

EasyWander - SDD

Gennaro Panico
0124002115

Michele Fiorentino
0124002085

Dario De Lauretis
0124002081

Giuseppe Iacomino
0124001769

January 2022



Indice

1	Introduzione	2
1.1	Scopo del sistema	2
1.2	Obiettivi di progettazione	2
1.3	Definizioni, acronimi e abbreviazioni	2
1.4	Definizioni	3
1.5	Panoramica	3
2	Sistema corrente	3
3	Sistema proposto	4
3.1	Panoramica	4
3.2	Decomposizione sottosistema	4
3.3	Hardware/Software Mapping	5
3.4	Gestione dati persistenti	6
3.5	Controllo degli accessi e sicurezza	7
3.6	Controllo software globale	7

3.7 Condizioni limite	7
4 Servizi del sottosistema	7
5 Glossario	8

1 Introduzione

1.1 Scopo del sistema

Negli ultimi anni è diventato sempre più popolare l'utilizzo di monopattini per spostarsi nei centri urbani. Le persone vogliono potersi spostare comodamente ma senza aver bisogno di portare con sé il proprio mezzo, per tal motivo si è diffuso il noleggio di monopattini. Analizzando il contesto in questione, non vi è ancora un'applicazione che fornisca all'utente una modalità di spostamento mediante l'utilizzo di monopattini elettrici che sia completamente comoda e immediata, per tal motivo nasce EasyWander. L'utente noleggerà il monopattino in uno dei vari punti di sosta predisposti dall'azienda, oppure può prelevare un qualsiasi monopattino presente in giro purché sia libero. Infatti, l'utente non sarà tenuto a riportare il monopattino in uno dei punti di sosta sebbene l'azione sarà incentivata (ad esempio, assegnando dei punti di credito).

1.2 Obiettivi di progettazione

- **Usability:** il sistema dovrebbe essere intuitivo da usare e l'interfaccia utente dovrebbe essere semplice da capire.
- **Portability:** il sistema deve essere sviluppato in modo da essere impiegato da hardware differenti senza subire modifiche.
- **Privacy:** il sistema non deve rendere pubblici i dati personali di chi sta usando i servizi forniti dall'applicazione.
- **Maintenance cost:** il sistema dovrebbe garantire costi contenuti di manutenibilità.
- **Response time:** Le operazioni di visualizzazione, aggiornamento, inserimento e rimozione dei dati devono essere garantite dal sistema in un intervallo di tempo accettabile.
- **Affidability:** il sistema deve essere affidabile

