

UNIVERSITÀ DEGLI STUDI DI NAPOLI PARTHENOPE
SCUOLA INTERDIPARTIMENTALE DELLE SCIENZE, DELL'INGEGNERIA E DELLA SALUTE
INFORMATICA
CORSO DI INGEGNERIA DEL SOFTWARE E INTERAZIONE UOMO-MACCHINA



System Design Document

Proponenti:

Mungari Alfredo	0124002134
Giordano Orsini Massimiliano	0124002214
Ferraro Dominick	0124002048
Caruso Denny	0124002062

Data di Consegna:

//2022

Anno Accademico:

2021 – 2022

Categoria:

Trasporto Urbano

Indice

1 - Introduzione.....	2
1.1 - Scopo del sistema.....	2
1.2 – Obiettivi di progettazione	3
1.3 – Definizioni, Abbreviazioni e Acronimi.....	4
1.3.1 - Definizioni	4
1.3.2 - Abbreviazioni	4
1.3.3 - Acronimi.....	4
1.4 - Riferimenti.....	5
2 – Sistema Corrente.....	7
3 – Sistema Proposto.....	9
3.1 – Panoramica.....	9
3.2 – Decomposizione del sistema.....	11
3.3 – Hardware/Software Mapping	14
3.4 – Gestione dei dati persistenti	15
3.5 – Controllo accessi e sicurezza.....	15
3.6 – Decisioni sul flusso di controllo globale	15
3.7 – Condizioni limite	16
4 – Servizi dei sottosistemi	18
5 – Glossario.....	20

Elenco delle figure

Figura 1.1: Diagramma delle classi fase di progettazione.....	5
Figura 1.2: Diagramma dei sottosistemi.....	11
Figura 1.3: Diagramma Sottosistemi software con annessi servizi offerti.....	18

Bibliografia

[1] Bernd Bruegge, Allen Dutoit - Object-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International Edition (Pearson, 2014)

[2] Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides - Design Patterns - Elements of Reusable Object-Oriented Software (1994, Addison-Wesley Professional)

Note Aggiuntive

Per una migliore visualizzazione dei diagrammi e delle figure riportate si consiglia di consultare i relativi file allegati.

Introduzione

1 – Introduzione

1.1 - Scopo del sistema

La mobilità cittadina è un servizio sempre più essenziale per l'uomo moderno, molto spesso però ci sono tanti problemi organizzativi e difficoltà che causano contrattempi indesiderati, generando disagi ai fruitori del sistema stesso, ovvero ai singoli cittadini che a vario titolo consultano il sistema per organizzare la propria vita privata ed i propri impegni, e gli “addetti ai lavori” ovvero tutti i lavoratori dell'azienda di trasporto urbano che a vario titolo utilizzano il sistema per svolgere le proprie mansioni.

Lo scopo del sistema IPT risiede nel fornire un supporto, ai cittadini e ai lavoratori dell'azienda, affidabile, sicuro e sempre funzionante. IPT si interfaccia sia con l'utente cittadino che con l'utente impiegato, fornendo semplicità e concretezza a chi lo utilizza.

È un sistema che permette la gestione interna di tutto ciò che riguarda l'organizzazione delle corse, dei mezzi, delle linee, dei turni lavorativi, degli impiegati e dall'altro lato permette all'utente cittadino di comprare un biglietto, visualizzare le corse, ricercare i percorsi percorribili con l'ausilio di mezzi di trasporto aziendali e altre molteplici funzioni, tutte da scoprire. È stata individuata una figura, il *Manager Aziendale*, il cui compito sarà gestire gli aspetti tecnici e gestionali di IPT.

