

Contratti delle Operazioni (UC1)

Contratti operazioni

Operazione: ottieniMetodiCifratura()

Riferimenti: UC:CreaSistemaCifratura

Precondizioni: nessuna

Postcondizioni: restituisce un elenco di oggetti Metodo (di cifratura)

Operazione: mostraSceltaChiave(metodo)

Riferimenti: UC:CreaSistemaCifratura

Precondizioni: *metodo* deve essere un Metodo (di cifratura) definito all'interno del sistema

Postcondizioni: restituisce/mostra/attua un elenco di Vincoli sulla tipologia di chiave che il *metodo* prevede di poter utilizzare.

Operazione: generaMappatura(chiave, metodo)

Riferimenti: UC:CreaSistemaCifratura

Precondizioni: *metodo* deve essere un Metodo (di cifratura) definito all'interno del sistema; *chiave* deve essere una Chiave che rispetta i Vincoli definiti dal *metodo*

Postcondizioni: crea un SistemaDiCifratura *sdc* con *metodo* e *chiave* e valorizza il suo attributo Mappatura applicando *metodo* a *chiave*.

Operazione: cifra(testo, sdc)

Riferimenti: UC:CreaSistemaCifratura

Precondizioni: il *testo* deve essere scritto sull'alfabeto per cui il sistema di cifratura *sdc* è stato creato (data l'assunzione iniziale che abbiamo solo un alfabeto, questo è naturalmente vero).

Postcondizioni: restituisce un nuovo testo che è il risultato dell'operazione di cifratura.

Operazione: inizializzaEditing(bozza)

Riferimenti: UC:InviaMessaggioCifrato

Precondizioni: Lo Studente *st* deve essere identificato; *bozza* è un Messaggio Mittente con *bozza*=y e che ha per mittente *st*, oppure è vuoto se l'utente sta creando un messaggio nuovo.

Postcondizioni: nessuna (dal punto di vista degli oggetti concettuali; possiamo supporre che avrà degli effetti sull'interfaccia)

Operazione: cifraMessaggio(messaggio)

Riferimenti: UC:InviaMessaggioCifrato

Precondizioni: Lo Studente *st* deve essere identificato; il *messaggio* deve essere un Messaggio Mittente con un testo non vuoto; inoltre i suoi mittente e destinatario devono essere specificati e il mittente deve coincidere con *st*.

Postcondizioni: il testo cifrato del *messaggio* viene creato applicando il sistema di cifratura condiviso da mittente e destinatario al testo di partenza.

Operazione: spedisce(messaggio)

Riferimenti: UC:InviaMessaggioCifrato

Precondizioni: Lo Studente *st* deve essere identificato; il testo cifrato del messaggio deve essere non vuoto; mittente e destinatario devono essere specificati e il mittente deve coincidere con *st*.

Postcondizioni: un Messaggio Ricevente *m*, uguale a *messaggio* se non per il fatto che il testo (in chiaro) viene rimosso, viene creato e messo nella casella di posta ricevuta del destinatario, con letto=n. Il *messaggio* di partenza viene inserito nella casella di posta inviata del mittente.

Operazione: salvaSistemaCifratura(sdc)

Riferimenti: UC:GestisciSistemiCifratura

Precondizioni: definiti sia chiave che metodo all'interno di *sdc*; Studente *st* identificato.

Postcondizioni: *st* viene impostato come creatore di *sdc*.

Operazione: elencaSistemiCifratura()

Riferimenti: UC:GestisciSistemiCifratura

Precondizioni: Studente identificato

Postcondizioni: nessuna

Operazione: visualizzaSistemaCifratura(sdc)

Riferimenti: UC:GestisciSistemiCifratura

Precondizioni: Studente *st* identificato, *sdc* è un Sistema di Cifratura esistente all'interno del sistema e *st* è il suo creatore.

Postcondizioni: nessuna

Operazione: eliminaSistemaCifratura(sdc)

Riferimenti: UC:GestisciSistemiCifratura

Precondizioni: Studente *st* identificato, *sdc* è un sistema di cifratura esistente all'interno del sistema e *st* è il suo creatore.

Postcondizioni: nessuna

Operazione: elencaDestinatari()

Riferimenti: UC:ProponiSistemaCifratura

Precondizioni: Studente identificato

Postcondizioni: nessuna

Operazione: proponiSistemaCifratura(sdc, partner)

Riferimenti: UC:ProponiSistemaCifratura

Precondizioni: metodo e chiave in *sdc* devono essere definiti; lo Studente proponente deve essere identificato.

