Relazione Progetto 2023/2024

USE CASE:

1. Registrazione mediante utilizzo della piattaforma desktop
2. Login mediante utilizzo della piattaforma desktop
3. Registrazione a una tratta mediante piattaforma desktop
4. Eliminazione di una registrazione a una tratta mediante piattaforma desktop
5. Logout mediante utilizzo della piattaforma Desktop
6. Login mediante utilizzo della piattaforma Telegram
7. Ricezione notifica mediante utilizzo della piattaforma Telegram
8. Logout mediante utilizzo della piattaforma Telegram

USE CASE 1:

1. L’utente inserisce i dati di registrazione (nome, cognome, e-mail, password) invia i dati effettuando una chiamata post server principale.
2. Il server principale effettua un hashing della password ed effettua una chiamata al gestore utente(server) passando i dati.
3. Il server che gestisce gli utenti verifica se i dati inseriti sono già presenti all’interno del database alla quale si collega o meno.
4. Se è presente riceve un feedback negativo a video e si chiede di inserire i dati.
5. Se non è presente, effettua la registrazione, salvando i dati nel DB e ricevendo un feedback positivo a video.
6. Si viene portati sulla pagina di login.

USE CASE 2:

1. L’utente inserisce i dati di login (e-mail, password) invia i dati effettuando una chiamata post server principale.
2. Il server principale effettua un hashing della password ed effettua una chiamata al gestore utente(server) passando i dati.
3. Il server che gestisce gli utenti verifica se l’utente esiste e se l’hash della password sia uguale a quello all’interno del database alla quale si collega.
4. Se no, riceve un feedback negativo a video e dice che non è stato autorizzato l’accesso.
5. Se è presente, effettua il login.
6. Si viene portati sulla pagina riguardante il pannello di controllo.

USE CASE 3:

1. Dopo aver effettuato il login e trovandosi nel pannello di controllo, si seleziona mediante dei menu a finestra la partenza e la destinazione.
2. Una volta selezionati partenza e destinazione si inviano al server principale mediante chiamata api.
3. Il server verifica se l’utente è già iscritto a tale tratta.
4. Se no, permette la registrazione e riceve un feedback positivo a video aggiungendo una riga dove nella sezione tratte iscritte.
5. Se è presente, notifica a schermo che non è possibile effettuare tale iscrizione poiché già presente.

USE CASE 4:

1. Dopo aver effettuato il login e trovandosi nel pannello di controllo, nella sezione riguardante le tratta alla quale siamo iscritti.
2. Possiamo selezionare una tratta inviando i dati di quest’ultima al server principale mediante chiamata api.
3. Il server verifica se presente.
4. Se no, riceve un feedback a schermo.
5. Se è presente, notifica a schermo che è stata rimossa tale iscrizione e la elimina dalla lista.

USE CASE 5:

1. L’utente clicca sul pulsante di logout
2. Una volta effettuato il logout viene chiusa la connessione al server e si torna alla pagina di login

USE CASE 6:

1. L’utente inserisce invia i dati di login (e-mail, password) dalla chat telegram al server telegram
2. I dati vengono inviati effettuando una chiamata post al server principale.
3. Il server principale effettua un hashing della password ed effettua una chiamata al gestore utente(server) passando i dati.
4. Il server che gestisce gli utenti verifica se l’utente esiste e se l’hash della password sia uguale a quello all’interno del database alla quale si collega.
5. Se no, riceve un feedback negativo a video e dice che non è stato autorizzato l’accesso.
6. Se è presente, effettua il login.

USE CASE 7:

1. Una volta effettuato il Login.
2. Viene aggiornato lo stato delle route alla quale si è iscritto.
3. Il server principale riceve tramite chiamata api i dati dell’utente.
4. Successivamente si connette al DB SQL e aggiorna il flag notify a 1, permettendo così la ricezione di notifiche.
5. Una volta fatto ciò se tutto è avvenuto correttamente il server Telegram smisterà i dati presenti sul server KAFKA all’utente loggato.

USE CASE 8:

1. L’utente utilizza il comando logout effettuando una chiamata post server principale.
2. Successivamente si connette al DB SQL e aggiorna il flag notify a 0, negando così la ricezione di notifiche.
3. Viene aggiornato lo stato delle route alla quale si è iscritto.
4. Il server principale riceve tramite chiamata api i dati dell’utente.
5. Una volta fatto ciò se tutto è avvenuto correttamente il server Telegram non smisterà i dati presenti sul server KAFKA all’utente che ha effettuato il logout.