1.3 Definizioni, acronimi e abbreviazioni

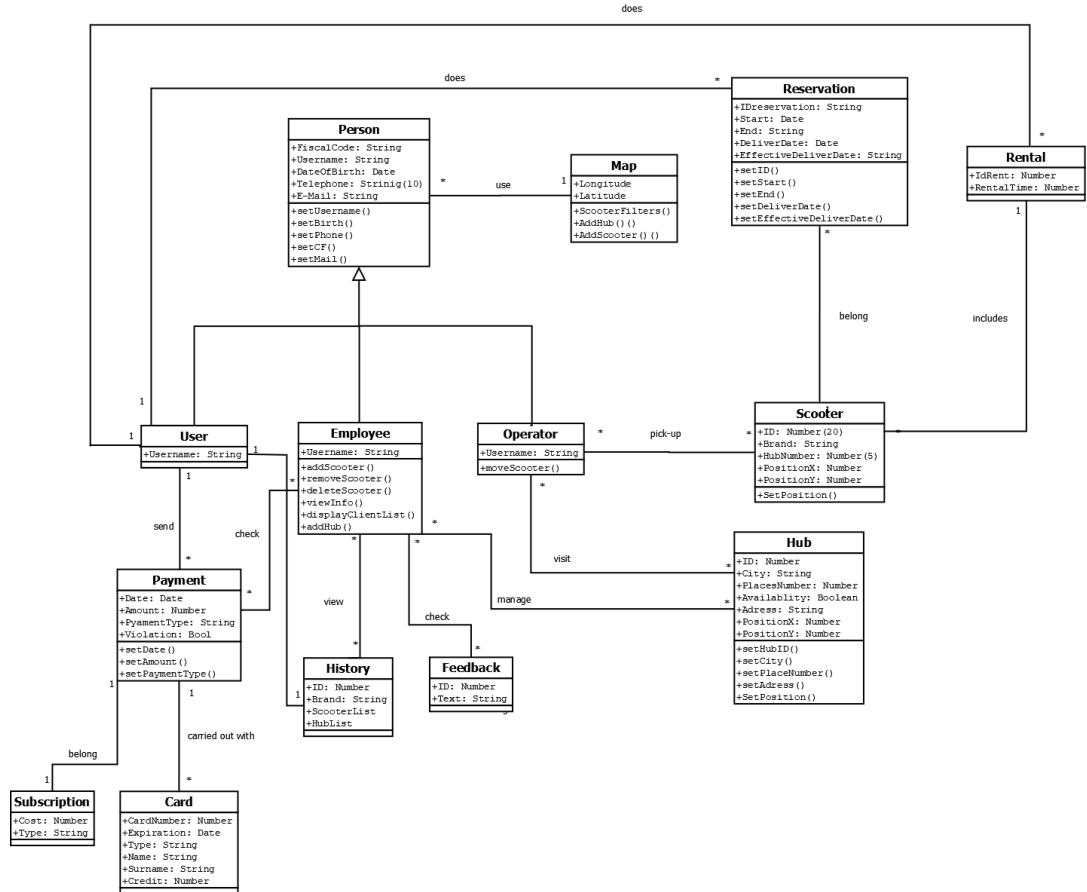
“EW” sta per EasyWander che è il nome dell' applicazione.

Non ci sono definizioni o abbreviazioni degne di nota. Per qualsiasi dubbio consultare il Glossario

1.4 Definizioni

Nessuna definizione degna di nota. Per qualsiasi dubbio consultare il Glossario

1.5 Panoramica



2 Sistema corrente

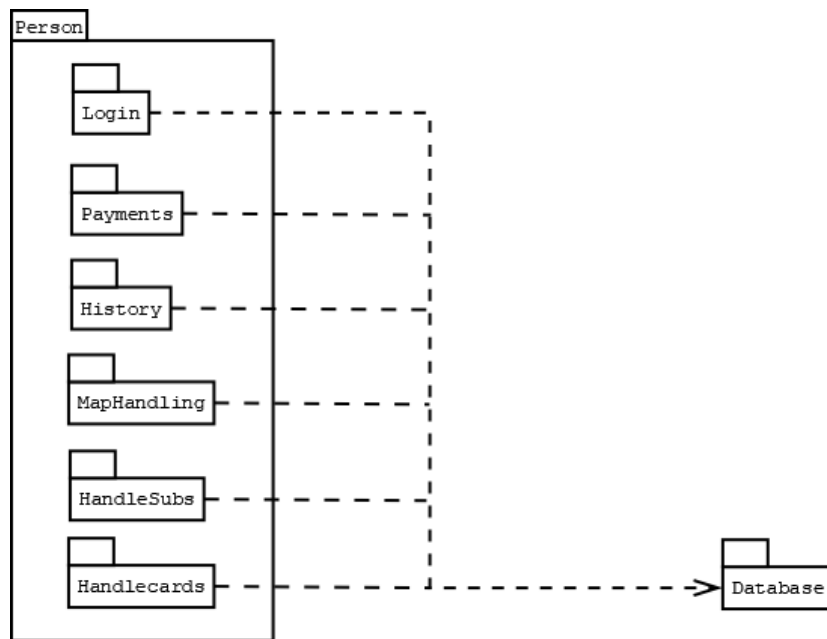
Il sistema è implementato dal nulla in quanto nessun sistema esiste in precedenza, per cui questa fase è di “Greenfield Engineering” e la raccolta dei requisiti viene fatta esclusivamente colloquiando con il cliente e cercando di estrarre i requisiti dalle sue richieste.

