

[Bozza di] Soluzione simulazione esame del 28/2/2019

Fonte della prova

La prova è stata pubblicata sul sito del ministero.

- Esempio 1 (Febbraio 2019) Pdf
- Esempio 1 (Febbraio 2019) Word

Analisi del testo e appunti vari

Si evidenziano/sottolineano le parti del testo che si ritengono interessanti e significative (sul cartaceo).

Infrastruttura informatica

Un server per ogni stazione che raccoglie e coordina le operazioni. Il server di stazione aggiorna il server centrale mediante connessione ad internet (eventuale vpn per aumentare la sicurezza). Il server di stazione deve verificare che il cliente sia solvibile prima di aprire lo slot.

Potrebbe bastare verificare che esista una prenotazione ed eventualmente fare la prenotazione insieme al ritiro del velocipede.

I software potrebbero essere gestiti mediante sistema docker opportuno, utilizzando si garantisce l'indipendenza dal sistema operativo installato e la scalabilità del sistema. Anche le configurazioni, i ripristini e l'amministrazione sarebbe semplificata.

Applicazioni previste

- sito internet per clienti
- sito promozionale
- sito per amministrazione
- app per smartphone per clienti
- database
- log e backup
- servizio di ticket per guasti e segnalazioni (es. manutenzione bicicletta)

Server di stazione

Software: Pacchetto docker

- controllo slot
- comunicazione con server centrale

Server centrale

Software: Pacchetto docker

- database SQL MariaDB
- server web apache + php

Sicurezza

Il testo chiede di memorizzare il numero di carta di credito valida. Vista la delicatezza di tale dato si ritiene opportuno registrare il dato su entità separata. Si consiglia comunque di codificare i dati prima della memorizzazione nel database. La codifica dovrebbe essere tale da permettere la decodifica ai soli operatori (software) autorizzati. Per motivi di tempo tale implementazione non viene esplicitata nel presente documento ma deve essere affrontata prima di realizzare un sistema che richieda la memorizzazione di tali dati.

Una alternativa che risolverebbe il problema del pagamento delle tariffe e che non richiede la memorizzazione di dati sensibili potrebbe passare attraverso lo studio di uno smart contract da far funzionare su blockchain ethereum o simile.

Prevedere un servizio di videosorveglianza.

Prevedere mancanza connessione dalla stazione alla centrale. Memorizzare gli eventi e inviarli appena possibile.

Serve una sincronizzazione del tempo.

Prevedere la gestione sicura delle copie di backup del database.

Prevedere un sistema di log che permettano la ricostruzione degli eventi in caso di contestazioni da parte degli utenti.

Continuità del servizio

- opportuni sistemi di backup
 - delle connessioni
 - dell'alimentazione
 - hardware In caso di problemi di connessione tra stazione e centrale l'utente deve poter affittare la bicicletta. Per evitare truffe bisognerebbe controllare che l'utente non sia segnalato come problematico (carta di credito non valida o altre problematiche). Tale funzionalità non viene implementata.

Modello concettuale

Si utilizza ferret per realizzare lo schema concettuale.

Utenti

Devono avere anche i recapiti per invio della documentazione. Per semplificare si utilizza un'unica entità anche se sarebbe opportuno separare i recapiti in tabelle differenti.

Si considera il servizio fruibile ai soli privati rimandando la gestione delle azienda ad una fase successiva. Il CF viene gestito con 32 Byte in quanto si vuole consentire l'inserimento anche in presenza di eventuali problematiche.

Con il CF andrebbero memorizzati anche i dati necessari a ricavarlo (data e luogo di nascita, sesso), non si implementa tale funzionalità per problemi di tempo a disposizione.

Per identificarli nelle applicazioni web si utilizza la coppia email/password. L'indirizzo mail va verificato ma la trattazione di questo aspetto esula dalla presente trattazione. La password viene memorizzata dopo aver applicato l'hash sha256, il campo è pertanto di tipo *char[32]*.

Smart Card

Anche le smart card verranno memorizzate su entità separata. Si inserisce la data di invio e la data di scadenza. Il codice registrato su tale carta viene memorizzato nel campo *sc_pin* di tipo *char(16)* al fine di consentire 256^{16} codici differenti. Per garantire la sicurezza sarebbe opportuno studiare un sistema a chiave privata/pubblica sfruttando le caratteristiche della smart card. Tale approccio esula dagli scopi della presente trattazione (forse nella parte di sistemi e reti?). Si suppone semplicemente che la smart card sia l'unica in grado di fornire tale codice al sistema informatico.

Carta di credito

La gestione della memorizzazione dei numeri di una carta di credito è una operazione delicata e rischiosa. I pericoli che si corrono nel memorizzare tali dati sono parecchi e vanno affrontati con cura e attenzione.

Bisogna predisporre una informativa sulla privacy e verificare di rispettare tutte le leggi in materia come il recente GDPR. Tale problematica non viene affrontata.

Per limitare le problematiche di cui sopra, si decide di memorizzare le carte di credito in una entità separata. Sarà possibile limitare i permessi da SQL in modo più flessibile.

