

# Documentazione progetto Ingegneria del Software

Massagrande Marco VR446285  
Vanini Samantha VR443381

Luglio 2021

## Indice

Requisiti ed interazioni utente-sistema	
Introduzione.....	3
Requisiti.....	3
Assunzioni.....	4
Specifiche casi d'uso.....	5
Casi d'uso relativi all'utente.....	7
Casi d'uso relativi all'amministratore .....	10
Diagrammi di attività .....	13
Sviluppo: progetto dell'architettura ed implementazione del sistema	
Note sul processo di sviluppo.....	16
Progettazione e pattern architetturali usati .....	17
Modellazione e gestione dei dati.....	19
Implementazione e design pattern usati.....	21
Attività di test e validazione	
Ispezione codice e documentazione.....	22
Test degli sviluppatori.....	22
Test utente generico.....	22

## Introduzione

L'obiettivo dell'elaborato è quello di creare un sistema gestionale per un servizio di organizzazione di vacanze studio per ragazzi. In questa relazione abbiamo raccolto la documentazione sviluppata in fase di progettazione del prototipo.

## Requisiti

Viene riportato il testo dell'elaborato, con i requisiti funzionali evidenziati:

Si vuole progettare un sistema informatico per gestire il servizio di organizzazione di vacanze studio per ragazzi. I ragazzi interessati possono iscriversi al servizio. Per ogni ragazzo vengono memorizzati i dati anagrafici, indirizzo, recapito telefonico personale (se presente), email, le eventuali allergie (nome allergene e descrizione delle precauzioni necessarie), gli hobby. Di ogni ragazzo si memorizzano anche le informazioni di almeno uno dei genitori: nome, cognome, telefono, indirizzo email. Una volta iscritti, i ragazzi possono accedere al sistema utilizzando email e password. Le vacanze possono essere visualizzate per data di partenza, per durata (numero settimane) o per luogo di destinazione. La vacanza è identificata da un codice ed è caratterizzata da città di permanenza, data di partenza, numero di settimane di durata e lingua straniera studiata. In fase di prenotazione, i ragazzi possono esprimere una preferenza per il tipo di alloggio (college o famiglia). In caso di vacanza in college occorre specificare se desiderano avere una camera singola o condivisa, per la vacanza in famiglia si possono specificare il nome e l'indirizzo email di un amico che andrebbe nella stessa famiglia. Ogni vacanza prevede anche una o più gite, per le quali si conoscono la destinazione, il costo, il numero di ore e una descrizione. Dei college si conoscono il nome, l'indirizzo e le attività proposte (nome e descrizione). Per ogni famiglia ospitante si memorizzano i dati del capo famiglia, il numero di componenti il nucleo familiare, presenza di animali domestici, il numero di camere disponibili, il numero di bagni e la distanza dal centro città. Prima della partenza, ogni iscritto riceve conferma del nome del college o della famiglia in cui soggiornerà e del nome dell'eventuale compagno con cui condividerà la stanza nel college o in famiglia. La vacanza può essere pagata con carta di credito o bonifico bancario (e si deve tenere traccia della scelta). Al termine del viaggio ogni ragazzo riceve un certificato che specifica il livello raggiunto (ad esempio B2, C1) nella lingua straniera studiata durante la vacanza e compila un questionario di gradimento sulla vacanza studio. In questionario comprende anche un voto complessivo da 0 a 10 da dare alla vacanza e permette ai ragazzi di specificare un eventuale commento libero. I responsabili dell'agenzia che offre il servizio, devono autenticarsi per poter accedere al sistema e inserire i dati delle vacanze disponibili. Il sistema permette ai responsabili di visualizzare il voto medio che i ragazzi hanno assegnato ad ogni vacanza offerta e i dati raccolti nei relativi questionari. Per i responsabili dell'agenzia si memorizzano i dati anagrafici, l'indirizzo email, il telefono e le credenziali di accesso (login e password). Il sistema deve permettere ai ragazzi iscritti di accedere al loro profilo, modificare i dati anagrafici, visualizzare le vacanze che hanno già fatto, con i relativi certificati ottenuti e questionari compilati. Il sistema deve permettere l'accesso a più ragazzi contemporaneamente

## **Assunzioni**

Per poter svolgere l'elaborato abbiamo dovuto effettuare delle scelte progettuali necessarie per chiarire i diversi flussi di esecuzione.

Le nostre assunzioni sono state:

