

“Software Engineering” Course a.a. 2019-2020

Lecturer: Prof. Henry Muccini (henry.muccini@univaq.it)

Progetto 5: Portami a destinazione

Date	13/02/2020
Deliverable	Documento Finale – D3
Team (Name)	Team Banana

Team Members		
Name & Surname	Matriculation Number	E-mail address
Tagliente Gabriele	247295	gabriele.tagliente@student.univaq.it
Innamorati Giorgio	253380	giorgio.innamorati@student.univaq.it
Sborlini Valentina	247975	valentina.sborlini@student.univaq.it

Table of Contents of this deliverable

List of Challenging/Risky Requirements or Tasks	3
A. Stato dell'Arte	5
B. Raffinamento dei Requisiti	7
B.1 Servizi (con prioritizzazione)	7
B.2 Requisiti non Funzionali	9
B.3 Scenari d'uso dettagliati	10
B.4 Excluded Requirements	12
B.5 Assunzioni	13
B.6 Use Case Diagrams	13
C. Architettura Software	14
C.1 The static view of the system: Component Diagram	14
C.2 The dynamic view of the software architecture: Sequence Diagram	17
D. Dati e loro modellazione	19
E. Design Decisions	20
F. Design di Basso Livello	21
G. Explain how the FRs and the NFRs are satisfied by design	24
H. Effort Recording	26
PERT	26
Logging	27
Categorization	30
Summary Statistics	30
Appendix. Prototype	31
Implementazione	31
Test delle Performance	33
Documentazione Postman	34

List of Challenging/Risky Requirements or Tasks

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Manutenzione del database	22/11/2019	25/11/2019	Il database un giorno prestabilito della settimana dovrà obbligatoriamente effettuare un backup su disco rigido. Così nel caso in cui il nostro database dovesse avere un guasto, non andremo a perdere tutti i dati e il lavoro precedentemente salvati.
Segnalazione di nuovi POI	23/11/2019	24/11/2019	Non offriamo l'opportunità allo studente di segnalare nuovi POI in quanto non abbiamo abbastanza personale in grado di recarsi in loco e controllare se effettivamente sia veritiero o meno. L'aggiunta di nuovi POI avverrà solo dopo una notifica da parte dell'Università stessa.
Chat Supporto Studente-Università	23/11/2019	26/11/2019	Non offriamo una chat studente-supporto applicazione in quanto, anche qui, siamo sprovvisti di personale e di conseguenza si rischierebbe che i messaggi non vengano letti o vengano letti troppo in ritardo rispetto all'invio. Però, in caso uno studente abbia dei problemi ha la possibilità di inviare un'email alla segreteria della studenti, e poi sarà lei a girare le relative segnalazioni ai gestori della piattaforma.
Utenti non registrati	07/12/2019	10/12/2019	Sarà loro concesso di utilizzare ogni funzione che non necessita di informazioni sull'utente (es. visualizzazione POI, ricerca, filtraggio, navigazione, ...) inibendo solamente le funzioni strettamente legate allo studente (es. filtraggio per CdL, aggiornamento dati personali, ...)
Online vs Offline	08/01/2020	10/01/2020	L'applicativo dovrà necessariamente funzionare con una connessione ad internet, soprattutto per:

			1) L'interrogazione del server per recuperare i POI, effettuare il login, ... 2) visualizzare la mappa
Mappa embedded	13/01/2020	14/01/2020	La mappa sarà importata dall'applicativo dinamicamente e tramite una connessione a internet.
Ricordo credenziali	16/01/2020	16/01/2020	Le credenziali di accesso, in quanto personali, sono molto delicate e quindi difficili nel trattarle. In un primo momento non potranno essere ricordate, sarà un progetto per il futuro o se la richiesta sarà sufficientemente alta.

A. Stato dell'Arte

Prima di creare questo sistema ci siamo guardati intorno, e navigando in rete abbiamo trovato queste altre università italiane che possono offrire agli studenti dei sistemi innovativi come POI.

Il migliore che si può trovare in rete da cui prendere esempio è quello della SAPIENZA DI ROMA, di cui lasciamo il link qui sotto.

<https://virtualtour.uniroma1.it/>

Infatti è possibile vedere tutti i POI su di una mappa, su cui l'utente può fare zoom-in, zoom-out e scorrere a piacimento. Inoltre dispone di una categorizzazione dei POI per servizio (consentendo il filtraggio degli stessi da visualizzare sulla mappa) e per Corso di Laurea. Infine è possibile cercare un POI per nome (anche parziale) e iniziare una navigazione per un POI.

Pro

- visualizzazione POI su mappa
- colori diversi per servizi diversi
- mappe di Google con utilizzo di GeoCMS

Contro

- non è possibile visualizzare informazioni sui POI
- non è possibile navigare verso un qualsiasi POI, ma solo verso edifici

Un altro sito di esempio è quello dell'Università di Ferrara.

<http://www.unife.it/it/internazionale/venire/mappa-di-ferrara>

Questo applicativo consente di visualizzare i POI su di una mappa realizzata con Google My Maps. Anche qui è possibile filtrare per tipo di servizio dei POI. La particolarità è che quando si vuole iniziare una navigazione verso un POI, si viene reindirizzati su Google Maps, sfruttando di fatto i servizi già offerti da Google.

Pro

- visualizzazione dei POI su mappa, con colori diversi in base al servizio
- nella scheda delle informazioni aggiuntive è possibile attivare una navigazione verso il POI
 - in tal caso viene aperto Google Maps
- mappe di GoogleMyMaps

Contro

- non è possibile effettuare una ricerca per nome

Un ultimo esempio è quello dell'Università di Bologna, a cui si affidano anche altre università come quella di Cesena, Forlì, Ravenna e Rimini.

<https://www.unibo.it/uniboWeb/unibomappe/default.aspx?site=bologna>

Questo mostra informazioni dei POI su di una mappa, ma purtroppo non sembra essere stato popolato di punti di interesse. Da qui possiamo imparare che, sebbene l'applicativo sia funzionante, è molto importante fare il data entry dei POI.

Pro

- informazioni aggiuntive cliccando sui POI
- possibilità di essere reindirizzati sulle mappe di Google
- ricerca per nome mostra sia una lista che i corrispondenti POI sulla mappa
- mappe di OpenStreetMaps

Contro

- filtraggio per servizio disponibile ma non funzionante
- mappa non popolata di POI
- ricerca non funzionante per sottostringhe
- OpenStreetMap non consente la creazione di mappe dinamiche, con caratteristiche di scalabilità

B. Raffinamento dei Requisiti

B.1 Servizi (con prioritizzazione)

- 1) Mappare tutti i POI di un Ateneo in una mappa. (Priorità **ALTA**)
- 2) Categorizzare i POI in base a CdL-specifico, Dipartimento-specifico, Università; (Priorità **MEDIA**)
- 3) Se lo studente è registrato prioritizzare i POI del CdL e Dipartimento; (Priorità **MEDIA**)
- 4) Permettere allo studente di localizzare in modo semplice quanto di suo interesse (ovvero, risalire dal servizio necessario, al POI che fornisce tale servizio); (Priorità **BASSA**)
- 5) Mappare i POI in mappa con icone diverse a seconda del servizio offerto; (Priorità **BASSA**)
- 6) Utilizzare comandi vocali; (Priorità **BASSA**)

REQUISITI FONDAMENTALI

1. L'Utente Generico dell'Applicazione

2. Deve poter Scaricare l'Applicazione

- 2.1. Può Accedere alla Mappa con tutti i POI interni alla sua Università
- 2.2. Deve poter visualizzare nella Mappa i POI colorati/rappresentati diversamente in base al servizio offerto
- 2.3. deve poter usare i comandi vocali:
 - 2.3.1. per inserire una query di ricerca
 - 2.3.2. per raggiungere un determinato POI
 - 2.3.3. per catalogare i poi in base ad una nuova ricerca
 - 2.3.4. per registrarsi nell'applicazione (caso non vedente)
 - 2.3.5. per loggarsi nell'applicazione (caso non vedente)

3. Lo Studente

3.1. Deve potersi Registrare (una tantum)

- 3.1.1. Deve fornire le proprie Credenziali Universitarie
- 3.1.2. Deve fornire i propri Dati Anagrafici e Recapiti (email)
- 3.1.3. Deve fornire i propri Dati Universitari (i.e. matricola, CdL, ...)

