Documentazione progetto Ingegneria del software

VR458509

VR460035

VR443470

settembre 2022

Indice

1 Ingegneria dei requisiti

I requisiti forniti al team di sviluppo, mediante testo, sono stati minuziosamente analizzati e successivamente raffinati tramite interviste chiuse (via e-mail) con il cliente. Il passo iniziale quindi, è stato quello di ottenere i diagrammi UML (*usecase and activity diagram*).

I passi successivi sono stati:

1. Stilare i requisiti come una lista di compiti (*task*) da eseguire;
2. Eseguire una stima del tempo richiesto per completare ciascun elemento;
3. Assegnare ciascun *task* ad ogni sviluppatore.

Le metodologie di sviluppo utilizzate verranno introdotte nei paragrafi successivi.

Sono stati creati vari prototipi con l'obbiettivo di essere presentati al cliente e di ottenere una convalida dei requisiti dedotti dal testo. Ogni prototipo ha dunque consentito di migliorare la specifica dei requisiti, migliorando conseguentemente il software in fase di sviluppo.

I vari prototipi sono stati sviluppati con lo stesso linguaggio di programmazione utilizzato per il software, ovvero Java insieme a JavaFX per il comparto grafico.

Nel momento in cui i requisiti sono stati completati sono stati riuniti tutti i prototipi, per poi perfezionarli e per creare il software vero e proprio.
Così facendo, sono state ottimizzate le fasi di ingegneria dei requisiti e sviluppo del software.
Inoltre, i vari diagrammi UML utilizzati in questa fase sono stati inseriti anche nella documentazione finale.

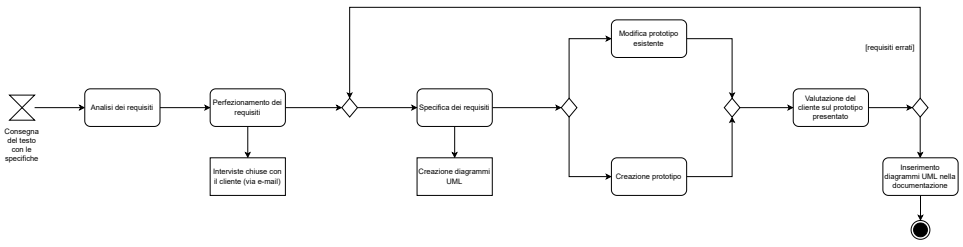


Figura 1: Activity diagram - Processo di ingegneria dei requisiti

1.1 Analisi dei requisiti

Si vuole progettare un sistema informatico di una agenzia che fornisce servizi di supporto alla ricerca di lavoro stagionale. I lavoratori interessati possono iscriversi al servizio, rivolgendosi agli sportelli dell'agenzia. Il sistema deve permettere la gestione delle anagrafiche e la ricerca di lavoratori stagionali, nei settori dell'agricoltura e del turismo. I responsabili, dipendenti dell'agenzia, devono inserire i dati dei lavoratori ed in particolare:

- → Dati anagrafici: nome, cognome, luogo e data di nascita, nazionalità;
- → Indirizzo;
- → [opzionale] Recapito telefonico personale;
- → E-mail;
- → [opzionale] Specializzazioni/esperienze precedenti (bagnino, barman, istruttore di nuoto, viticultore, floricultore);
- → Lingue parlate;
- → Tipo di patente di guida;
- → Se automunito;
- → Periodi e zone (comuni), per il quale il lavoratore è disponibile;
- → Almeno un contatto d'emergenza:
  1. Nome;
  2. Cognome;
  3. E-mail;
  4. Telefono (opzionale);
- → [opzionale] Lavori stagionali svolti negli ultimi 5 anni.

I dipendenti devono autenticarsi per poter accedere al sistema e inserire i dati dei lavoratori. Per ogni lavoro svolto vanno registrati:

- ➛ Periodo;
- ➛ Nome dell'azienda;
- ➛ Mansioni svolte;
- ➛ Luogo di lavoro;
- ➛ Retribuzione lorda giornaliera;

Invece, per i dipendenti dell'agenzia si memorizzano:

- ➟ Dati anagrafici;
- ➟ E-mail;
- ➟ Telefono;
- ➟ Credenziali di accesso (login e password);

Una volta registrate le informazioni sui lavoratori, il personale dell'agenzia può effettuare ricerche rispetto a possibili profili richiesti. In particolare, il sistema deve permettere ai dipendenti di effettuare ricerche per:

- ❒ Lavoratore;
- ❒ Lingue parlate;
- ❒ Periodo di disponibilità;
- ❒ Mansioni indicate;
- ❒ Luogo di residenza;
- ❒ Disponibilità di auto/patente di guida;

Il sistema deve permettere di effettuare ricerche complesse, attraverso la specifica di differenti condizioni di ricerca (sia in AND che in OR).

1.2 Perfezionamento dei requisiti

L'intervista con il cliente è stata eseguita tramite uno scambio di messaggi via e-mail.
Qui di seguito si lascia un elenco con domanda e risposta:

- ✍ Quando vengono definiti i dati da memorizzare per ogni lavoratore, viene menzionato luogo di nascita. Invece, quando vengono elencati i parametri di ricerca, viene richiesto il luogo di residenza. Se quest'ultimo non viene inserito, come è possibile ricavarlo?

  - ☛ L'indirizzo del lavoratore è ovviamente corrispondente al luogo di residenza.

- ✍ Il sistema non richiede esplicitamente un portale per la gestione dei responsabili, ovvero l'inserimento dei loro dati e le relative credenziali. Provvederà il team di sviluppo a implementare tale funzione in modo dettagliato o sarà lasciato come opzione ad ogni dipendente?

  - ☛ Un responsabile **non** deve poter avere accesso ad una eventuale gestione dei dipendenti. Tuttavia, la decisione su come gestire, inserire, modificare e/o eliminare un responsabile, non è ben chiara al cliente.
    Quindi, si lascia carta bianca al team di sviluppo.

- ✍ Le eventuali esperienze/specializzazioni richieste sono indicate tra parentesi. L'agenzia è interessate solo a queste o anche ad altre?

  - ☛ Le specializzazioni, o esperienze passate, sono quelle indicate tra parentesi. Il target a cui è rivolto questo software non ha interesse ad essere a conoscenza di altro.

- ✍ La gestione delle anagrafiche deve consentire la modifica di **tutti** i dati inseriti o solo alcuni?

  - ☛ Tutti i dati inseriti. È possibile che nel momento dell'inserimento un responsabile faccia un errore di battitura, oppure che il lavoratore cambi eventuali dati (nome, indirizzo, telefono, ecc.). Il sistema deve quindi consentire di avere il pieno controllo.

1.3 Specifica dei requisiti

1.3.1 Casi d'uso generali

Si presenta un unico diagramma raffigurante i casi d'uso esistenti. Il sistema si pone l'obbiettivo di supportare l'azienda fornendo servizi per la ricerca di lavoro stagionale.
Il sistema ha come protagonista due attori: il "responsabile" e "admin". Il primo, come dice la parola stessa, è il responsabile che ha il compito di registrare eventuali lavoratori, di effettuare modifiche o ricerche. Il secondo, invece, è l'amministratore (*admin*) che gestisce i vari responsabili modificando, eliminando o inserendone di nuovi.

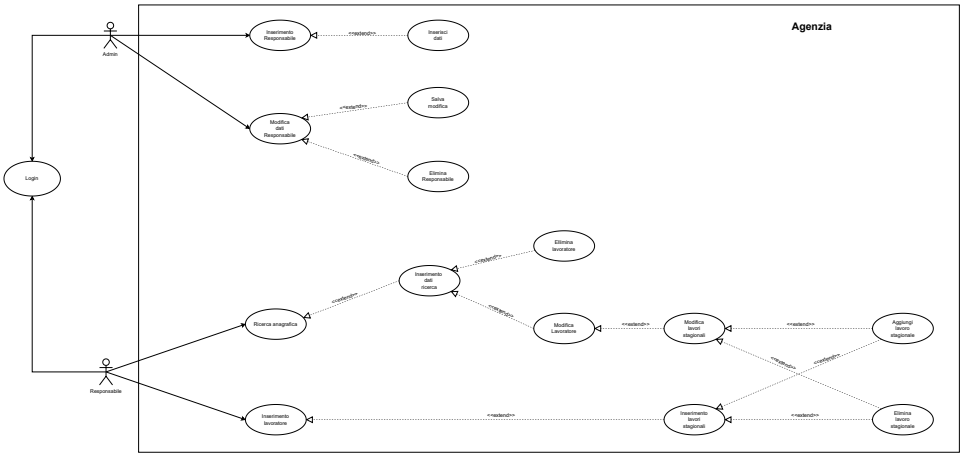


Figura 2: Use case

Accesso al sistema - Login

Sia il responsabile che admin, per poter accedere al sistema, devono inserire le loro credenziali. Una volta inserite, viene premuto LOGIN per eseguire l'accesso. Se sono corrette, l'utente accede al sistema, altrimenti dovrà riprovare.

Dato che il login è identico sia per il responsabile che per admin, il caso d'uso e il relativo diagramma di flusso vengono rappresentati in modo generale.