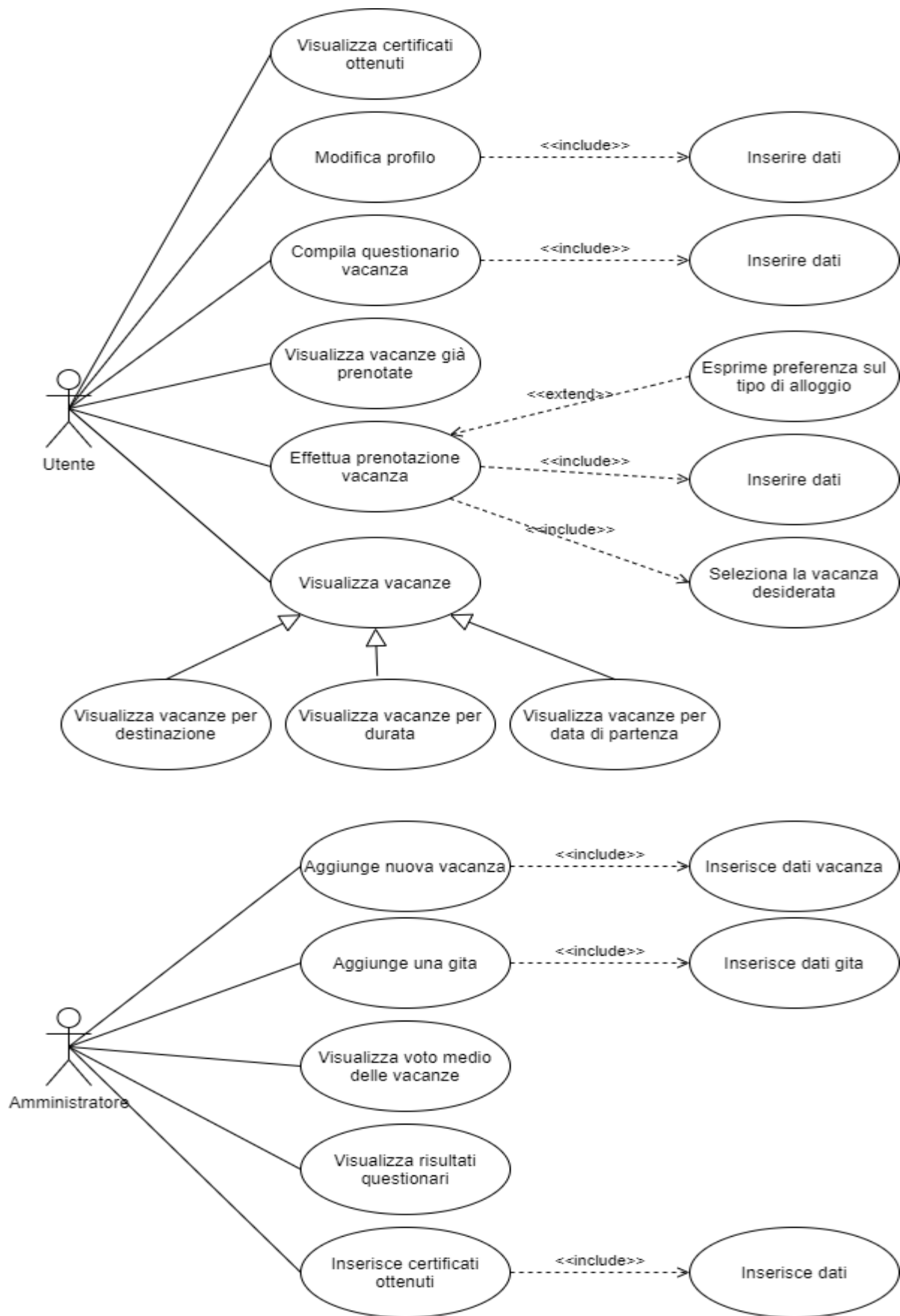
1. Il ragazzo che si iscrive non può inserire autonomamente le allergie ed il loro trattamento, in quanto un semplice errore ortografico nel nome porterebbe alla creazione di un nuovo record nel database, causando l'eventuale presenza di dati duplicati e spreco di spazio
2. La conferma del nome del college o della famiglia avverrà separatamente a mezzo mail, momento nel quale verranno indicate le coordinate per il pagamento a chi ha scelto la modalità bonifico bancario, oppure verrà mandato il link per inserire i dati della carta di credito a chi ha scelto questa modalità
3. I ragazzi non potranno cancellare o modificare la prenotazione tramite applicativo, ma solamente contattando l'agenzia
4. Per compilare il questionario i ragazzi dovranno conoscere il numero di prenotazione, che verrà loro comunicato al termine della vacanza, per evitare che questo venga compilato prima della partenza

## **Interazioni utente-sistema**

### **Specifiche casi d'uso**

#### **Note generali:**

Il sistema proposto supporta l'utilizzo da parte sia dell'utente medio del software, ovvero il ragazzo che vuole prenotare una vacanza, e sia degli amministratori del software che sono responsabili di aggiungere le informazioni delle relative vacanze. Entrambe le categorie hanno delle credenziali, (l'utente se non detiene delle credenziali di accesso può crearle mentre gli amministratori del sistema hanno delle credenziali predefinite) con cui possono effettuare l'autenticazione. Nel caso in cui l'autenticazione va a buon fine, chi ha eseguito l'accesso verrà reindirizzato alla rispettiva schermata iniziale, a seconda del tipo di utente associato alle credenziali.



Use-case diagram: use-case di Utente e Amministratore

### **Casi d'uso relativi all'utente**

Dopo opportuna autenticazione, l'utente viene introdotto ad una interfaccia che permette di scegliere la vacanza da prenotare.

#### **Effettuare prenotazione**

Gli utenti possono effettuare una prenotazione.

In fase di prenotazione è possibile esprimere una preferenza per il tipo di alloggio (college o famiglia).

**Attori:** Utente

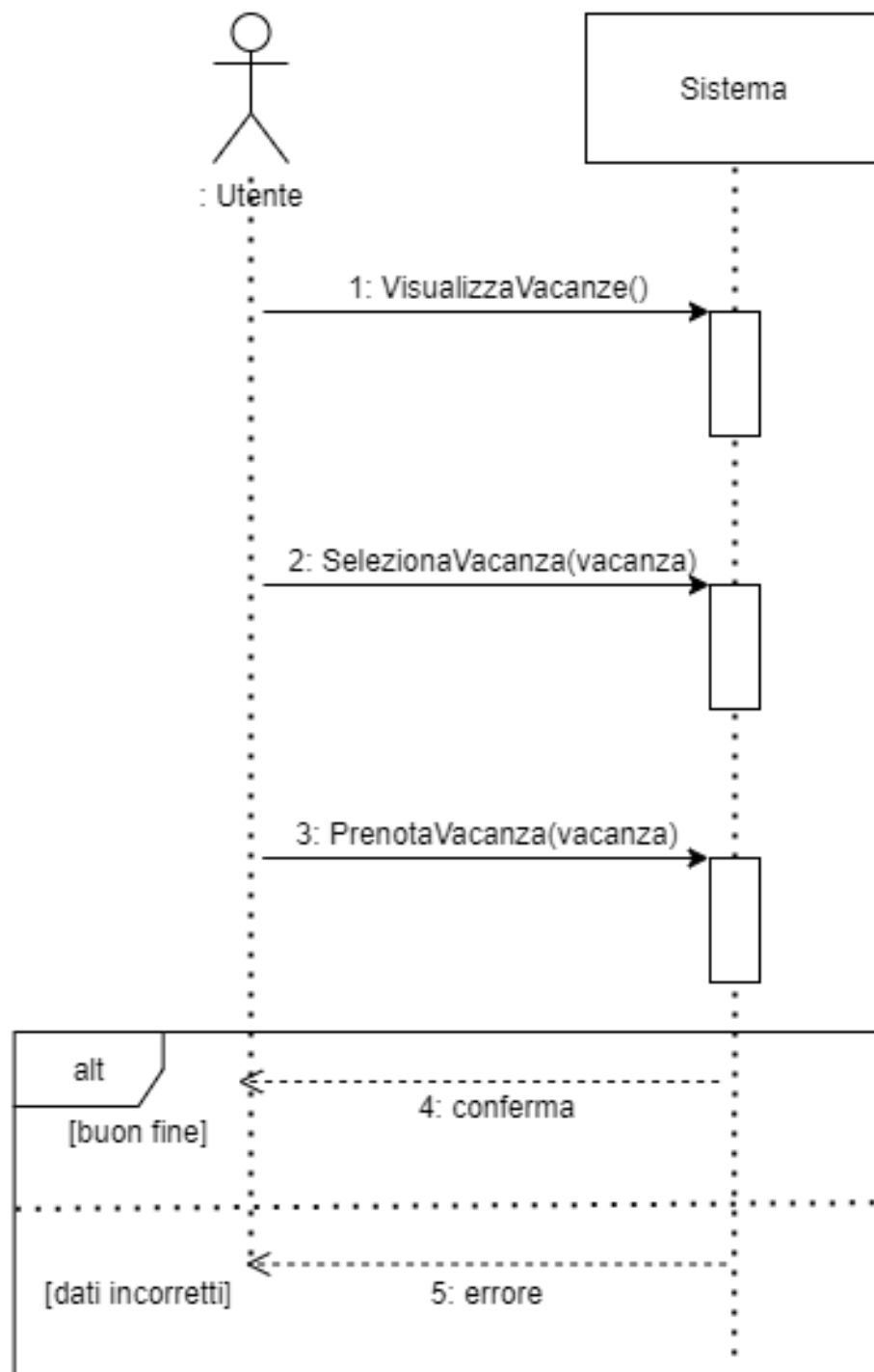
**Precondizioni:** L'utente deve essersi autenticato