3.2. Deve potersi Loggare

- 3.2.1. Deve poter Accedere all'Applicazione attraverso le Credenziali Universitarie

3.3. Può Prioritizzare nella Mappa i POI in base ai proprio CdL, Dipartimento e Sede di appartenenza

3.4. Può Selezionare i propri Interessi in una sezione dedicata nel proprio profilo personale

- 3.4.1. Può Filtrare i POI in base a tali interessi

4. L'Amministratore

4.1. Deve poter accedere ad un'Interfaccia di amministrazione

4.2. Può inserire/modificare/eliminare i POI

B.2 Requisiti non Funzionali

- Per creare un account bisogna essere in possesso di credenziali Universitarie.
 - Il sistema deve rispettare il Regolamento Europeo in materia di Privacy UE 2016/679
 - Il sistema deve rispettare il Regolamento Europeo in materia dei cookie UE 2016/679
 - Il sistema è indipendente da ogni singola università
 - Il sistema deve avere una mappa perennemente aggiornata della propria università
 - Il sistema si deve integrare con il calendario docenti, in modo tale per permettere di trovare il Docente (POI) nel proprio orario di ricevimento all'interno della mappa
 - Deve poter funzionare con molteplici sedi, una quantità importante di POI e numerosi studenti
 - Deve poter funzionare con molteplici dispositivi (IOS, Android, ...)
 - Il sistema deve garantire una garanzia di utilizzo 16 ore su 24 (sono escluse le ore notturne in cui gli studenti non possono essere in università, in quanto quest'ultima è chiusa).
 - Il sistema deve prevedere un backup dei dati in caso di guasti
 - Il sistema deve avere un database capace di elaborare la richiesta di ogni singolo studente in pochi secondi.
 - Il sistema deve assicurarsi che nel caso un POI non è disponibile al momento (ristrutturazione, ...) non venga visualizzato nella mappa.
-

B.3 Scenari d'uso dettagliati

- Registrazione/Login utente tramite credenziali terze
 - un utente può effettuare la registrazione/login in qualsiasi momento cliccando sull'apposito pulsante
 - verrà mostrata una pagina in cui gli verrà chiesto di inserire le proprie credenziali universitarie
 - il sistema controllerà che le credenziali siano corrette e in tal caso loggerà l'utente, altrimenti verrà chiesto di reinserirle
 - in fase di registrazione il Sistema preleverà dal database universitario (se il sistema dell'università lo permette) le informazioni dello Studente quali matricola, username, email, password, corso di laurea, ...
- Operazioni su mappa

Rappresenta lo scenario più comune. Sulla mappa l'utente è in grado di fare:

 - scroll della mappa
 - l'utente può muovere la mappa a suo piacimento per visualizzare zone diverse e di suo interesse
 - l'applicazione, a questo punto, caricherà la mappa di quella determinata area e contestualmente recupererà i POI associati da visualizzare
 - pinch sulla mappa
 - l'utente può effettuare uno zoom-in o zoom-out della mappa
 - l'applicazione caricherà una versione più o meno dettagliata della mappa, in base a quanto richiesto
 - tap su un POI
 - l'utente ha la possibilità di accedere alle informazioni di un determinato POI semplicemente cliccando su di esso
 - nell'applicazione apparirà una scheda a comparsa con le informazioni del POI selezionato
 - alla fine di queste informazioni saranno presenti dei pulsanti di azioni dove l'utente potrà scegliere il da farsi (es. navigazione, ulteriori informazioni, aggiungere/eliminare ai POI preferiti)
 - Se l'utente, invece, pensa di aver concluso l'operazione con quel determinato POI può semplicemente chiudere la scheda tramite il pulsante 'x' e tornare alla mappa

- selezionare i servizi da visualizzare
 - l'utente ha la possibilità di filtrare i POI in base ai servizi desiderati (es. caffetteria, biblioteca, parco, fermate autobus, ...)
 - a questo punto, l'applicazione visualizzerà sulla mappa solo i POI che offrono i servizi scelti
 - i POI relativi ai diversi servizi verranno visualizzati sulla mappa con colori/icone diversi per aumentare la velocità di comprensione da parte dell'utente
- ricerca
 - l'utente può inserire nella barra di ricerca una stringa, ossia la parola chiave
 - l'applicazione mostrerà come risultato una lista di POI filtrati in base alla stringa inserita
 - l'utente ha la possibilità di cliccare su un elemento della lista per visualizzare il POI sulla mappa e le relative informazioni associate
- Navigazione verso un POI
 - l'utente clicca sul pulsante per avviare la navigazione verso un POI
 - verrà chiesto di inserire la posizione di partenza
 - se il servizio di geolocalizzazione è attivo, allora sarà possibile selezionare la posizione attuale dell'utente
 - altrimenti, l'utente dovrà inserirla manualmente
 - facendo clic sul pulsante di start, verrà avviata la navigazione
 - sulla mappa verrà mostrato il percorso per raggiungere la destinazione
 - verranno inoltre mostrate le indicazioni passo passo per raggiungere la destinazione
 - l'utente può, inoltre, avviare la navigazione vocale, che lo guiderà a destinazione tramite comandi vocali
- Prioritizzazione per CdL
 - un'opzione che può essere utilizzata solo da utenti che avranno effettuato il login
 - dopo essersi loggato, lo studente potrà filtrare i POI in base al proprio Corso di Laurea tramite un flag
 - lo studente può cliccare su un apposito pulsante per disattivare o attivare l'opzione in qualsiasi momento
 - quando è attiva:
 - sulla mappa verranno visualizzati solo i POI inerenti il CdL dello studente
 - la ricerca sarà effettuata solo sui POI inerenti il CdL dello studente

B.4 Excluded Requirements

Tra i **Servizi Esclusi**, possiamo trovare:

Recensioni dei POI.

Ogni Studente, una volta arrivato a destinazione non potrà effettuare recensioni sull'ultimo POI recentemente visitato, in quanto sennò bisognerebbe creare una sezione apposita.

Magari in un'altra versione dell' applicazione.

Segnalazione nuovi POI

Ogni Studente che vuole segnalare un nuovo POI non potrà in alcun modo segnalare nuovi POI attraverso form interne al sistema in quanto, potrebbero arrivare numerose segnalazioni fittizie e non avendo personale a disposizione sarebbe abbastanza complicato gestirle.

Chat Supporto Studente-Gestore App

Nel caso in cui uno studente non riesce ad accedere all'Applicazione, Non gli vengono aggiornati i POI in base al suo CDL o sede universitaria, oppure semplicemente non trova i POI di suo interesse, potrà soltanto inviare una mail di assistenza alla segreteria studenti e non potrà ricevere una live chat di supporto, sempre per il motivo di mancanza di personale a disposizione.

Si riparerà il guasto non appena arriverà segnalazione dalla segreteria studenti.

No Criptazione dei dati

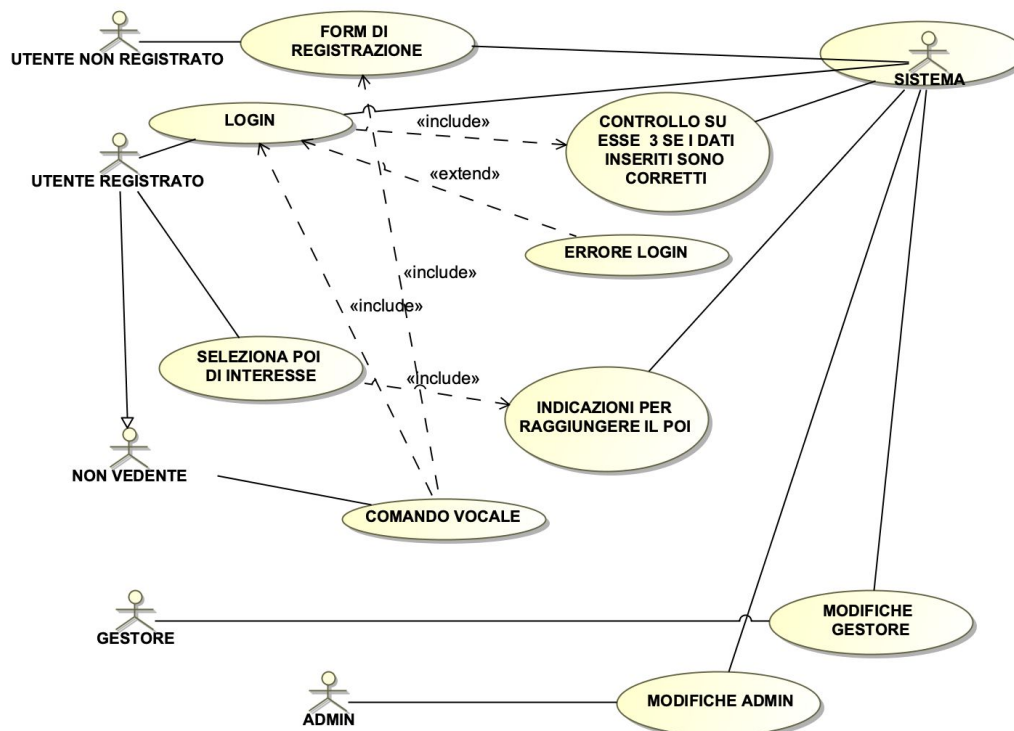
Abbiamo deciso in gruppo che non verrà effettuata nessuna criptazione dei dati, in quanto il sistema non presenta dati che possano essere definiti SENSIBILI.