1.2 – Obiettivi di progettazione

- Response time: IPT deve fornire soluzioni in tempi brevi rispetto ad eventi straordinari che potrebbero verificarsi, ad esempio come nei sistemi real-time.
- Maintenance cost: IPT dovrebbe garantire costi convenienti di manutenibilità.
- Security: IPT dovrebbe garantire la sicurezza dei dati nel rispetto della privacy di tutti gli utenti utilizzatori.
- Portability: IPT dovrebbe essere un sistema multiplatforma.
- Extensibility: IPT dovrebbe essere tale da permettere l'aggiunta di nuove funzionalità e comportamenti in modo semplice, senza dover modificare l'intero sistema.
- Availability: IPT dovrebbe garantire, per quanto possibile, il funzionamento del servizio. Pertanto, deve rispondere con particolare resilienza ad eventi straordinari come guasti ai mezzi di trasporto oppure imprevisti da parte degli autisti.
- Look and feel: le interfacce di IPT dovrebbero essere *responsive* e rispettare tutti i canoni di progettazione moderna.
- Engagement: IPT dovrebbe invogliare l'utilizzo all'utente cittadino ogni volta che deve usare i mezzi.
- Accessibility: IPT dovrebbe ispirarsi alla metodologia progettuale DFA, garantendo l'utilizzo semplice ed immediato per tutti.

1.3 – Definizioni, Abbreviazioni e Acronimi

All'interno di questo documento troveremo un gergo riguardante il mondo del trasporto urbano. Sono state quindi stilate delle sezioni contenenti definizioni, acronimi e abbreviazioni per orientare meglio il lettore.

1.3.1 - Definizioni

- **Checking:** fase all'interno della quale una figura lavorativa checker si occupa di verificare il corretto funzionamento di alcuni mezzi di trasporto aziendali. Qualora dovessero esserci guasti, il checker provvede a segnalarli al sistema. Se il guasto è di tipo ordinario, cioè tale da richiedere manutenzione ordinaria, il checker provvede anche alla risoluzione.
- **Abbonamento:** clausola di una convenzione che un cliente contrae con un fornitore di servizi o beni al fine di poter accedere, per un certo periodo, a multipli di essi secondo una tariffa stabilita.
- **Scheduling:** fase all'interno della quale una figura lavorativa scheduler si occupa di organizzare la giornata di lavoro assegnando mezzi e autisti per un turno lavorativo.
- **GPS:** è un sistema satellitare per la navigazione, la misura del tempo, la misura della distanza e il posizionamento globale.

1.3.2 - Abbreviazioni

- **MO:** abbreviazione per "Manutenzione ordinaria"
- **MNO:** abbreviazione per "Manutenzione non ordinaria"
- **DB:** abbreviazione per "Database"
- **Log:** abbreviazione per "Login"
- **QR:** abbreviazione per "Quick Response Code"
- **GPS:** abbreviazione per "Global Positioning System"

1.3.3 - Acronimi

- **D.F.A.** acronimo per "Design For All", stile di progettazione "open user".
- **I.P.T.** acronimo per "Ingegneria SW, Programmazione III, TecWeb", l'insieme degli esami che compongono il progetto. Così è stato rinominato il sistema.
- **A.P.I** acronimo per "Application Programming Interface" si indica un insieme di procedure (in genere raggruppate per strumenti specifici) atte all'espletamento di un dato compito; spesso tale termine designa le librerie software di un linguaggio di programmazione.

1.4 - Riferimenti

Si faccia riferimento al RAD per avere una visione globale del sistema e per avere una comprensione maggiore del lavoro esposto.

1.5 - Panoramica

Un ulteriore affinamento di quanto ottenuto in fase di analisi ha portato alla seguente suddivisione in classi. Si veda “Design Patterns - Elements of Reusable Object-Oriented Software” (1994, Addison-Wesley Professional) per confrontare i design pattern sfruttati: Strategy, Facade, Mediator, Abstract Factory, Singleton, Observer, Memento, Iterator.

