**MONEYMATE**

Munfuletto Ilenia (0744613)

**GENERAL DESCRIPTION:**

Il Progetto di sistemi web richiesto consiste nell’implementazione di una versione base di un’applicazione per la gestione di carte prepagate che permetta a chiunque la ricerca di una carta ,noto il numero, per conoscerne il saldo.

L’app deve anche permettere ad un admin loggato, di poter sbloccare/bloccare carte esistenti e di crearne di nuove o creare nuovi utenti (titolari o negozianti).

Un negoziante loggato invece, deve poter realizzare ricariche o addebiti su una carta noto il numero.

Infine, un titolare deve essere in grado di visualizzare le ultime transazioni di tutte le sue carte.

**SYSTEM ARCHITECTURE**

**Tecnologie utilizzate:**

1. Classi Java servlet, per l’accettazione di richieste http, per la costruzione delle relative pagine web di risposta e per il processo di validazione ed elaborazione dei dati inseriti dal client;
2. HTML e CSS per la realizzazione grafica dell’interfaccia utente, con framework Bootstrap 5.3.3;
3. JavaScript, in particolare JQuery, per associare gli eventi alle azioni dell’utente e per effettuare dei controlli di validazione, AJAX per effettuare le richieste http e JSON per la gestione di oggetti;
4. Apache Tomcat 10.1.2, quale middleware per il deployment della web application;
5. Database relazionale MySQL, gestito da MySQL Workbench, per la memorizzazione dei dati;
6. IntelliJ IDEA, come IDE per la scrittura del codice e la gestione delle risorse specificate precedentemente.

**FUNCTIONAL REQUIREMENTS**

**Generali:**

1. L’applicazione web sviluppata implementa un’ app di gestione di carte prepagate che permetterà ad un cliente Titolare di accedere all’app, tramite apposito form di login presente nella navbar del sito e visualizzare gli ultimi movimenti delle sue carte registrate, nella sezione “Ultimi Movimenti”.
2. L’applicazione, effettuando il login come Amministratore, sempre tramite la navbar del sito, permetterà anche di gestire l’inserimento di nuove carte in “Aggiungi Carta”, bloccare o sbloccare le presenti in “Autorizza Carte”, di inserire nuovi titolari o nuovi negozianti nella sezione “Gestione Utenti”.
3. L’applicazione, effettuando il login come Negoziante, attraverso la navbar, permetterà di effettuare ricariche o addebiti per la simulazione di un pagamento ad una carta, noto il numero, nella sezione personale denominata “Addebita/Ricarica Carta”.
4. Chiunque invece, potrà accedere alla sezione pubblica “Ricerca Carta”, collocata nella navbar, in cui poter cercare tramite il suo numero, una carta per conoscerne il saldo, il titolare e la data di scadenza.

**Client side:**

1. Vista index: visualizza la homepage del sito costituito da una navbar pubblica (“*Navbar.jsp*” nella cartella componenti) dalla quale è possibile loggarsi, navigare tra i contenuti della pagina ed accedere alla sezione di “Ricerca Carta”. Inoltre, presenta un footer (“*Footer.jsp*” nella cartella componenti) nel quale sono presenti link per contatti e social.
2. Vista ricercaCarta: visualizza la pagina per la ricerca della carta, costituita sempre dalla navbar pubblica (“*Navbar.jsp*” nella cartella componenti),dal footer (“*Footer.jsp*” nella cartella componenti) e da un form per l’inserimento del numero della carta. La pagina mostrerà il saldo della carta, se il numero inserito esiste ed è valido effettuando una richiesta post alla servlet corrispondente oppure segnalerà l’errore nell’input.
3. Vista errore: visualizza la pagina per gli errori 404 – Not Found o 500 – Internal Server Error, costituita sempre dalla navbar pubblica (“*Navbar.jsp*” nella cartella componenti),dal footer (“*Footer.jsp*” nella cartella componenti), da una descrizione generica dell’errore verificatosi e da un bottone per tornare alla homepage dell’app.
4. Componente cookieBanner: visualizza il banner per un possibile utilizzo dei cookie, incluso nella precedenti viste pubbliche e presente nella cartella componenti, con un pulsante di “Ok ” per l’accettazione di questi e la loro impostazione tramite script (“*cookie.js*”).
5. Vista Admin: visualizza la pagina riservata a seguito dell’accesso di un amministratore, costituita dalla navbar specifica dell’admin (“*NavbarAdmin.jsp*” nella cartella componenti) e dal classico footer (“*Footer.jsp*” nella cartella componenti). La pagina contiene quattro cards:

* “Aggiungi Carta”: rimanda al form di AggiungiCartaAdmin.
* “Autorizza Carte”: rimanda al form di BloccaSbloccaCartaAdmin.
* “Gestione Utenti” : rimanda alla pagina di GestisciUtenteAdmin.
* “Esci”: esegue il logout dell’admin richiamando la servlet corrispondente.