Use case: Login
Attori: Utente
Precondizioni: Inserimento delle credenziali
Sequenza degli eventi: 1. L'utente clicca "Login" 2. Se le credenziali sono corrette: 2.1. Viene dato l'accesso alla interfaccia successiva 3. Altrimenti viene restituito un errore
Sequenza alternativa: 1. L'utente può in qualsiasi momento terminare il programma
Postcondizioni: 1. Verrà generata una nuova interfaccia in base al tipo di credenziali: amministrative o responsabile

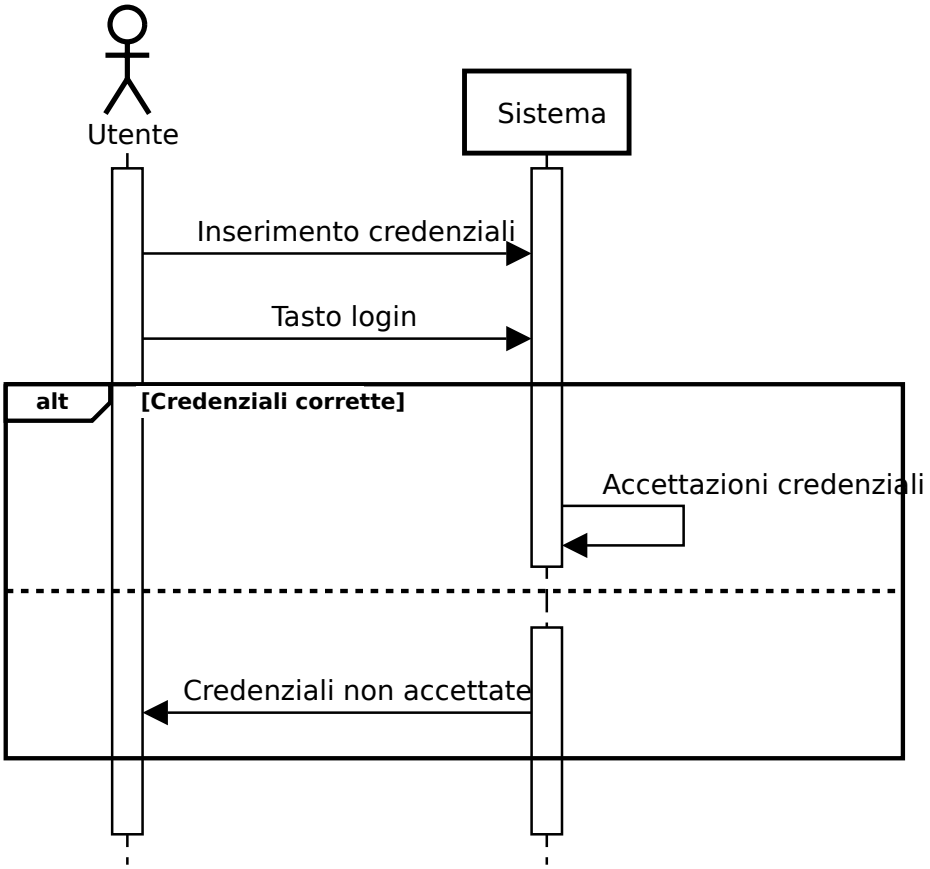


Figura 3: Sequence Diagram - Login

1.3.2 Casi d'uso admin

Inserimento responsabile

L'unico attore per questo caso d'uso è admin.

Il sistema fornisce la possibilità di:

- ✓ Visualizzare l'intero elenco di responsabili presenti nel sistema;
- ✓ Inserire nuovi responsabili;
- ✓ Modificare i responsabili;
- ✓ Eliminare i responsabili.

Quindi, il software garantisce un controllo totale dei responsabili.
Inoltre, per motivi di privacy, le password preesistenti non possono essere visualizzate da admin, ma solamente sovrascritte con una nuova.

Use case: Inserimento responsabile

Attori: Admin

Precondizioni:

Login effettuato con successo utilizzando le credenziali amministrative

Sequenza degli eventi:

1. L'amministratore inserisce i dati del nuovo responsabile
2. Fintantochè l'amministratore è nella pagina di inserimento, Se preme il tasto "salva":
   - 2.1. Le modifiche vengono salvate permanentemente
3. Se preme il pulsante "indietro":
   - 3.1. Ritorna alla pagina iniziale di login
4. Se preme il tasto "modifica"
   - 4.1. I campi vengono autocompilati con quelli del responsabile
   - 4.2. L'amministratore li modifica in accordo alle sue specifiche
5. Se il responsabile clicca il pulsante "elimina":
   - 5.1. Il responsabile viene eliminato dalla lista

Sequenza alternativa:

1. L'amministratore clicca il pulsante indietro e torna alla pagina di login scartando ogni modifica effettuata o terminare il programma

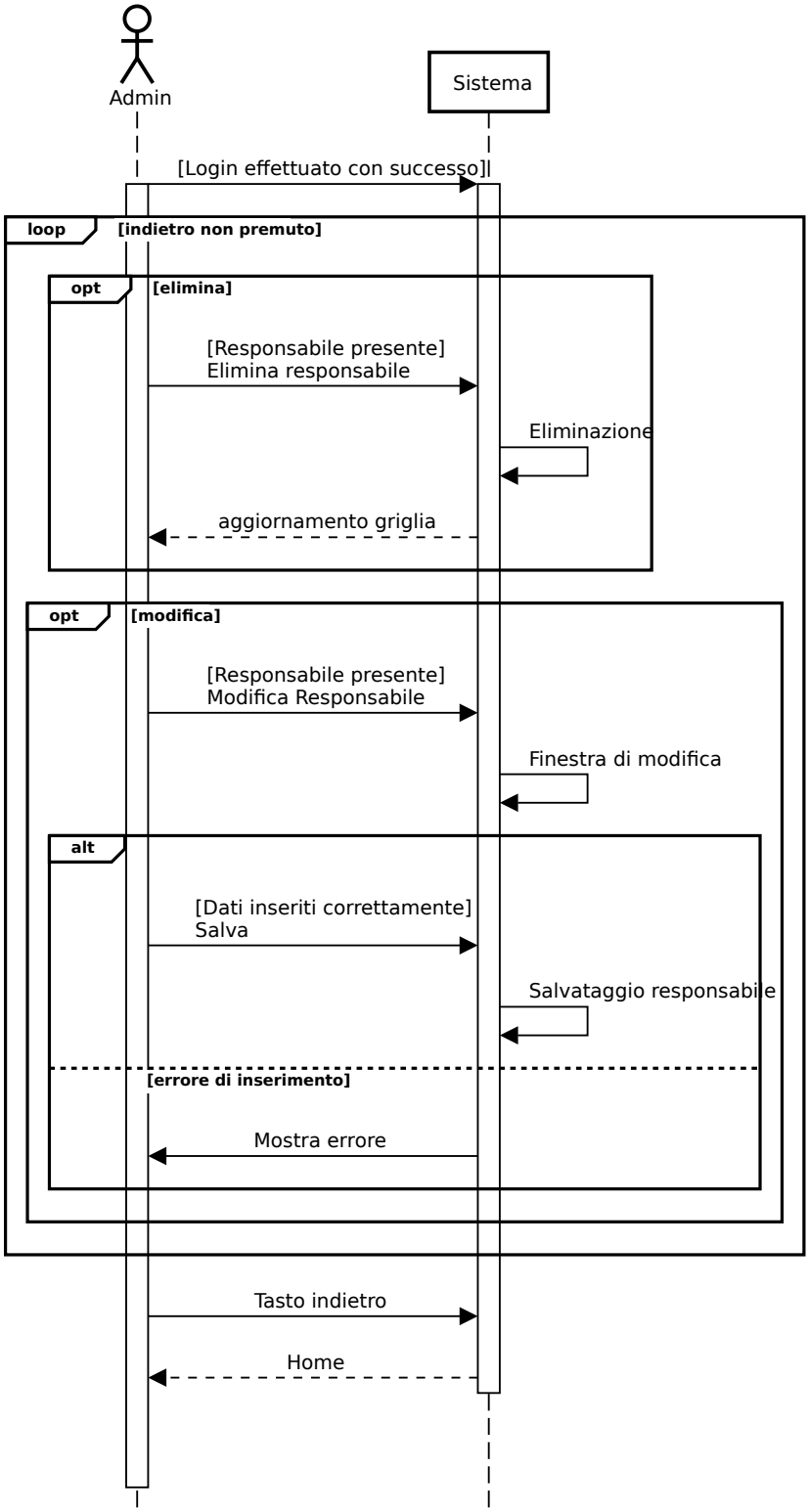


Figura 4: Sequence Diagram - Inserimento responsabile

1.3.3 Casi d'uso responsabile

Inserimento lavoratore

Una volta effettuato il login e selezionato l'inserimento del lavoratore, il responsabile ha la possibilità di inserire tutti i dati di un lavoratore.
In questa schermata viene data la possibilità di inserire tutti i dati del lavoratore eccetto i lavori stagionali.
Infine, esiste la possibilità, in qualsiasi momento, di tornare alla schermata precedente grazie al pulsante INDIETRO.

NOTA: il sequence diagram è stato creato in unione con il caso d'uso "Inserimento lavoro stagionale" (prossima pagina).