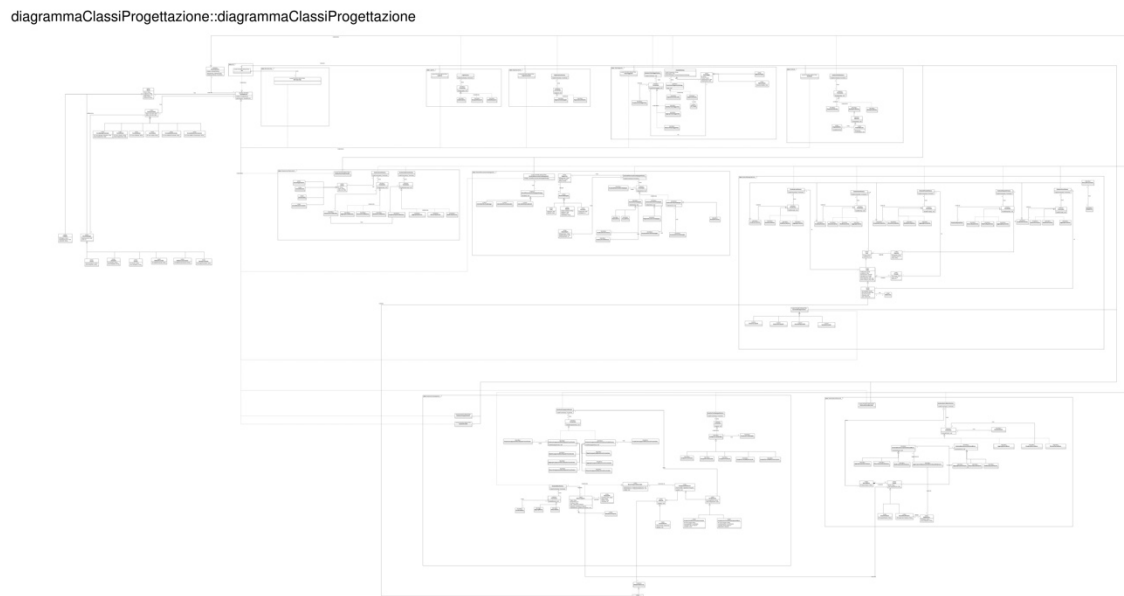


Figura 1.1: Diagramma Classi Progettazione

Sistema corrente

2 - Sistema corrente

IPT è frutto di un progetto greenfield: non si basa per tanto su di un'architettura preesistente, sollevando il lavoro dalla necessità di rimanere all'interno di vincoli precedentemente imposti. Lo sviluppo del sistema introdotto parte da zero e non si basa su sistemi precedentemente esistenti. Si parla quindi di Greenfield Engineering. Infatti, in base a “Object-oriented software engineering using UML, Patterns, and Java” (2014, Pearson):

“In greenfield engineering, the development starts from scratch, no prior system exists, so the requirements are extracted from the users and the client. A greenfield engineering project is triggered by a user need or the creation of a new market. [...] Developers must first gain a solid knowledge of the application domain before the direct approach can be used.”

Sistema proposto

3 - Sistema proposto

3.1 – Panoramica

Per il sistema proposto è stato scelto il modello client/server. Si sono individuati una moltitudine di client che possono utilizzare diverse funzionalità in maniera semplice e veloce. Di seguito viene analizzato nel dettaglio un breve riepilogo dei vari utenti del sistema:

- **Cittadino:** tramite il software richiesto, è in grado di visionare biglietti, abbonamenti e altri eventuali titoli di viaggio messi a disposizione dall'azienda, di effettuarne l'acquisto e il rinnovo. L'azienda prevede un abbonamento unico che può essere utilizzato per viaggiare su diverse linee. L'acquisto dei titoli di viaggio all'interno del sistema software richiesto, si appoggia su un'infrastruttura già presente fornita dall'azienda di trasporto e quindi il sistema software richiesto non è tenuto a provvedere tali funzionalità, ma è necessario solo collegare il sistema richiesto con il sistema preesistente. Il sistema software deve essere in grado di sfruttare questa tecnologia per condividere in tempo reale le posizioni di tutti i suoi mezzi.

La posizione ottenuta mediante GPS dai mezzi di trasporto non è sfruttata solo internamente dall'azienda di trasporto urbano al fine di gestire e migliorare il servizio, ma anche al fine di fornire un servizio all'utenza che sarà in grado di tracciare i mezzi ai quali è interessata, valutare percorsi, eventuali ritardi e così via.

- **Scheduler:** si occupa dello scheduling dei checker e degli autisti
- **Checker:** si occupa della verifica dei mezzi di trasporto prima del loro impiego
- **Autista:** si occupa della guida dei mezzi di trasporto
- **AddettoPersonale:** si occupa della gestione degli impiegati dal punto di vista remunerativo
- **AddettoComunicazioni:** si occupa di informare l'utenza di eventi e imprevisti che accadono su linee, corse, fermate, e così via
- **ManagerAziendale:** si occupa della registrazione degli impiegati, della loro assunzione e del loro licenziamento, approva le buste paga elaborate dagli "AddettoPersonale", gestione delle aree geografiche, delle linee, dei mezzi, dei depositi, delle fermate e degli impiegati in termini di registrazione o rimozione.