Ad eccezione dei dati di login che per **LEGGE** devono essere nascosti ai proprietari dell' applicazione.

B.5 Assunzioni

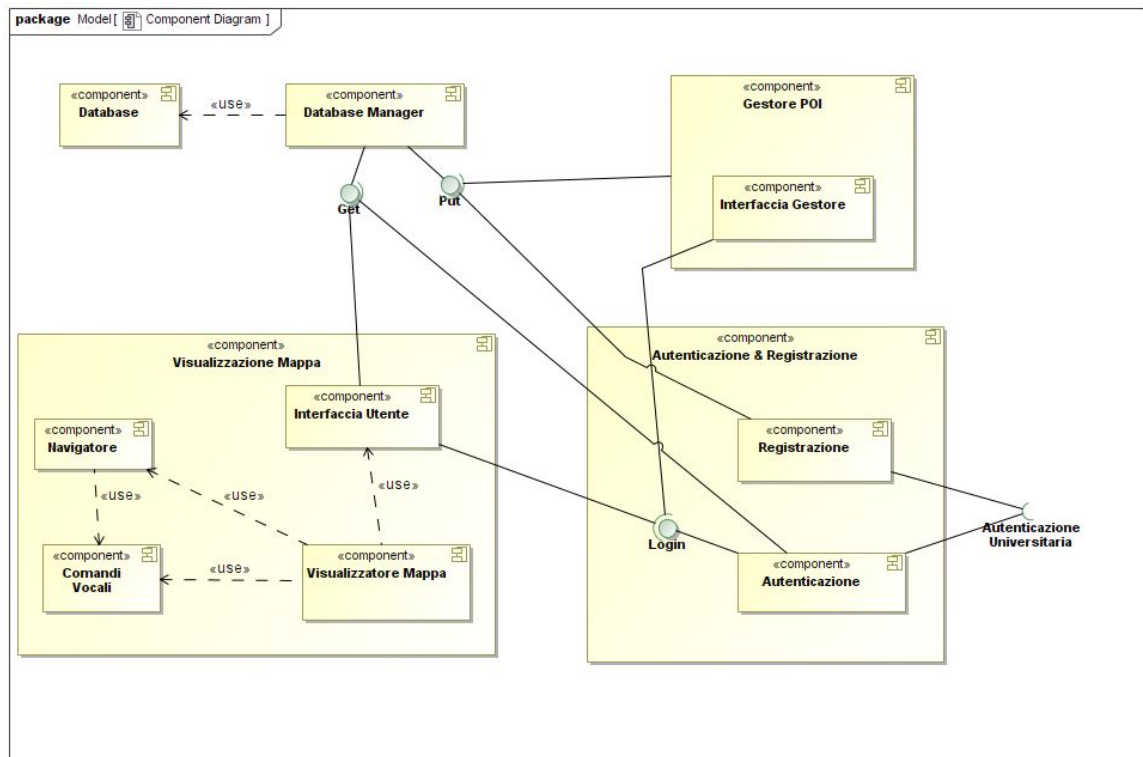
- **Online:** supponiamo che il dispositivo dispone di una connessione a internet, per l'interrogazione del server
- **Mappa:** supponiamo di poter utilizzare e importare le mappe stradali/satellitari messe a disposizione da soggetti terzi (e.g. Google, Bing, OpenStreetMaps)
Per ovvie ragioni queste mappe saranno utilizzate online

B.6 Use Case Diagrams



C. Architettura Software

C.1 The static view of the system: Component Diagram



Il component Diagram viene utilizzato per rappresentare la struttura interna dell'applicazione, modellando i componenti principali e le loro relazioni.

I componenti principali del nostro software sono i seguenti:

1. Login/Registrazione:

I dati che arrivano dall'autenticazione tramite credenziali universitarie (p.e. ESSE3 o simili) per le università convenzionate vengono processate dal sistema per controllare la veridicità dei dati inseriti nel form.

È quella parte del sistema che si occupa della gestione dell'utente, della sua autenticazione e del recupero dei dati nel database.

2. Mappa:

Viene intesa come quella parte di software che comprende tutti i componenti che costituiscono l'interfaccia di un utente generico o studente.

3. Interfaccia Utente:

Si occupa di recuperare tutti i dati che verranno visualizzati sulla mappa, gestendo sia l'utente generico che lo studente loggato.

In tal caso si occupa di filtrare in base al proprio CDL, alla sede universitaria di appartenenza oppure in base ai propri interessi.

4. Visualizzatore Mappa:

Costituisce la componente che visualizza i POI sulla mappa e rappresenta l'interfaccia di un utente generico o di uno studente.

5. Navigatore:

Viene definito Tragitto tutta quella parte software chiamata tramite API a tool esterno che porta il nostro Utente (Studente Universitario) dalla propria posizione corrente a destinazione (POI selezionato).

6. Comando Vocale:

Questa componente invece, si occupa dell'utilizzo dell'applicazione da parte degli utenti non vedenti, e li aiuta ad interagire con la mappa sia in input (parlando) che in output (ascoltando). Il comando vocale non è altro che un traduttore che trasforma voce in testo e invia i dati inseriti dall'utente con il database.

7. Gestore POI:

Questa componente consente l'amministrazione e la gestione dei POI, nonché di tutti i dati loro relativi come i servizi di appartenenza, la disponibilità, ...
Rappresenta dunque il backoffice del sistema.

8. Interfaccia Gestore:

È la parte del sistema accessibile dai gestori, essi dopo aver superato la fase di login, potranno andare a modificare la parte dei dati relativa ai corsi di studi degli studenti, oppure potranno direttamente agire sui POI (cancellarli, aggiungerli, segnalarli temporaneamente offline, ...).

9. Database Manager:

Componente che si occupa della gestione del database, dell'inserimento e della presentazione dei dati in esso contenuti, offrendo un'interfaccia unica per tutte le componenti che necessitano di interagire con il DB. L'interfaccia sarà realizzata tramite le API.

10. Autenticazione & Registrazione:

Questo modulo raccoglie tutte le componenti che si occupano della registrazione e autenticazione degli utenti, e della gestione degli stessi.

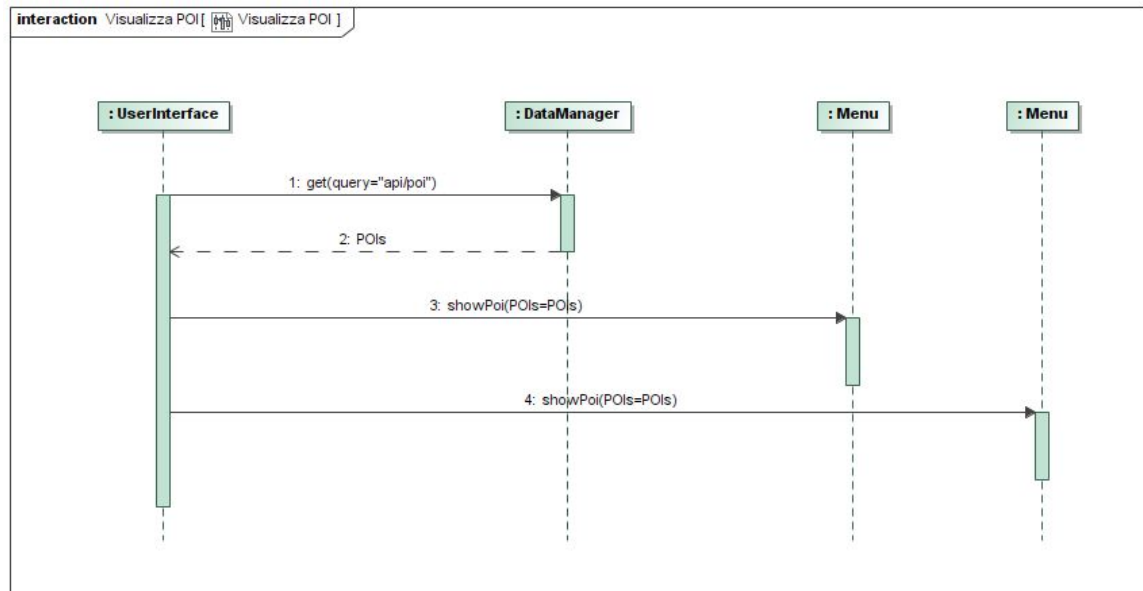
11. Registrazione:

Componente che si occupa della registrazione di un nuovo utente. Nel caso sia un utente locale, esso viene semplicemente inserito nel database di utenti. Nel caso in cui sia un studente che utilizza le credenziali di ateneo, questo modulo si occuperà prima di verificare la correttezza delle credenziali, e solo successivamente della creazione di un entry nel database.

