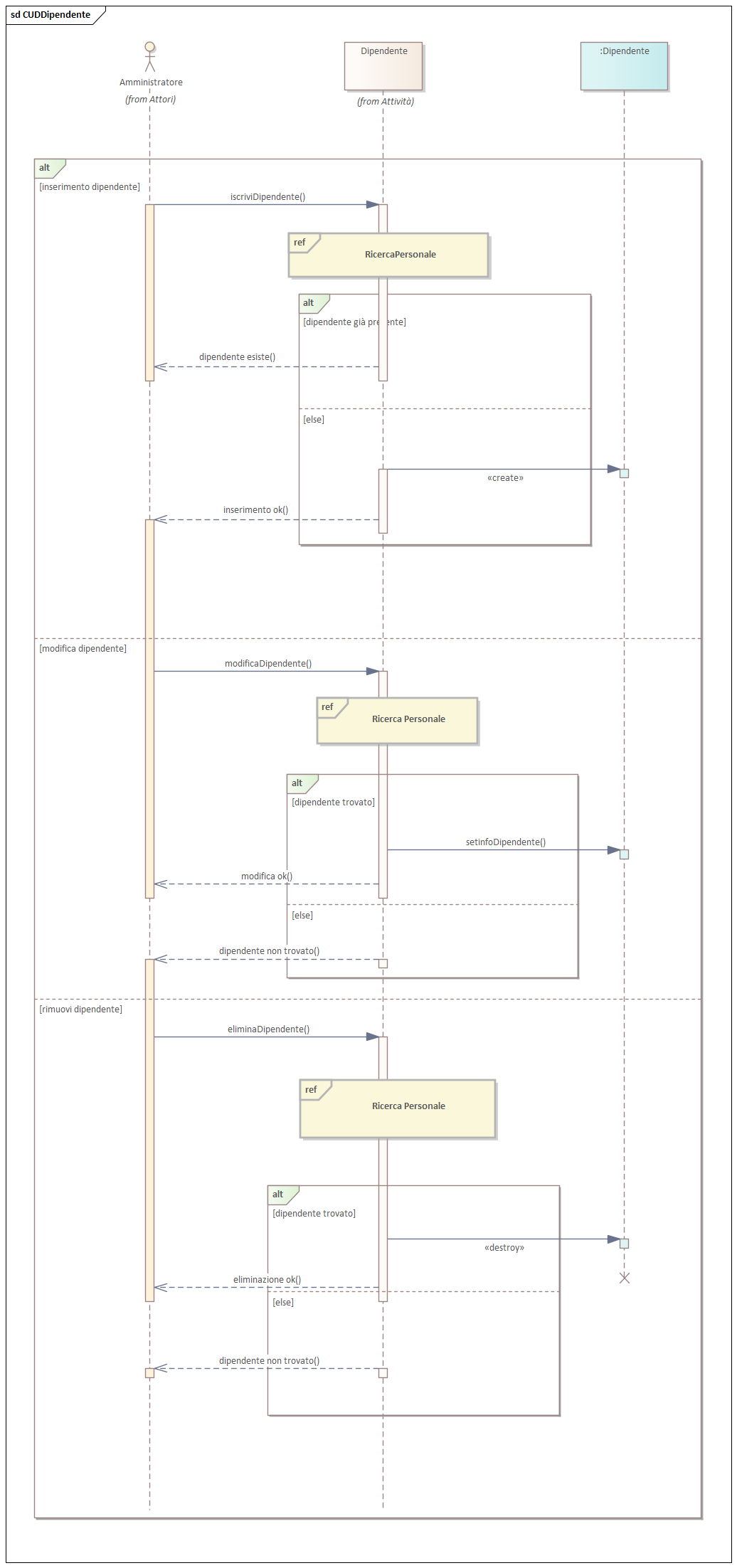
****

**DIAGRAMMI DI SEQUENZA**

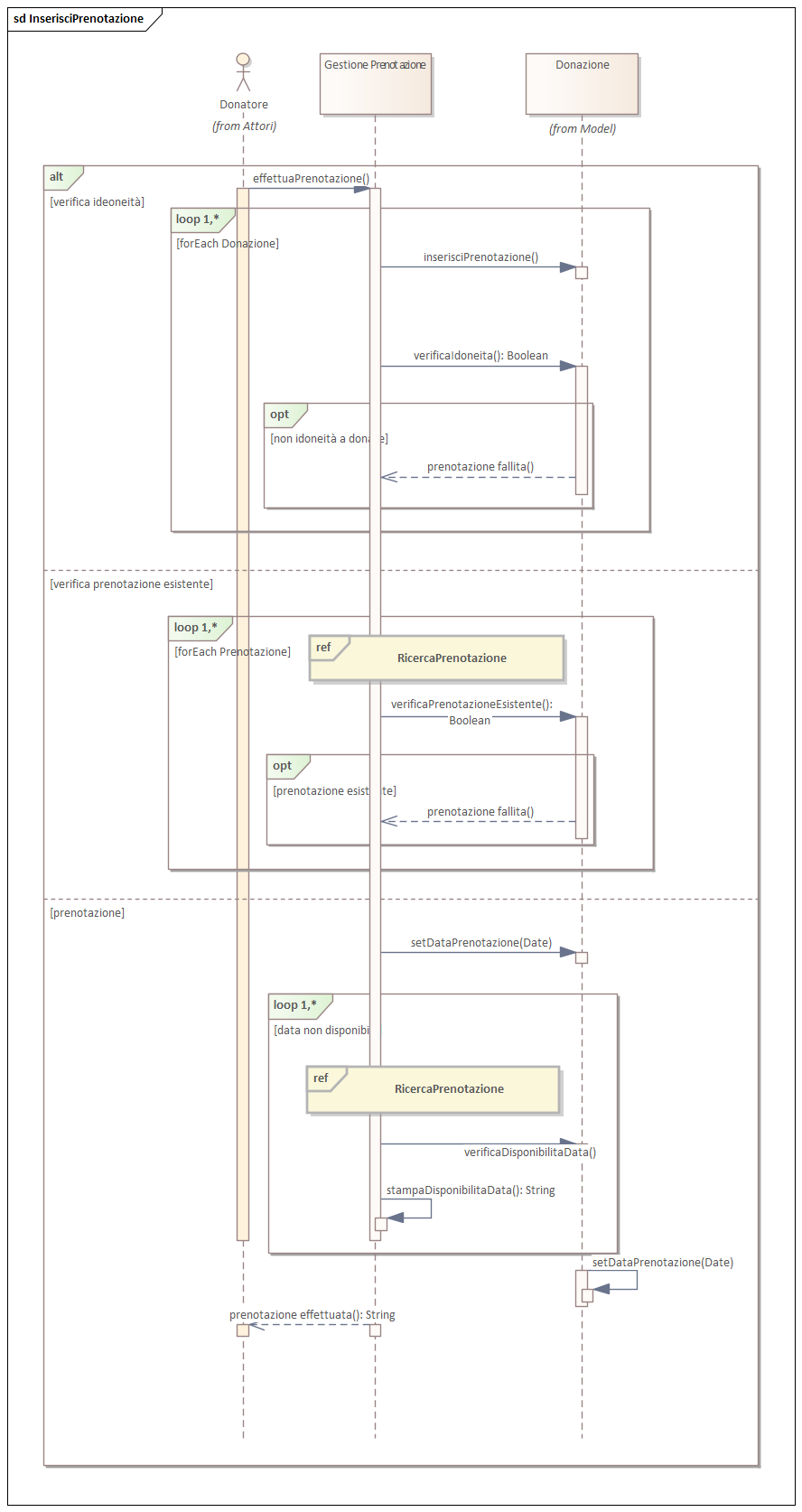
**BACKUP**

****

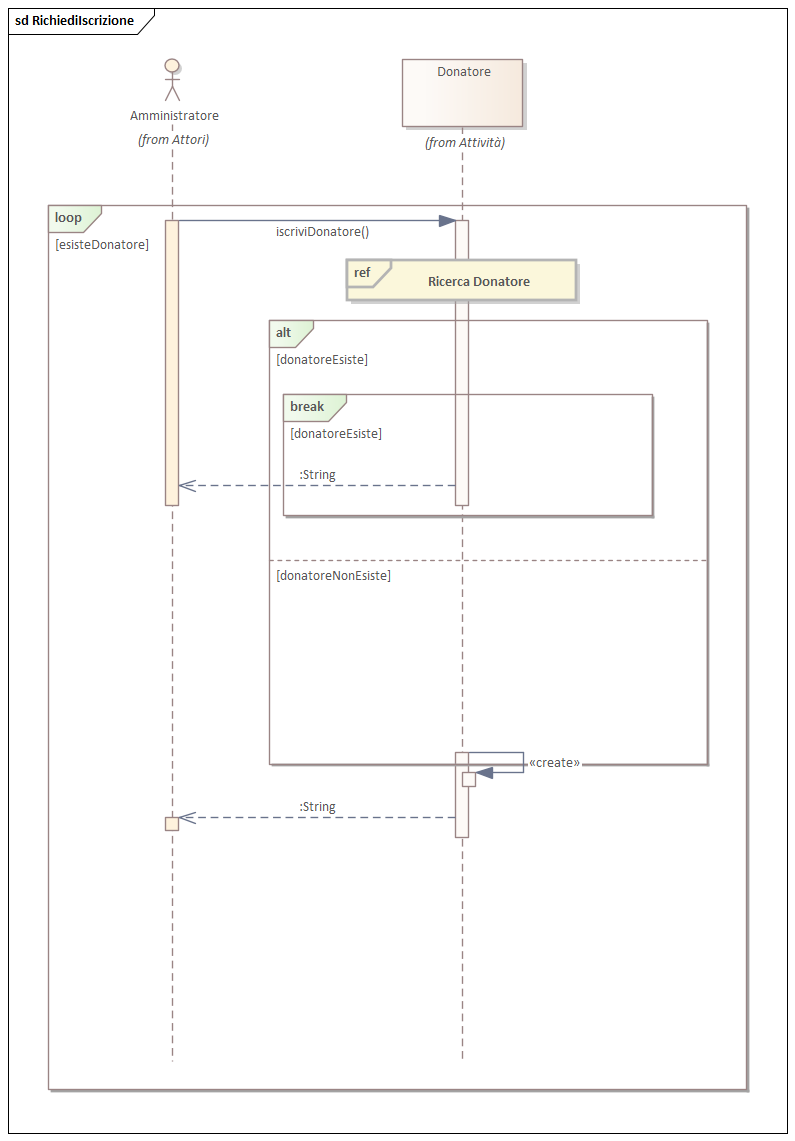
**CUDDIPENDENTE**

****

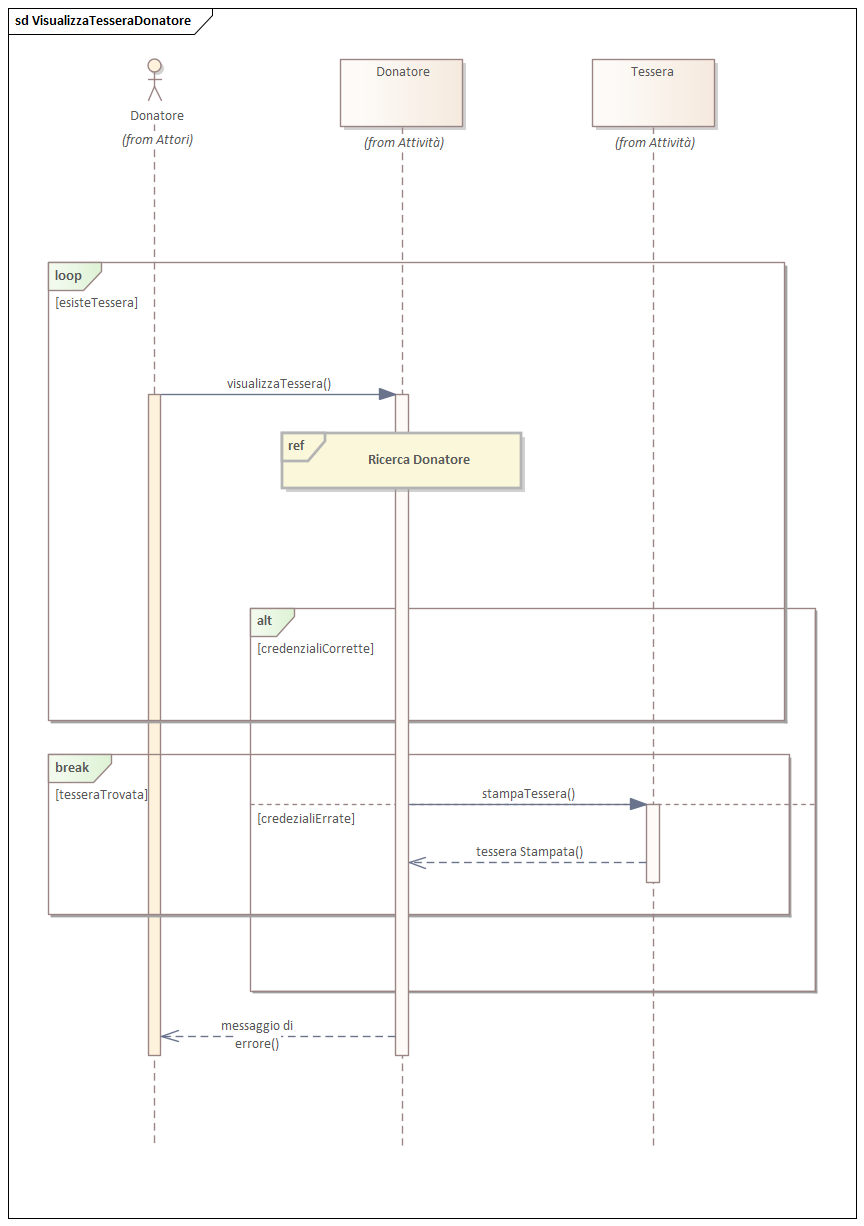
**INSERISCIPRENOTAZIONE**

****

**RICHIEDIISCRIZIONE**

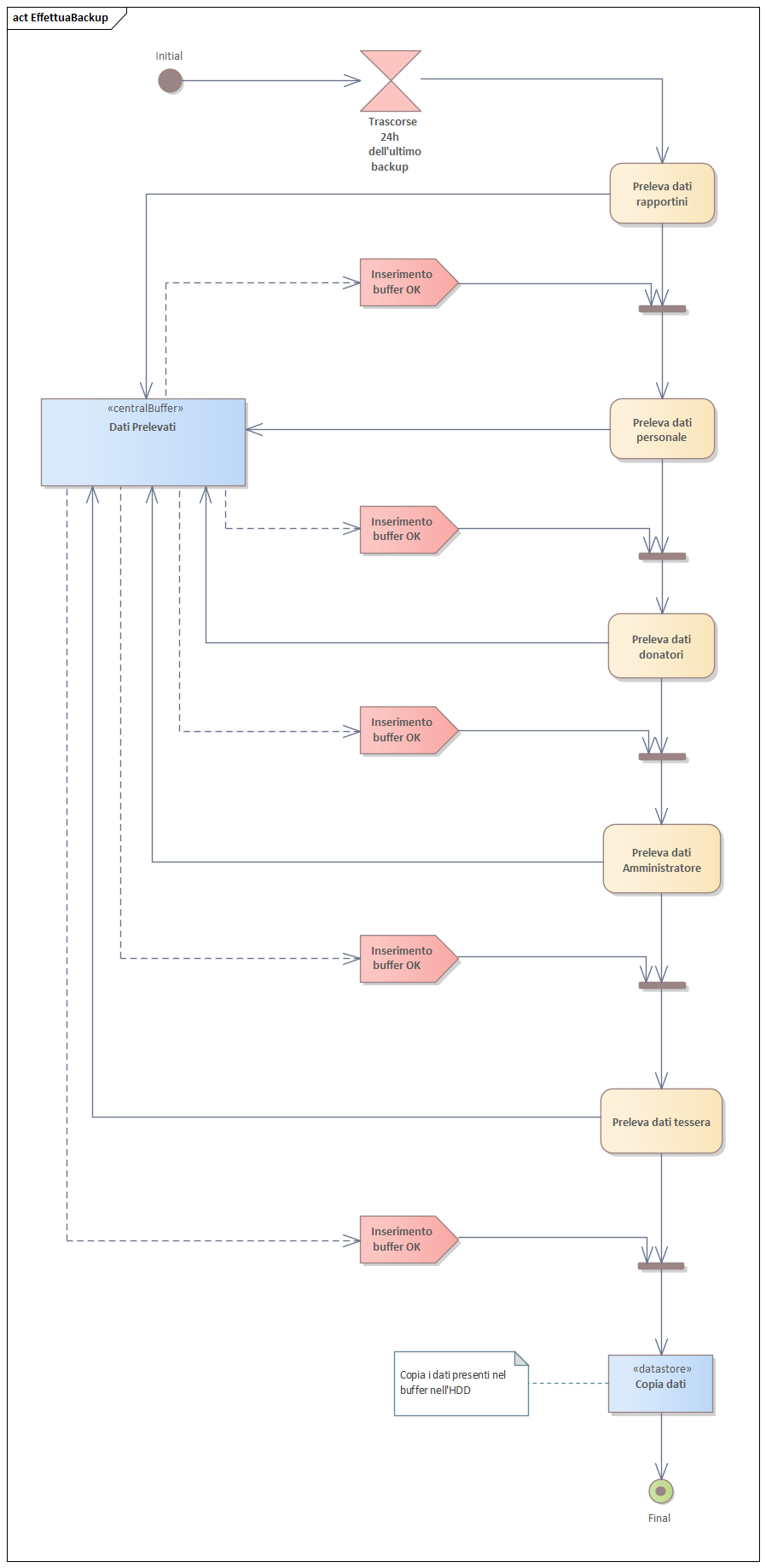
****

**VISUALIZZATESSERADONATORE**

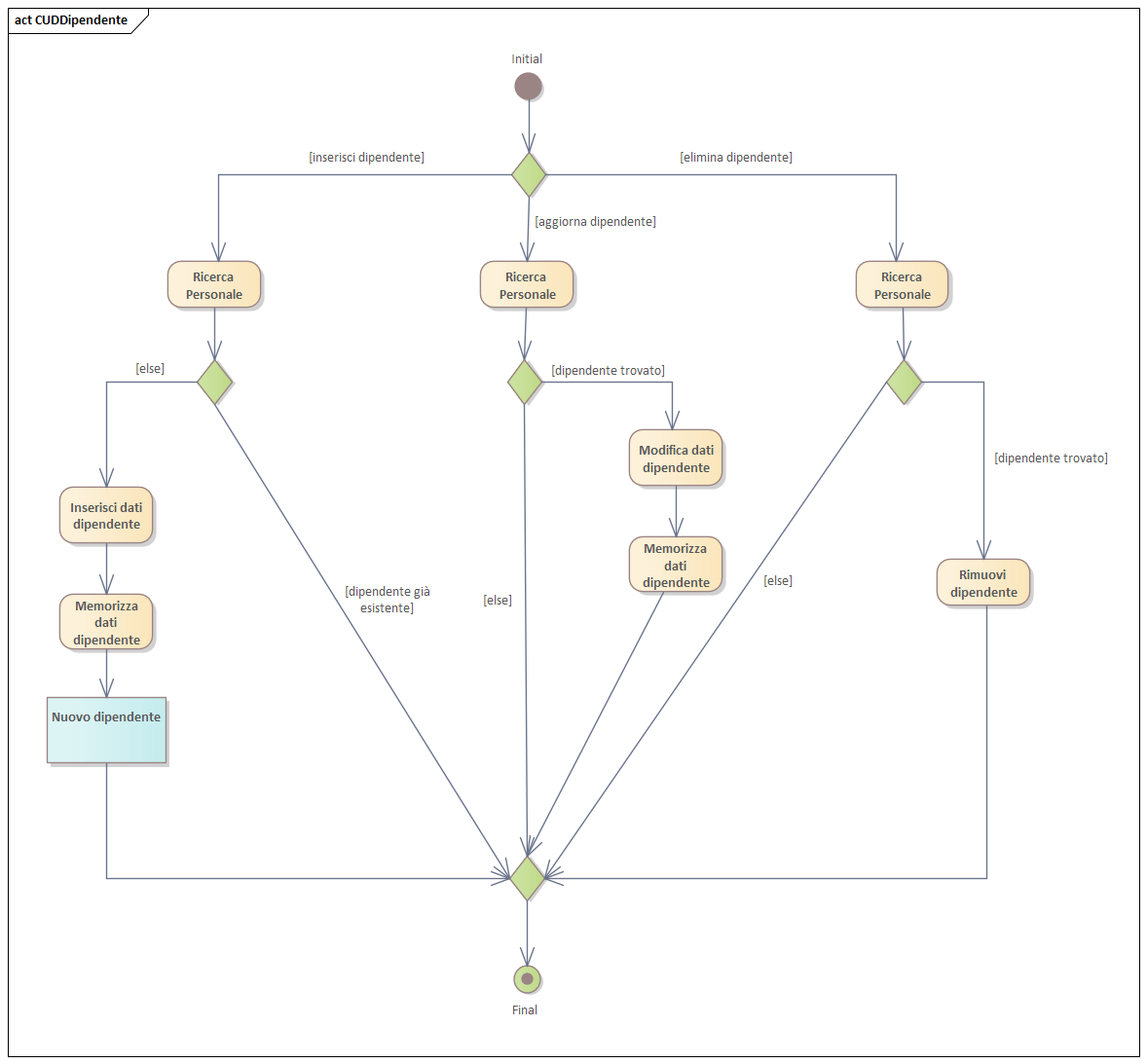