Il server dispone di un API capace di fornire una serie servizi ai vari client; inoltre, gestisce l'accesso al Database IPT. Tra le ragioni che hanno fatto preferire un'applicazione con interfacciamento sulla rete piuttosto che un software stand-alone, troviamo la possibilità di suddividere i compiti tra diverse tipologie di utenti, garantendo maggiore usabilità, e la necessità di aumentare la disponibilità, rendendo possibile l'accesso al sistema da qualunque dispositivo.

È necessario precisare che le funzionalità alle quali ogni client può accedere sono ben determinate dalla fase di login che avviene nel sistema stesso. Una volta effettuata tale fase, il sistema riconosce la tipologia di utente e mette a disposizione un'interfaccia ben definita. Chiaramente, nel momento in cui il cittadino può anche non usufruire della registrazione e quindi del login, viene visto dal sistema come un utente anonimo. I dati e le preferenze non risulteranno essere persistenti a livello di sistema, ma solo a livello di singolo dispositivo. Inoltre, tale utente cittadino che non sceglie di registrarsi ha un'interfaccia più ridotta rispetto all'utente cittadino che sceglie di registrarsi come per esempio la mancata possibilità di effettuare acquisti di titoli di viaggio per il trasporto urbano previsto dall'azienda. In questo senso un utente è sempre "loggato" al sistema software come impiegato aziendale, con i differenti incarichi ricoperti, come cittadino registrato o come cittadino non registrato.