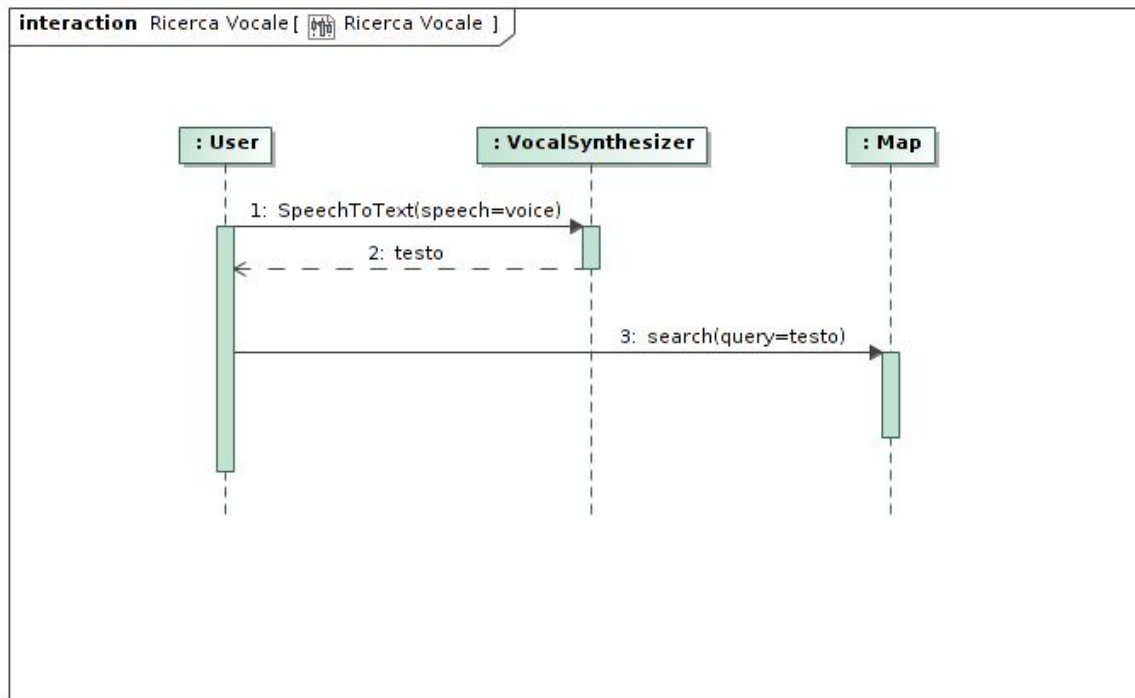
12. Autenticazione:

Componente che effettua l'autenticazione di un utente, attraverso l'apposito form di login. L'autenticazione può avvenire o con credenziali salvate sul database o con credenziali esterne salvate sul sistema gestito dall'università.

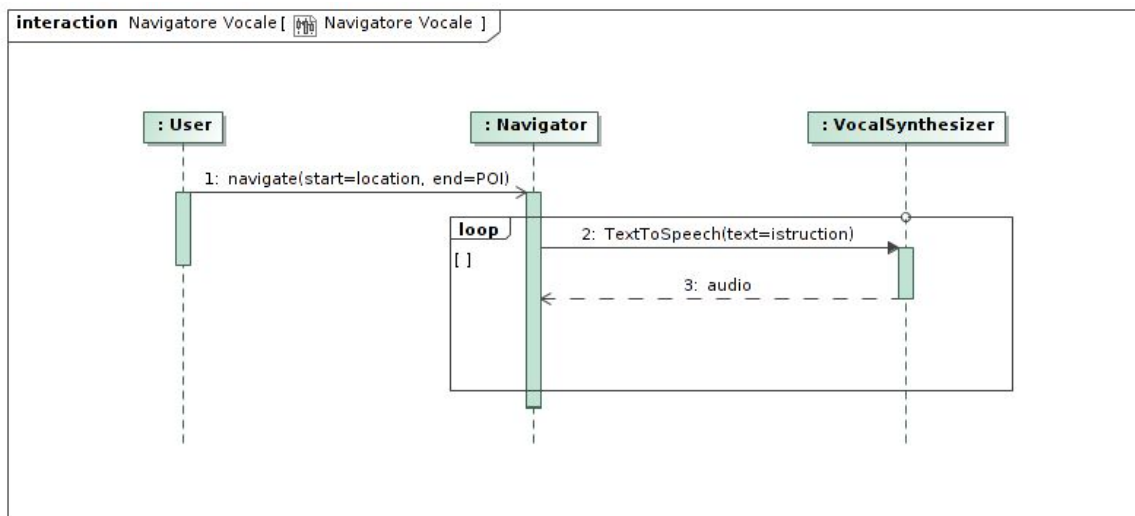
C.2 The dynamic view of the software architecture: Sequence Diagram



Presentazione dei POI sulla mappa: L'interfaccia dell'utente recupera i POI tramite le API offerte dal DataManager. Successivamente vengono passati alla mappa e al menù per essere elaborati e mostrati.

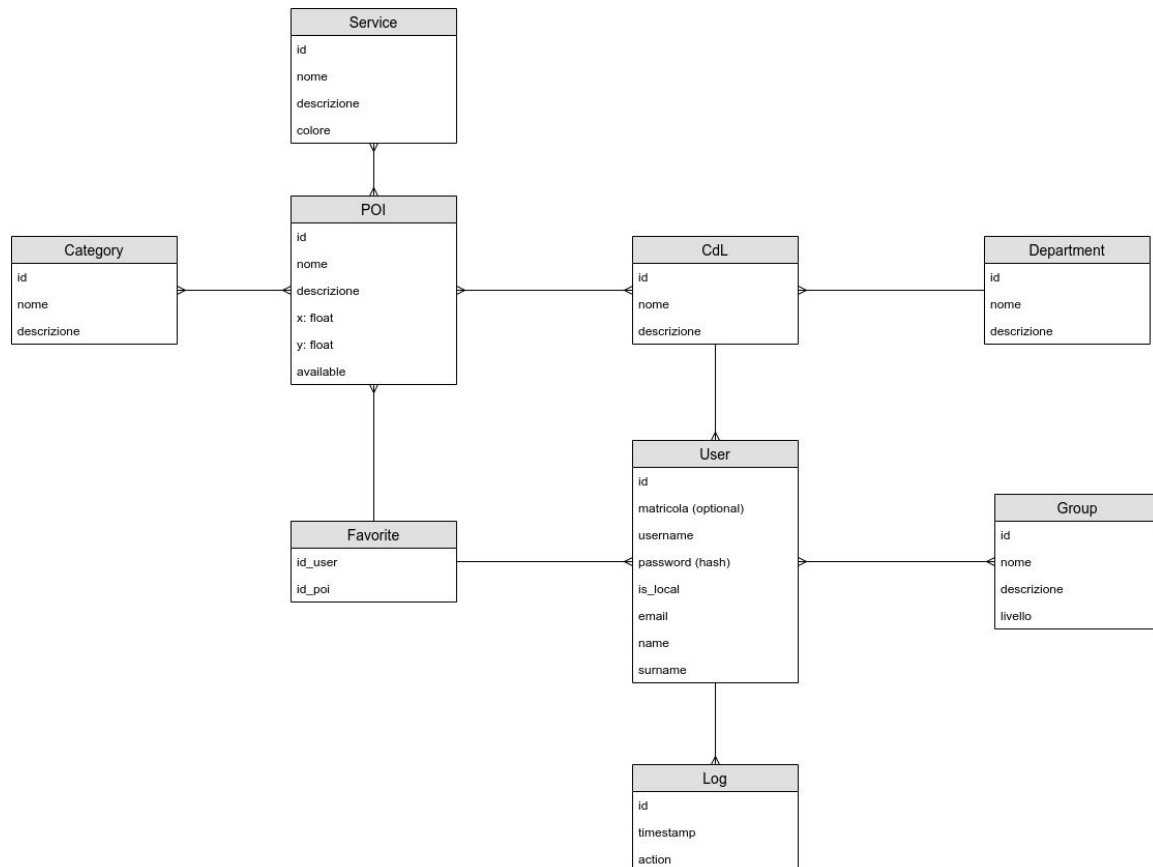


Ricerca di POI tramite input vocale: l'utente richiede al sintetizzatore vocale di convertire la voce in testo, il quale sarà utilizzato come query di ricerca sulla mappa.



