Documentazione progetto Ingegneria del Software

Angelo Marano VR456447
Elisa Campostrini VR456441
Stefano Gennarelli VR456907

Luglio 2022

Indice generale

Specifiche casi d’uso

Note generali

Il sistema è sviluppato per i dipendenti di un’agenzia del lavoro . Ai dipendenti vengono fornite le credenziali per accedere al sistema. Se il dipendente accede per la prima volta e l’autenticazione va a buon fine, il dipendente si deve registrare inserendo i propri dati anagrafici, l’indirizzo email , il recapito telefonico e le credenziali di accesso fornite. Se non è la prima volta che il dipendente accede e l’autenticazione va a buon fine il dipendente viene indirizzato al menù (schermata iniziale).

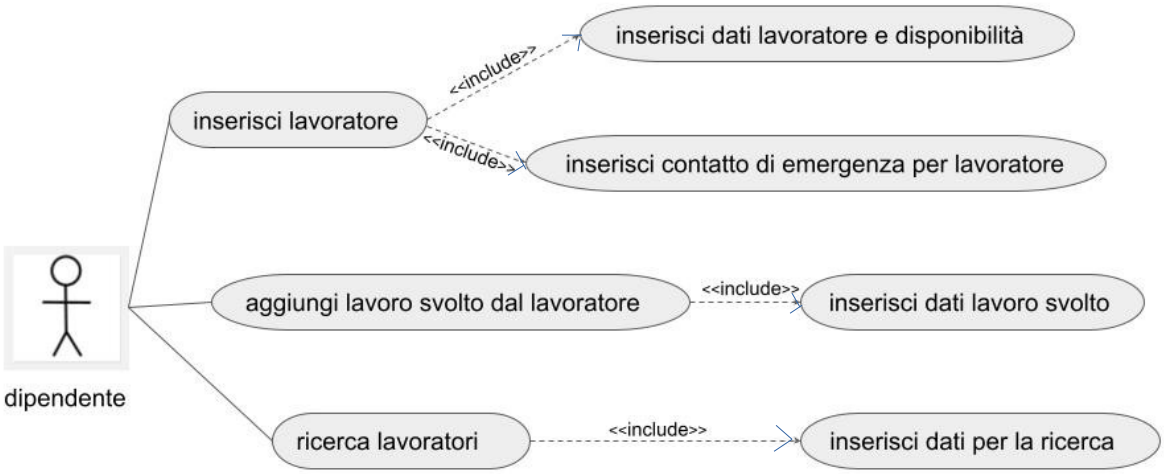

Figura 1. casi d’uso

Inserisci lavoratore

I dipendenti dell'agenzia devono poter registrare nuovi lavoratori. Per aggiungere nuovi lavoratori devono inserire i dati anagrafici del lavoratore, l'indirizzo, il recapito telefonico (se presente) , l'email ,le eventuali specializzazioni/esperienze precedenti, le lingue parlate, il tipo di patente, se automunito o meno , il periodo di disponibilità e un contatto di emergenza.

Attori: *Dipendente dell'agenzia*

Precondizioni: *Il dipendente deve essersi autenticato.*

Passi:

1. *Il dipendente accede al sistema.*
2. *Il dipendente è introdotto all'interfaccia di base.*
3. *Il dipendente accede all'interfaccia per l'aggiunta di un nuovo lavoratore .*
4. *Il dipendente inserisce i dati relativi al lavoratore.*
5. *Il dipendente clicca su next.*
6. *Il dipendente inserisce il contatto di emergenza.*
7. *Il dipendente clicca su next.*