**Passi:**

1. L'utente accede al sistema
2. L'utente è introdotto all'interfaccia di base
3. L'utente accede all'interfaccia per la visualizzazione delle vacanze
4.
  - a. L'utente seleziona una preferenza sul tipo di alloggio
  - b. L'utente visualizza le vacanze disponibili
  - c. L'utente seleziona la vacanza tra quelle visualizzate da display
5. L'utente esegue la prenotazione

**Postcondizioni:** la prenotazione è effettuata con successo

interaction generaPrenotazione



Sequence diagram: Fai una prenotazione

## Modificare profilo

L'utente può modificare le proprie credenziali nell'interfaccia di modifica del profilo.

**Attori:** Utente

**Precondizioni:** L'utente deve essersi autenticato

**Passi:**

1. L'utente accede al sistema
2. L'utente è introdotto all'interfaccia di base
3. L'utente accede all'interfaccia per la modifica del profilo
4. L'utente modifica il proprio profilo

**Postcondizioni:** la modifica è effettuata con successo

## Compilare questionario vacanza

L'utente può compilare il questionario vacanza

**Attori:** Utente

**Precondizioni:** L'utente deve essersi autenticato

**Passi:**

1. L'utente accede al sistema
2. L'utente è introdotto all'interfaccia di base
3. L'utente accede all'interfaccia per la compilazione del questionario vacanza
4. L'utente compila il questionario vacanza

**Postcondizioni:** la compilazione è effettuata con successo

## Altre funzionalità del sistema che l'utente può utilizzare

L'utente nell'interfaccia principale può inoltre visualizzare i certificati acquisiti, può visualizzare le vacanze per data di partenza, per durata e per destinazione. L'utente inoltre può visualizzare le vacanze già prenotate.

**Attori:** Utente

**Precondizioni:** L'utente deve essersi autenticato

**Passi:**

1. L'utente accede al sistema
2. L'utente è introdotto all'interfaccia di base
3.
  - a. L'utente può visualizzare i certificati acquisiti
  - b. L'utente può visualizzare le vacanze per data di partenza
  - c. L'utente può visualizzare le vacanze per destinazione
  - d. L'utente può visualizzare le vacanze per durata



e. L'utente può visualizzare le vacanze già prenotate

**Postcondizioni:** nessuna

## **Casi d'uso relativi all'amministratore**

### **Inserimento nuova vacanza**

Il sistema deve fornire un meccanismo di gestione per l'inserimento di una nuova vacanza.

Il sistema, una volta che l'autenticazione è andata a buon fine, permette all'amministratore di inserire i dati relativi alla nuova vacanza da inserire.