Postcondizioni: viene creato un nuovo oggetto Proposta (proponente, partner e sdc come specificati nell'input dell'operazione) con stato = "pendente"

Operazione: vediNotificheAccettazioneProposte()

Riferimenti: UC:ProponiSistemaCifratura

Precondizioni: lo Studente è identificato.

Postcondizioni: le Proposte dell'elenco vengono segnate come notificate

Operazione: vediProposteSistemiCifrature()

Riferimenti: UC:Accetta/RifiutaSistemaCifratura

Precondizioni: lo *Studiante* è identificato

Postcondizioni: nessuna (a livello di modello concettuale, l'interfaccia mostrerà qualcosa...)

Operazione: `comunicaDecisione(decisione)` OOOOPS! qui bisognerebbe dire anche a cosa è relativa la decisione!!! → `comunicaDecisione(proposta, decisione)`

Riferimenti: UC:Accetta/RifiutaSistemaCifratura

Precondizioni: lo *Studiante* *st* è identificato; la *proposta* ha *st* come partner ed è in stato "pendente"

Postcondizioni: lo stato di *proposta* viene modificato per essere uguale a *decisione* e la *proposta* viene segnata come non notificata

Operazione: `elencaMessaggiBozza()`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: Lo *Studiante* è identificato

Postcondizioni: nessuna

Operazione: `salvaMessaggioBozza(messaggioModificato)`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: definito il titolo del *messaggioModificato*; *Studiante st* identificato e corrispondente al mittente .

Postcondizioni: *messaggioModificato* viene inserito nella casella bozze di *st*?

Operazione: `eliminaMessaggioBozza(messaggio)`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: *Studiante st* identificato, *messaggio* è un *Messaggio Mittente* della casella bozze di *st*

Postcondizioni: Il *messaggio* non è più nella casella bozze di *st*

Operazione: `apriMessaggioBozza(messaggio)`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: *Studiante st* identificato, *messaggio* è un *Messaggio* della casella bozze di *st*

Postcondizioni: nessuna

Operazione: `elencaMessaggiInviati()`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: Lo Studente è identificato

Postcondizioni: nessuna

Operazione: `eliminaMessaggioInviato(messaggio)`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: Studente *st* identificato, *messaggio* è un Messaggio Mittente della casella messaggi inviati di *st*

Postcondizioni: Il *messaggio* non è più nella casella messaggi inviati di *st*

Operazione: `visualizzaMessaggioInviato(messaggio)`

Riferimenti: UC:GestisciMessaggiMittente

Precondizioni: Studente *st* identificato, *messaggio* è un Messaggio Mittente della casella messaggi inviati di *st*

Postcondizioni: nessuna

Operazione: `elencaMessaggiRicevuti()`

Riferimenti: UC:GestisciMessaggiDestinatario

Precondizioni: Lo Studente è identificato

Postcondizioni: nessuna

Operazione: `eliminaMessaggioRicevuto(messaggio)`

Riferimenti: UC:GestisciMessaggiDestinatario

Precondizioni: Studente *st* identificato, *messaggio* è un Messaggio Destinatario della casella messaggi ricevuti di *st*

Postcondizioni: Il *messaggio* non è più nella casella messaggi ricevuti di *st*

Operazione: `apriMessaggioRicevuto(messaggio)`

Riferimenti: UC:GestisciMessaggiDestinatario

Precondizioni: Studente *st* identificato, *messaggio* è un Messaggio Destinatario della casella messaggi

ricevuti di *st*

Postcondizioni: lo stato del messaggio diventa pari a “letto”

Operazione: `decifraMessaggio(messaggio)` <-- Cambiato nome operazione per simmetria con “`cifraMessaggio`”

Riferimenti: UC: `RiceviMessaggioCifrato`

Precondizioni: Studente *st* identificato, *messaggio* è un Messaggio Destinatario della casella messaggi ricevuti di *st*

Postcondizioni: il testo del *messaggio* viene stabilito applicando a rovescio il sistema di cifratura condiviso da mittente e destinatario al testo cifrato. Lo stato del messaggio diventa pari a “letto”

Publicato da [Google Drive](#) – [Segnala una violazione](#) – Aggiornato automaticamente ogni 5 minuti