Postcondizioni: *il lavoratore viene inserito.*

Diagramma di sequenza per l’inserimento di un lavoratore

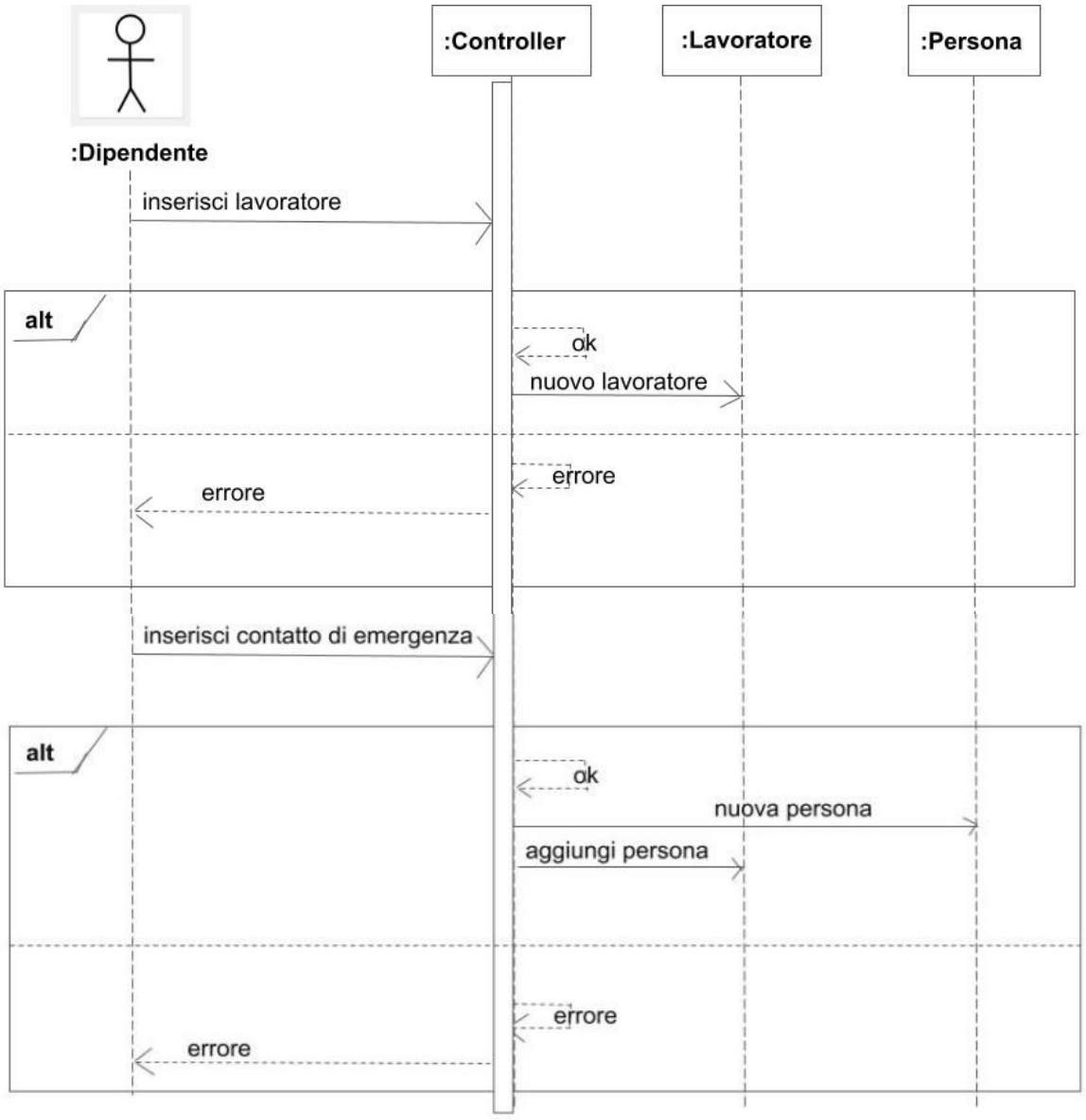

Figura2 : diagramma di sequenza inserimento nuovo lavoratore

Si inseriscono i dati anagrafici , l’indirizzo, il recapito telefonico (se presente) , l’email ,le eventuali specializzazioni/esperienze precedenti, le lingue parlate, il tipo di patente, se automunito o meno e il periodo di disponibilità del lavoratore. I dati inseriti vengono controllati e se corretti si crea un nuovo lavoratore. Solo dopo aver creato il nuovo lavoratore si passa all’inserimento del contatto di emergenza. Si verificano i dati del contatto di emergenza e solo se corretti si aggiunge al nuovo lavoratore il contatto di emergenza.

Aggiungi lavoro svolto del lavoratore

Il dipendente dell'agenzia deve poter aggiornare le anagrafiche con tutti i lavori che i lavoratori hanno svolto negli ultimi 5 anni. Per ogni lavoro svolto il dipendente dell'agenzia deve inserire : il periodo,il nome dell'azienda, le mansioni svolte, il luogo di lavoro, la retribuzione lorda giornaliera.