3.2 - Decomposizione del sistema

diagrammaPackage::diagrammaPackage

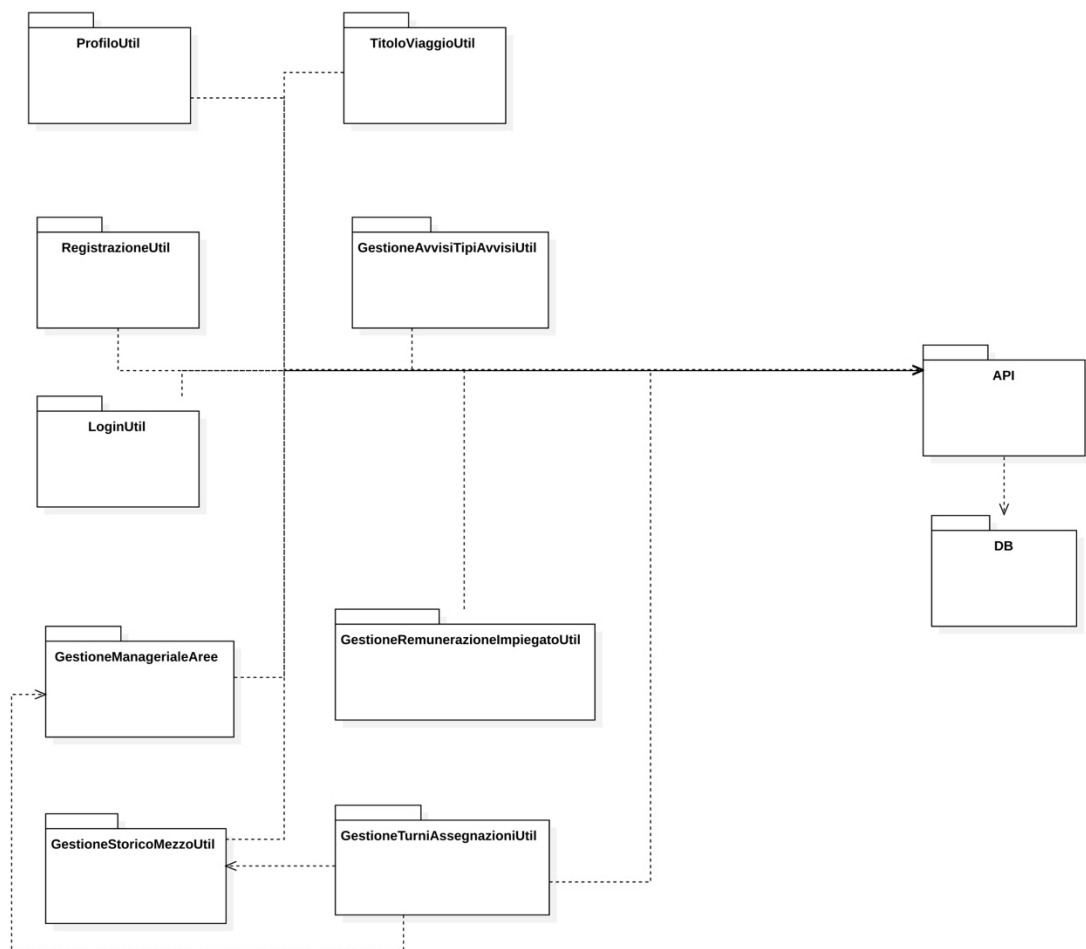


Figura 1.2: Diagramma Sottosistemi Software

Analizziamo nel dettaglio i vari sottosistemi individuati:

- ProfiloUtil è il sottosistema software all'interno del quale si gestisce tutto l'interfacciamento con l'utente in merito alla modifica, eliminazione e visualizzazione del suo profilo
- TitoloViaggioUtil è il sottosistema software all'interno del quale viene integrata la parte del sistema software che riguarda l'acquisto di titoli di viaggio, quindi sia titoli di viaggio singoli che abbonamenti, la loro rimozione e visualizzazione
- RegistrazioneUtil è il sottosistema software che prevede l'interfacciamento con l'utente per la fase di registrazione
- LoginUtil è il sottosistema software che prevede l'interfacciamento con l'utente per la fase di login
- GestioneAvvisiTipiAvvisiUtil è il sottosistema software all'interno del quale si gestisce l'interfacciamento con l'utente in merito alle seguenti operazioni: aggiunta, modifica, visualizzazione e rimozione di avvisi in bacheca, aggiunta, modifica, visualizzazione e rimozione di tipi di avvisi che è possibile inserire in bacheca
- GestioneManagerialeAree è il sottosistema software all'interno del quale si gestisce l'interfacciamento con l'utente in merito alle seguenti operazioni: aggiunta, rimozione, modifica e visualizzazione di aree, aggiunta, rimozione, modifica e visualizzazione di linee, aggiunta, rimozione, modifica e visualizzazione di corse, aggiunta, rimozione, modifica e visualizzazione di depositi
- GestioneRemunerazioneImpiegato è il sottosistema software all'interno del quale si gestisce l'interfacciamento con l'utente in merito alle seguenti operazioni: calcolo remunerazione impiegato secondo diversi approcci, richieste di giorno/i di malattia, di giorni di ferie, di ore di permesso, concessione di ferie e permessi, visualizzazione e inserimento di straordinari effettuati da un impiegato, visualizzazione della remunerazione dell'impiegato
- GestioneStoricoMezzoUtil è il sottosistema software all'interno del quale si gestisce l'interfacciamento con l'utente in merito alle seguenti operazioni: aggiunta, modifica, rimozione, visualizzazione di guasti ordinari e non ordinari, aggiunta, modifica, rimozione e visualizzazione dello storico di uno o più mezzi di trasporto aziendali, aggiornamenti dovuti all'interfacciamento con una ditta specializzata di riparazione per i casi di manutenzione non ordinaria
- GestioneTurnoAssegnazioneUtil è il sottosistema software all'interno del quale si gestisce l'interfacciamento con l'utente in merito alle seguenti operazioni: aggiunta, rimozione, ripetizione e visualizzazione di assegnazioni "MezzoCorsaTurnoAutista" e di assegnazioni "MezziDepositoTurnoChecker", aggiunta e visualizzazione di turni per checker, scheduler, addetto comunicazioni, addetto personale, autista; aggiunta, rimozione e visualizzazione dei mezzi di trasporto aziendali, nonché la segnalazione di guasti a un mezzo di trasporto aziendale riscontrato da un checker o da uno autista e conseguente notifica alla figura scheduler e all'autista eventualmente interessato.