Use case: Inserimento lavoratore
Attori: Responsabile
Precondizioni: Login effettuato con successo
Sequenza degli eventi: 1. Il responsabile clicca "Inserimento dati lavoratori": 2. Inserisce i dati del lavoratore da aggiungere alla anagrafica 3. Clicca "avanti" e procede alla pagina di aggiunta dei lavori stagionali
Sequenza alternativa: 1. Il responsabile può in qualsiasi momento tornare indietro alla pagina home o terminare il programma
Postcondizioni: 1. Il lavoratore viene salvato temporaneamente senza alcun lavoro stagionale inserito

Inserimento lavoro stagionale

Questa schermata consente di inserire molteplici lavori stagionali.
Il responsabile ha dunque la possibilità di aggiungere un lavoro stagionale ed eventualmente eliminarlo nel caso in cui abbia commesso errori durante l'inserimento.
Il sistema, dopo che il responsabile ha cliccato sul tasto AGGIUNGI, inserisce il lavoro stagionale nell'apposita tabella.
Il software è flessibile, dunque non costringe l'utente ad inserire obbligatoriamente almeno un lavoro stagionale, ma lascia la libertà di salvare, tramite il tasto FINE, in qualsiasi momento.
Infine il sistema consente di tornare indietro, mantenendo i dati in precedenza inseriti, alla pagina di inserimento.

Use case: Inserimento lavoro stagionale
Attori: Responsabile
Precondizioni: Login effettuato con successo
Sequenza degli eventi: 1. Il responsabile compila i campi preposti con i dati relativi al lavoro stagionale 2. Preme il tasto "aggiungi" ed esegue la aggiunta del lavoro effettuato 3. Fintantochè resta nella pagina di inserimento, Se preme il tasto "fine": 3.1. Rende permanente la aggiunta del lavoratore e dei lavori stagionali 3.2. Torna alla pagina home con la scelta della operazione che può effettuare 4. Se preme il tasto "elimina" accanto a un lavoro stagionale: 4.1. Elimina il lavoro stagionale inserito
Sequenza alternativa: 1. Il responsabile può in qualsiasi momento tornare indietro alla pagina home o terminare il programma

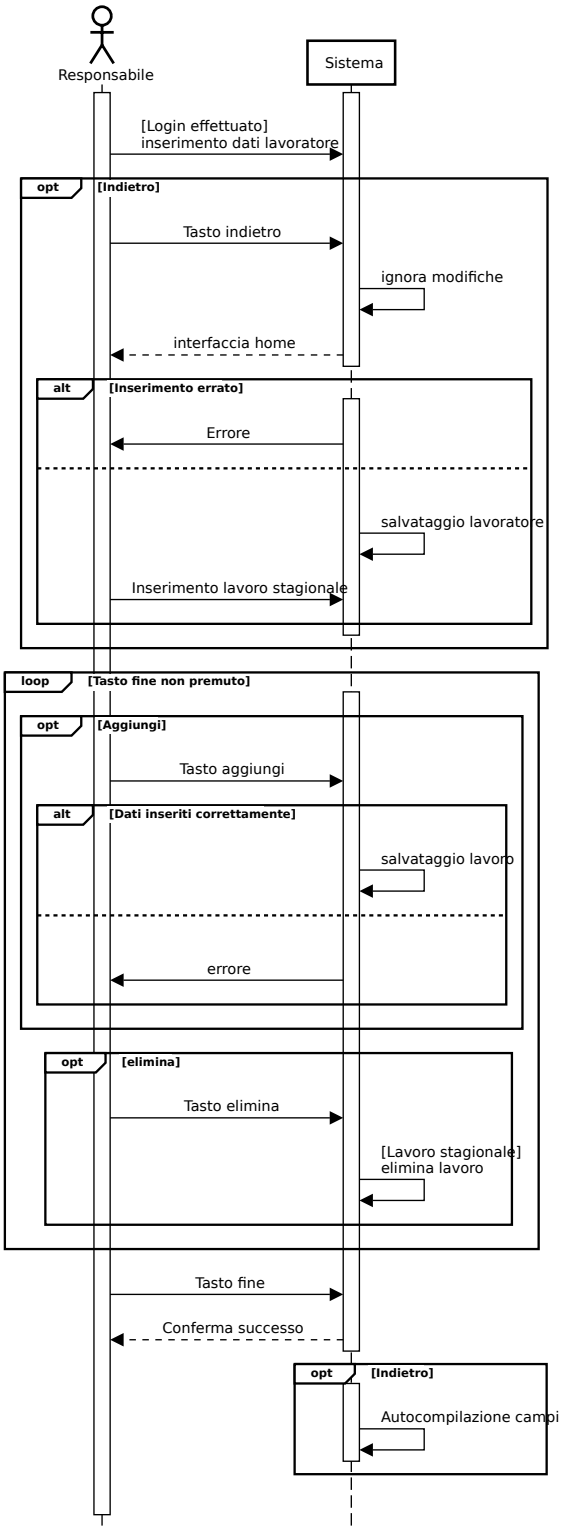


Figura 5: Sequence Diagram - Inserimento lavoratore e lavoro stagionale

Ricerca anagrafica

Questa schermata consente di eseguire una ricerca dei lavoratori all'interno di tutto il sistema.
Il software è flessibile e consente al responsabile di effettuare ricerche molto avanzate utilizzando operatori logici: AND e OR. Inoltre, ogni campo può essere combinato con altri tramite questi due operatori per creare ricerche ancora più dettagliate.
Il sistema consente anche di modificare ed eliminare i lavoratori.

Use case: Ricerca anagrafica
Attori: Responsabile
Precondizioni: Login effettuato con successo
Sequenza degli eventi: 1. Il responsabile clicca "Ricerca e aggiornamento anagrafiche" 2. Fintantochè resta nella pagina di aggiornamento e ricerca, Se preme il tasto "mostra tutti" 2.1. Visualizza tutti i lavoratori attualmente presenti nella anagrafica 3. Se preme il tasto "cerca": 3.1. Mostra i risultati della ricerca effettuata in base ai campi compilati 4. Se preme il tasto "modifica" a fianco di un lavoratore: 4.1. Verrà rimandato alla pagina di inserimento del lavoratore con i campi precompilati per procedere alla modifica 5. Se preme il tasto "elimina": 5.1. Il lavoratore con i relativi dati e i lavori stagionali eventualmente inseriti verrà eliminato definitivamente
Sequenza alternativa: 1. Il responsabile può in qualsiasi momento tornare indietro alla pagina home o terminare il programma

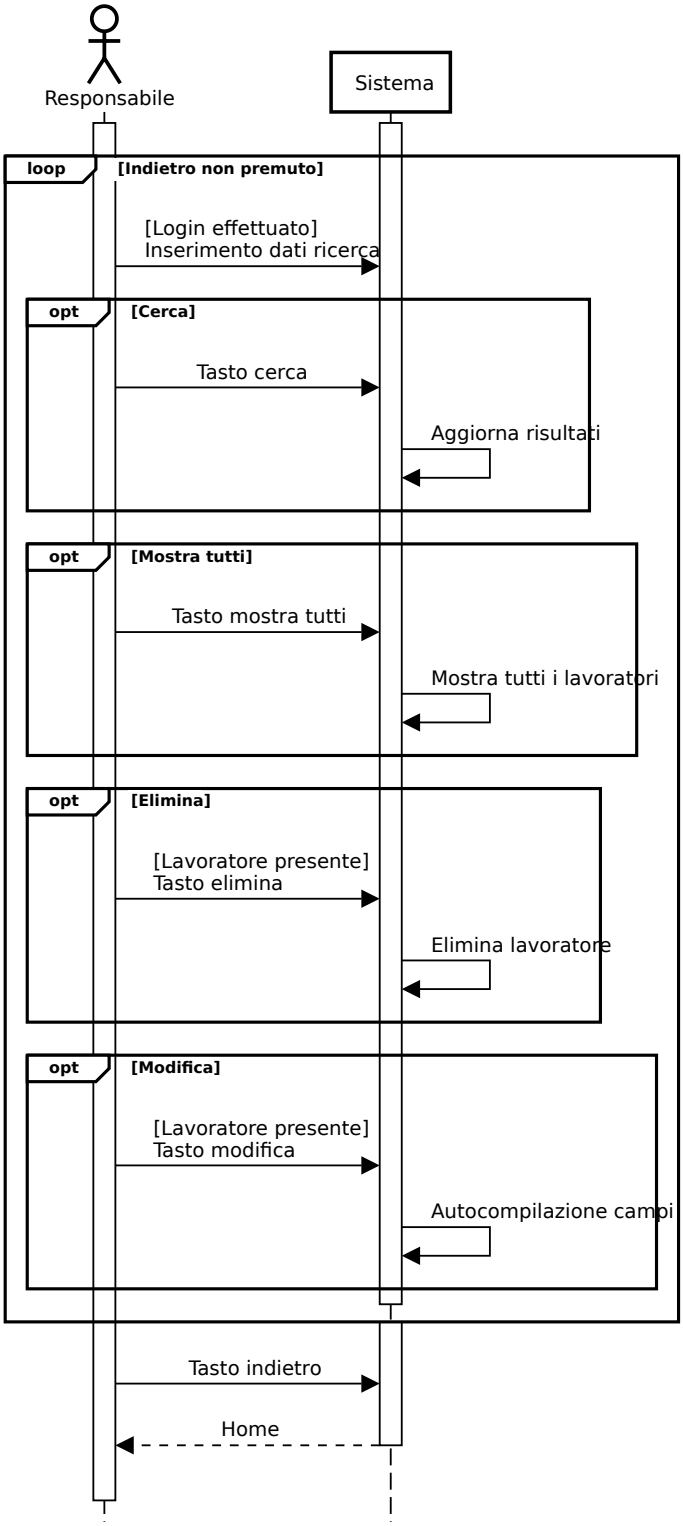


Figura 6: Sequence Diagram - Ricerca anagrafica

1.3.4 Diagramma di attività

NOTA: I seguenti diagrammi hanno l'obbiettivo di essere completi e chiari al lettore. Tuttavia, non è stato possibile inserire casi particolari, come la chiusura del software, o l'esecuzione di molteplici operazioni in sequenza casuale. Sono stati previsti i casi in cui si ripeta l'operazione più di una volta, ma si tratta comunque di situazioni semplici.
Queste scelte sono state effettuate per evitare di rendere difficile la lettura dei diagrammi.