Attori: *Dipendente dell'agenzia*

Precondizioni: *Il dipendente deve essersi autenticato.*

Passi:

1. *Il dipendente accede al sistema.*
2. *Il dipendente è introdotto all'interfaccia di base.*
3. *Il dipendente accede all'interfaccia per l'aggiunta di un nuovo lavoro.*
4. *Il dipendente inserisce i dati relativi al lavoro svolto*
5. *Il dipendente clicca su next.*

Postcondizioni: *il lavoro viene inserito.*

Ricerca lavoratori

I dipendenti dell'agenzia devono poter effettuare ricerche rispetto ai profili richiesti. I dipendenti devono poter fare ricerche per lavoratore, per lingue parlate,per periodo di disponibilità,per mansioni indicate,per luogo di residenza,per disponibilità di auto e/o patente di guida. I dipendenti devono poter fare ricerche complesse (sia in and che in or).

Attori: *Dipendente dell'agenzia*

Precondizioni: *Il dipendente deve essersi autenticato.*

Passi:

1. *Il dipendente accede al sistema.*
2. *Il dipendente è introdotto all'interfaccia di base.*
3. *Il dipendente accede all'interfaccia per la ricerca di lavoratori.*
4. *Il dipendente inserisce i campi desiderati selezionando il metodo di ricerca (and o or).*
5. *Il dipendente clicca su cerca.*

Postcondizioni: *vengono stampati a video i lavoratori che soddisfano il profilo richiesto.*

Diagrammi di attività

Vengono rappresentati di seguito i diagrammi delle singole attività che il dipendente dell’agenzia può svolgere. Il dipendente dell’agenzia attraverso la schermata iniziale di menù può scegliere quale attività svolgere. Alla conclusione dell’attività si ritorna sempre al menù e si può scegliere nuovamente l’attività da svolgere.

