

# [I.T.I.S. P. Paleocapa](https://www.itispaleocapa.edu.it)

# CORSO DI ISTRUZIONE SECONDARIA SUPERIORE AD INDIRIZZO INFORMATICO

**APPLICAZIONE GESTIONE ALBERGO**

**(UN PROTOTIPO)**

CoordinatoreCandidato ***Prof. Renato Cortinovis*** ***Melnic Filip***

Anno Scolastico 2020-2021

**INDICE**

**ABSTRACT**

Questa tesina ha l’obbiettivo di presentare il prodotto, o in questo caso specifico, il sistema e i metodi utilizzati per la sua realizzazione. Si tratta di un sistema informatico sotto forma di gestionale alberghiero, in grado di gestire le prenotazioni dei clienti, gli accessi alla struttura e alle stanze e il check-in/check-out. Il sistema implementerà delle funzionalità che rendono tutte le procedure dei clienti, dalla prenotazione al check-out, completamente autonome. Per implementare queste funzionalità ho utilizzato degli strumenti informatici di ultima generazione ampiamente utilizzati dalla maggioranza dei software developer odierni. Per quanto riguarda un’ipotetica commercializzazione del prodotto sul mercato, esso dovrà essere presentato sotto forma di un pacchetto “all-in-one”, che comprenderà sia l’istallazione del sistema elettronico che l’implementazione del front-end e back-end in base alle necessità del cliente/compratore. L’obiettivo implicito del progetto è di “dipingere un quadro” sull’inevitabilità di un futuro caratterizzato dall’automazione a tutti i livelli commerciali.

**INTRODUZIONE**

Vorrei partire facendo un salto indietro nella storia.

Ci troviamo in Giappone,705 CE. Siamo dei pellegrini ed è da tanto che non abbiamo fatto una pausa, in più sta per piovere e si sta facendo notte, cosa facciamo? Se solo qualcuno avesse inventato l’hotel…

-Niente problemi! Nishiyama Onsen Keiunkan a vostra disposizione! Come possiamo aiutarvi?

Nishiyama Onsen Keiunkan è stato riconosciuto ufficialmente da Guinness World Records come l’hotel più antico al mondo. Ma in Europa a livello di massa, l’hotel come lo conosciamo oggi divenne popolare soltanto agli inizi del XIX-imo secolo in seguito alla nascita della società di massa dopo la seconda rivoluzione industriale. L’hotel veniva gestito da un personale che era responsabile di tutti i processi di management, dalla prenotazione dei clienti al loro check-out. Da allora sono cambiate tante cose, ma generalmente il modo in cui l’hotel va gestito è rimasto lo stesso. A mio avviso questo deve e sicuramente cambierà, soprattutto con l’avvento dell’automazione in seguito alla rivoluzione digitale.

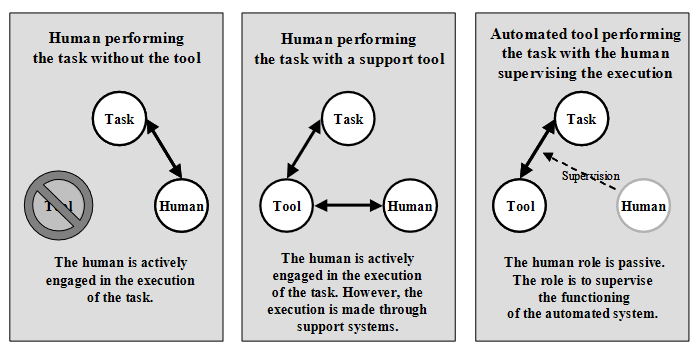
Di seguito mostrerò un possibile sistema alberghiero futuristico che inevitabilmente sostituirà il modello per eccellenza di oggi. Un modello frutto dell’avvento dell’era digitale che è stato possibile implementare grazie ai traguardi informatici raggiunti nell’ultimo decennio. Si tratta di un sistema informatico nell’ambito alberghiero che sfrutta le potenzialità messe a disposizione dal mondo digitale, più specifico un modello che rende la gestione di un hotel quasi del tutto automatizzata.