Accesso al sistema - Login

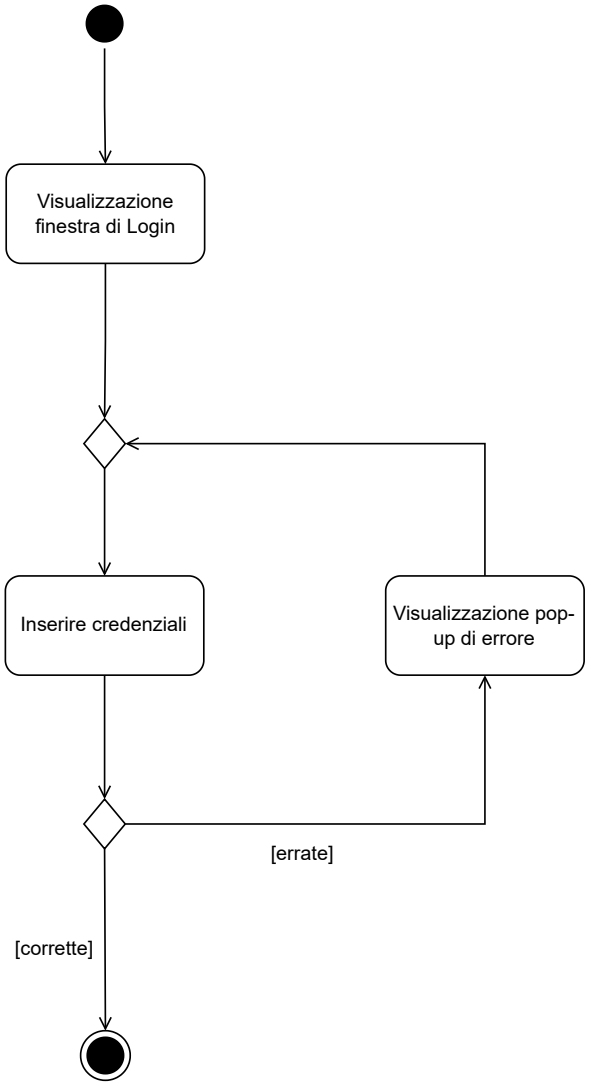


Figura 7: Activity Diagram - Accesso al sistema - Login

Attività del responsabile

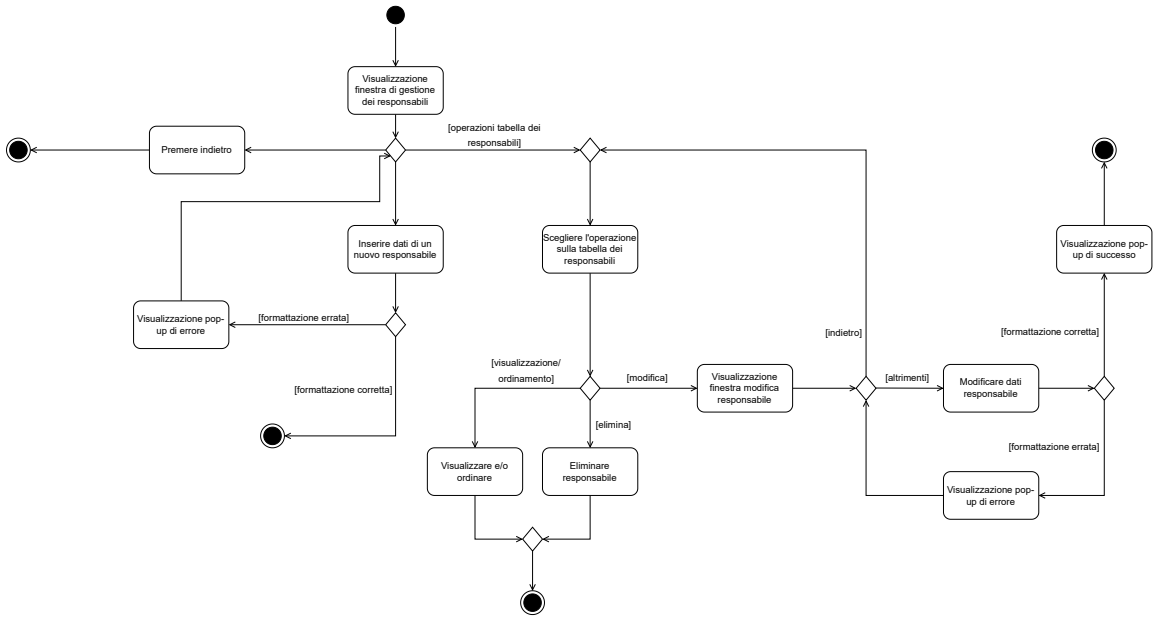


Figura 8: Activity Diagram - Attività del responsabile

Selezione dell'operazione da parte del responsabile

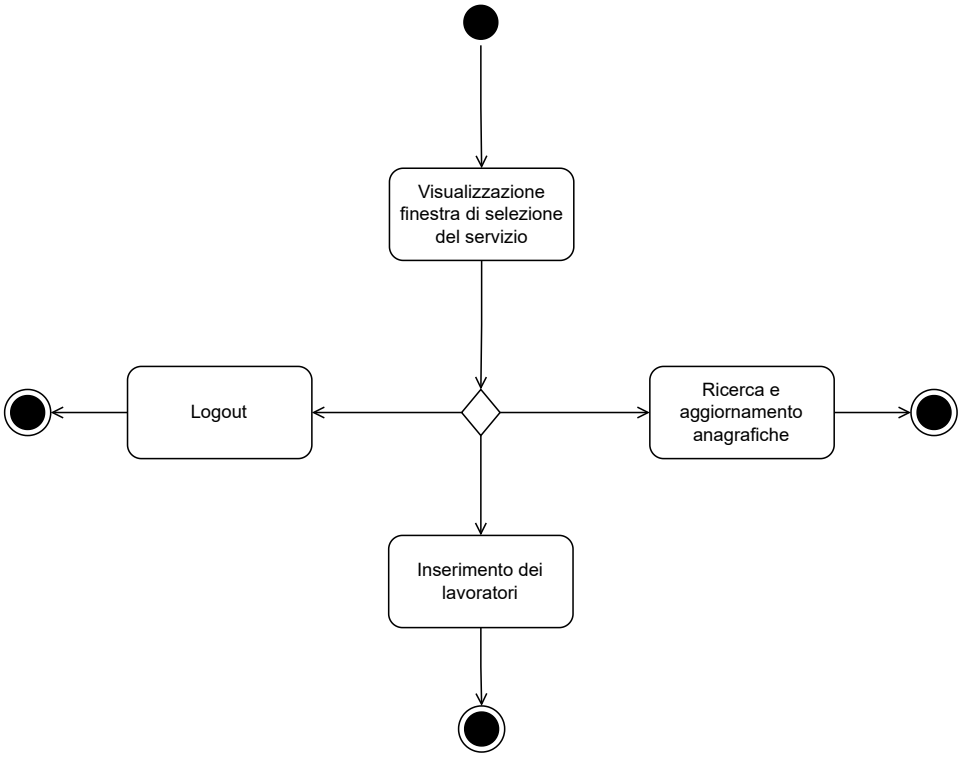


Figura 9: Activity Diagram - Selezione dell'operazione da parte del responsabile

Inserimento lavoratore da parte del responsabile

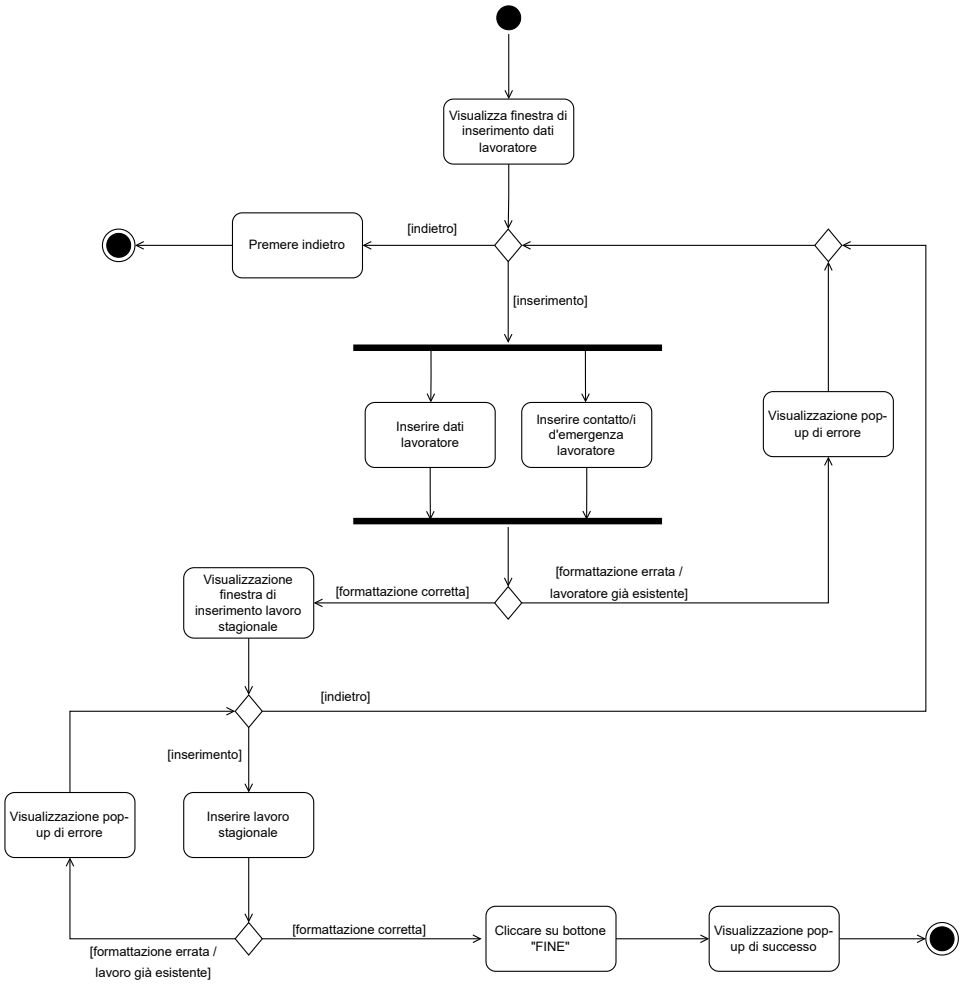


Figura 10: Activity Diagram - Inserimento lavoratore da parte del responsabile

Ricerca anagrafica da parte del responsabile

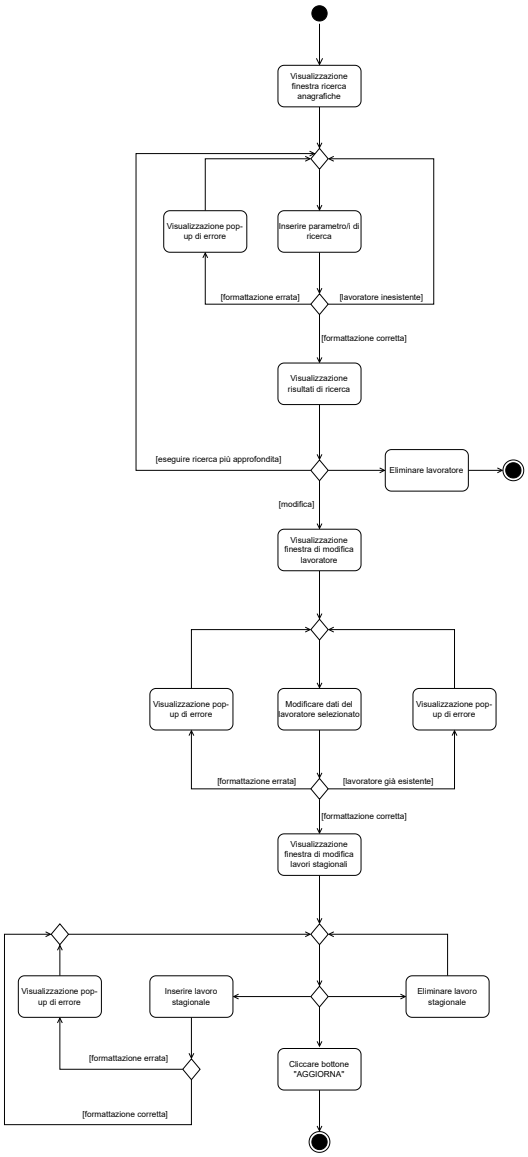


Figura 11: Activity Diagram - Ricerca anagrafica da parte del responsabile

2 Implementazione e sviluppo software

2.1 Processo di sviluppo

Durante il processo di sviluppo è stato utilizzato il metodo **Agile ed Incrementale**.
Il metodo agile scelto è stato **Scrum**. Nonostante sia consigliato per aziende di medie/grandi dimensioni grazie alla sua chiara visibilità esterna, il team di sviluppo ha ottenuto ottimi risultati con l'applicazione di questo sistema.

Il team di sviluppo è composto da tre persone e inizialmente è stato stilato un ***product backlog***, ovvero una lista di elementi di cui si deve occupare il team.
Una volta definiti i compiti, sono stati programmati degli ***sprint***, cioè dei periodi di tempo in cui il team lavorava per completare un compito assegnatoli. Ovviamente, prima di eseguire gli ***sprint***, sono state prese in considerazione le varie velocità di ciascun membro, così da calibrare al meglio il lavoro.
