



UNIVERSITÀ DI FIRENZE

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea Triennale in Ingegneria Informatica

Gestionale per un parco dei divertimenti

Autori:
Mathilde Patrissi
Samuele Marrani

Corso:
Ingegneria del Software

ANNO ACCADEMICO 2023/2024

Indice generale

1 Introduzione.....	2
1.1 Descrizione ed obiettivi del progetto.....	2
1.2 Architettura.....	2
2 Analisi e progettazione.....	2
2.1. Diagramma dei casi d'uso.....	2
2.2. Pattern.....	4
2.3 Testing.....	4

1 Introduzione

1.1 Descrizione ed obiettivi del progetto

L'obiettivo del progetto in questione è la realizzazione di un applicativo per la gestione di un parco giochi al fine di automatizzare la gestione dei clienti, dei loro acquisti e dei vari servizi offerti.

Vi è infatti innanzitutto la possibilità di acquistare tipologie diverse di biglietti per l'accesso al parco, ma anche abbonamenti e servizi aggiuntivi come braccialetti salta fila e buoni usufruibili presso i ristoranti e negozi del parco.

Il software permette inoltre la gestione delle attrazioni, degli spettacoli ed eventi. Questi possono essere prenotati, avendo un numero limitato di posti, ma per alcuni l'accesso è vincolato anche dalla tipologia di biglietto/abbonamento.

Ciascuna attrazione ha un orario di apertura e di chiusura ed è presieduta da un dipendente che può segnalare eventuali guasti.

La gestione degli acquisti, ciascuno dei quali ha un proprio costo e una durata temporale, è a carico dei gestori di "AmusementPark".

1.2 Architettura

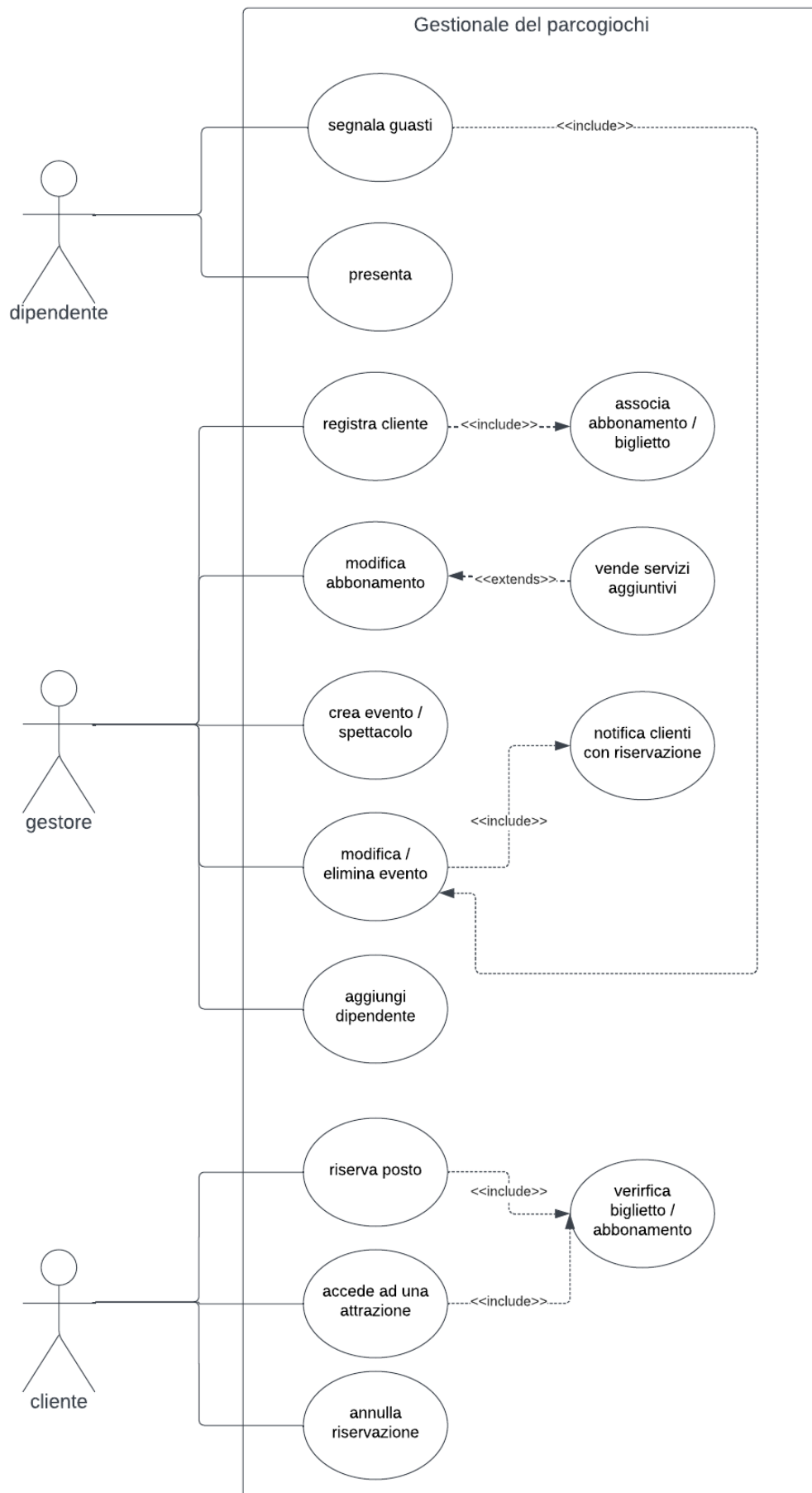
Il software sarà realizzato tramite il linguaggio Java e per garantire la persistenza dei dati verrà utilizzato un database SQLite, la cui connessione avverrà con JDBC.

2 Analisi e progettazione

2.1. Diagramma dei casi d'uso

Nel sistema sono presenti tre attori, ovvero Clienti, Dipendenti ed Gestori, con distinti ruoli. Il seguente diagramma dei casi d'uso mostra gli attori coinvolti e le possibili interazioni fra questi.

Ad esempio: un gestore può vendere i servizi, organizzare nuovi eventi e riservare posti ai clienti, i quali devono essere precedentemente registrati.



2.2. Pattern

Verranno poi utilizzati i seguenti design patterns:

a) Decorator.

Verrà implementata una struttura di “tickets” tramite il pattern decorator.

Questo permetterà la costruzione di un biglietto/abbonamento scegliendo uno o più aggiunte per modificarne le funzionalità base. Sarà possibile anche analizzare le preferenze sugli acquisti dei clienti per notificare eventuali offerte o eventi in programma.

b) Data Access Object (DAO).

Tale pattern che fornirà un'interfaccia per gestire la persistenza dei dati la quale avverrà tramite l'utilizzo di un database SQLite.

Il pattern in questione rende possibile offrire alle varie classi le operazioni CRUD effettuabili sui dati senza conoscere i dettagli implementativi del database.

c) Observer.

Questo design pattern consente la definizione di una dipendenza uno a molti tra oggetti. Ciò implica che se un oggetto cambia il proprio stato, gli oggetti da esso dipendenti vengono notificati e aggiornati automaticamente.

Nel gestionale del parcogiochi un gestore che ha creato un evento ha anche la possibilità di annullarlo per varie motivazioni, tra cui la rilevazione di guasti da parte di un dipendente. Per tanto si mostra necessario notificare i clienti che hanno effettuato una riservazione per l'evento/attrazione in questione. Tale funzionalità sarà implementata tramite il pattern Observer.

2.3 Testing

Saranno poi implementati gli opportuni test al fine di verificare il corretto comportamento del gestionale.