Vediamo un attimo i possibili vantaggi di un sistema analogo per l’amministrazione del hotel:

* La riduzione del carico di lavoro
* Con i sistemi automatizzati, le risorse attentive dell’operatore possono essere assegnate ad altre attività eseguite contemporaneamente
* La riduzione del fattore di stress indotto dalla posta in gioco della situazione.
* La riduzione del fattore di fatica.
* I sistemi automatizzati forniscono un certo livello di stabilità nell’esecuzione di un’attività
* I sistemi automatizzati eliminano gli errori umani

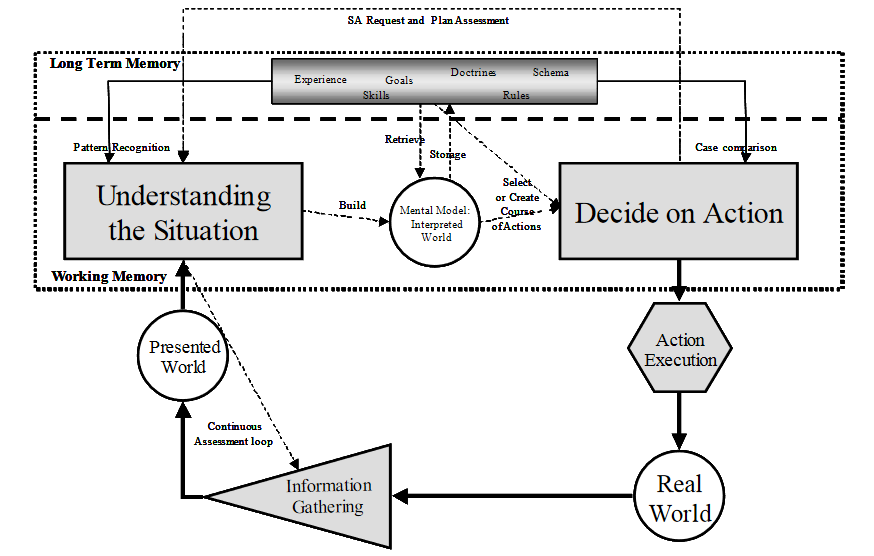
Diamo uno sguardo più approfondito:

**L’interazione tra i sistemi umani e automatizzati/di supporto:**



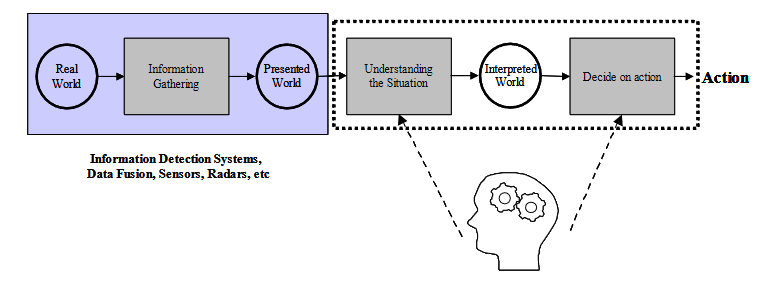
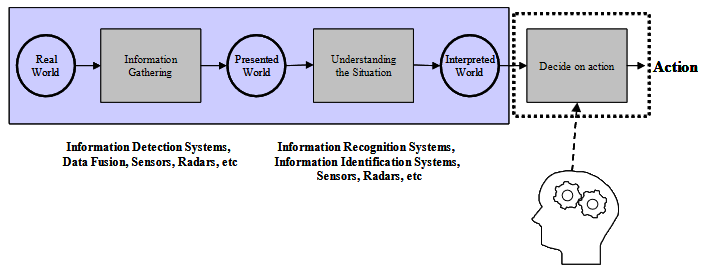
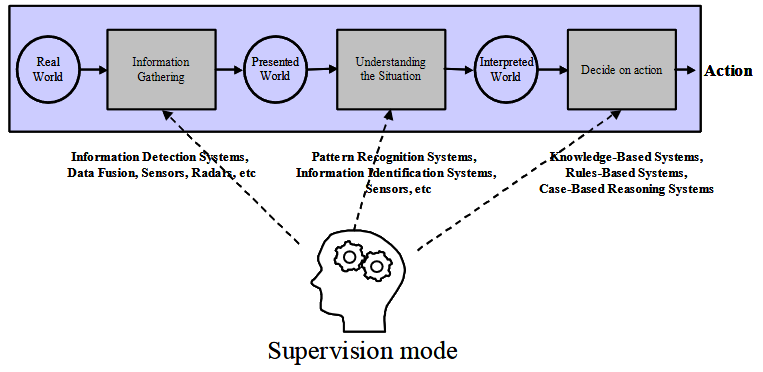
**La scomposizione del processo decisionale:**