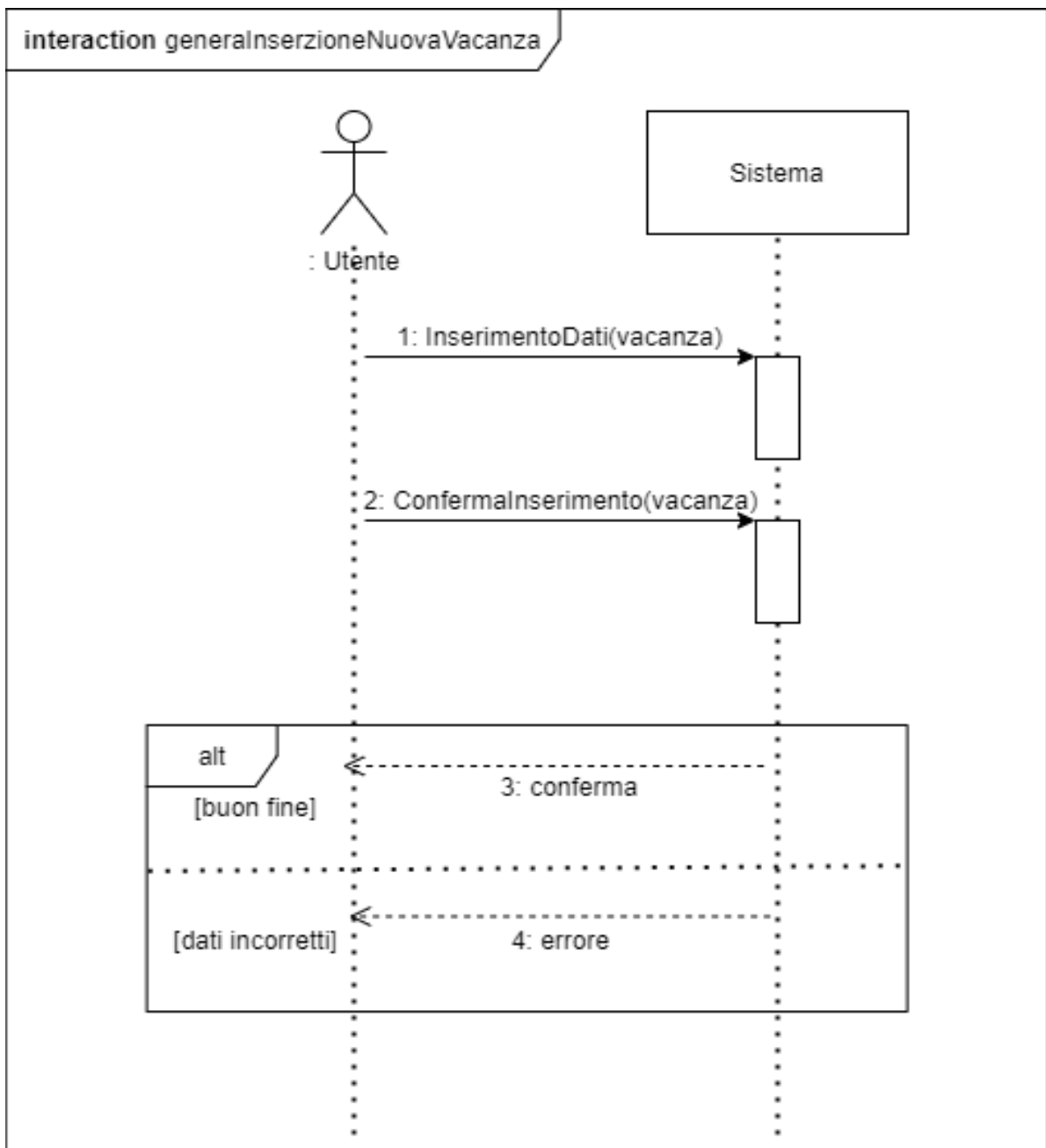
**Attori:** Amministratore

**Precondizioni:** L'amministratore deve essersi autenticato

**Passi:**

1. L'amministratore accede al sistema
2. L'amministratore è introdotto all'interfaccia di base
3. L'amministratore accede all'interfaccia per l'inserimento dei dati della nuova vacanza
4. L'amministratore conferma l'inserimento della nuova vacanza

**Postcondizioni:** Nuova vacanza inserita con successo



Sequence diagram: Inserimento nuova vacanza

## **Inserisci certificati ottenuti**

L'amministratore una volta autenticato può inserire i certificati ottenuti dagli utenti.

**Attori:** Amministratore

**Precondizioni:** L'amministratore deve essersi autenticato

**Passi:**

1. L'amministratore accede al sistema
2. L'amministratore è introdotto all'interfaccia di base
3. L'amministratore accede all'interfaccia per l'inserimento dei certificati ottenuti dagli utenti
4. L'amministratore inserisce il certificato

**Postcondizioni:** Inserimento del certificato avvenuto con successo

## **Altre funzionalità del sistema che l'amministratore può utilizzare**

L'amministratore una volta entrato nell'interfaccia principale può inserire i dati di una nuova gita, visualizzare il voto medio delle vacanze, visualizzare i risultati dei questionari.

**Attori:** Amministratore

**Precondizioni:** L'Amministratore deve essersi autenticato

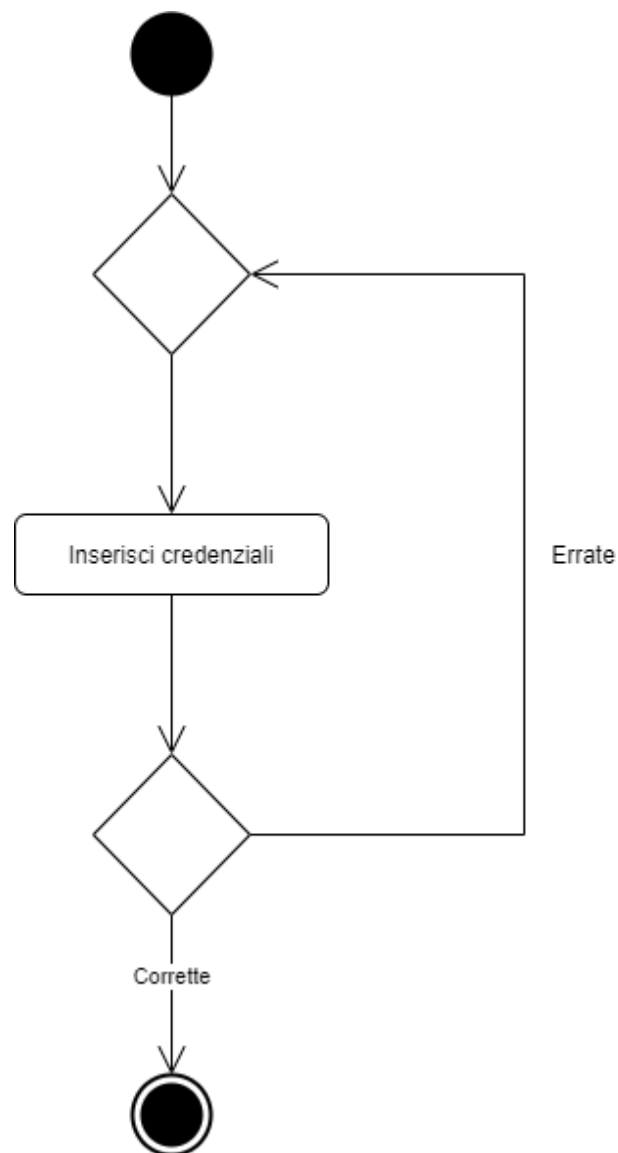
**Passi:**

1. L'Amministratore accede al sistema
2. L'Amministratore è introdotto all'interfaccia di base
3.
  - a. L'Amministratore può visualizzare il voto medio delle vacanze
  - b. L'amministratore può visualizzare i risultati dei questionari