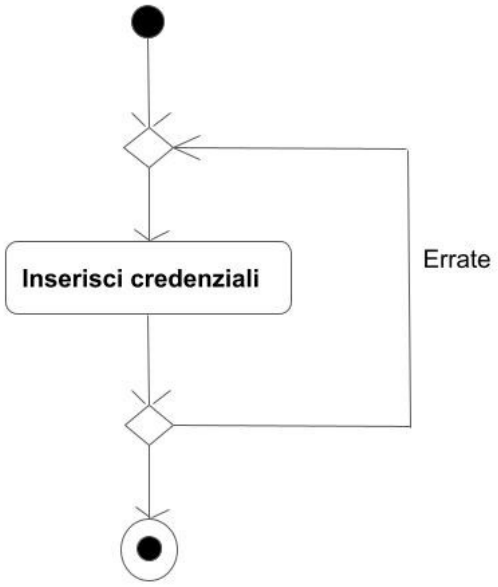

Figura 3: diagramma attività login

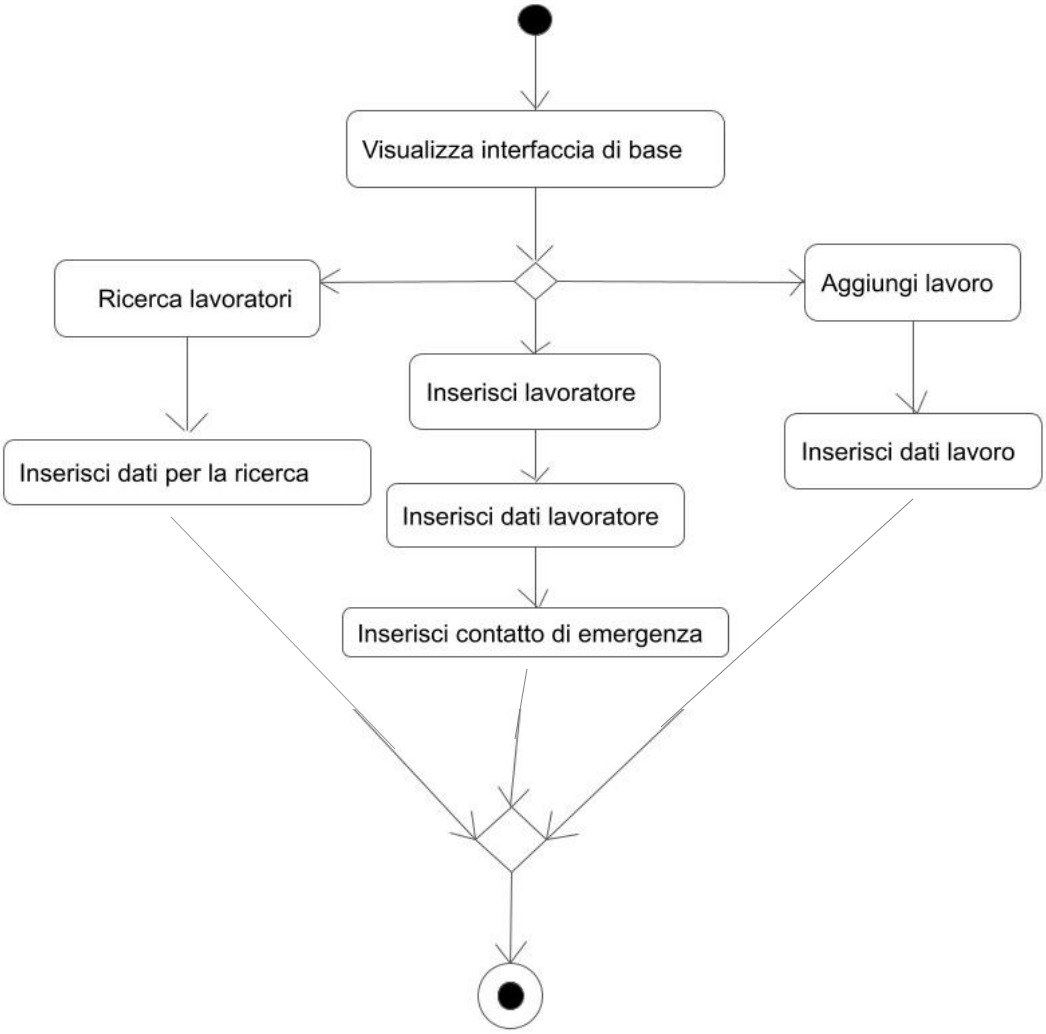

Figura 4: Attività dipendente

Sviluppo del progetto dell’architettura e implementazione del sistema

Il processo di sviluppo è stato essenzialmente di tipo *Agile* ed *Incrementale*. Ciononostante, si è cercato quanto più possibile di mantenere sequenziali le fasi di progettazione, implementazione e validazione. Questo è stato fatto cercando di procedere sempre nello stesso modo: progettando prima e implementando dopo. Dopo ogni modifica significativa è stata fatta una breve attività di test .Abbiamo implementato i vari casi d’uso e magari sviluppandone uno ,abbiamo rivisto qualche parte di codice implementato precedentemente , svolgendo , quindi, l’attività di *refactoring*.
Prima di cominciare il ciclo di sviluppo principale, abbiamo condotto una fase di analisi dei requisiti, generando i relativi use-case e i diagrammi di attività riportati precedentemente nella documentazione.
Anche la progettazione architetturale è stata fatta prima di iniziare il lavoro centrale, in modo da assicurarsi di trovarsi per lo meno in una situazione solida da quel punto di vista. Per quanto riguarda l’implementazione, non sono state fatte grosse divisioni o piani di sviluppo programmatici.

Progettazione e pattern architetturali usati

Il sistema è stato progettato utilizzando le tecniche di modellazione ad oggetti. Dal punto di vista architetturale, si è scelto di utilizzare il pattern MVC.
Utilizzando le librerie grafiche di JavaFx abbiamo deciso di adottare il pattern MVC.
Le tre parti sono implementate all’interno dello stesso package. Il Modello è implementato nella classe Main, la Vista è rappresentata dai file fxml presenti e il Controllore è implementato nella classe Controller.
• Modello: Gestisce le informazioni immagazzinate dal sistema e le loro strutture .
• Vista: Rappresenta graficamente il modello.
• Controllore: Definisce la logica del sistema,ovvero il comportamento del sistema a
fronte degli stimoli esterni.
Le azioni dell’utente(input) saranno gestite dalla classe denominata “Controllore” la quale modifica le informazioni contenute nel Modello e aggiorna la vista.
Di seguito viene riportato uno schema dell’architettura.