***Il modello del processo decisionale proposto da Keus e Breton***



**L’automazione dei processi decisionali**

***1.L’automazione del processo di raccolta delle informazioni***

***2.L’automazione della raccolta di informazioni e della comprensione dei compiti di situazione**3.L’automazione di tutti i processi che portano all’attuazione dell’azione***

*Illustrazioni tratte dall’articolo academico di Dr Richard Breton e Dr Eloi Bossè “The role of Humans in Intelligent and Automated Systems” pubblicato il 9 Ottobre 2002*

**In Conclusione:**

Massimizzando l’automazione, l’amministrazione avrà soltanto il ruolo di supervisionare il sistema ed è questo il risulatato che vorrei ottenere con questo progetto. Limitare al massimo l’implicazione dell’essere umano vuol dire anche ridurre al massimo il fattore dell’errore umano.Vano anche considerate le implicazioni di un sistema del genere sul comfort e la convenienza di avere tutto alla portata di un click per il cliente. In fine rimane anche da dire che dall’altra parte , l’automazione rende l’adattamento a situazioni impreviste e guasti di sistema molto difficile, quindi il ruolo del supervisore rimane imperativo per la loro risoluzione.

**Considerazioni(prefazione) preliminari:** Il sistema che ho implementato io è soltanto un prototipo, un primo esemplare di un sistema con infinte possibilità di applicazione. Bisogna anche considerare l’arco di tempo limitato che ho avuto a disposizione per la realizzazione di questo progetto, oltre alle nuove conoscenze che ho dovuto acquisire per adattare tale sistema ai continui mutamenti dell’ambiente operativo e del dominio applicativo. Di conseguenza l’effettivo sistema che ho creato è una versione “in miniatura” di un modello ideale che richiederebbe molto più tempo, più considerazioni e delle conoscenze ulteriormente approfondite dell’ambiente operativo.

**Vincoli/Dominio Progetto**

Visto le considerazioni preliminari precedenti ho deciso di impostare dei vincoli/un dominio per l’esistenza del sistema creato.

**Hotel:**

* L’hotel mette a disposizione soltanto 7 camere;
* Ogni camera è identificata da un nome che corrisponde a un colore;
* Non viene offerto alcun tipo di pasto.

**Front-End:**

* Il sito sarà minimalistico da un punto di vista funzionale. Serve soltanto a illustrare il processo semplificato di una richiesta di prenotazione.
* Per la prenotazione il cliente non deve registrarsi/autenticarsi.
* La prenotazione non può essere cancellata una volta fatta e visto l’assenza di un personale presso la struttura va pagata in anticipo al momento della prenotazione stessa.
* Come sostituto del processo di check-in, l’utente inserisce il numero del suo passaporto.
* Un cliente può prenotare soltanto una stanza alla volta.