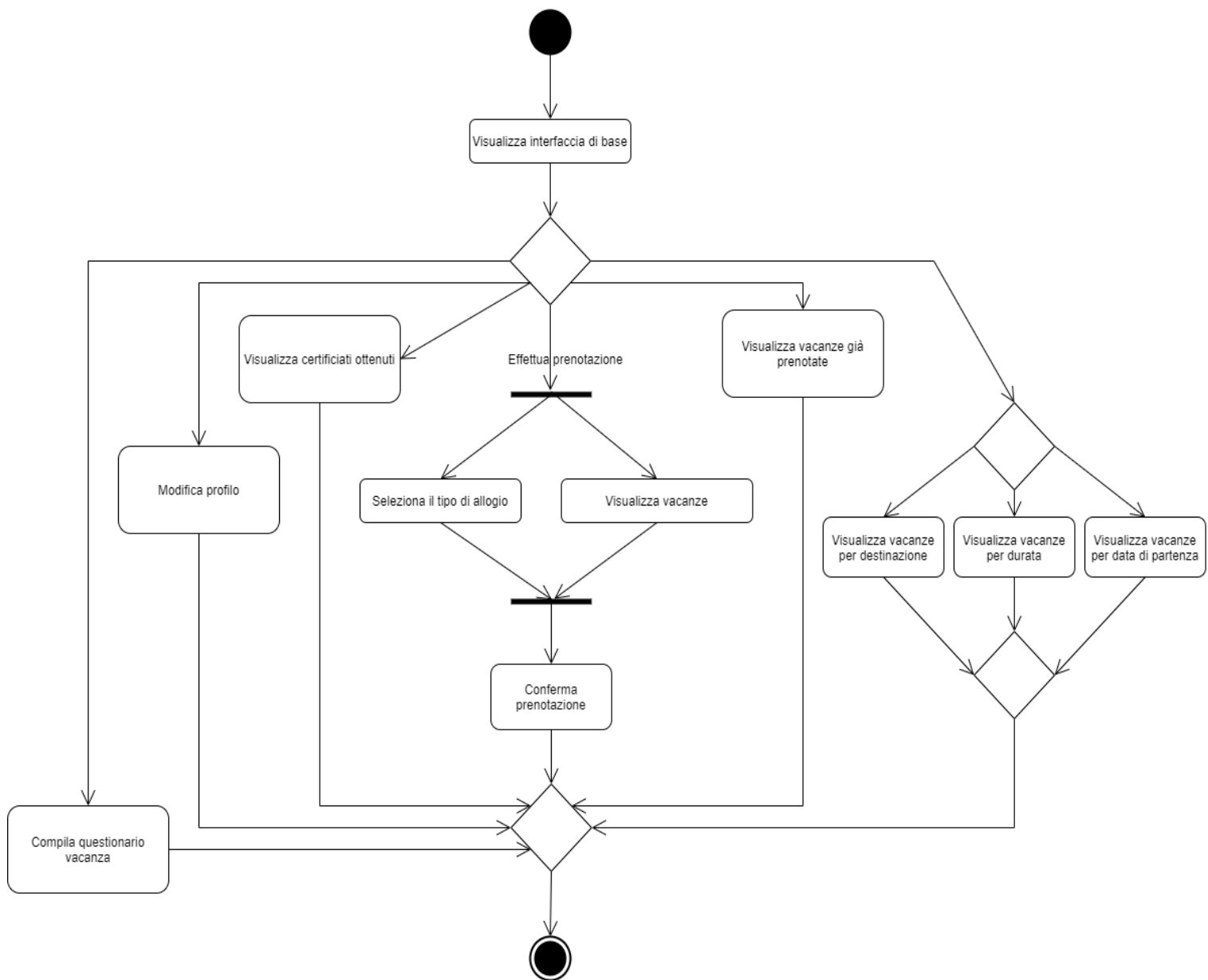
**Postcondizioni:** nessuna

## Diagrammi di attività

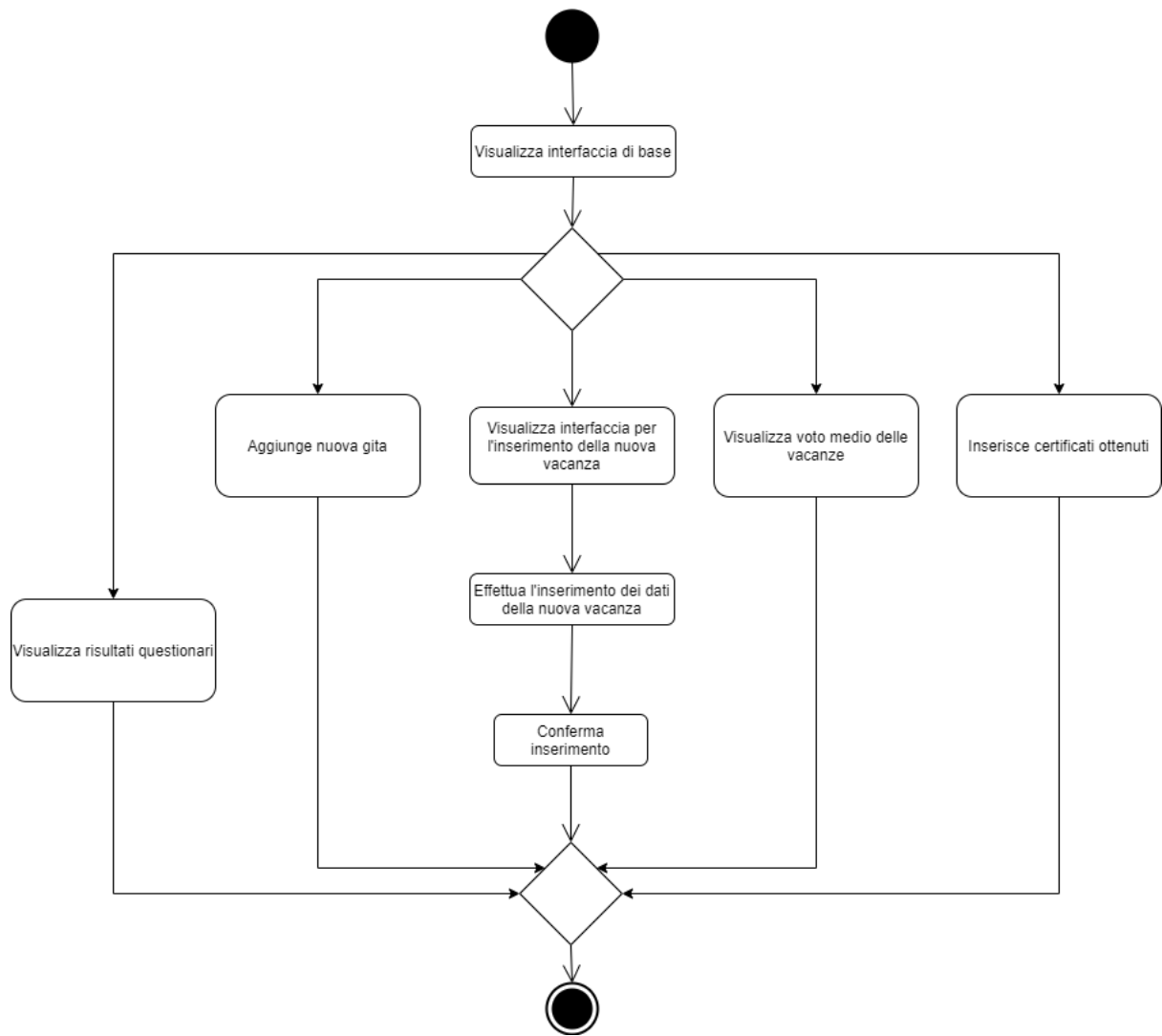
**Nota:** i seguenti diagrammi catturano una singola attività di un utente rispetto al sistema. Non è stata rappresentata nel diagramma la possibilità di ripetere più volte la stessa operazione, in sequenza, senza chiudere il software. Questo per semplici ragioni di chiarezza e leggerezza; si considerano quindi le singole attività d'interazione.