1. Vista Negoziante: visualizza la pagina riservata a seguito dell’accesso di un negoziante, costituita dalla navbar specifica del negoziante (“*NavbarNegoziante.jsp*” nella cartella componenti) e dal classico footer (“*Footer.jsp*” nella cartella componenti). La pagina contiene due cards:

* “Addebita/Ricarica Carta”: rimanda al form di AddebitaRicaricaCartaNegoziante.
* “Esci”: esegue il logout del negoziante richiamando la servlet corrispondente.

1. Vista Titolare: visualizza la pagina riservata a seguito dell’accesso di un titolare, costituita dalla navbar specifica del titolare (“*NavbarTitolare.jsp*” nella cartella componenti) e dal footer (“*Footer.jsp*” nella cartella componenti). La pagina contiene due cards:

* “Ultimi Movimenti”: rimanda al form di UltimiMovimentiTitolare.
* “Esci”: esegue il logout del titolare richiamando la servlet corrispondente.

1. Vista AggiungiCartaAdmin: visualizza la pagina di aggiungi carta riservata all’admin, contenente (“*NavbarAdmin.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) ed un form di registrazione di una nuova carta ed un bottone, che richiamerà la servlet corrispondente.
2. Vista BloccaSbloccaCartaAdmin: visualizza la pagina di blocca/sblocca carta riservata all’admin, contenente (“*NavbarAdmin.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) ed un form per l’inserimento del numero della carta di cui si vuole modificare lo stato. La pagina mosterà le informazioni della carta cercata (saldo, email titolare, data di scadenza e stato) inviando la richiesta alla servlet corrispondente , se il numero inserito esiste ed è valido, altrimenti segnalerà l’errore nell’input. Sarà presente poi un bottone di “Blocca” o “Sblocca” in base allo stato corrente della carta, per rispettivamente bloccarla o sbloccarla.
3. Vista GestioneUtentiAdmin: visualizza la pagina di gestione degli utenti riservata all’admin, contenente (“*NavbarAdmin.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) e due cards:

* “Aggiungi Titolare”: rimanda al form di AggiungiTitolareAdmin.
* “Aggiungi Negoziante”: rimanda al form di AggiungiNegozianteAdmin.

1. Vista AggiungiTitolareAdmin: visualizza la pagina di aggiungi titolare riservata all’admin, contenente (“*NavbarAdmin.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) ed un form di registrazione di un nuovo titolare ed un bottone, che richiamerà la servlet corrispondente.
2. Vista AggiungiNegozianteAdmin: visualizza la pagina di aggiungi negoziante riservata all’admin, contenente (“*NavbarAdmin.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) ed un form di registrazione di un nuovo negoziante ed un bottone, che richiamerà la servlet corrispondente.
3. Vista AddebitaRicaricaCartaNegoziante: visualizza la pagina di ricarica/addebita carta riservata al negoziante, contenente (“*NavbarNegoziante.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) ed un form per l’inserimento del numero della carta da ricaricare o addebitare. La pagina mosterà due checkbox per la scelta dell’operazione da eseguire, un input dedicato all’importo da ricaricare/addebitare ed una textarea per la causale. Sarà presente poi un bottone di “Ricarica” o “Addebita” in base all’ operazione selezionata, per inoltrare la richiesta alla servlet corrispondente.
4. Vista UltimiMovimenti: visualizza la pagina degli ultimi movimenti delle carte riservata al titolare, contenente (“*NavbarTitolare.jsp*” nella cartella componenti), (“*Footer.jsp*” nella cartella componenti) ed una tabella di riepilogo, ordinata per data, di tutte le transazioni effettuate da tutte le carte del titolare.
5. Componente Modal: visualizza il modale di notifica sull’esito dell’operazione corrente, presente nella cartella componenti.
6. Componente head delle pagine: contiene tutti i tag necessari a importare stili CSS, Bootstrap e personalizzati compresi nel file (“*style.css*”), e gli script jQuery.
7. Validazione client-side (tramite attributi HTML e jQuery).
8. Tecnologie:

* HTML
* CSS
* JavaScript
* jQuery
* AJAX
* JSON
* Framework Bootstrap per stili e script dell’interfaccia

**Server side:**

1. Il model sarà suddiviso nelle seguenti classi Java Bean:

* Utente :
* Admin
* Negoziante
* Titolare
* Carta
* Transazione

**DATA MODEL**

Schema logico per il database.

Immagine che contiene testo, diagramma, schermata, Carattere

Descrizione generata automaticamente