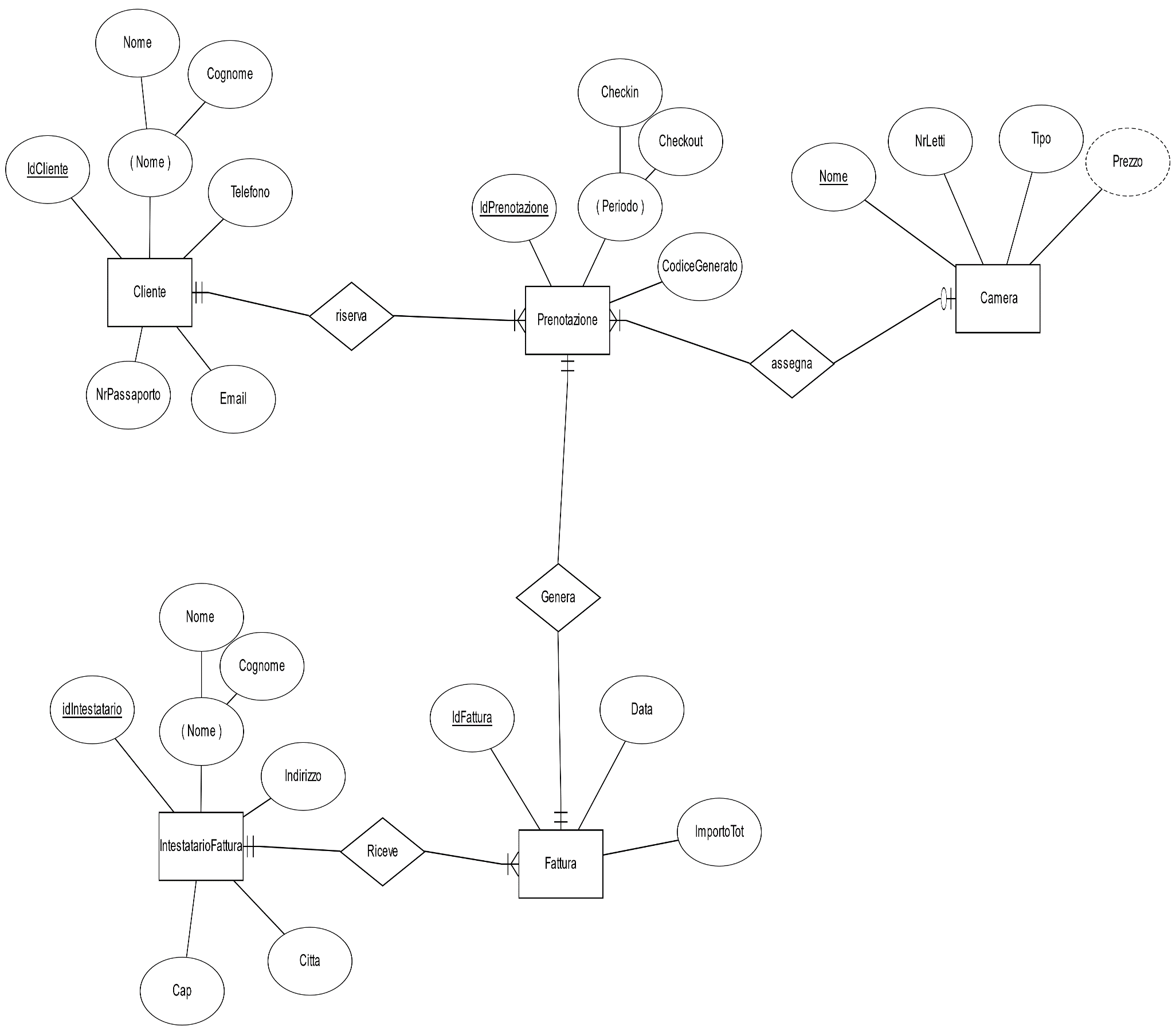
**Back-End:**

* Il database è anch’esso minimalistico. Serve a illustrare un modello semplificato per la gestione delle prenotazioni.
* Il database non è normalizzato visto la semplicità del dominio.

(Il campo IntestatarioFattura.Indirizzo non è atomico, dipendenze transitive : Camera.prezzo -> Camera.Tipo ; Citta,indirizzo -> Cap).

* Il database non registra il numero della carta di credito crittografato come garanzia visto il pagamento in anticipo.
* Non ho inclusa alcuna API del settore alberghiero (Booking.com API, PayPal API etc.)

Tutti questi vincoli potrebbero essere sollevati senza particolari difficoltà, ma come già detto, sono necessari visto il tempo stretto che ho avuto a disposizione.