Activity diagram: Autenticazione



Activity diagram: Attività utente



Activity diagram: Attività amministratore

# **Sviluppo: progetto dell'architettura ed implementazione del sistema**

## **Note sul processo di sviluppo**

Il processo di sviluppo del software è stato agile ed incrementale. Il sistema è stato sviluppato come una serie di versioni o incrementi che venivano realizzati quotidianamente per poter ricoprire ogni richiesta di implementazione richiesta.

Nonostante l'approccio incrementale è stata fatta una analisi iniziale delle richieste realizzando una sintesi generale dei requisiti totali del cliente in modo da poter avere una visione del prodotto finale e per poter chiarire i dubbi da parte delle componenti del nostro team.

La realizzazione di molte versioni non ci ha permesso di realizzare una documentazione iniziale bensì solo al termine dello sviluppo siamo stati in grado di realizzare un documento che descrive al meglio ogni aspetto.

Seguendo un approccio agile la specifica, la progettazione, l'implementazione e i test sono interfogliati, ad ogni versione è stato eseguito un serie di test unitari ai singoli output che venivano restituiti dai singoli processi.

L'obiettivo nella realizzazione del software è stato di mantenere la semplicità nello sviluppo e nei processi di sviluppo, cercando anche di realizzare una documentazione chiara e compatta per poter migliorare considerevolmente la comprensione da parte degli stakeholder.

## Progettazione e pattern architetturali usati

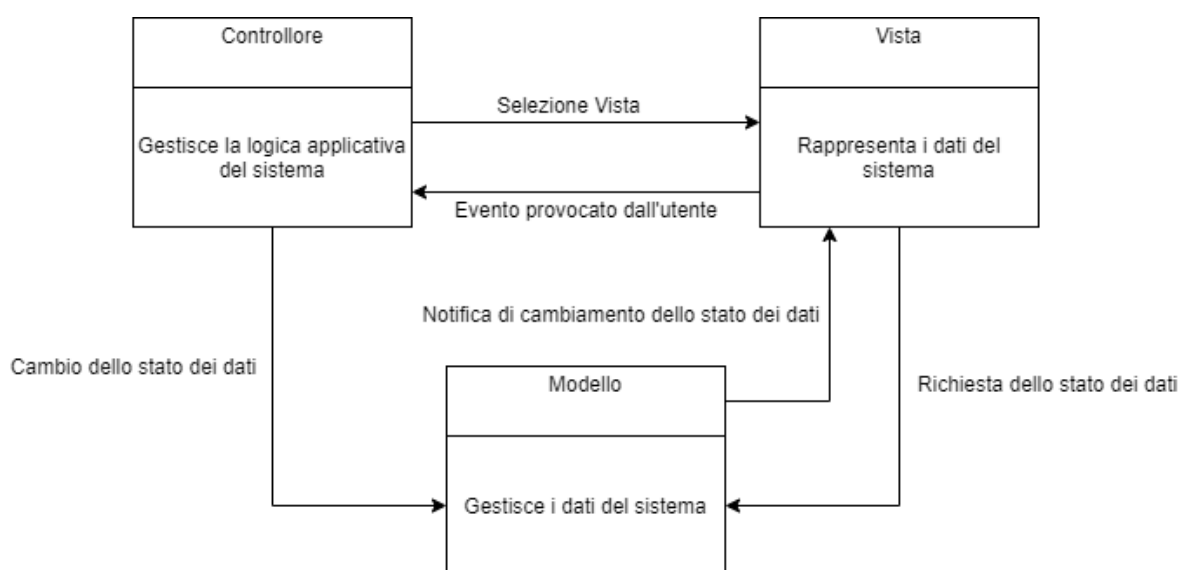
Il sistema è stato progettato utilizzando le tecniche di modellazione ad oggetti.

Dal punto di vista architetturale, si è scelto di utilizzare il pattern MVC.

La motivazione principale per cui è stato utilizzato questo pattern è la possibilità di realizzare sequenzialmente le componenti, questa indipendenza tra le componenti ha garantito la flessibilità nella realizzazione dell'interfaccia d'accesso del sistema dell'utente. In una fase iniziale infatti la grafica dei dati era molto scarna, sufficiente per verificare che la componente di gestione dei dati fosse funzionante, poi in una fase finale il layout della componente View è stato raffinato e reso semplice e chiaro. Ogni componente del sistema dovrebbe essere inserito in un package separato, per evidenziare la separazione tra le 3 componenti, ma non sempre questo è possibile.