Navigazione con comandi vocali: quando l'utente inizia la navigazione verso il POI desiderato, il navigatore oltre a fornire le indicazioni sullo schermo le “legge” con l'ausilio del sintetizzatore vocale. In particolare cicla su tutte le istruzioni per convertire le stringhe di testo in audio da poter riprodurre.

D. Dati e loro modellazione



I dati per il login utente, quali email e password, verranno inseriti per la prima volta dall'utente non autenticato, in fase di registrazione. Questi dati possono essere utilizzati per recuperare le altre informazioni relative allo studente, attraverso un sistema centralizzato di gestione dei dati dell'università (e.g. ESSE3, ma non solo).

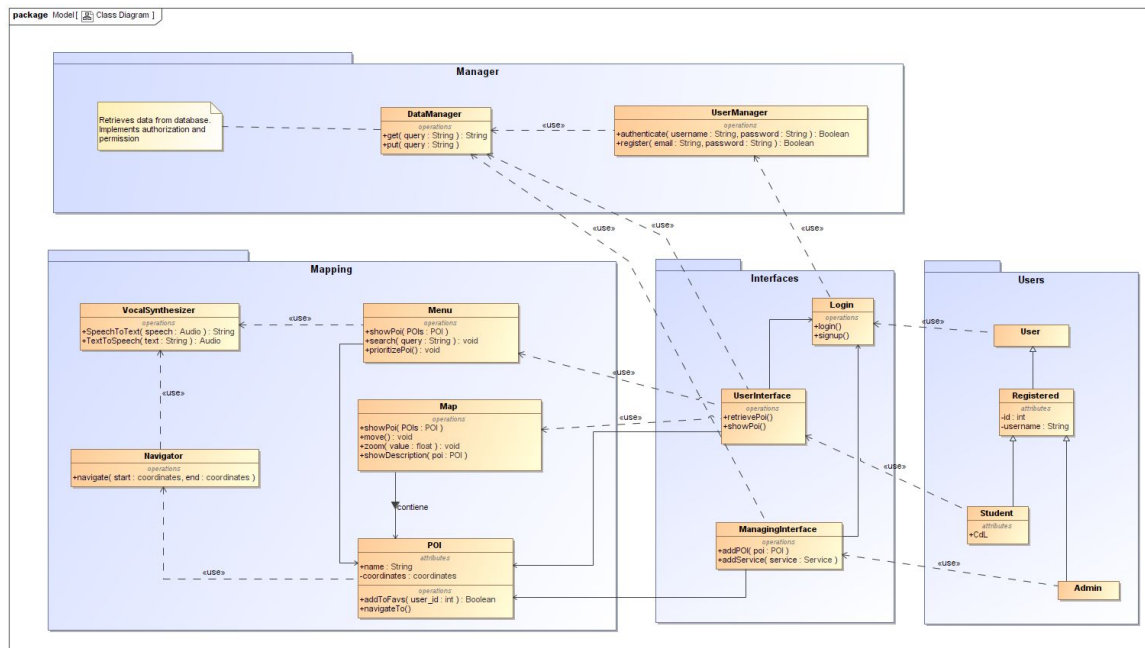
Se l'università non dispone di tale sistema dovrà essere disposto un data entry delle informazioni sul corso di laurea frequentato direttamente dall'utente utilizzatore.

E. Design Decisions

Per quanto riguarda il Design Decision, avevamo pensato di proporre:

1. Un'interfaccia amichevole e con caratteri ben leggibili, così da poter essere utilizzata da tutti gli utenti senza fatica e in modo veloce.
2. La schermata di Login presenterà 2 opzioni: login tramite ESSE3 e registrazione.
3. Nella fase della registrazione saranno richiesti pochi elementi, come il nome, cognome, email e password, in quanto deve essere un'applicazione di utilizzo veloce.
4. La schermata principale presenterà il logo dell'università nel momento in cui si sta caricando la mappa (BUFFERING).
Una volta caricata la Mappa, l'app presenterà subito la mappa con i POI.
5. L'utente sarà in grado di cliccare su di un POI sulla mappa per mostrare informazioni aggiuntive, come la descrizione e i servizi offerti, nonché per avviare una navigazione verso il Punto scelto.
6. A sinistra della mappa sarà presente un menù ad hamburger che permette la ricerca dei POI, la visualizzazione del proprio account, il logout dall'account e la possibilità di inoltrare una richiesta di assistenza (presente anche nella schermata di Login).
7. Per quanto riguarda la ricerca di un POI, l'utente potrà inserire una stringa di ricerca nell'apposita barra e successivamente verrà mostrato
 - da un lato l'elenco di Punti che contengono nel proprio nome la stringa cercata, prioritizzati per il Corso di Laurea dell'utente che effettua la ricerca,
 - mentre al centro la mappa con soli i Punti che corrispondono ai criteri di ricerca
8. Nel momento del logout l'utente dovrà confermare l'uscita, tramite una finestra a comparsa

F. Design di Basso Livello



Il diagramma sopra raffigurato è diviso in quattro package:

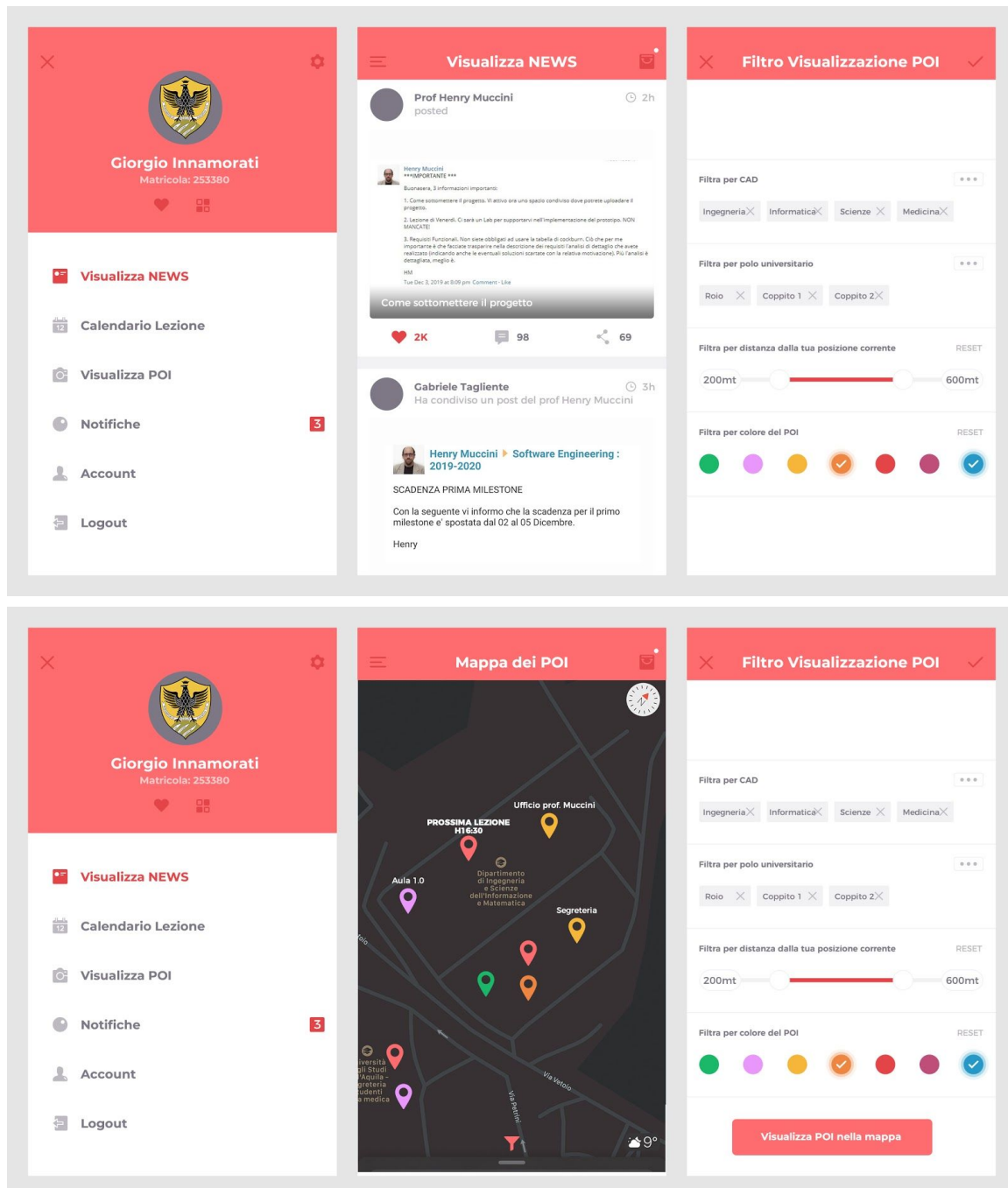
1. **Users:** contiene le classi che definiscono le diverse situazioni di utente che viene ad interagire con il sistema;
2. **Interfaces:** contiene le classi che implementano l'interfaccia utente
3. **Mapping:** contiene le classi che implementeranno la mappa e tutte le funzioni ad essa collegate;
4. **Manager:** contiene quelle classi che servono per la gestione del sistema, quali il gestore del database e il gestore delle utente.