A fine di ogni sprint, il team ha ottenuto spesso un software potenzialmente rilasciabile con poche criticità.
Quasi giornalmente, veniva effettuata una ***scrum***, ovvero una riunione in cui ogni membro presentava i vari avanzamenti effettuati e venivano stabilite le varie priorità del lavoro da svolgere. Purtroppo, queste ***scrum*** sono state effettuate a distanza, utilizzando un software professionale per i meeting online: Zoom.
Il team di sviluppo era anche il ***product owner***, quindi ogni membro doveva continuamente identificare delle caratteristiche o dei requisiti del prodotto, stabilirne le priorità e rivedere continuamente il ***product backlog*** per garantire che il progetto continuasse a soddisfare i requisiti.
Infine, la presenza della figura ***ScrumMaster*** non è stata presente, poiché durante ogni ***scrum*** venivano prese decisioni in base alle opinioni dei membri del team.

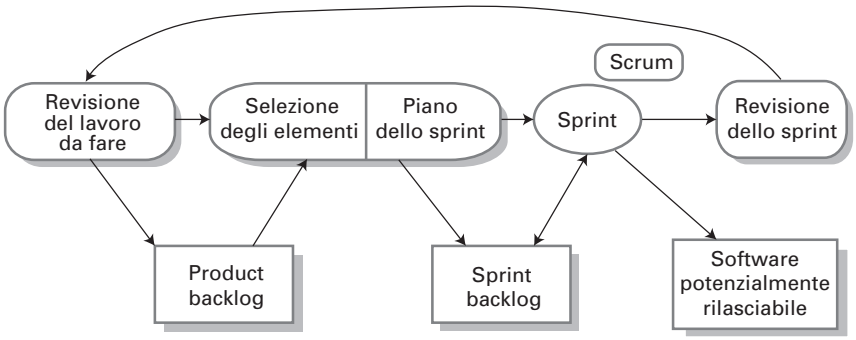


Figura 12: Activity Diagram - Metodo Scrum

In alcuni casi, sono state effettuate sessioni di ***pair programming*** (programmazione a coppie), in cui i membri del team si incontravano per migliorare il software o creare parti di documentazione.

2.2 Progettazione architetturale e pattern utilizzato

Il sistema è stato sviluppato utilizzando le tecniche di modellazione ad oggetti.