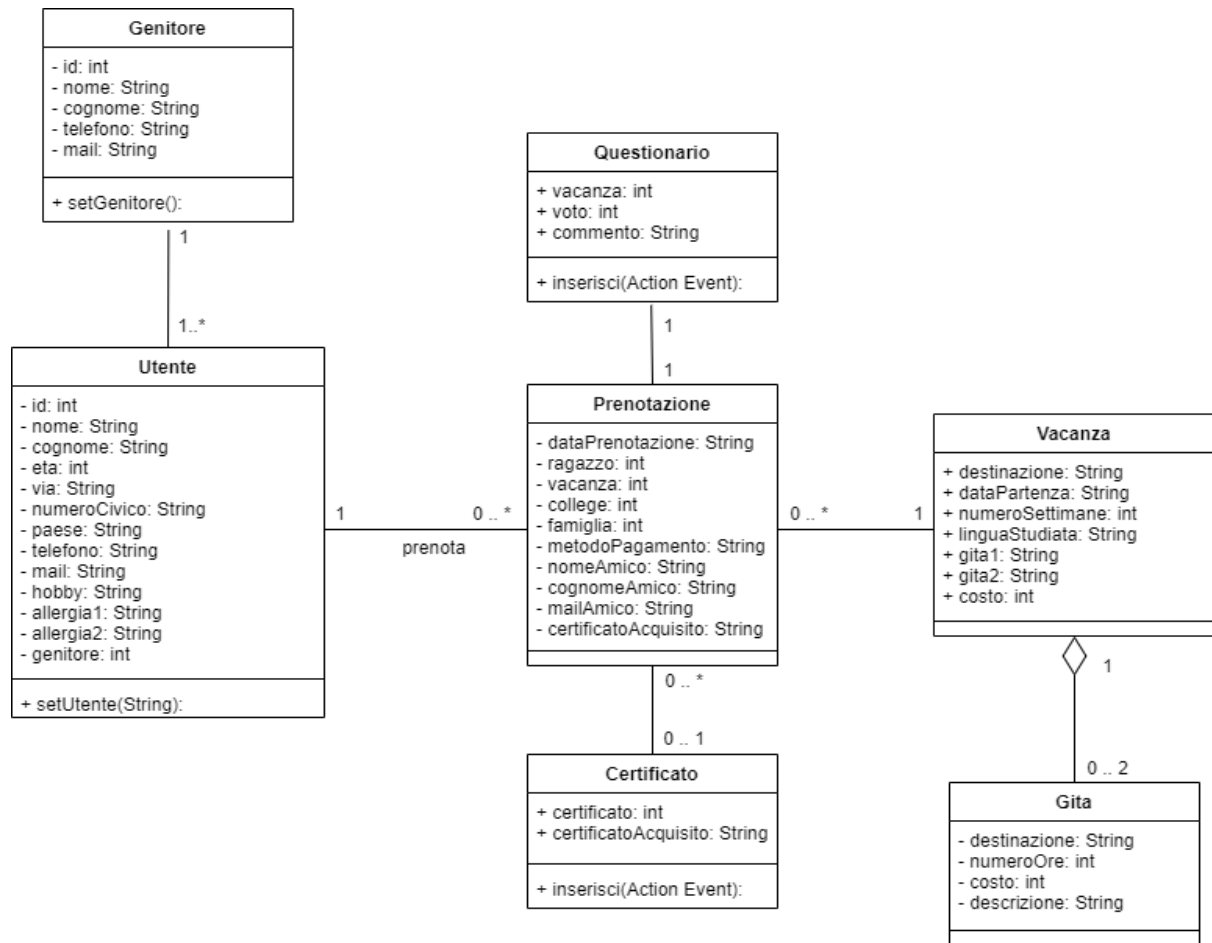
I package solitamente sono :

- **Modello:** sottoparte del sistema che si occupa di gestire i dati del sistema e le operazioni associate ai dati
- **Vista:** sottoparte del sistema che definisce e gestisce come i dati sono rappresentati all'utente
- **Controllore:** sottoparte del sistema che si occupa di gestire le interazioni tra l'utente e i dati del sistema, notificando tali interazioni alle componenti Modello e Vista