Mese e anno di scadenza vengono gestiti con due interi al posto di una data.

Nel campo *log_creazione* inseriamo i dati necessari ad identificare l'utente che l'ha inserita, indirizzo IP, data, ora e ogni altro dato si riterrà utile memorizzare al fine di tutelare sia la privacy degli utenti che la sicurezza del sistema informatico.

Biciclette

campo “rfid”: Il codice univoco memorizzato nell’RFID. Il modello: si inserisce tutto in un char tenendo conto anche della marca e di ogni altra caratteristica, andrebbe migliorata la gestione di queste informazioni normalizzando la tabella in modo opportuno, si rimanda ad altra trattazione tale adempimento. Tale pratica renderebbe anche più agevole la realizzazione del sito internet a scopo pubblicitario.

rfid è una chiave candidata adatta ma si opta per creare una chiave artificiale (id) al fine di consentire la sostituzione del chip in caso di rottura o malfunzionamento oppure il riutilizzo dell’rfid in caso di smaltimento della bicicletta. Si noti che tali opzioni non sono gestite, si può **considerare il campo rfid unico**.

Il campo codice è stato inserito per memorizzare il numero della bicicletta in modo che sia facilmente leggibile dall’utente (per poterla riconoscere agevolmente qualora sia parcheggiata con altre biciclette identiche).

Prevedere un trigger per aggiornare campo slot da impostare NULL quando la bici è in giro.

Al ritiro della bicicletta prevedere un servizio di segnalazione dello stato della bici, in caso di problemi il cliente potrà decidere se rifiutare la bici al fine di non essere ritenuto responsabile del degrado del mezzo. L’intervento di un operatore potrebbe risolvere eventuali diatribe prima che diventino difficilmente documentabili. La stazione dovrebbe comunque avere un sistema di videosorveglianza.

Stazioni

Per memorizzare la posizione esatta si possono usare tipi di campo specifici, sono reperibili sul sito di MariaDB e richiedono approfondimenti ulteriori. Ho inserito un campo di testo “coordinate” per gestire tali dati.

slot

E’ stata realizzata una entità slot. I campi libero e funzionante sono definiti come interi ma vengono gestiti principalmente come boolean. Libero va aggiornato con un trigger mentre funzionante serve per escludere eventuali slot difettosi dal conteggio di quelli disponibili. Anche eventuali bici intrappolate da malfunzionamenti.

Noleggio (ritiro e restituzione)

Supponendo di limitare ai soli due eventi del noleggio (ritiro e restituzione) si utilizza un’unica entità con i campi opportuni. Sebbene la traccia porti a consigliare la realizzazione di una entità eventi nella quale registrare gli eventi singoli, si ritiene che tale approccio semplifichi la trattazione delle interrogazioni.

Potrebbe risultare interessante prevedere anche la prenotazione di una bicicletta ad una certa stazione al fine di non arrivare “troppo tardi” alla stazione per scoprire che le biciclette “sono terminate”. L’importo si può calcolare considerando

le tre date: prenotazione, ritiro e consegna. La tariffazione dovrebbe dipendere anche dal modello di bicicletta noleggiato e dallo stato di conservazione alla consegna e al ritiro.

La consegna potrebbe essere gestita senza richiedere l'utilizzo della smart card. Si chiude il noleggio dell'utente che aveva in carico la bicicletta restituita. L'utente andrebbe avvisato della restituzione: mediante mail o altro messaggio push.

Si potrebbe prevedere una sorta di prepagato e un sistema per garantire la società da eventuali furti. La carta di credito può risolvere il problema del furto (o smarrimento) della bicicletta.

Modello logico

Ferret fatica a gestire la doppia relazione tra noleggio e stazione (per ritiro e consegna).

Comandi SQL per creare le tabelle

Nell'SQL generato vanno sistemati i soliti problemi:

- PK Autoincrement per tutte le tabelle
- nomi campi per relazioni 1aN
- ferret fatica a gestire la doppia relazione tra noleggio e stazione (anche campo n_box che indica il box nel quale prelevare/inserire la bicicletta).
- I campi che devono essere **UNIQUE** vanno specificati

Progetto di pagina web

Parte da completare. Indicare gli strumenti che si intende utilizzare, bozza dell'aspetto grafico atteso, struttura delle pagine e sezione amministratore e utenti registrati. Quali framework utilizzeresti? Come tieni in considerazione l'ottimizzazione per la visualizzazione sui dispositivi mobili? Intendi hostare i framework o utilizzare una CDN?

A partire da una mappa

Si vuole verificare la disponibilità di biciclette in una stazione selezionata

Visualizzare le biciclette in uso

Parte amministratore, elenco bici in uso, utente e stazione di prelievo.

Argomenti utili non ancora affrontati

Al fine di migliorare la trattazione del tema affrontato potrebbero risultare utili i seguenti argomenti.

Trigger

Sito di approfondimento

Stored procedure

Sito di approfondimento

Viste

Sito di approfondimento

Stored Functions

Sito di approfondimento