****

**DIAGRAMMI DI ATTIVITÀ**

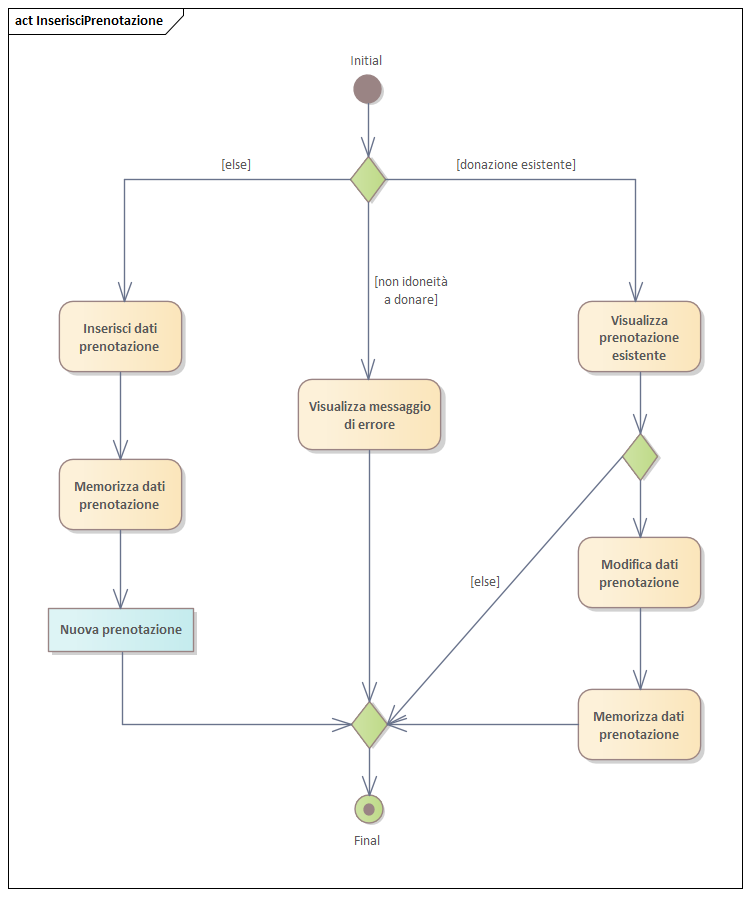
**EFFETTUABACKUP**

****

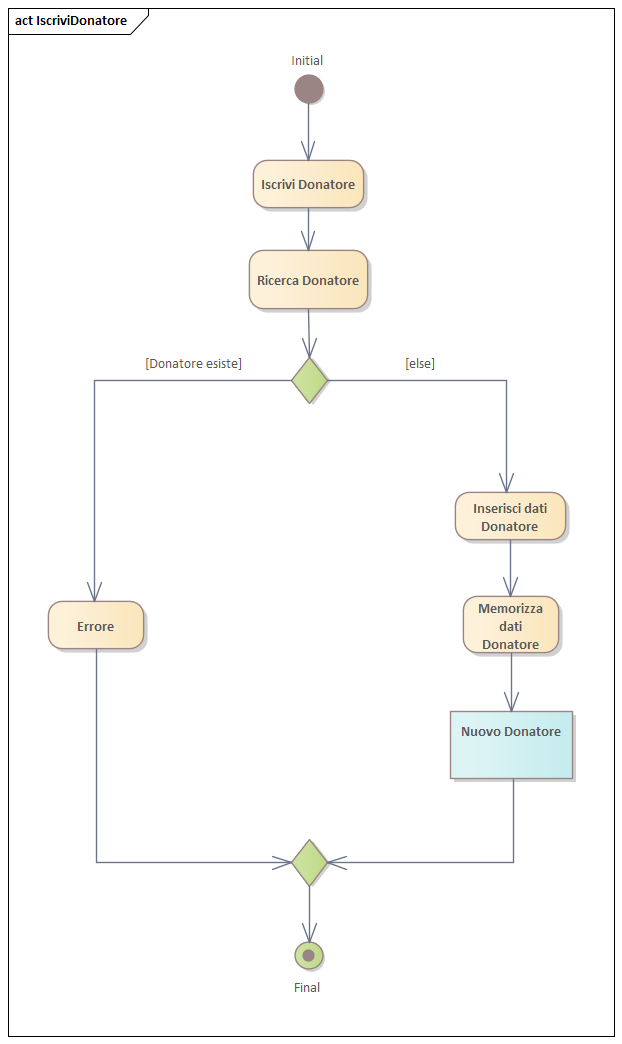
**CUDDIPENDENTE**



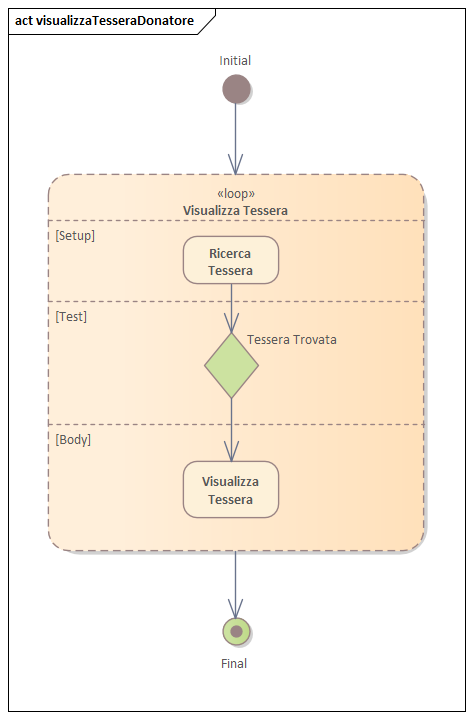
**INSERISCIPRENOTAZIONE**

****

**ISCRIVIDONATORE**

****

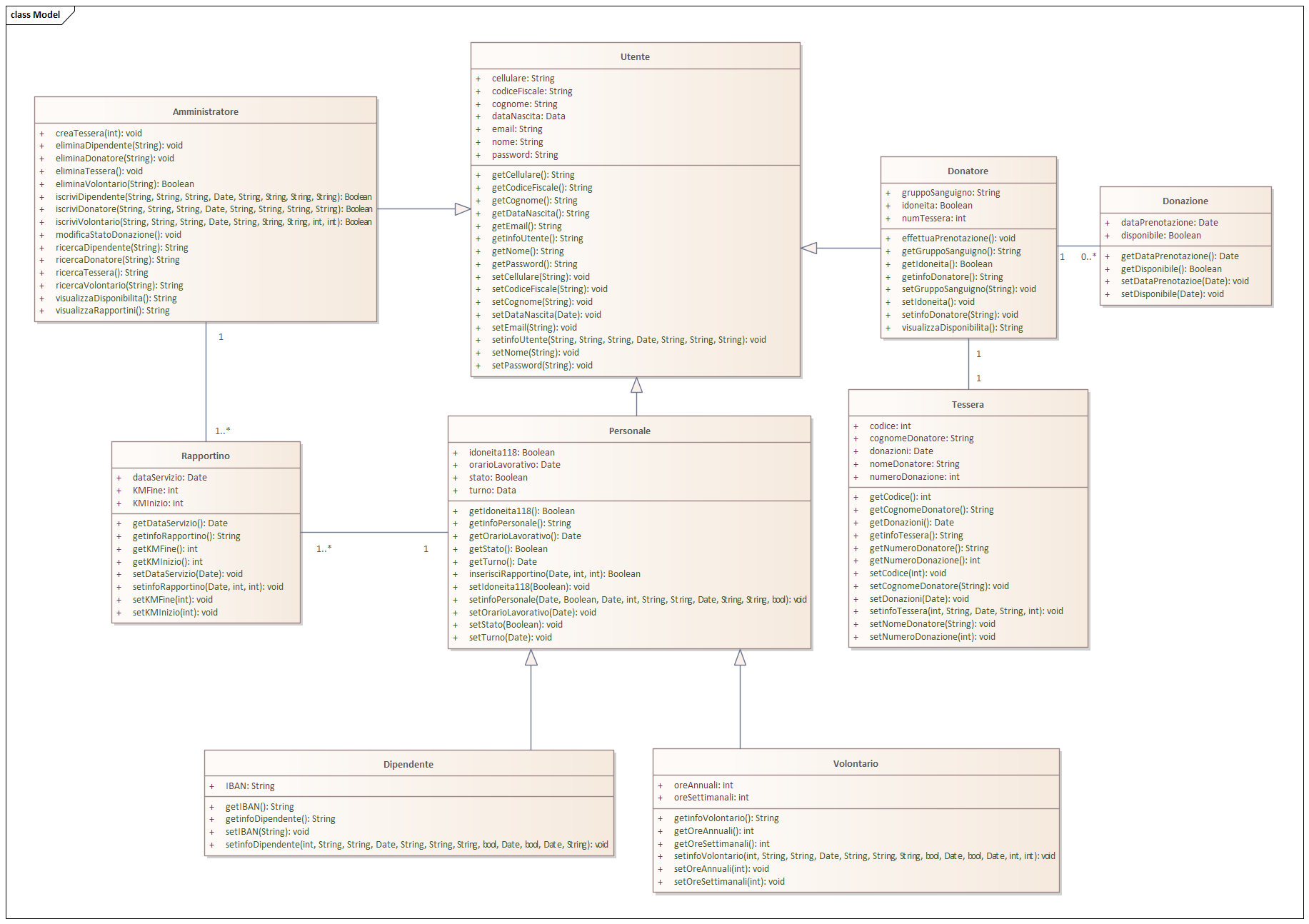
**VISUALIZZATESSERADONATORE**

****

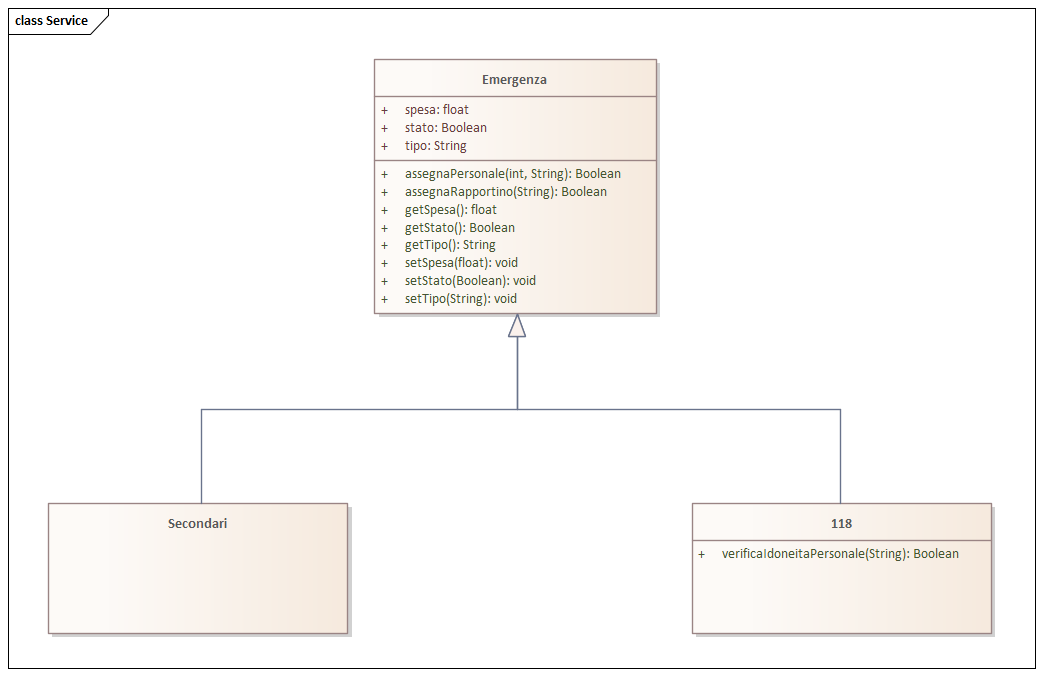