**Schema Concettuale**

**Schema Logico**

* Prenotazione (**IdPrenotazione**, Checkin, Checkout, CodiceGenerato, IdClienteFK, IdCameraFK)
* Cliente (**IdCliente**, Nome, Cognome, Telefono, Email, NrPassaporto)
* Camera (**Nome**, NrLetti, Tipo, Prezzo)
* Fattura (**IdFattura**, Data, ImportoTot, IdPrenotazioneFK, IdIntestatarioFK)
* IntestatarioFattura (**IdIntestatario**, Nome, Cognome, Indirizzo, Citta, Cap)

**Analisi Progettazione Database**

**Entità:**

* Cliente – contiene tutte le informazioni base del cliente che effettua la prenotazione;
* Camera – contiene le informazioni relative alle stanze che l’hotel mette a disposizione;
* Fattura – memorizza le fatture ed il prezzo totale dei soggiorni che vengono prenotati;
* IntestatarioFattura – memorizza i dati relativi all’intestatario della fattura, ovvero la persona intestataria della carta utilizzata come metodo di pagamento.
* Prenotazione – memorizza i dati relativi al periodo di soggiorno dei clienti, quindi registra tutti i soggiorni prenotati;

**Attributi:**

**Cliente**:

* IdCliente – identifica univocamente ogni cliente che effettua una prenotazione;
* Nome – Il nome del cliente che effettua la prenotazione;
* Cognome – Il cognome del cliente che effettua la prenotazione;
* Telefono – Il numero di telefono che effettua la prenotazione;
* Email – Identifica l’indirizzo mail del cliente che effettua la prenotazione;
* NrPassaporto – Identifica il numero di passaporto del cliente che effettua la prenotazione. Questo campo funge da sostituto per il meccanismo di check-in;

**Camera**:

* Nome – identifica univocamente ogni stanza dell’albergo;
* NrLetti – identifica il numero di letti che ogni stanza mette a disposizione;
* Tipo – identifica la tipologia di stanza. Es. Doppia, Singola etc.;
* Prezzo – identifica il prezzo/notte di ciascuna stanza.

**Prenotazione**

* IdPrenotazione – identifica univocamente ciascuna prenotazione effettuata, presente, passato e futuro;
* Checkin – identifica la data di arrivo del cliente relativa al soggiorno del cliente;
* Checkout - identifica la data di partenza del cliente relativa al soggiorno del cliente;
* CodiceGenerato – identifica il codice generato per ciascuna prenotazione.Il codice è operativo soltanto all’interno del periodo di soggiorno del cliente;

**Fattura**

* IdFattura – identifica univocamente ogni fattura relativa alla prenotazione del cliente;
* Data – identifica la data di generazione della fattura;
* ImportoTot – identifica il costo totale del soggiorno relativo alla prenotazione del cliente;

**IntestatarioFattura**

* IdIntestatario – identifica univocamente ciascun intestatario della fattura, in questo caso specifico identifica il titolare della carta di credito utilizzata come metodo di pagamento;
* Nome – identifica il nome dell’intestatario della fattura;
* Cognome – identifica il cognome dell’intestatario della fattura;
* Indirizzo – identifica l’indirizzo dell’intestatario che serve come indirizzo di fatturazione;
* Citta – identifica la citta dell’intestatario che serve alla fatturazione;
* Cap – identifica il cap dell’intestatario che serve alla fatturazione;

**Associazioni:**

**Cliente-Prenotazione**

* Tra l’entità Cliente e l’entità Prenotazione esiste un’associazione riserva di cardinalità uno a molti (1:N);

Di fatto l’associazione diretta è univoca poiché una prenotazione può essere prenotata soltanto da un cliente, mentre l’associazione inversa è multipla visto che un cliente può prenotare tante prenotazioni.