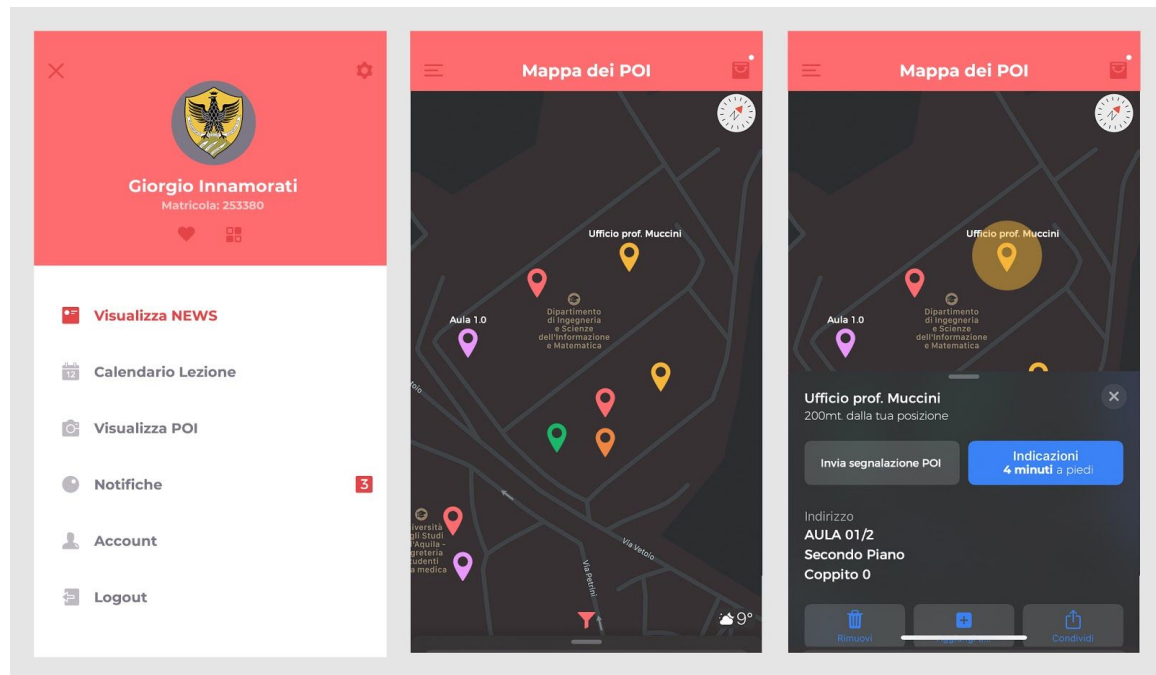
Un utente generico (non autenticato) accederà ad un'interfaccia tramite la quale può decidere se autenticarsi oppure proseguire sulla mappa e avere accesso alle funzionalità disponibili. Alcune di queste funzionalità sono gestite da classi separate, come il navigatore o il sintetizzatore vocale (per permettere all'utente di interagire tramite comandi vocali). Se autenticato, uno studente potrà inoltre prioritizzare i POI in base al suo CdL e aggiungere un POI ai suoi preferiti.

La gestione dei POI e dei servizi è affidata solo ad utenti amministratori che accedono al sistema tramite un'interfaccia di gestione.

Ad ausilio delle altre classi, per la gestione di particolari componenti, definiamo una classe per la gestione del database, alla quale ci relazioniamo attraverso opportune API,

ed un'altra per la gestione delle utenze (e.g. registrazione, autenticazione, mantenimento della sessione)





G. Explain how the FRs and the NFRs are satisfied by design

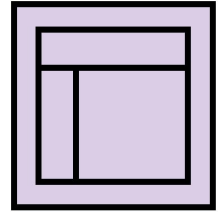
QUALITÀ DEL PRODOTTO SISTEMA/SOFTWARE

- Efficienza nella prestazioni
Per quanto riguarda il time behaviour, deve essere breve in quanto lo studente deve ricevere una risposta in poco tempo da parte del sistema. Quindi la richiesta dell'utente viene mandata al sistema, che deve elaborare la richiesta in pochi secondi, ed inviare la risposta sul display.
Il database e il sistema devono essere in grado di gestire una quantità abbastanza elevata di dati nel più breve tempo possibile.
- Usabilità
Il progetto presenta un'interfaccia basilare e semplice così da poter essere utilizzata da tutti gli studenti.
E' veloce, in quanto vengono salvate nel database le preferenze e gli storici dei singoli utilizzatori.
- Affidabilità
E' la capacità di un sistema o componente di eseguire le funzioni richieste in base a quanto indicato per un periodo di tempo specificato, per questo, il progetto presenta un tempo medio accettabile per guasti e la capacità di rilevare guasti specifici o di resistere ad 'attacchi' esterni.
- Sicurezza
Non è uno dei punti fondamentali e forti di questo progetto, in quanto non vengono trattati dati sensibili, ma solo dati anagrafici che vengono 'protetti' in base alle norme presenti sulla privacy.
- Portabilità
Presenta un'alta portabilità, in quanto può essere usata su diverse piattaforme come ad esempio IOS e Android.

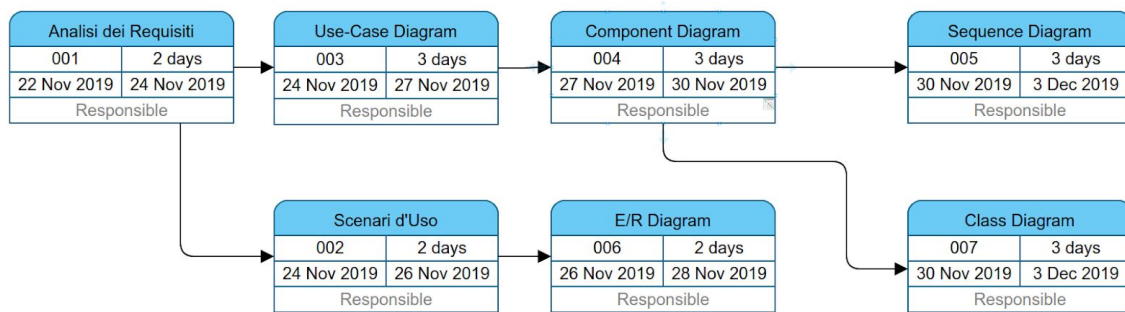
QUALITÀ IN USO

- Efficacia ed efficienza
Vogliamo avere il massimo risultato con il minimo sforzo, per questo vogliamo offrire un servizio che sia di gradimento all'utilizzatore finale. Deve avere quantità e qualità.
- Soddisfazione
Altro punto fondamentale, lo studente deve essere soddisfatto del servizio offerto, deve sentirsi sicuro nel immettere i propri dati e fornire la sua posizione. Per l'utente deve poter essere un piacere utilizzare il sistema e non essere un peso.
- Libero da rischi
L'utilizzatore dell'app, non va incontro a dei particolari rischi, ad esempio economici in quanto non inserisce dati relativi ai propri conti in bancari, di salute psico-fisica in quanto non sono previsti pop-up o pubblicità che possono nuocere ad uno studente più sensibile.
- Copertura del contesto
Il progetto, copre unicamente il contesto universitario, quindi come già detto si può accedere all'app solamente nelle ore in cui l'ateneo è aperto al pubblico.