**DIAGRAMMI DI PROGETTAZIONE**

**DIAGRAMMI DELLE CLASSI DI PROGETTAZIONE**

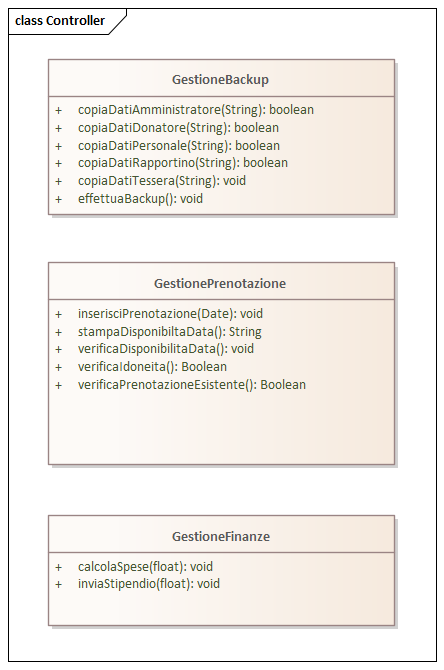
**PACKAGE MODEL**

****

**PACKAGE SERVICE**

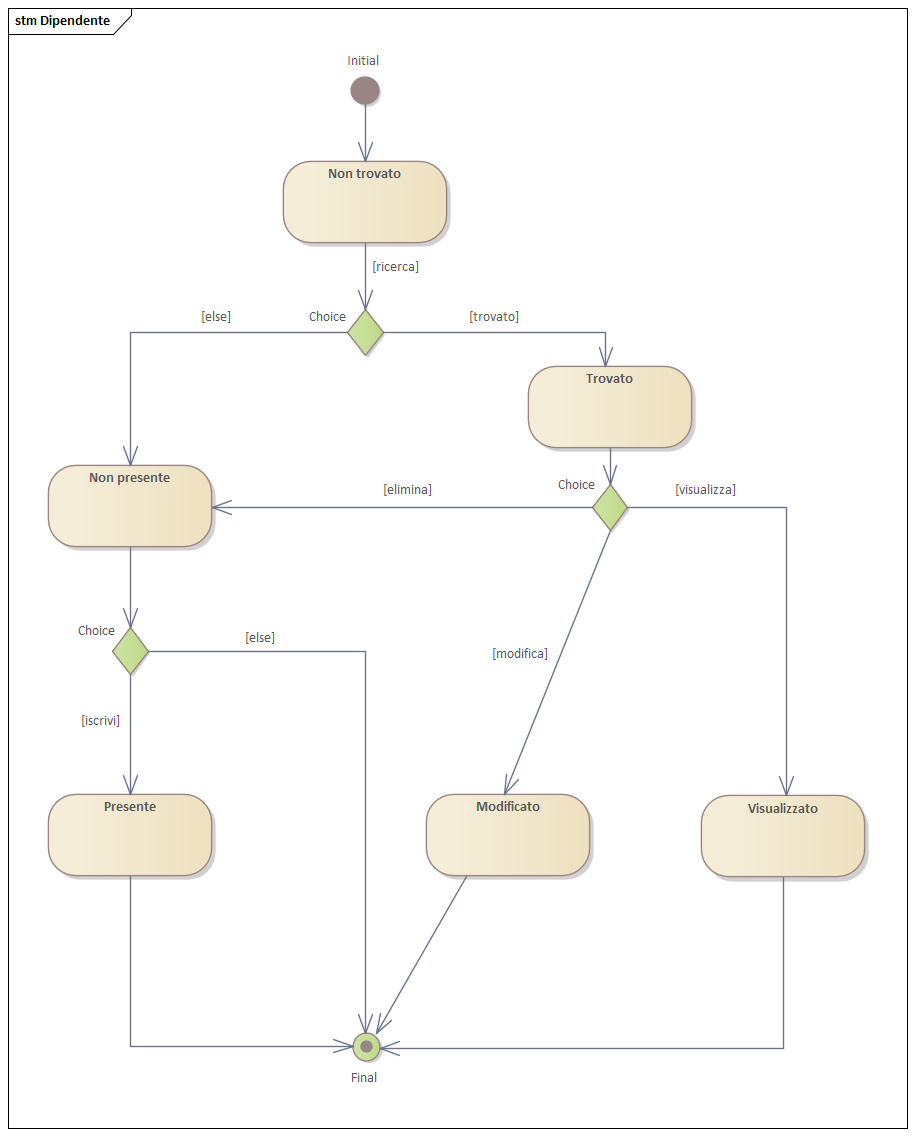
****

**PACKAGE CONTROLLER**

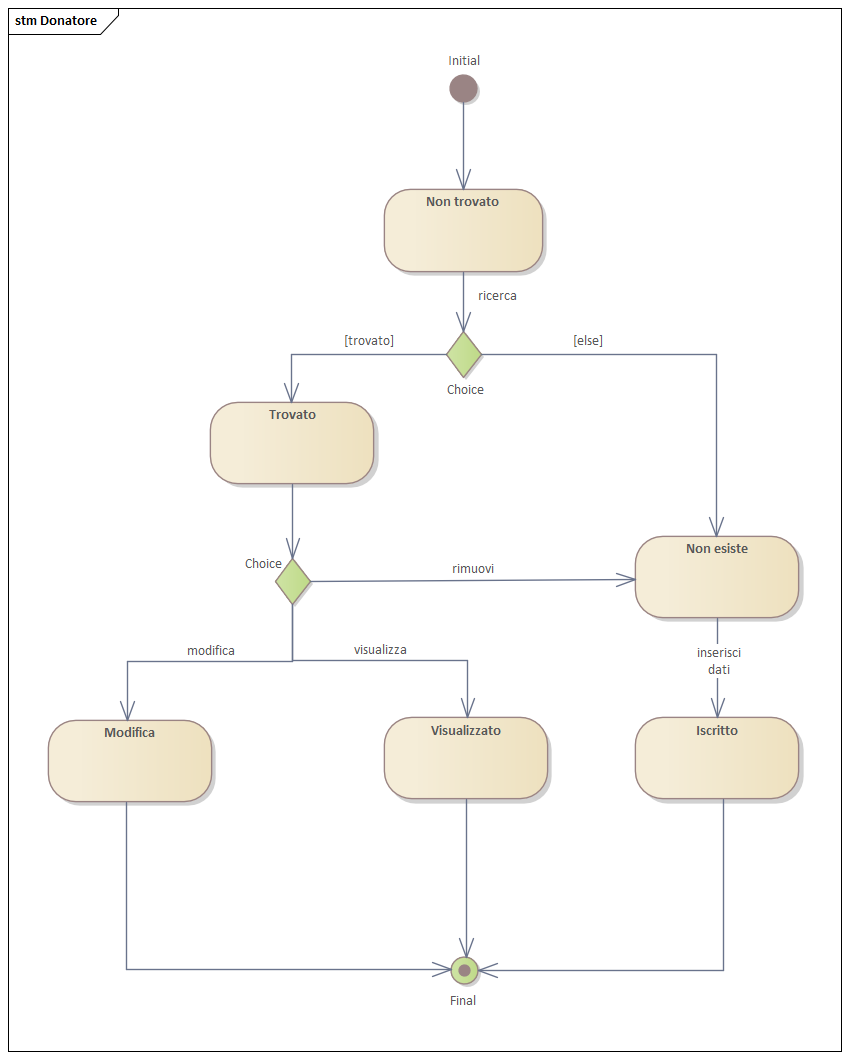
****

**DIAGRAMMI DELLE MACCHINE A STATI**

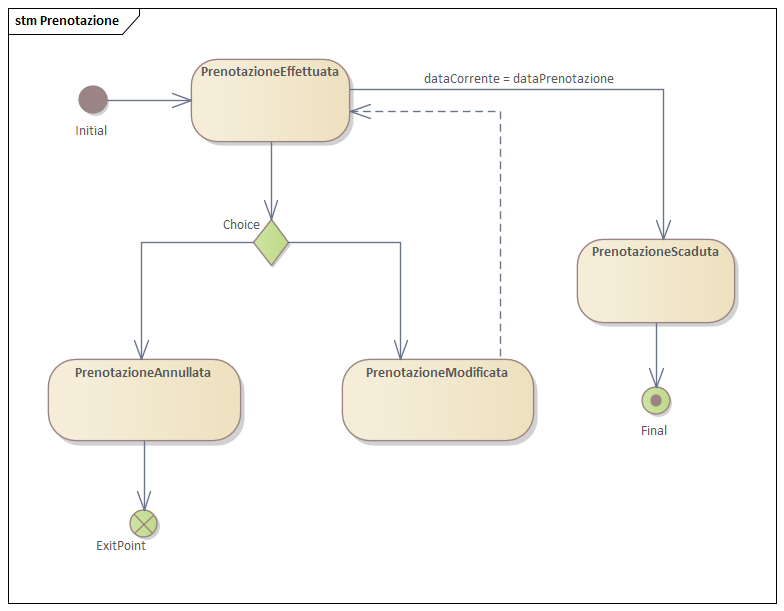
**DIPENDENTE**

****

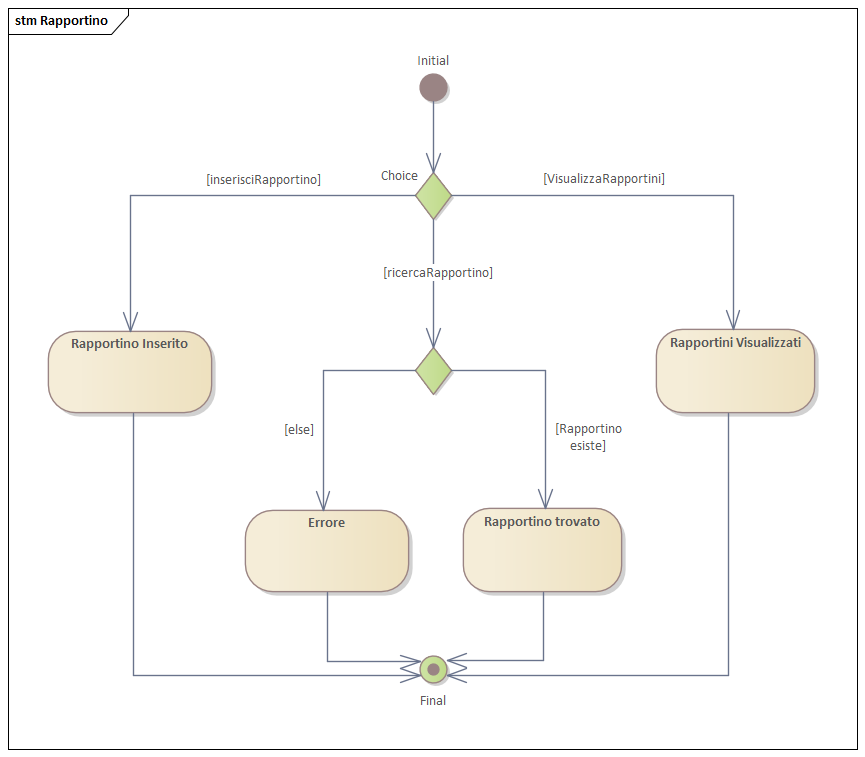