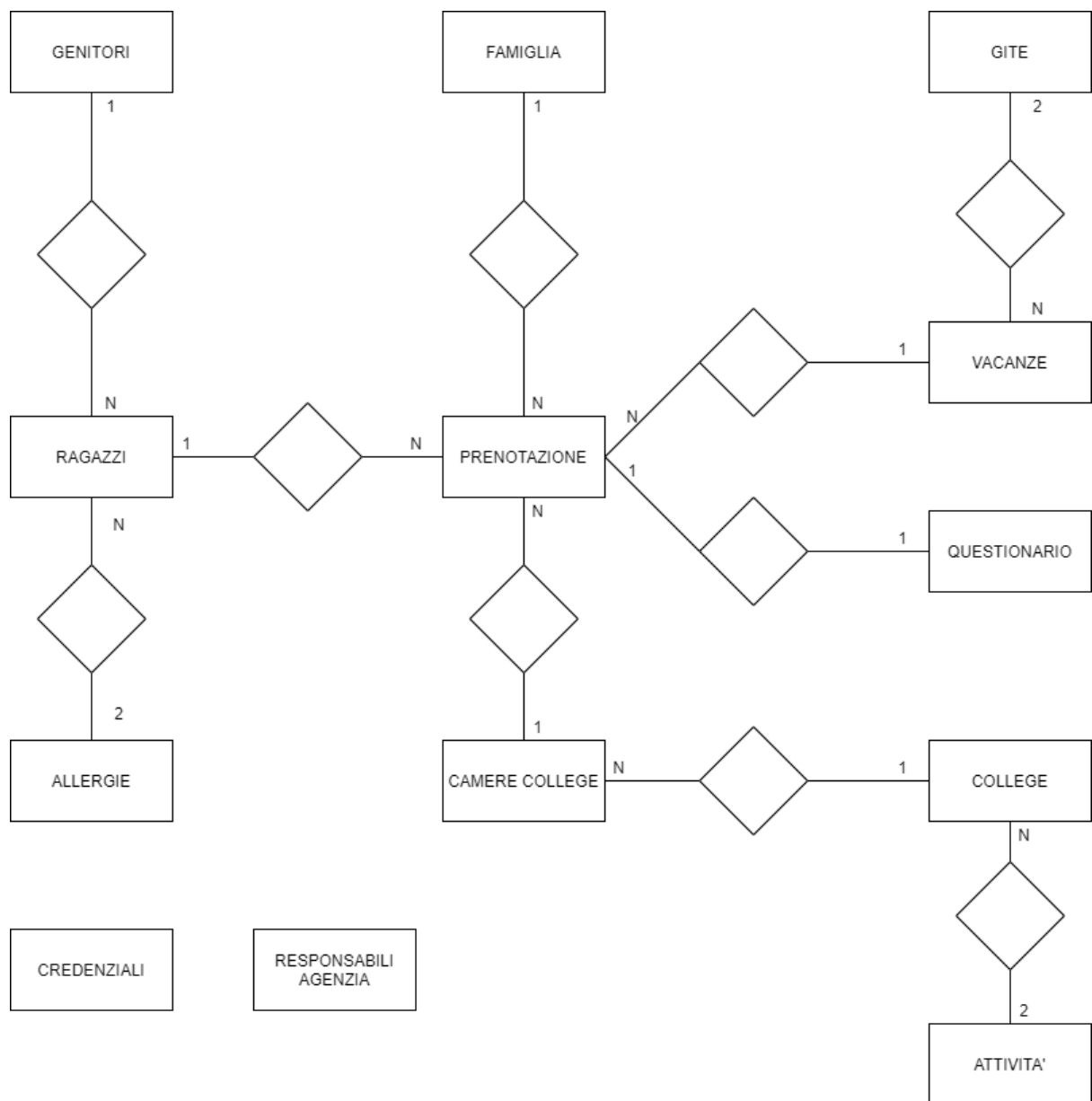
Modello MVC





Class diagram: Classi del modello

## Modellazione e gestione dei dati



Schema ER

## Nota:

Lo schema relazionale, in cui si mettono in evidenza le chiavi primarie, le chiavi esterne e la composizione delle tabelle è il seguente:

Genitori ( id [PK], Nome, Cognome, Telefono, Mail)

Ragazzi ( id [PK], Nome, Cognome, Via, NumeroCivico, Paese, Telefono, Mail, Hobby, Allergia1 [FK], Allergia2 [FK], Genitori [FK])

Allergie (Nome [PK], Trattamento)

Prenotazione ( id [PK], Data, Ragazzo [FK], Vacanza [FK], College [FK], Famiglia [FK], MetodoPagamento, NomeAmico, CognomeAmico, MailAmico, CertificatoAcquisito, Questionario [FK])

Questionario ( id [PK], Voto, Commento)

Vacanze ( id [PK], Destinazione, DataPartenza, NumeroSettimane, LinguaStudiata, Gita1 [FK], Gita2 [FK], Costo)

Gite (Destinazione [PK], Costo, NumeroOre, Descrizione)

Famiglia ( id [PK], Nome, Cognome, Telefono, Mail, PaeseResidenza, Componenti, AnimaliDomestici, NumeroCamere, NumeroBagni, DistanzaDalCentro)

Camere ( id [PK], Tipologia, College [FK])

College (Nome [PK], Città, Indirizzo, Attività1 [FK], Attività2 [FK])

Attività (Nome [PK], Descrizione)

Credenziali (Mail [PK], Password [PK])

Responsabili Agenzia ( Login [PK], Nome, Cognome, Telefono, Password)

Poi per la gestione della concorrenza è stata aggiunta la tabella lock, in cui sono presenti i campi Tabella [PK] e modifica, che ha il compito di simulare la funzionalità di un semaforo per l'accesso alla sezione critica.

La sezione critica nel nostro caso è composta da tutte le istruzioni che comportano la modifica del database, in quanto accessi concorrenti possono portare ad inconsistenza dei dati.

## Implementazione e design pattern usati

In questa sezione si discute l'implementazione e l'applicazione dei vari pattern.

- Collegandosi brevemente al discorso sull'architettura, si è scelto di utilizzare il **pattern singleton** garantendo che una classe possa generare un unico oggetto e che non si possano creare ulteriori istanze della stessa classe, è stato utile nel nostro caso perchè grazie a questo pattern abbiamo potuto, quando necessario, invocare la classe e ottenere i dati relativi in qualsiasi parte del programma senza avere problemi di mantenimento dei dati.
- Per l'accesso al database è stato usato il **pattern Data Access Object (DAO)**, per separare gli accessi al database a basso livello dai servizi offerti all'utente.

## **Attività di test e validazione**

Per verificare la solidità del software , si sono svolte le seguenti attività:

1. Verifiche multiple sul codice ad ogni ciclo di sviluppo
2. Verifica delle implementazioni delle classi e della correttezza dell'implementazione delle richieste
3. Ispezione del codice, verifica della correttezza dei pattern
4. Test degli sviluppatori del software
5. Test utente generico sul software

### **Ispezione codice**

In questa fase è stato rivisto il codice di ogni componente del software e si è cercato di identificare le componenti di codice incorretto, si è inoltre cercato di migliorare l'indentazione del codice per renderlo più leggibile e gestibile nel caso in cui il codice venisse rivisto da altri sviluppatori.

### **Test degli sviluppatori**

In questa fase lo sviluppatore ha immesso nel sistema degli input limite ovvero valori che richiedono l'esecuzione dei controlli da noi implementati nel codice per verificare che il sistema sia consistente e riesca a gestire input errati.

### **Test utente generico**

In questa fase, che ricopre la fase finale dello sviluppo di questo software, è stato fatto testare il codice ad individui che non appartengono al mondo dell'informatica, in modo da avere l'impressione della persona più vicina all'effettivo utilizzatore di questo prodotto una volta distribuito. L'utente è stato libera di utilizzare il software, in modo da verificare che il sistema sia chiaro e intuitivo, grazie a questo passaggio siamo stati in grado di migliorare alcuni aspetti relativi alla fruizione dei dati e anche ad aggiungere delle funzionalità che anche se sottili sono sufficienti a rendere l'utente a proprio agio con il prodotto.