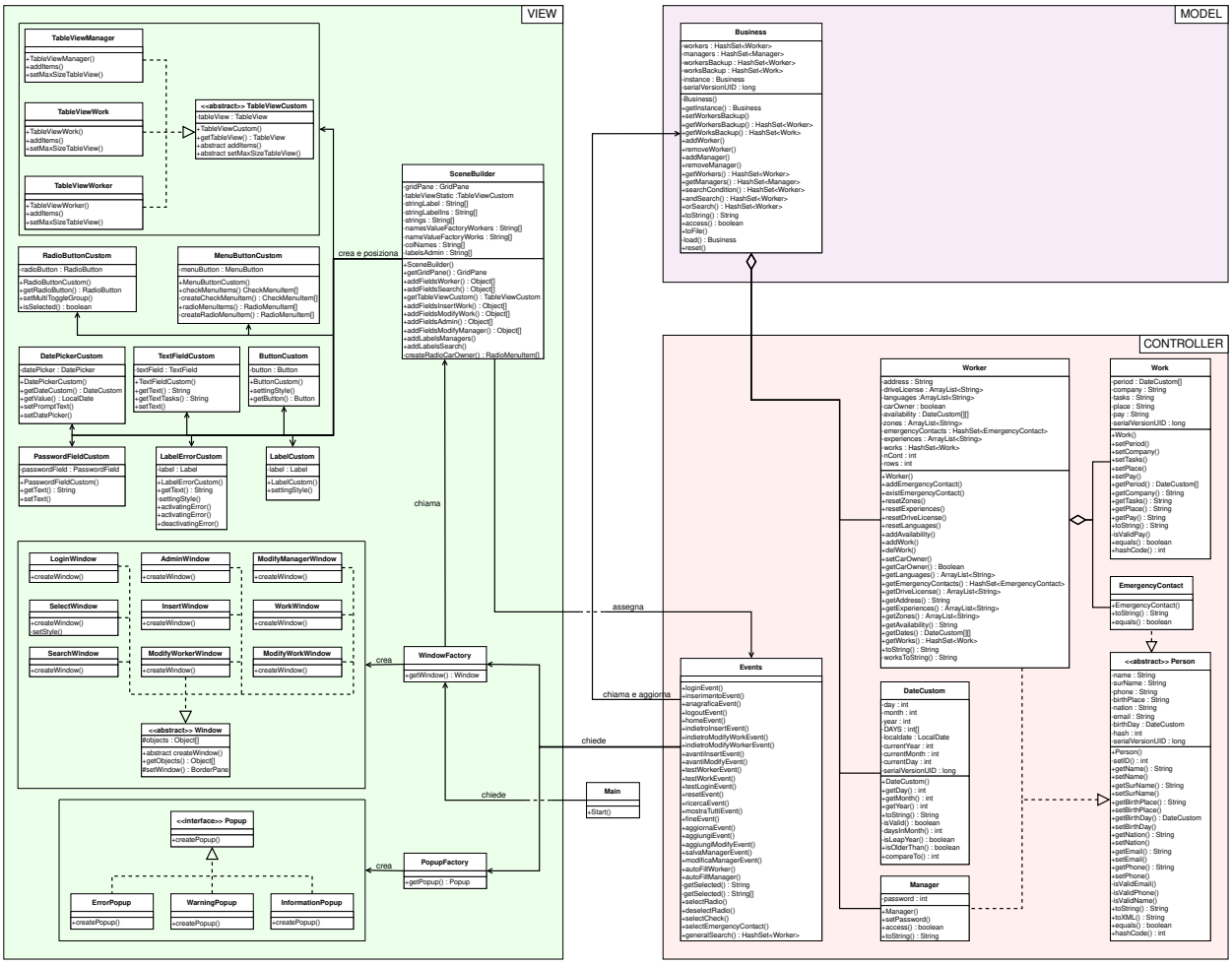
Per facilitare lo sviluppo, è stato scelto il pattern architetturale ***MVC***, dividendo però tra più classi i compiti di ciascun componente seguendo la filosofia del *divide et impera*. La suddivisione dei componenti in classe, successivamente approfondita nel ***Class Diagram***, è la seguente:

- **Model**: Si occupa di gestire i dati utilizzati dall'applicazione. Nel progetto è costituito dalla classe Business, un singleton rappresentante l'agenzia che si occupa di fornire il servizio di gestione dei lavoratori. Contiene tutti i dati inerenti ai responsabili e lavoratori registrati, insieme a tutti i metodi necessari alla gestione di questi. La memorizzazione dei dati è implementata tramite serializzazione.

- **View**: Si occupa della visualizzazione dei dati sull'interfaccia utente e della creazione e gestione di quest'ultima. Nel progetto è costituita dalle classi Window, SceneBuilder e Popup. Le classi Window sono implementate seguendo il factory pattern e si occupano di creare una nuova finestra, chiamando i metodi di SceneBuilder per posizionare i vari campi e pulsanti. Le classi Popup, anch'esse un factory pattern, si occupano della creazione di una finestra popup per dare messaggi di errore, avviso o successo all'utente.

- **Controller**: Si occupa di ricevere tramite l'interfaccia gli input dell'utente e dopo averli elaborati li converte in comandi per Model e View. Nel progetto è costituito dalla classe Events e dalle classi che contengono i dati di lavoratori e responsabili. Queste ultime si occupano, quando viene chiamato il loro costruttore, di controllare i dati passati in input dall'utente, lanciando un'eccezione con messaggio in caso di dati non validi. Le eccezioni vengono poi fermate così che il componente View possa mostrare a schermo un popup di errore con il messaggio dell'eccezione. La classe Events invece contiene i metodi che assegnano ad un pulsante il proprio comportamento, come ad esempio prelevare i dati inseriti dall'utente.

2.3 Class Diagram



2.4 Design pattern

– **Pattern Singleton**: Il *Singleton* è utilizzato per assicurare l'esistenza di una ed unica istanza di un oggetto con un punto di accesso comune. Nel progetto è stato utilizzato per la classe *Business* in quanto questo ne rende più comodo l'accesso, semplificando e rafforzando la sicurezza del sistema di salvataggio tramite serializzazione.

– **Pattern Factory**: Questo pattern è utilizzato per non esporre al client le logiche e i metodi di creazione degli oggetti permettendo però di utilizzare un'interfaccia comune per riferirsi ad essi. In questo caso è stato utilizzato in maniera piuttosto estensiva per la creazione delle classi *Popup* e per le classi *Window*, che si occupano di creare tutte le finestre grafiche dell'applicazione.

2.5 Sequence Diagram del software progettato

Seguono i Sequence Diagram del software. Sono state riportate, seppur in modo sufficientemente dettagliato, solo le interazioni più importanti.
Si noti che tutte le classi di tipo *Window* sono chiuse poco dopo la creazione, senza ulteriori scambi di dati, questo perché si occupano di posizionare finestra e campi sullo stage passato come parametro, una volta ottenuti i campi appena posizionati sulla finestra non vengono più richiamate.
Sono inoltre stati omessi, per semplificare i diagrammi, eventuali loop che indichino la ripetibilità di un'azione da parte dell'utente. I diagrammi mostrano quindi una semplice sequenza con le singole possibili scelte dell'utente.