**DONATORE**

****

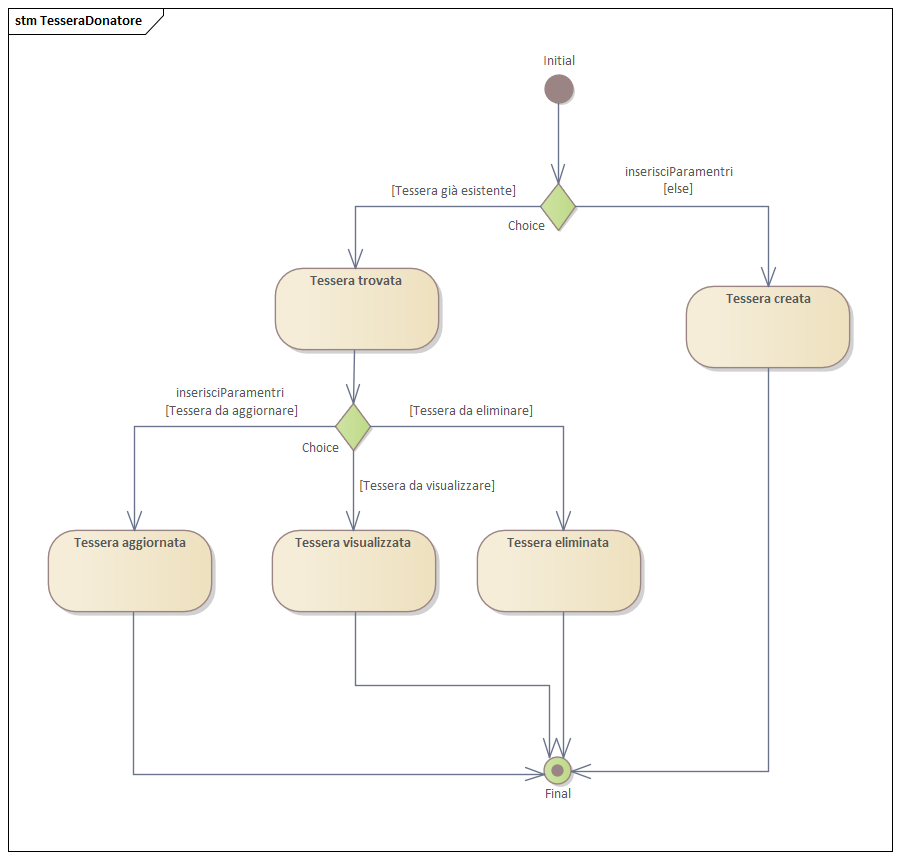
**PRENOTAZIONE**



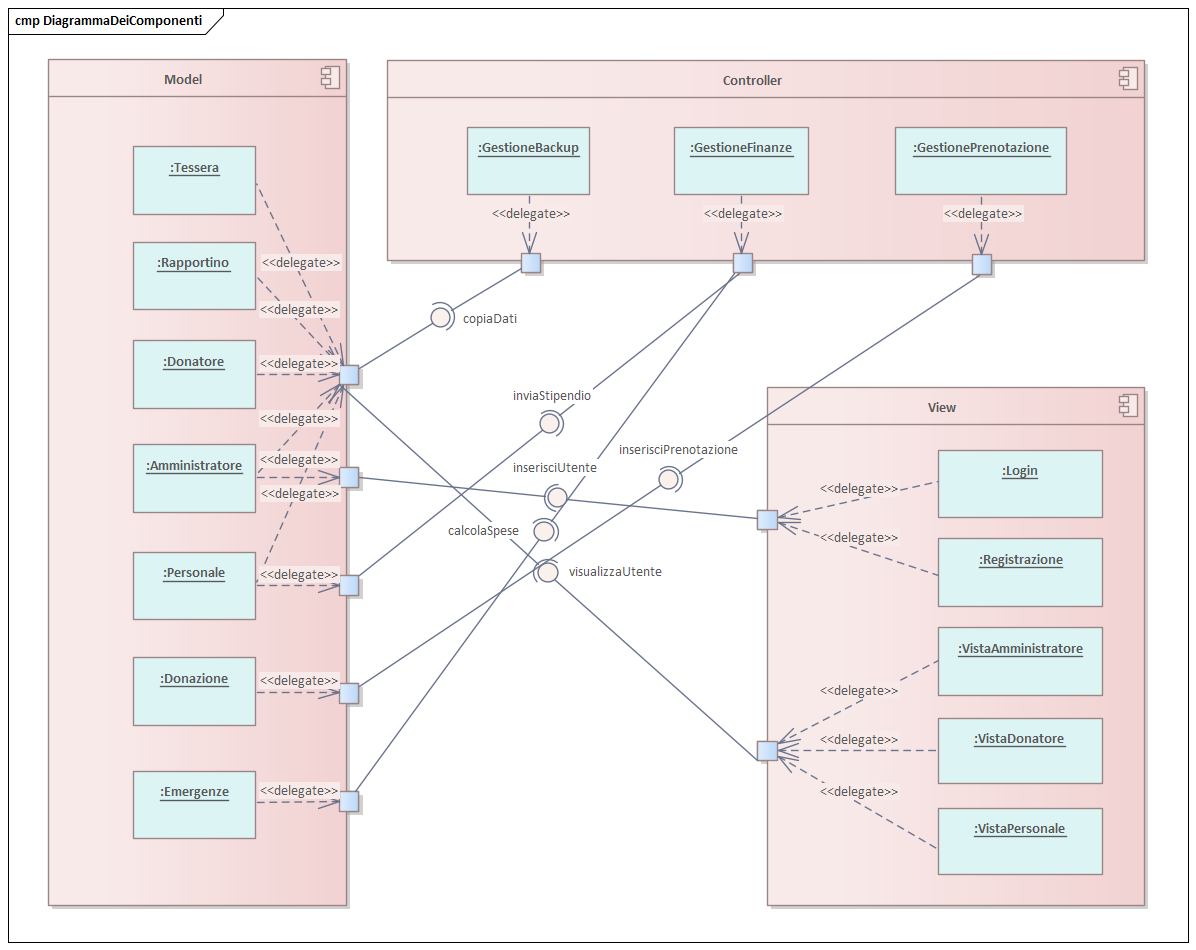
**RAPPORTINO**



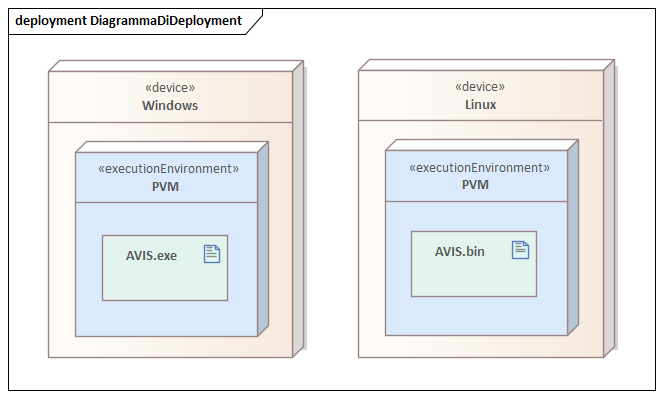
**TESSERADONATORE**



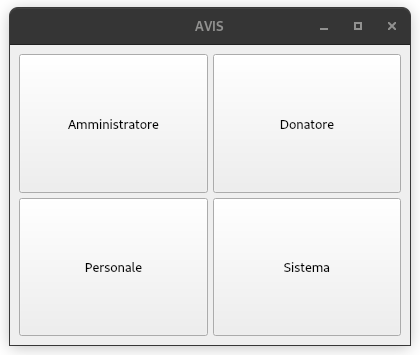
**DIAGRAMMA DEI COMPONENTI**

****

**DIAGRAMMA DI DEPLOYMENT**

****

**MOCKUP**

 Immagine che contiene testo

Descrizione generata automaticamente