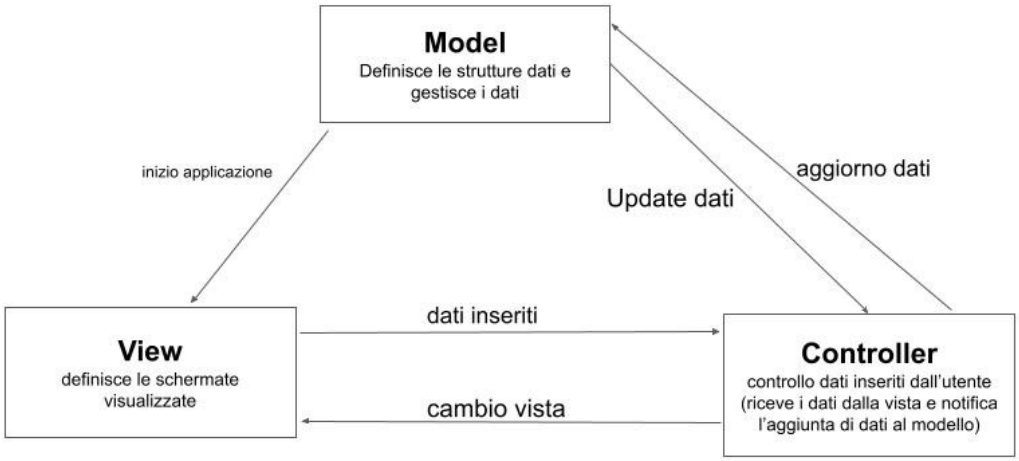

Figura 5 :Architettura MVC

Diagramma delle classi

Nel diagramma delle classi per ragioni di spazio e per rendere il tutto più comprensibile non vengono inseriti i parametri di cui i metodi necessitano, inoltre tutti i campi inseriti nei file fxml che vengono controllati nel Controller no vengono inseriti nella classe.