H. Effort Recording



PERT



Logging

Personal Journal

Team (number and name): TEAM BANANA
 Student name: GIORGIO INNAMORATI
 Student number: 253380
 Email: giorgio.innamorati@student.univaq.it

When (Month/Day)	Time spent	Partners (please report how many people have been working)	Brief Description of the performed task	Category	Sub-Category
11	22	1H e 30MIN	GABRIELE T	ANALISI DEI REQUISITI	LEARNING
11	24	0H e 45MIN		USE CASE DIAGRAM	LEARNING
11	25	2H e 00MIN		USE CASE + REQUISITI	DOING
11	28	2H e 30MIN	GABRIELE T	COMPONENT DIAGRAM	DOING
11	29	0H e 30MIN		COMPONENT DIAGRAM	DOING
11	30	1H e 30MIN	VALENTINA	BRIEFING – CLASS DIAGRAM	LEARNING
12	1	2H e 00MIN		REVISIONE LAVORO	LEARNING
12	2	1H e 30MIN		SEQUENCE DIAGRAM	DOING
12	3	1H e 30MIN		SEQUENCE DIAGRAM	DOING
12	4	2H e 00MIN		REVISIONE LAVORO	DOING
12	5	0H e 30MIN		REVISIONE LAVORO	DOING
12	10	1H e 40MIN		STATO DELL'ARTE	DOING
12	15	2H e 30MIN		API	LEARNING
12	19	2H e 00MIN		DEFINIZIONE API	DOING
12	20	2H e 40MIN	GABRIELE T, VALENTINA	DOCUMENTAZIONE API	DOING
1	8	1H e 30MIN		REQUISITI FUNZIONALI	DOING
1	10	1H e 00MIN		DATI	LEARNING
1	10	1H e 30MIN		DATI	DOING
1	14	2H e 00MIN		API	LEARNING
1	15	3H e 00MIN		DESIGN BASSO LIVELLO	DOING
1	17	3H e 00MIN	GABRIELE T, VALENTINA	API	DOING
1	21	3H e 30MIN		API	DOING
1	23	3H e 00MIN	GABRIELE T	API	DOING
1	30	2H e 45MIN		PROTOTIPO	DOING
2	1	3H e 00MIN		API	DOING
2	4	3H e 15MIN		API	DOING
2	10	3H e 30MIN	GABRIELE T	PROTOTIPO	DOING
2	12	3H e 00MIN	GABRIELE T, VALENTINA	REVISIONE LAVORO	DOING

Personal Journal						
Team (number and name): team banana						
Student name: Gabriele Tagliente						
Student number: 247295						
Email: gabriele.tagliente@student.univaq.it						
When (Month/Day)	Time spent	Partners (please report how many people have been working)	Brief Description of the performed task	Category	Sub-Category	
11 22	01:00	1	sviluppo del grafico PERT	Doing	Effort Recording	
11 22	01:30	2: Giorgio	elenco dei requisiti funzionali e non funzionali	Doing	Analisi dei Requisiti	
11 25	02:00	1	modellazione e UML	Learning	Modeling	
11 26	01:00	1	ricerca e studio di applicativi esistenti	Doing	Stato dell'Arte	
11 26	01:30	2: Valentina	descrizione degli scenari d'uso	Doing	Scenari D'Uso	
11 27	03:00	1	Use-Case, Component e Class Diagrams	Learning	Diagrammi	
11 28	02:30	2: Giorgio	sviluppo Component Diagram	Doing	Component Diagram	
12 1	02:30	1	sviluppo Class Diagram	Doing	Design di Basso Livello	
12 10	03:00	1	costruzione API	Learning	Web Services	
12 13	01:30	1	revisione e miglioramento	Doing	Scenari D'Uso	
12 15	02:00	1	revisione e miglioramento	Doing	Component Diagram	
12 17	01:30	1	revisione e miglioramento	Doing	Class Diagram	
12 19	02:00	3: Giorgio, Valentina	definizione API	Doing	Prototipo	
1 9	01:30	1	revisione e miglioramento	Doing	Stato dell'Arte	
1 11	01:00	1	strutturazione e contenuti	Learning	Scenari D'Uso	
1 12	02:45	2: Valentina	revisione e miglioramento	Doing	Scenari D'Uso	
1 14	01:15	1	revisione e miglioramento	Doing	Design Decisions	
1 15	02:30	2: Giorgio	revisione e miglioramento	Doing	Class Diagram	
1 16	02:00	1	definizione API	Learning	API	
1 17	03:00	3: Giorgio, Valentina	definizione API	Doing	Prototipo	
1 23	03:00	2: Giorgio	gestione Laravel e Posman	Doing	API	
1 27	03:00	1	implementazione API e esempi	Doing	API	
1 31	03:30	1	implementazione mappa	Doing	Prototipo	
2 4	03:00	1	implementazione gestione utente	Doing	Prototipo	
2 7	02:30	1	implementazione servizi	Doing	Prototipo	
2 10	03:30	2: Giorgio	implementazione interfaccia grafica	Doing	Prototipo	
2 12	03:00	3: Giorgio, Valentina	revisione finale	Doing	Deliverable	

Personal Journal

Team (number and name): team banana

Student name: Valentina Sborlini

Student number: 247975

Email: valentina.sborlini@student.univaq.it

When (Month/Day)	Time spent	Partners (please report how many people have been working)	Brief Description of the performed task	Category	Sub-Category
11 23	01:10		Modellazione casi d'uso	learning	Use Case Diagram
11 26	01:30	Tagliente	Scenari d'uso dettagliati	doing	Scenari D'Uso
11 26	01:15		Descrizione principali assunzioni	doing	Assunzioni
11 27	02:00		Modellazione class diagram	learning	Class Diagram
11 28	01:40		Dati e loro modellazione	doing	ER Diagram
11 29	02:00		Design Decisions	doing	Design Decisions
11 30	01:30	Innamorati	Design di Basso Livello	doing	Class Diagram
12 3	01:00		Spiegazione di come FRs e NFRs sono stati soddisfatti	doing	FR e NFR soddisfatti
12 12	01:15		Dati e loro modellazione	doing	ER Diagram
12 15	01:05		Descrizione fonti	doing	ER Diagram
12 19	02:00	Tagliente, Innamorati	Definizione API	doing	Prorotype
1 10	02:00		Descrizione dei casi trovati	doing	Challenges and Risks
1 12	02:45	Tagliente	Scenari d'uso dettagliati	doing	Scenari D'Uso
1 14	01:00		Ripasso e dettagli	learning	Assunzioni
1 14	01:30		Descrizione principali assunzioni	doing	Assunzioni
1 16	01:30		Definizione API	learning	API
1 17	03:00	Tagliente, Innamorati	Definizione API	doing	API
1 23	02:30		Gestione Laravel	doing	Prorotype
1 27	03:00		Definizione API e esempi	doing	API
1 30	02:00		Implementazione	doing	Prorotype
2 6	03:00		Implementazione	doing	Prorotype
2 10	03:00		Sistemazione diagrammi	doing	Diagrammi
2 12	03:00	Tagliente, Innamorati	Revisione	doing	Deliverable

Categorization

Learning <ul style="list-style-type: none"> · Requirements Engineering · Non functional Requirements · Tool study · Diagrams <ul style="list-style-type: none"> ○ Use Case ○ Component ○ Class · Use Scenarios · API · 	Doing: <ul style="list-style-type: none"> · State of the Art · Requirements discovery · Requirements Modeling (UC diagrams) · Diagrams <ul style="list-style-type: none"> ○ Use Case ○ Component ○ Class ○ E/R · Use Scenarios · API Definition & Implementation · Prototype Implementation
---	---