Figura 1: **Vista: Home** Figura 2: **Vista: Login**

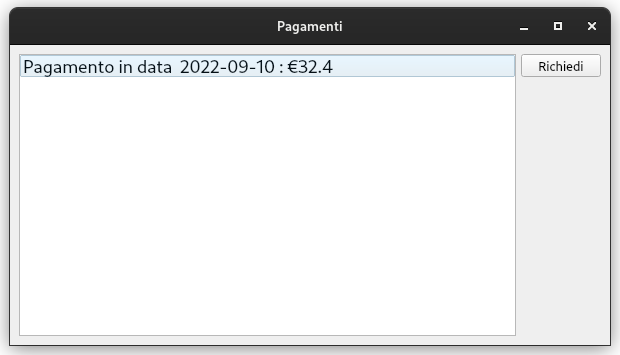


Figura 3: **Vista: Amministratore**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 4: **Vista: Amministratore > Gestione Turni**

 Immagine che contiene testo

Descrizione generata automaticamente

Figura 5-6: **Vista: Amministratore > Gestione Pagamenti**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 7: **Vista: Amministratore > Gestione Donazioni**

Immagine che contiene testo

Descrizione generata automaticamente Immagine che contiene testo

Descrizione generata automaticamente

Figura 8-9: **Vista: Amministratore > Gestione Rapportini**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 10: **Vista: Donatori**



Figura 11: **Vista: Donatore > Nuovo Donatore**

Immagine che contiene testo