3 Sistema proposto

3.1 Panoramica

Lo stile architetturale scelto è il client/server, individuando tre tipologie di client: un primo, pensato per l'utilizzo dell'employee, che gestisce e monitora le funzionalità e i contenuti del sistema; il secondo per l'utilizzo dell'operatore che è incaricato allo spostamento fisico degli scooter e infine il terzo che è destinato agli utenti che usufruiscono del servizio. Il server si occuperà di fornire una servizi ai client; inoltre, gestisce l'accesso al Database EW. La ragione che ci ha spinto a scegliere questo stile architetturale è perché tale stile ha diversi obiettivi, cioè è: portabile, scalabile, flessibile, affidabile, di posizione trasparente, e ad elevate prestazioni.

3.2 Decomposizione sottosistema



Login: Fase in cui viene gestito l'accesso ai profili personali degli utenti con l'inserimento delle credenziali o di eventuali richiesta di recupero di esse.

Payments: Area del sistema in cui viene gestito il pagamento degli scooter attraverso il credito dell'utente.

HandleSubs: Area del sistema che gestisce l'abbonamento mensile degli utenti e che ne permette quindi l'acquisto o la rimozione.

HandleCards: Area del sistema che permette all'utente di aggiungere o rimuovere carte di credito al profilo dell'utente.

History: Area del sistema che registra la cronologia dei monopattini selezionati dall'utente a cui può accedere anche l'employee.

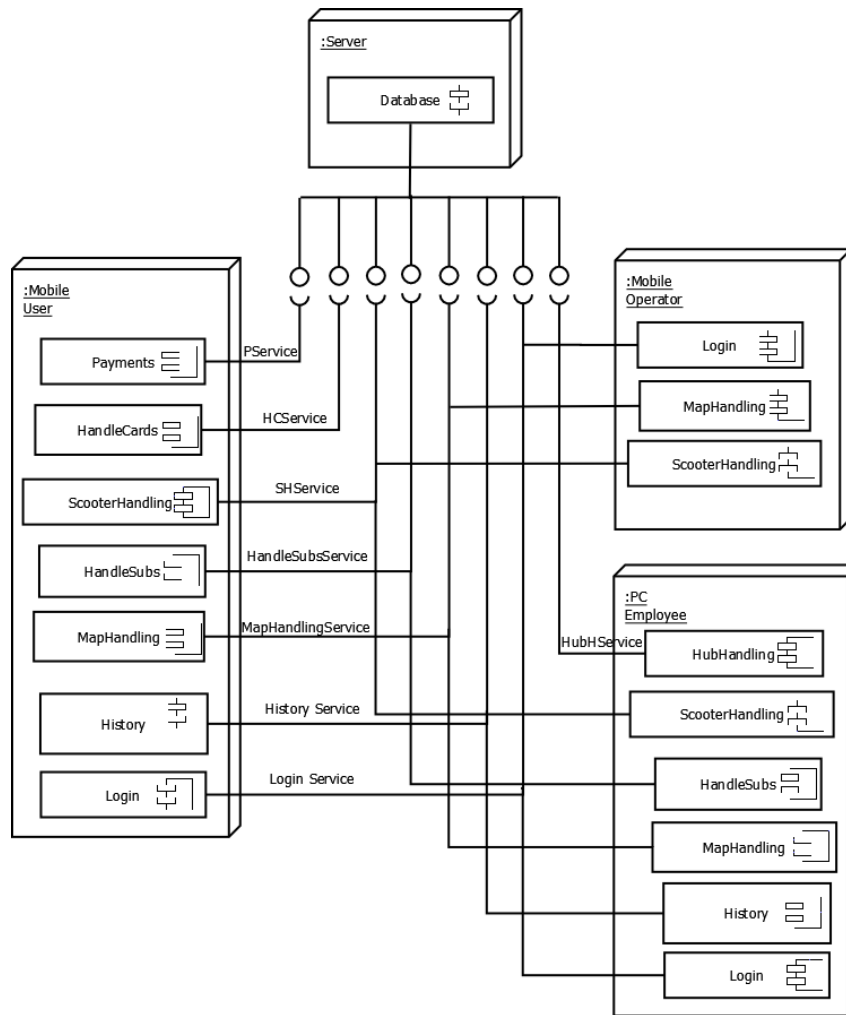
MapHandling: Area del sistema gestita da un employee che permette all'utente di visualizzare una mappa dove sono identificati con segnalini gli scooter e gli hub.

