

Scopo del documento

Il presente documento riporta la definizione dell'architettura del progetto BookEasy - Antico Granaio usando diagrammi delle classi in UML ("Unified Modeling Language") e codice in OCL ("Object Constraint Language"). Nel precedente documento è stato presentato il diagramma degli use case, il diagramma di contesto e quello dei componenti. Ora, tenendo conto di questa progettazione, viene definita l'architettura del sistema dettagliando da un lato le classi che dovranno essere implementate a livello di codice e dall'altro la logica che regola il comportamento del software. Le classi vengono rappresentate tramite un digramma delle classi in linguaggio UML. La logica viene descritta in OCL perché tali concetti non sono esprimibili in nessun altro modo formale nel contesto di UML.

1. Diagramma delle classi

Nel presente capitolo vengono presentate le classi previste nell'ambito del progetto BookEasy - Antico Granaio. Ogni componente presente nel diagramma dei componenti diventa una o più classi. Tutte le classi individuate sono caratterizzate da un nome, una lista di attributi che identificano i dati gestiti dalla classe e una lista di metodi che definiscono le operazioni previste all'interno della classe. Ogni classe può essere anche associata ad altre classi e, tramite questa associazione, è possibile fornire informazioni su come le classi si relazionano tra loro.

Riportiamo di seguito le classi individuate a partire dai diagrammi di contesto e dei componenti. In questo processo si è proceduto anche nel massimizzare la coesione e minimizzare l'accoppiamento tra classi.

1.1. Utenti e sistemi esterni

Analizzando il diagramma di contesto realizzato per il progetto BookEasy - Antico Granaio si nota la presenza di **due attori "user" e "ristoratore"**.

Lo "user" è colui che utilizza l'applicazione per prenotare un tavolo del ristorante e il "ristoratore" è chi gestisce il ristorante. Entrambi questi attori hanno specifiche funzioni e attributi ma hanno anche molto in comune. Sono state quindi individuate due classi `user` e `ristoratore` con funzioni e attributi specifici e una classe `utente` con funzioni e attributi in comune collegate tramite una generalizzazione.

Nella classe `user` sono state aggiunte anche le operazioni verso il **sistema esterno QR code** anch'esso presente nel diagramma di contesto.

IMMAGINE

1.2. Gestione autenticazione

Il diagramma di contesto analizzato presenta un ultimo **sistema subordinato** denominato **"sistema credenziali universitarie"**. Questo elemento rappresenta il meccanismo di autenticazione degli utenti attraverso un sistema esterno di gestione delle credenziali universitarie. È stata identificata una classe `autenticazione`, che si interfacerà con questo sistema di gestione delle credenziali di ateneo. Il software sviluppato per il progetto

ACCESS light non gestirà e non memorizzerà username e password inserite dall'utente ma, tramite questa classe, passerà solo i dati di autenticazione ad un sistema paritario esterno, il sistema credenziali universitarie, che valuterà questi dati e risponderà specificando se le credenziali inserite sono valide o meno. Di seguito il dettaglio di questa classe con i propri attributi e metodi.

IMMAGINE

1.3. Pagina Check-In Edificio

Dopo aver riportato tutti gli elementi del diagramma di contesto nel diagramma delle classi, facciamo la stessa operazione con il diagramma dei componenti. Analizzando il componente **“Pagina Check-In Edificio”** sono state identificate due classi per gestire la procedura che l'utente segue quanto entra in un edificio (inquadrare il codice QR in portineria e compilare la form sullo stato di salute). Di seguito il dettaglio di queste classi con i propri attributi e metodi.

IMMAGINE

1.4. Pagina gestione autorizzazioni utente

Analizzando il componente **“Pagina gestione autorizzazioni utente”** che prevede la richiesta di autorizzazione per l'accesso agli edifici da parte degli utenti e la verifica dello stato delle autorizzazioni per le proprie richieste. Definiamo le classi che permettono all'utente la creazione di nuove richieste di accesso e la visualizzazione del loro stato con tutti gli attributi necessari.

IMMAGINE

1.5. Gestione responsabili edificio

Il componente **“Gestione responsabili edificio”** prevede che il responsabile di edificio possa autorizzare gli accessi degli utenti, confermando o meno la richiesta di accesso degli utenti.

Definiamo ora le classi che permettono queste funzionalità riservate ai responsabili di edificio. Da notare che, in questo caso, vista la complessità dei dati da gestire per ogni stanza, sono state previste 3 classi opportunamente relazionate tra loro.

IMMAGINE

1.6. [... specificare una o più classi per tutti gli altri componenti]

...

1.7. Diagramma delle classi complessivo

Riportiamo di seguito il diagramma delle classi con tutte le classi fino ad ora presentate. Oltre alle classi già descritte, sono state inserite classi ausiliarie, ad esempio `tipo_data`. Queste classi servono per descrivere eventuali tipi strutturati usati, ad esempio, negli attributi delle altre classi.

IMMAGINE

2. Codice in Object Constraint Language

In questo capitolo è descritta in modo formale la logica prevista nell'ambito di alcune operazioni di alcune classi. Tale logica viene descritta in Object Constraint Language (OCL) perché tali concetti non sono esprimibili in nessun altro modo formale nel contesto di UML.

2.1. Compilazione stato salute

La form per compilare lo stato di salute è accessibile solo se un'utente è autorizzato ad accedere, il giorno in cui è autorizzato ad eccedere. Questa condizione su queste classi:

IMMAGINE

è espressa in OCL attraverso una preconditione con questo codice:

2.2. Conferma entrata edificio

Nella classe `user` appena riportata è presente anche il metodo `conferma_entrata_edificio`. L'esecuzione di questo metodo comporta, tra l'altro, l'inizializzazione del valore dell'attributo `presenza_in_edificio` a `true`. Questa condizione è espressa in OCL attraverso una postcondizione con questo codice:

2.3. Stato contatori postazioni autorizzate

Ci sono due condizioni che devono essere sempre verificate per ogni stanza: il numero di postazioni autorizzate deve essere minore o uguale al numero di postazioni previste per l'emergenza COVID e il numero di postazioni attualmente occupate deve essere minore o uguale al numero di postazioni autorizzate. Queste due condizioni fanno riferimento a questa classe:

IMMAGINE

e sono espresse in OCL attraverso un'invariante con questo codice:

2.4. [... specificare altri concetti in OCL per le altre operazioni]

...

3. Diagramma delle classi con codice OCL

Riportiamo infine il diagramma delle classi con tutte le classi fino ad ora presentate ed il codice OCL individuato.

IMMAGINE

Alessandro Busola -> 21/11/2023 -> 10.55-11.50 -> copia incolla iniziale + inizio modifiche + inizio lucichart