L’associazione diretta è totale poiché affinché un cliente venga registrato nel database, il cliente deve necessariamente riservare una prenotazione;

L’associazione inversa è totale poiché una prenotazione per esistere deve essere necessariamente riservata da un cliente;

**Prenotazione-Camera**

* Tra l’entità Prenotazione e Camera esiste un’associazione assegna di cardinalità uno a uno (1:N);

Di fatto l’associazione diretta è univoca poiché una prenotazione può assegnare al cliente soltanto una stanza, mentre l’associazione inversa è multipla visto che a una prenotazione possono essere assegnate più stanze;

L’associazione diretta è totale poiché una prenotazione deve avere assegnata una camera al momento della prenotazione;

L’associazione inversa è parziale poiché una camera per esistere nel database non deve essere necessariamente assegnata a una prenotazione;

**Prenotazione-Fattura**

* Tra l’entità Prenotazione e Fattura esiste un’associazione genera di cardinalità uno a uno (1:1);

Di fatto l’associazione diretta è univoca poiché una prenotazione può generare soltanto una fattura. L’associazione inversa è univoca anch’essa poiché una fattura può essere associata soltanto a una prenotazione;

L’associazione diretta è totale poiché una prenotazione deve generare una fattura al momento della prenotazione;

L’associazione inversa è anch’essa totale poiché una fattura deve essere generata necessariamente da una prenotazione;

**IntestatarioFattura-Fattura**

* Tra l’entità IntestatarioFattura e Fattura esiste un’associazione riceve di cardinalità uno a molti (1:N);

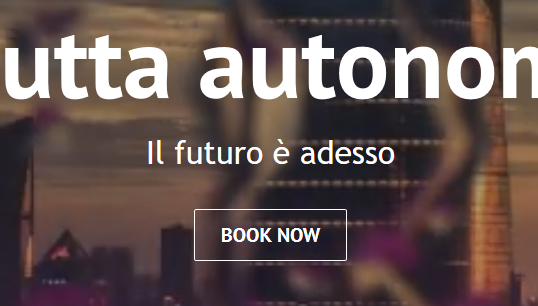
Di fatto l’associazione diretta è multipla poiché un intestatario fattura può ricevere più fatture, mentre l’associazione inversa è univoca poiché una fattura può essere ricevuta soltanto da un intestatario;

L’associazione diretta è totale poiché un intestatario deve necessariamente ricevere una fattura ogni volta che essa viene generata.

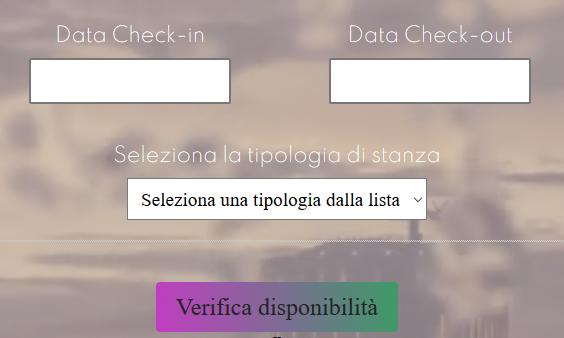
L’associazione inversa è anch’essa totale poiché una fattura deve necessariamente essere ricevuta da un intestatario ogni volta che essa viene generata;

Sequenza Eventi Cliente

1. **Il cliente accede al sito web. (Es.** [**www.\*\*\*\*\*hotel.it**](http://www.*****hotel.it)**);**
2. **Per riservare una prenotazione il cliente preme il pulsante “Book Now”:**



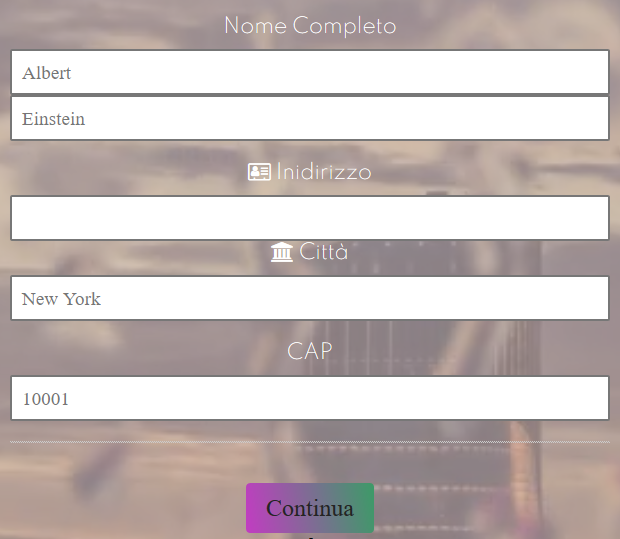
1. **Il cliente verifica la disponibilità della stanza compilando il primo form:**