ScooterHandling: Operazione riservata agli operatori che si occupano di aggiungere ,rimuovere o semplicemente spostare gli scooter da una zona all'altra.

HubHandling: Operazione di gestione degli hub riservata agli employee che permette loro di aggiungere hub nella mappa.

3.3 Hardware/Software Mapping

Il sistema è pensato per essere usufruito principalmente dispositivi mobile, ad eccezione dell'Employee che può utilizzarlo anche su computer.



3.4 Gestione dati persistenti

EW è un'applicazione che si occupa di tener traccia di un insieme di scooter e Hub in giro per il mondo, oltre che dei noleggi e delle informazioni degli utenti. Vista la grande quantità di dati da gestire, si è deciso di usare un database relazionale.

3.5 Controllo degli accessi e sicurezza

Oggetti/Attori	Employee	Operator	User
<u>Rent</u>	-	-	scooterRented manageRentedScooter
Login	login()	login()	login()
Payment	-	-	setDate() setAmount() setPaymentType()
Hub	addHub() setHubID() setCity() setplaceNumber() setAddress()	-	-
Scooter	addScooter() removeScooter() deleteScooter() ScooterFilters()	moveScooter()	-
<u>Reservation</u>	-	-	setID() setStart() setEnd() setDeliverDate() setEffectiveDeliverDate()
Feedback		viewFeedBack()	feedback()

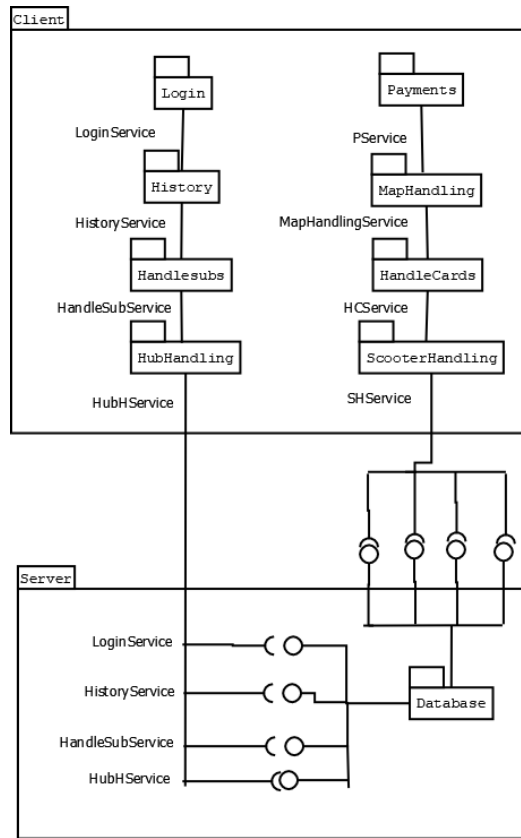
3.6 Controllo software globale

La numerosità di richieste sottomesse al server porta alla naturale scelta di un flusso di controllo thread based, e dunque è necessario gestire la concorrenza di queste richieste.

3.7 Condizioni limite

EW è un'applicazione che dispone di un proprio server proprietario, dunque la gestione e la manutenzione del server, come l'inizializzazione e la terminazione, è gestita dai tecnici stessi.

4 Servizi del sottosistema



5 Glossario

User: Entità generalizzata che rappresenta il client utente con i suoi attributi.

Employee: Impiegato che gestisce e monitora le funzionalità ed i contenuti del sistema.

Operator: Operatore che viene incaricato dall'Employee di effettuare una serie di operazioni riguardanti gli scooter, ad esempio: riportare uno scooter nell'hub di appartenenza o controllare lo stato dello scooter stesso nel momento in cui viene prelevato.