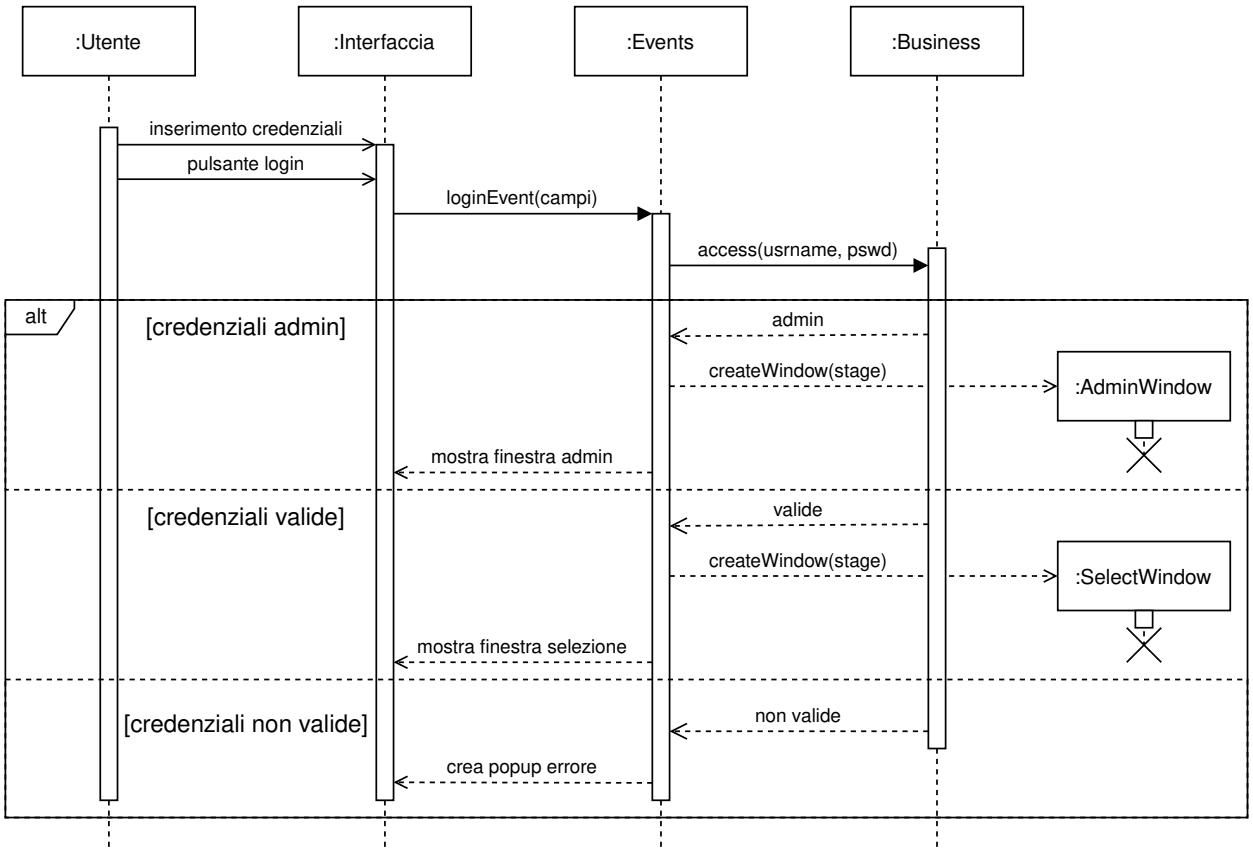


Figura 13: Sequence Diagram - Login di admin e responsabile

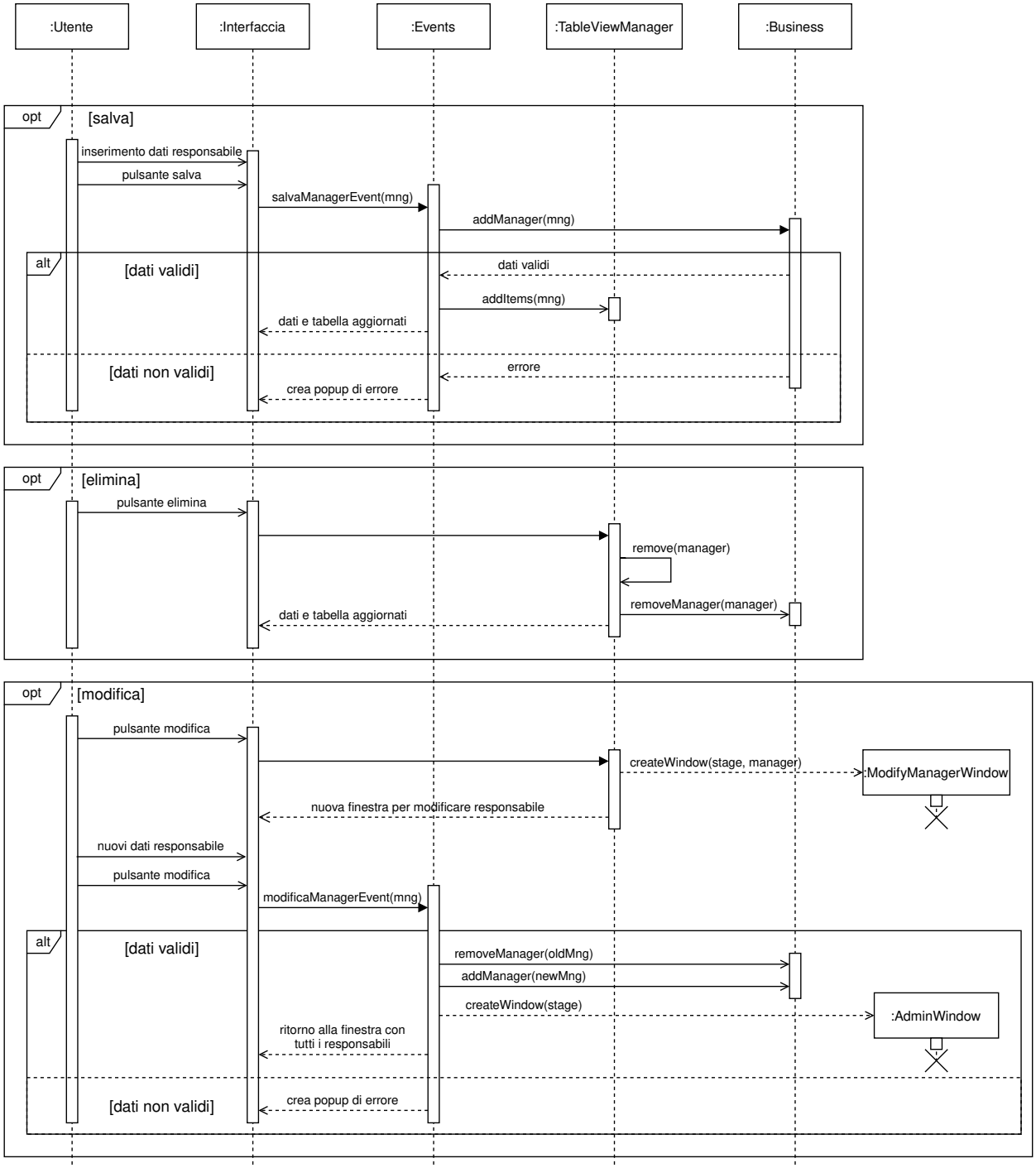


Figura 14: Sequence Diagram - Gestione dei responsabili

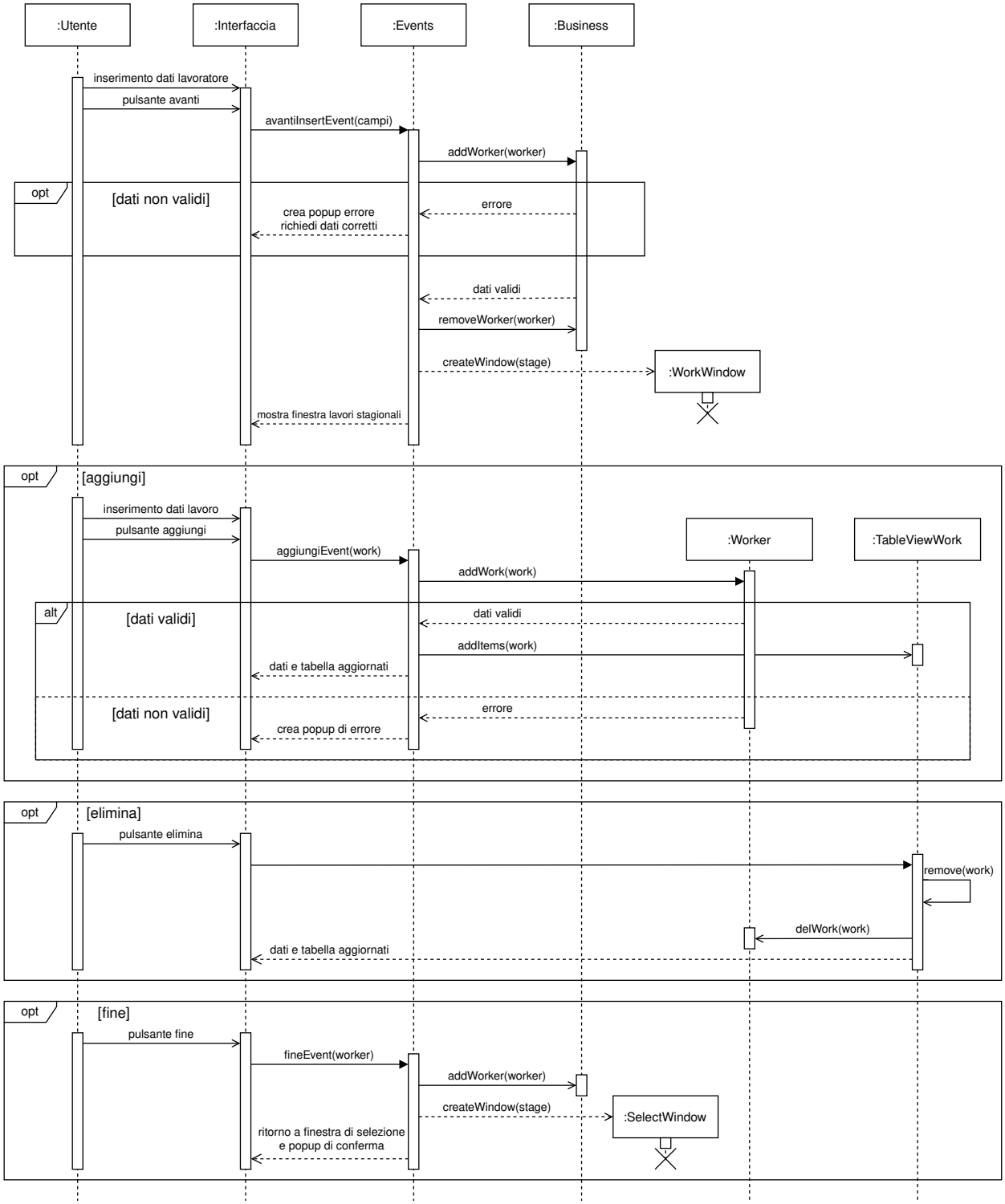


Figura 15: Sequence Diagram - Inserimento di un lavoratore e dei suoi lavori stagionali