1. **Il cliente inserisce i suoi dati personali:**



1. **Vengono inseriti i dati dell’intestatario fattura:**

****

1. **Vengono inseriti i dati della carta di credito e si riserva la prenotazione.**



Sequenza Eventi Sistema

1. Al premere del pulsante “Verifica Disponibilità” il back-end controlla il database per confermare la disponibilità del periodo selezionato.

\**Questo passaggio utilizza un algoritmo per verificare se due intervalli si sovrappongono.*

*(InizioIntA <= FineIntB) e (FineIntA >= InizioIntB) : Int – intervallo;*

*Dimostrazione:*

*Consideriamo due condizioni affinché la sovrapposizione non si verifichi:*

*Condizione A – L’intervalloA è completamente dopo l’intervalloB : InizioIntA > FineIntB;*

*Condizione B - L’intervalloA è completamente preccedente all’intervalloB : FineIntA > InizioIntB;*

*Quindi la sovrapposizione esiste se né la condizione A né la condizione B sono vere.*

*Secondo la legge di Morgan:*

*Not (A or B) ⬄ Not A and Not B*

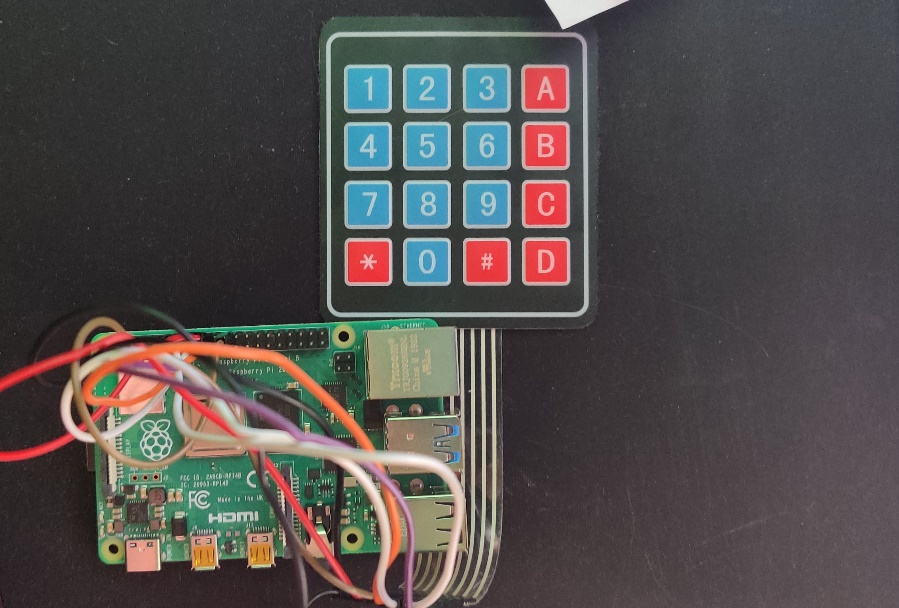
*Quindi nel nostro caso:*

*(InizioIntA <= FineIntB) and (FineIntA >= InizioIntB)*

1. Al premere del pulsante “Paga e Prenota” i dati del cliente, dell’Intestatario fattura e della prenotazione vengono inseriti nel database.
2. In seguito, il back-end invia al cliente i dati relativi alla sua prenotazione + la fattura;
3. Il back-end invia al cliente un codice generato, valido per la durata del suo soggiorno con il quale il cliente può accedere alla struttura.



1. Il cliente inserisce il codice all’arrivo presso la struttura alberghiera.



1. Il Raspberry comunica con il server per validare la password.
2. *Welcome!*

Immagine che contiene testo

Descrizione generata automaticamente