- API è il sottosistema che fa da ponte tra cliente e server. Permette di ridurre i costi di manutenibilità nell'ipotesi che IPT decida di affidarsi ad un diverso provider, poiché garantisce un minore accoppiamento tra i sottosistemi client e il sottosistema Database, e incrementa la portabilità di IPT. Inoltre offre un'interfaccia alle funzioni effettuabili da uno dei client menzionati nelle sezioni precedenti.
- DB è il sottosistema "Database" che permette il salvataggio dei dati del sistema in maniera persistente.

C'è da fare una piccola considerazione di carattere pratico: per ogni sottosistema le funzionalità presenti non sono accessibili da tutti gli utenti. Per prevenire l'accesso a funzioni non previste per uno dei determinati utenti del sistema, si effettua un controllo "a monte" sul tipo di utente "loggato" e di conseguenza tramite l'adattamento dinamico delle interfacce gli si permette l'accesso solo a parte di essa e gli si nega l'accesso alla restante parte del sistema.

3.3 - Hardware/Software mapping

Il sistema è pensato per poter garantire massima disponibilità. Per tanto è utilizzabile da qualunque notebook o altro device per i quali è stato sviluppato il sistema software con il soli vincolo di possedere il software installato e per accedere alla maggior parte dei servizi un determinato account utente, a seconda dei servizi ai quali è necessario accedere.

Uno scenario che si può immaginare è una macchina server sul quale è presente la parte server del sistema software ovvero il database e l'API. A tale macchina server è possibile collegarsi mediante la rete da un device dell'utente che può avere più o meno prestazioni avanzate. In ogni caso il segmento utente è necessario che abbia le capacità necessarie per interfacciarsi con il servizio GPS mediante apposito ricevitore, qualora si voglia usufruire dei servizi ad esso connessi.

Inoltre, per l'elaborazione della mappa in tempo reale con il dinamismo dei mezzi di trasporto che effettuano le proprie corse, sarà necessario anche un buon quantitativo di potenza in termini di processore e memoria in merito alle macchine sulle quali è installato il segmento client. In maniera analoga la macchina server dovrà avere le capacità opportune al fine di garantire i criteri di accettazione del cliente finale che ha commissionato il sistema software e i NFR che riguardano gli aspetti "URPS" del paradigma "FURPS".

Qualora dovesse essere necessario, il tasso di risposta desiderato dal sistema può essere ottenuto con il potenziamento dell'hardware esistente e presente e dell'ampiezza di banda di comunicazione. Per quanto riguarda il software, l'applicazione di algoritmi efficienti in termini di tempo e spazio, permetterà di ottimizzare le prestazioni lato software.

3.4 - Gestione dei dati persistenti

IPT è un sistema software molto complesso e dalla dimensiona mastodontica. Prevede la memorizzazione di numerosissimi dati in maniera persistente, l'interfacciamento, la sincronizzazione e la notifica di più attori coinvolti e per di più in un ambito professionale del tutto estraneo al mondo dell'informatica. Ragion per cui si è scelto un database relazionale al fine di mantenere traccia di tutti i dati generati, richiesti, da elaborare, da trasmettere e per gestire in maniera più semplice ed efficiente gli accessi concorrenti.

Si prevede che molte delle informazioni visualizzabili dagli utenti sono ottenibili mediante l'esecuzione di determinate query sul database, ragion per cui si prevede un alto tasso di utilizzo del suddetto. Molte delle classi facenti parte del modello a oggetti sono state progettate tenendo presente anche il mapping con le tabelle di un eventuale database, scelta fatta che si è rivelata ottima a posteriori.

3.5 - Controllo accessi e sicurezza

A questo punto della trattazione è necessario stabilire come determinare quale attore può accedere a quale servizio offerto da quale sottosistema. Come detto già precedentemente ogni sottosistema può avere a disposizione l'offerta di più servizi. Si “GestioneManagerialeAree” e “GestioneTurnoAssegnazioneUtil”, ognuno dei quali non può essere disponibile e offerto a tutti gli utenti. In base al login effettuato da un determinato tipo di utente è possibile stabilire quale interfaccia e servizio del sottosistema o intero sottosistema può o non può accedere. Tale gestione può essere simulata mediante una ACL (Access Control List) per ogni sottosistema associando una lista di coppie (attore, operazione) ad ogni classe da accedere. Ogni volta che un utente loggato accede al sistema è controllata la lista degli accessi, il conseguente attore e la relativa operazione invocata determinando o meno la possibilità d'accesso.

