Immagine che contiene testo

Descrizione generata automaticamente

Studenti

*Berardi Martin*

*Bigelli Leonardo*

*Dragne Bogdan Andrei*

*Pecmarkaj Arlind*

*Petrelli Tommaso*

**Progetto**

***Corso di Programmazione e Modellazione ad Oggetti***

*Sessione invernale 2021/2022*

Docente: Montagna Sara

**Indice:**

1. Analisi
   1. Requisiti
   2. Modello del dominio
2. Design
   1. Architettura
   2. Design dettagliato
3. Sviluppo
   1. Testing automatizzato
   2. Metodologia di lavoro
   3. Note di sviluppo
4. **Analisi**

Il progetto consiste nell'implementazione di un gestionale di parcheggi.

* Ogni parcheggio dispone di posti per auto e per moto, con la possibilità di effettuare il noleggio di monopattini elettrici.
* All'ingresso di ogni parcheggio è presente un sensore che rileva l'altezza di ogni auto, in modo tale da permettere l'ingresso al parcheggio solamente alle auto che non superano il limite massimo di altezza. In ogni posto è presente un sensore che, in base all'emissione dei gas di scarico, rileva il tipo di carburante utilizzato dall'auto. Nei parcheggi sotterranei non sarà possibile parcheggiare con macchine a GPL o a metano.
* Ogni proprietario ha la possibilità di fare un abbonamento per il parcheggio. L'abbonamento è strettamente associato ad un solo veicolo.
* Un utente, se dispone di abbonamento per il parcheggio, ha la possibilità di noleggiare un monopattino con una certa tariffa oraria. Si permette l'acquisto di un abbonamento premium che elimina la tariffa oraria di utilizzo del monopattino.
* Se all'uscita dal parcheggio il veicolo non risulta associato a nessun abbonamento, verrà calcolato il costo di utilizzo del parcheggio secondo la tariffa oraria imposta.
* I dati riguardanti il parcheggio vengono salvati in un file, che viene caricato durante l'apertura dell'applicazione e che viene aggiornato alla chiusura di quest'ultimo.
* E' previsto l'utilizzo di un'interfaccia grafica (GUI).

**1.1 Requisiti**

L’applicazione dovrà essere in grado di gestire l’utilizzo di parcheggi per auto e per moto, con la possibilità di effettuare il noleggio di monopattini elettrici, se disponibili.

I dati immessi e successivamente elaborati dovranno essere salvati in un file di testo, che verrà caricato durante l’apertura dell’applicazione e che verrà aggiornato alla chiusura di quest’ultimo.

L’applicazione disporrà di un’interfaccia grafica (GUI) per visualizzare lo stato del programma e per semplificare l’interazione dell’utente verso l’applicazione stessa.



Use Case Diagram

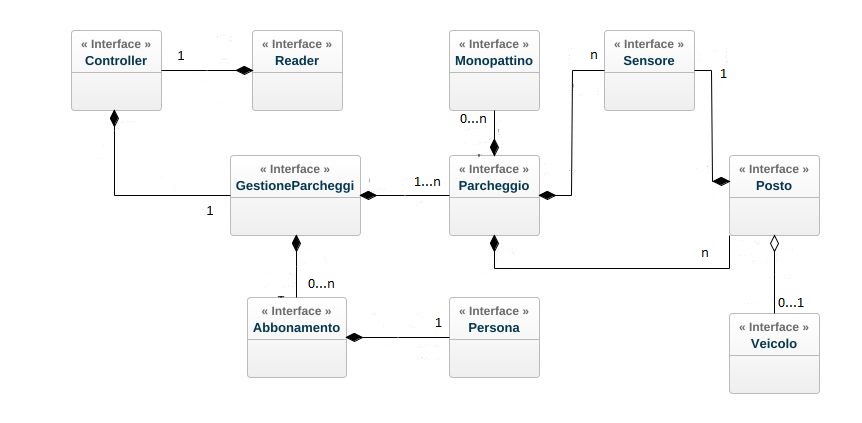
**1.2** **Modello del dominio**

Ogni parcheggio dispone di posti per auto e per moto, con la possibilità di effettuare il noleggio di monopattini elettrici.

All'ingresso di ogni parcheggio è presente un sensore che rileva l'altezza di ogni auto, in modo tale da permettere l'ingresso al parcheggio solamente alle auto che non superano il limite massimo di altezza. Inoltre, in ogni posto è presente un sensore che, in base all'emissione dei gas di scarico, rileva il tipo di carburante utilizzato dall'auto. Nei parcheggi sotterranei non sarà possibile parcheggiare con macchine a metano.

Ogni utente ha la possibilità di effettuare un abbonamento per il parcheggio. L'abbonamento è strettamente associato ad un solo veicolo. Un utente, se dispone di abbonamento per il parcheggio, ha la possibilità di noleggiare un monopattino secondo una certa tariffa oraria. È possibile acquistare un abbonamento premium che elimina la tariffa oraria di utilizzo del monopattino. Se all'uscita dal parcheggio il veicolo uscente non risulta associato a nessun abbonamento, verrà calcolato il costo di utilizzo del parcheggio secondo la tariffa oraria imposta.

L’immagine che segue rappresenta il diagramma delle classi del nostro scenario.



1. **Design**

La classe principale della struttura è Parcheggio. In essa vengono istanziati i vari posti per le auto e quelli per le moto, quest’ultimi definiti tramite delle classi apposite, PostoAuto e PostoMoto, esse sono realizzate partendo da una classe astratta utilizzata come template (Posto). La scelta di utilizzare uno scheletro di questo tipo è data dal fatto che non ha senso istanziare un posto generico, in quanto i posti potranno essere per le auto o per le moto, che avranno proprietà diverse ma anche qualche metodo in comune (motivazione principale dell’utilizzo di una classe astratta più generica).

Il parcheggio è identificato da un codice alfanumerico univoco (id). Una caratteristica fondamentale è data dal primo carattere di questo codice:

* ‘S’ 🡪 Identifica che il parcheggio sarà sotterraneo con conseguenza che le auto a metano e a GPL non potranno sostarsi;
* ‘A’ 🡪 Identifica un generico parcheggio all’aperto.

Per rappresentare tutti i veicoli in generale è stata creata una classe Veicolo. La creazione delle Auto e delle Moto è data dall’estensione (extends) della classe Veicolo. Per rappresentare tutti i possibili tipi di carburante è stato scelto di utilizzare un tipo enum Alimentazione, in modo tale da avere accesso esplicito al tipo di carburante utilizzato dal veicolo.

I sensori sono stati implementati grazie all’utilizzo di un’interfaccia parametrizzata. Le classi dei sensori per il rilevamento dell’altezza dell’auto e per l’emissione del carburante implementano entrambe l’interfaccia parametrizzata. Essa ha un metodo parametrizzato che ritorna il tipo double o Alimentazione, a seconda che ad usarla sia un sensore per l’altezza o per l’emissione del carburante.