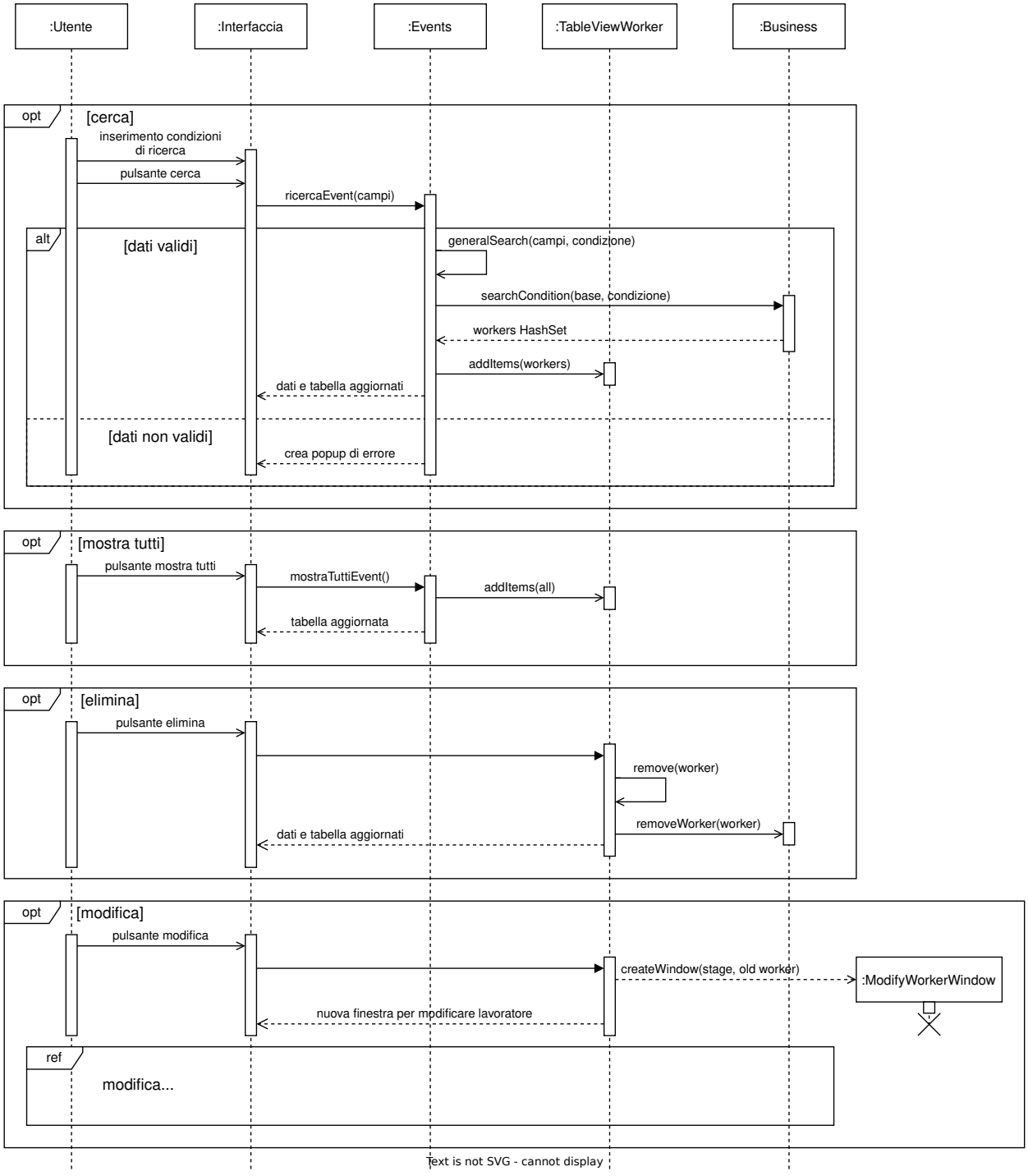


Figura 16: Sequence Diagram - Ricerca e visualizzazione dei lavoratori

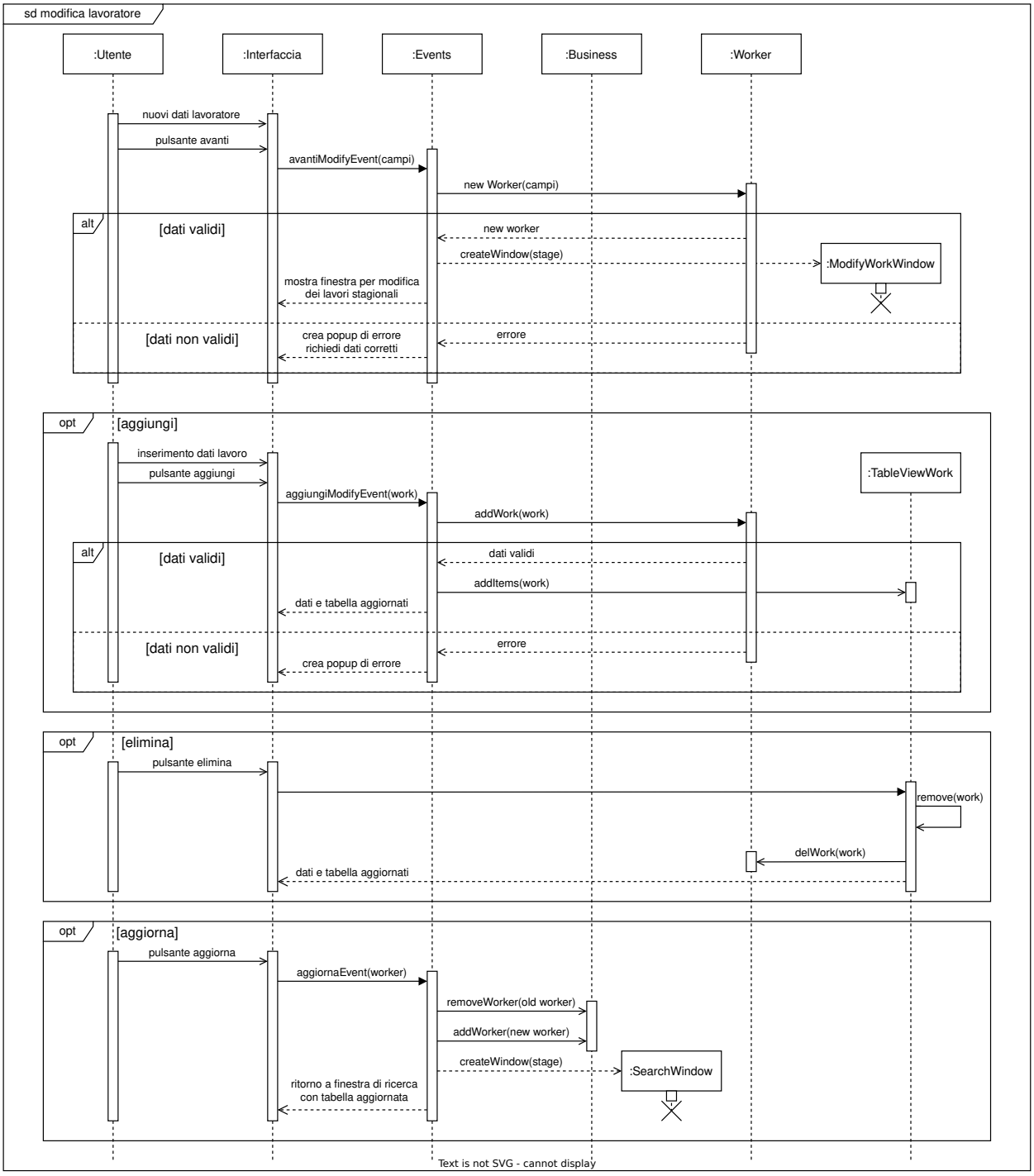


Figura 17: Sequence Diagram - Modifica di un lavoratore e dei suoi lavori stagionali

3 Test sul sistema

Il sistema è stato scrupolosamente sfruttato ad ogni rilascio di versione e dopo ogni completamento di task da parte dei programmatori. Da notare che i test sono stati in primo luogo effettuati internamente al termine di ogni nuova implementazione, in seguito all'unione delle varie parti di codice completate dal team e poi insieme al cliente. Quest'ultimo ha suggerito di volta in volta eventuali nuove revisioni di funzionalità inserite e quelle da aggiungere al sistema in base ai suoi bisogni.
Una volta raggiunta l'approvazione del cliente si è proceduto al successivo ciclo di sviluppo e all'implementazione delle funzionalità successive.
Il ciclo di test prevede, oltre ai controlli di qualità e funzionalità standard, un approfondito ciclo di *Unit test* in modo da assicurare il funzionamento corretto delle singole componenti prima dell'unione al software unitario così da accelerare i tempi di presentazione al cliente.
Il cliente infine ha il compito di verificare, tramite utilizzo standard e con configurazioni particolari, eventuali situazioni non previste dagli sviluppatori o errori non catturati dai quality assesment effettuati internamente.

3.1 Test eseguiti dagli sviluppatori

Di seguito una lista comprensiva delle verifiche effettuate dagli sviluppatori sugli input inseribili nel programma e la capacità di questo nel riceverli e gestirli:

- **Verifica di impossibilità di accesso da parte di un utente non autorizzato**
- **Salvataggio dei dati**
- **Eliminazione del salvataggio ed esecuzione del programma "a vuoto"**
- **Verifica della validità nell'inserimento di nome e cognome**: Nello specifico che non ci siano numeri o caratteri speciali inseriti
- **Verifica della validità del numero di telefono**: Il valore inserito deve essere esclusivamente numerico
- **Verifica della validità delle date**: Non deve essere possibile inserire date arbitrare, i periodi di inizio e fine validità devono essere coerenti, se una data si riferisce al giorno corrente e questa viene scritta nel passato non deve essere valida (caso del periodo di fine disponibilità da parte del lavoratore)
- **Verifica di validità dell'inserimento di una mail**
- **Impossibilità di lasciari incompleti i campi obbligatori**
- **Funzionalità corretta della funzione di ricerca**: Correttezza nelle ricerche utilizzando vari operatori di ricerca AND e OR, ricerche parziali completando solo alcuni dei campi disponibili.
- **Modifica ed eliminazione dei valori**: Valido per tutti gli oggetti inseriti che lo prevedono, come i lavori stagionali e i lavoratori, controllare che l'autocompletamento dei campi nelle interfacce di modifica avvenga con successo, così come l'eliminazione degli oggetti che deve avvenire in maniera puntuale, corretta e senza errori
- **Possibilità di navigazione a ritroso delle finestre**: Dove previsto, tornare indietro alla finestra precedente deve conservare correttamente i dati inseriti e la navigazione deve avvenire puntualmente secondo l'ordine stabilito e senza errori.
- **Verifica del salvataggio con chiusura**: alla chiusura del programma, da qualsiasi punto, tutti i dati confermati dall'utente devono essere salvati senza errori e recuperati correttamente al riavvio.