3.6 - Decisioni sul flusso di controllo globale

La numerosità di richieste sottomesse al server porta alla naturale scelta di un flusso di controllo thread based. Ciò porta ad una necessaria gestione della concorrenza di tali richieste. Il pattern Facade, utilizzato tra gli altri con l'oggetto API, individua quindi un thread control che deve garantire l'accesso da parte di più utenti in modo concorrente ai servizi del server.

3.7 - Condizioni limite

Il sistema software IPT è pensato per essere installato, avviato ed eventualmente modificato e/o riavviato in caso di manutenzione dello stesso da parte del gruppo di sviluppo IPT o in alternativa da una figura tecnica adeguata individuata presso l'azienda di trasporto urbano di competenza. Il sistema IPT, infatti, benché molto complesso nella sua struttura, prevede un'installazione e un avvio molto semplice.

Qualora l'azienda di trasporto urbano dovesse avere già a disposizione una o più macchine server sul quale installare tale sistema software IPT, è possibile usarle in parte o completamente a tale scopo. Infine, si prevede che il sistema una volta installato, venga affiancato da un'unità di backup persistente sufficientemente ampia che sistematicamente realizza il backup di parte o di tutti i dati fino a quel punto elaborati.

Servizi del sottosistema

4 - Servizi del sottosistema

diagrammaSottosistemiServiziOfferti::diagrammaSottosistemiServiziOfferti

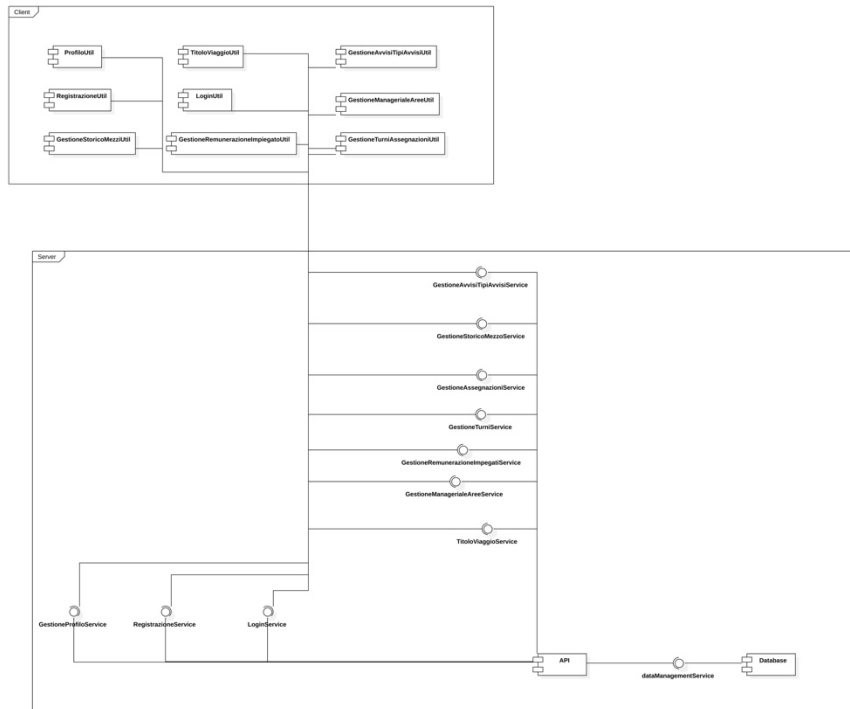


Figura 1.3: Diagramma Sottosistemi software con annessi servizi offerti

Glossario

5 – Glossario

La specifica della terminologia utilizzata è stata già affrontata a più riprese durante la stesura del documento RAD e del presente documento SDD. Si consulti rispettivamente il capitolo 1 e 4 del RAD e il capitolo 1 e 3 del presente SDD.