Descrizione generata automaticamente Immagine che contiene testo

Descrizione generata automaticamente

Figura 12: **Vista: Donatore** Figura 13: **Vista: Donatore > Tessera**

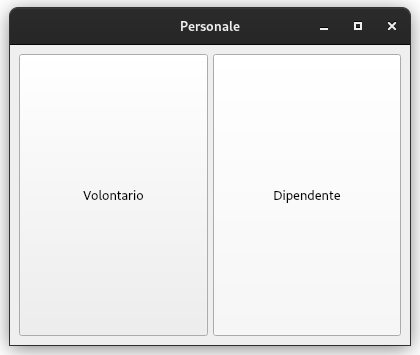


Figura 14: **Vista: Personale**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 15: **Vista: Personale > Volontari**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 16: **Vista: Volontario**

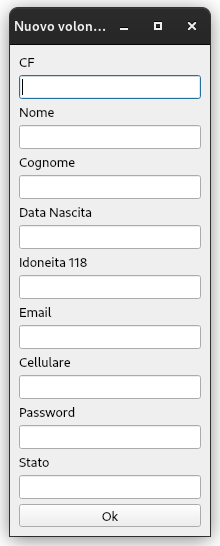


Figura 17: **Vista: Volontario > Nuovo Volontario**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 18: **Vista: Personale > Dipendenti**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 19: **Vista: Dipendente**



Figura 20: **Vista: Dipendente > Nuovo Dipendente**

Immagine che contiene testo

Descrizione generata automaticamente

Figura 21: **Vista:Nuovo Rapportino**