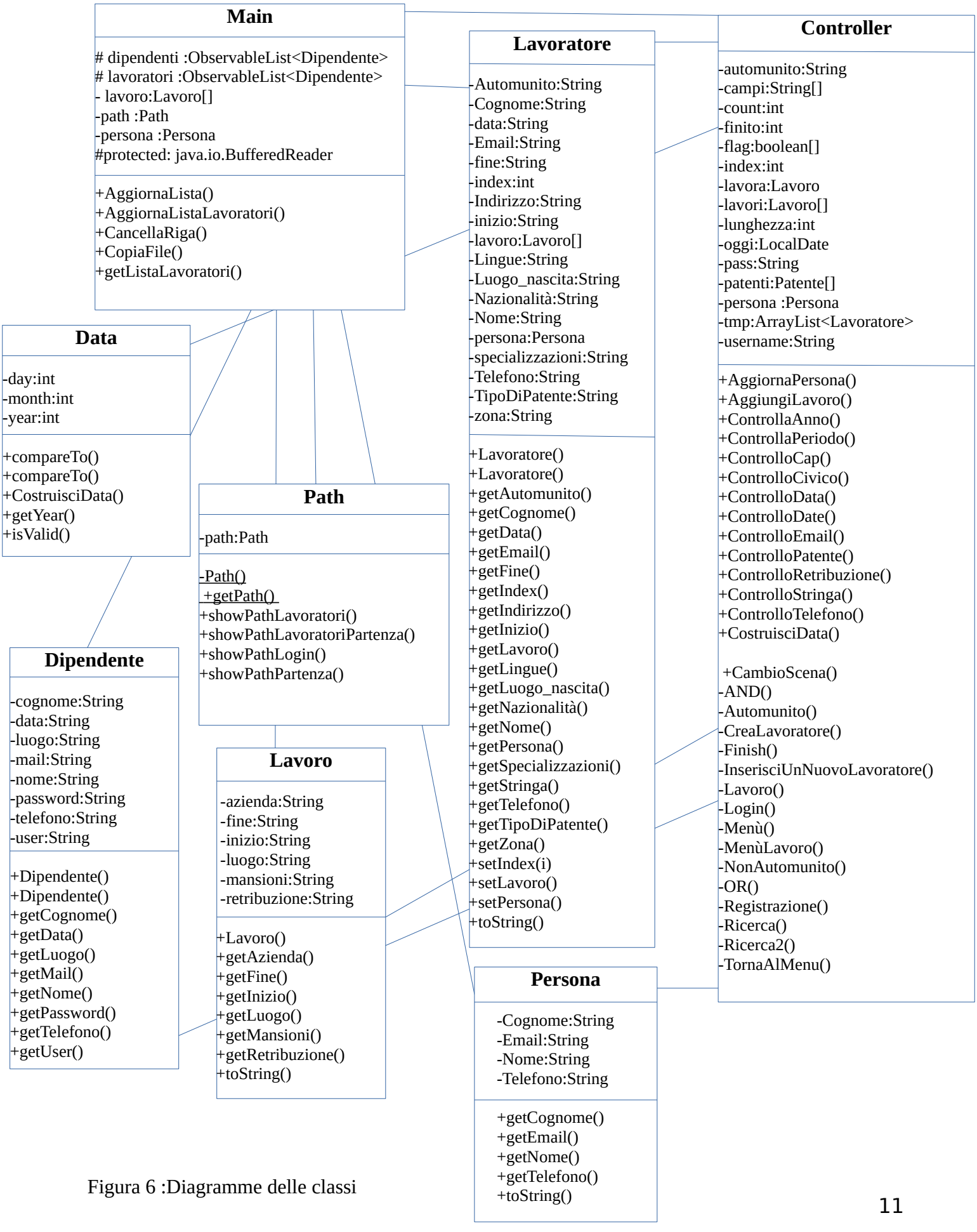

Figura 6 :Diagramme delle classi

Commento sui metodi

Vengono tralasciati i parametri utilizzati dai metodi

Classe Main:

+AggiornaLista() aggiorna i dati di login dei dipendenti
+AggiornaListaLavoratori() aggiorna i dati dei lavoratori
+CancellaRiga() toglie il prima lavoro inserito nell'array del lavoro del lavoratore , quando l'array è pieno
+CopiaFile() aggiorna tutti i file contenenti i dati
+getListaLavoratori() legge da file contenente la lista dei lavoratori e li aggiunge alla lista dei lavoratori(viene fatto solo all'avvio dell'applicazione)

Classe Controller:

+AggiornaPersona() aggiunge il contatto di emergenza al lavoratore
+AggiungiLavoro() aggiunge il lavoro svolto al lavoratore

Seguono i metodi per il controllo dei campi inseriti dall'utente:
+ControllaAnno()
+ControllaPeriodo() controlla che la data di fine sia dopo la data di inizio
+ControlloCap()
+ControlloCivico()
+ControlloData()
+ControlloDate()
+ControlloEmail()
+ControlloPatente()
+ControlloRetribuzione()
+ControlloStringa()
+ControlloTelefono()
+CostruisciData()

Tutti i metodi seguenti gestiscono il cambio della Vista
+CambioScena()
-AND()
-Automunito()
-CreaLavoratore()
-Finish()
-InserisciUnNuovoLavoratore()
-Lavoro()
-Login()
-Menù()
-MenùLavoro()
-NonAutomunito()
-OR()

-Registrazione()
-Ricerca()
-Ricerca2()
-TornaAlMenu()

Tutte le altre classi contengono metodi che ritornano la variabile richiesta. Nella classe Dipendente e nella classe Lavoratore i primi due metodi rispettivamente Dipendente() e Lavoratore() sono due costruttori e si differenziano per il numero di variabili dati in input.

Design pattern utilizzati

Nel progetto viene utilizzato il **pattern Singleton** per fare in modo che i file csv contenenti i dati dei dipendenti dell'agenzia e quelli dei lavoratori , siano unici.
Viene adottata anche la visione del **pattern factory** , solo per quanto riguarda le generazione delle due liste contenenti i lavoratori e i dipendenti dell'agenzia. Queste liste , infatti, vengono create con protezione package , in modo che non sia possibile accedervi dall'esterno. Si è , quindi, adottata solamente l'idea di fondo del pattern e non la sua implementazione per una questione di semplicità e chiarezza del codice.
Nel contesto dei lavoratori per quanto riguarda l'assegnamento del contatto di emergenza e dei lavori svolti viene utilizzata l'idea del **pattern data access object (DAO)**.Infatti non viene implementata l'interfaccia che definisce le operazioni standard per non complicare ulteriormente la struttura del prototipo.
La classe che acquisisce i dati inseriti dall'utente è la classe Controller, mentre la classe che rappresenta l'informazione che sto considerando è rappresentata dalla classe Lavoratore.
La base della programmazione di java , inoltre, comprende anche **l'iterator pattern** che viene utilizzato nella forma di for/each e while.