Hub: Aggregazione che comprende un dato numero di scooter convenzionati col servizio EasyWander. Negli hub vengono posizionati gli scooter e di ciascuno di essi se ne memorizza l'ubicazione e la disponibilità. È identificato da un Id o nome(da chiedere). Anche questi ultimi sono gestiti dagli impiegati relativamente all'aggiunta o alla rimozione.

Feedback: Valutazione dello stato dello scooter da parte dell'utente. In base alla valutazione l'impiegato sceglie se mandare in manutenzione lo scooter oppure se sostituirlo definitivamente con uno nuovo.

History: Cronologia degli scooter noleggiati dall'utente. Conterrà informazioni riguardo l'identificativo dello scooter noleggiato, il costo, la data e il tempo per cui è stato noleggiato.

Payment: Pagamento effettuato da un cliente riguardo un noleggio da lui sostenuto.

Rental: Insieme di informazioni relative ad un noleggio effettuato da un cliente iscritto al servizio.

FacadeService: Interfaccia che conosce le classi nel sottosistema che sono responsabili di una richiesta e che delega le richieste del client agli oggetti appropriati.

ORACLESQL: Entità che rappresenta il punto di ingresso al database ORACLE. Contiene tutte le funzioni che permettono di effettuare operazioni direttamente sul database.

LoginService: Entità che rappresenta il punto di accesso al sottosistema LoginService e ne implementa le funzionalità. Il sottosistema rappresenta la fase in cui viene gestito l'accesso ai profili personali degli utenti con l'inserimento delle credenziali o di eventuali richieste di recupero di esse.

HistoryService: Entità che rappresenta il punto di accesso al sottosistema HistoryService e ne implementa le funzionalità. Il sottosistema rappresenta l'area del sistema che registra la cronologia dei monopattini selezionati dall'utente a cui può accedere anche l'employee.

HandleSubService: Entità che rappresenta il punto di accesso al sottosistema HandleSubService e ne implementa le funzionalità. Il sottosistema rappresenta l'area del sistema che gestisce l'abbonamento mensile degli utenti e che ne permette quindi l'acquisto o la rimozione.

HubHandlingService: Entità che rappresenta il punto di accesso al sottosistema HubHandlingService e ne implementa le funzionalità. Il sottosistema rappresenta l'operazione di gestione degli hub riservata agli employee che permette loro di aggiungere hub nella mappa.

PaymentService: Entità che rappresenta il punto di accesso al sottosistema PaymentService e ne implementa le funzionalità. Il sottosistema rappresenta l'area del sistema in cui viene gestito il pagamento degli scooter attraverso il credito dell'utente.

MapHandlingService: Entità che rappresenta il punto di accesso al sottosistema MapHandlingService e ne implementa le funzionalità. Il sottosistema rappresenta l'area del sistema gestita da un employee che permette all'utente di visualizzare una mappa dove sono identificati con segnalini gli scooter e gli hub.

HandleCardsService: Entità che rappresenta il punto di accesso al sottosistema HandleCardsService e ne implementa le funzionalità. Il sottosistema rappresenta l'area del sistema che permette all'utente di aggiungere o rimuovere carte di credito al profilo dell'utente.

ScooterHandlingService: Entità che rappresenta il punto di accesso al sottosistema ScooterHandlingService e ne implementa le funzionalità. Il sottosistema rappresenta l'operazione riservata agli operatori che si occupano di aggiungere, rimuovere o semplicemente spostare gli scooter da una zona all'altra.

Form: Interfaccia implementata da oggetti che rappresentano form.

LoginForm: interfaccia adibita al login.

PaymentForm: Interfaccia adibita al pagamento.

HistoryForm: Interfaccia adibita alla cronologia dei monopattini.

SubForm: Interfaccia adibita all'abbonamento.

CardForm: Interfaccia adibita all'aggiungimento o alla rimozione di carte di credito dell'utente.

ScooterForm: Interfaccia adibita al noleggio dei monopattini.

HistoryView: Classe rappresentante la visualizzazione della cronologia da parte dell'employee.