Attività di Test e Validazione

Per verificare la correttezza del software prodotto sono state svolte le seguenti attività:

1. verifica che il software rispetti le specifiche date
2. verifica della consistenza tra diagrammi inseriti nella documentazione e il codice prodotto
3. verifica dei pattern
4. test degli sviluppatori
5. test dell'utente

Per quanto riguarda la verifica della consistenza tra diagrammi e codice prodotto ci si è basati sulla documentazione javadoc prodotta (inserita nel progetto).

Test degli Sviluppatori

Nella fase di implementazione del codice ad ogni cambiamento significativo è stato svolto un test in modo da verificare la correttezza delle azioni svolte. Lo sviluppatore in questa fase inserisce i campi richiesti corretti andando a verificare che il sistema inserisca i dati nel file csv e svolga i cambi di vista correttamente. In un secondo momento , poi ,sono stati inseriti uno a uno i campi, errando, per verificare che i controlli funzionassero correttamente.
Per esempio :

- nell'accesso si inseriscono credenziali errate
- al posto dei *campi che richiedono l'inserimento di una stringa viene messo un numero , oppure una stringa contenente un numero
- viene messa una data di nascita contenente un numero di giorni sbagliato , o il mese sbagliato o un anno futuro
- nella data di fine periodo di disponibilità viene messa una data antecedente alla data di inizio
- viene messa una email sbagliata (senza stringa.string@stringa.stringa)
- viene messo un numero di telefono con più di 10 cifre , o un numero contenente un carattere
- viene messa una patente che non è tra le patenti esistenti
- vengono lasciati liberi alcuni campi che non sia solamente il telefono
- nel caso dell'aggiunta di un nuovo lavoro svolto si è provato a immettere cinque lavori e poi aggiungerne uno in modo da osservare se viene eliminato il primo lavoro registrato , in modo da mantenere sempre 5 lavori e non più.
- Nel caso dell'aggiunta di un nuovo lavoro è stato messo un anno precedente agli ultimi cinque per verificare che solamente i lavori degli ultimi cinque anni possano essere registrati.
- Nel caso della ricerca si è provato a condurre ricerche inserendo campi in end e altri campi in or per verificare la correttezza delle ricerche condotte

Tutti questi casi sono stati testati singolarmente e poi collettivamente in modo da verificare l’assenza di eventuali errori.

*campi che richiedono l’inserimento di una stringa:

- nome, cognome, luogo di nascita,nazionalità,via, città,specializzazioni,lingue parlate e zona di preferenza per quanto riguarda il lavoratore.
- Nome , cognome per il contatto di emergenza.
- Nome dell’azienda , mansioni svolte,luogo di lavoro per l’aggiornamento dei lavori svolti.

Test effettuato dell’utente

Gli utenti coinvolti sono stati individui di diversa età con una limitata conoscenza dell’ambito informatico. Non è stata data loro alcuna informazione se non l’ambito in cui il software deve lavorare .

Questa fase è stata molto utile in quanto, attraverso le loro segnalazioni , è stato possibile migliorare la disposizione dei campi da inserire e rendere le schermate più intuitive.
I test effettuati degli utenti sono di diverso genere e riprendono quelli che erano già stati fatti dagli sviluppatori. I test effettuati sono stati :

- inserimento di credenziali errate
- inserimento di date sbagliate(15/05/2023 , 15/8/2021)
- campi lasciati vuoti (assenza di patente/zona di preferenza)
- inserimento mail sbagliata
- inserimento di un anno antecedente agli ultimi 5 nell’aggiunta di un lavoro svolto.

Grazie ai test svolti dagli utenti è stato possibile individuare controlli che non erano stati implementati come il procedere lasciando tutti i campi vuoti. Questo controllo è stato aggiunto e testato anche dagli sviluppatori.