Summary Statistics

Total time spent: 164h 15m

Total time Learning: 27h 55m

Total time Doing: 136h 20m

Appendix. Prototype

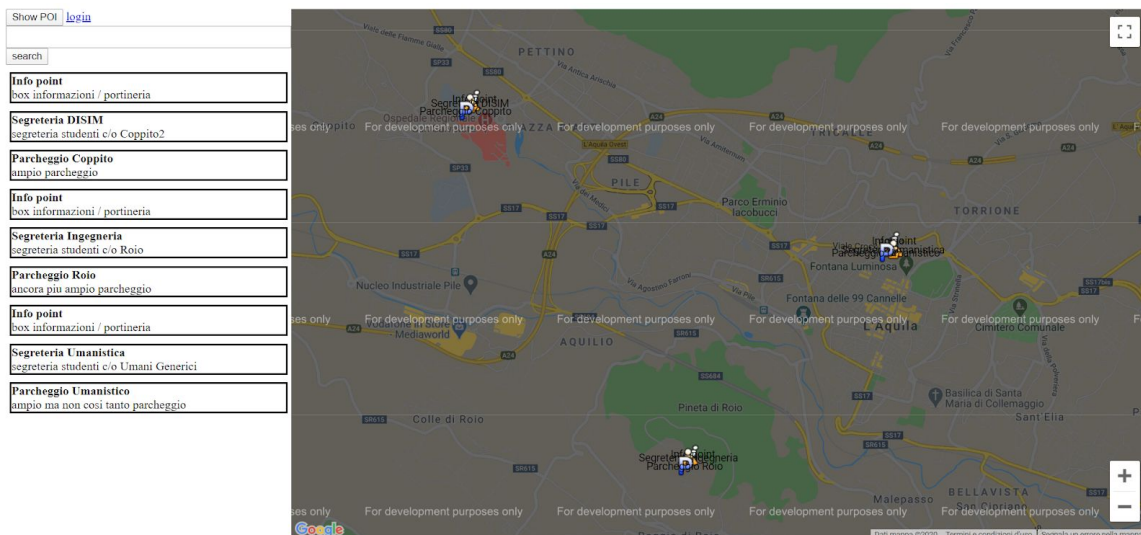
Il codice sorgente del prototipo può essere visualizzato nel repository GitHub:

https://github.com/realgabriele/Software_Engineering_19-20

Implementazione

1. Mappa

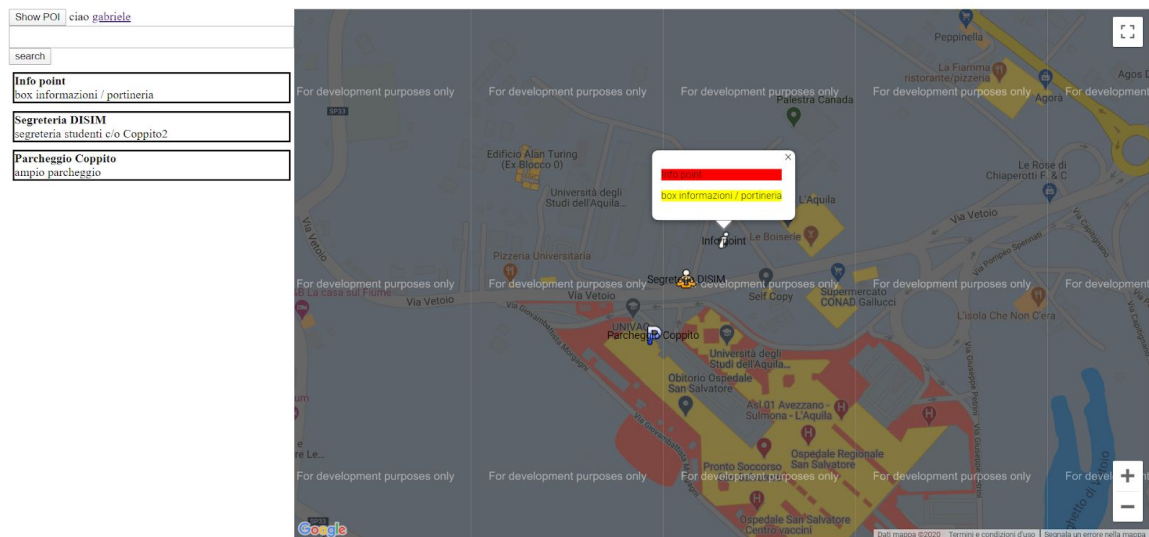
1.1. Utente generico



L'utente generico può accedere alla mappa generale, dove sono mostrati tutti i POI nella loro esatta posizione geografica, con simboli diversi che rappresentano i diversi servizi offerti dal singolo POI. In questa schermata si può muovere e zoomare la mappa a piacimento. Cliccando su di un POI, viene mostrata una finestra che mostra ulteriori dettagli sul POI selezionato.

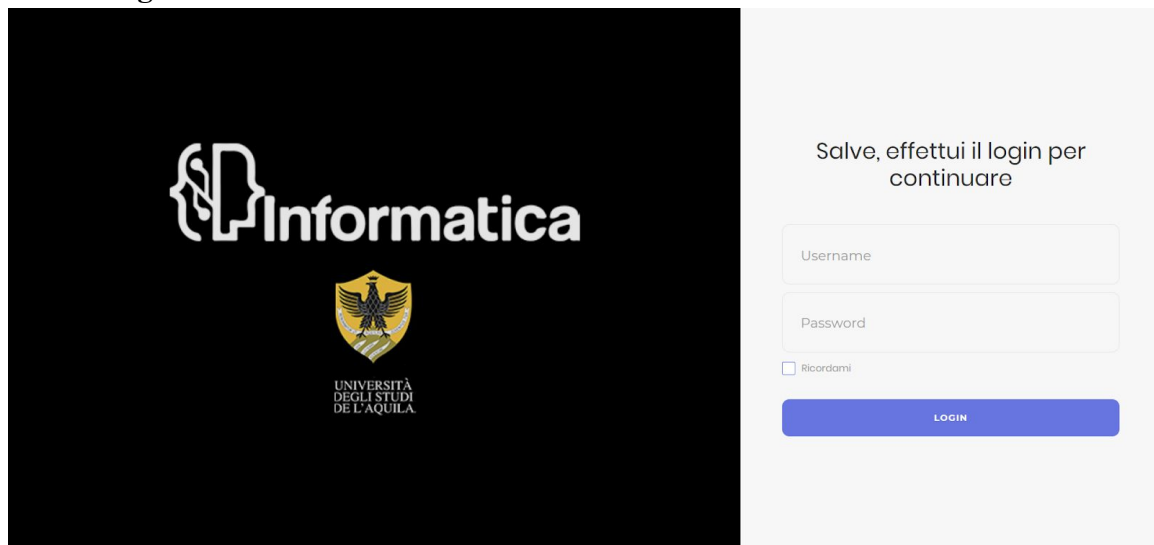
Inoltre è possibile filtrare i POI in base ad una parola chiave da inserire nella barra di ricerca. Saranno quindi mostrati solo i POI pertinenti, sia sulla lista laterale che sulla mappa.

1.2. Studente autenticato



Lo studente che si sia autenticato vedrà solamente i POI di suo interesse sulla mappa, filtrati in base al CdL di appartenenza. Tutte le altre funzioni rimangono invariate all'utente generico.

2. Login



Pagina tramite la quale effettuare il login tramite username e password, per autenticarsi come studente.

3. Pagina Personale

[LogOut](#)



User Info

username: gabriele

name: Gabriele

surname: Tagliente

email: gabriele@tagliente.it

Informatica ▼

password:

Edit

Mappa POI

Uno studente che si sia autenticato può accedere ad una pagina personale dove poter visualizzare i propri dati anagrafici e il suo Corso di Laurea, ed eventualmente modificarli.

nota: scritta “**For development purposes only**”:

La mappa embedded di Google Maps mostra la didascalia sopra riportata. Questo perché, sebbene le funzioni che utilizziamo siano tutte gratuite, è necessario creare un “billing account” per utilizzare le API Google.

Per motivi logistici è stato scelto di non completare questo passaggio per la realizzazione del prototipo, tenendo conto che andrà fatto prima di entrare in produzione.

Test delle Performance

Abbiamo effettuato un test delle performance del sistema per verificare che fossero soddisfatti i requisiti non funzionali descritti.

Di tutto il sistema, la componente che richiede un lavoro sostanzialmente maggiore delle altre è quella che mostra tutti i POI disponibili su una mappa.

La logica del sistema per mostrare i POI sulla mappa prevede 3 fasi:

- 1) Recupero della lista dei POI da mostrare tramite le API
- 2) Visualizzazione della mappa sullo schermo
- 3) Elaborazione dei POI e visualizzazione sulla mappa in base alla loro posizione geografica.

Poiché il punto 2) prevede l'utilizzo di API di servizi esterni (in questo caso Google Maps) e il punto 3) viene fatto utilizzando risorse client-side, riteniamo che il passaggio più delicato possa essere quello al punto 1) in caso di molteplici utenti che vogliono collegarsi nello stesso momento.

Il test è stato effettuato con le seguenti caratteristiche

- **3000** record POI: la tabella dei POI nel database è stata popolata con 3000 righe generate casualmente;
- **1000** utenti: sono state effettuate 1000 richieste API concorrenti;
- **30** secondi: abbiamo restritto il tempo in cui effettuare tutte le richieste a 30 secondi.

Disclaimer: il software non è stato testato su una macchina server ad-hoc, ma è stato fatto girare su una macchina generica aventi il seguente hardware:

CPU: Intel Core I5-8250U 1.6GHz ; RAM: 16GB ; OS: Windows10 Home 64bit

Il PDF riportante i risultati di tutte le richieste effettuate durante il test è possibile consultarlo nell'apposita directory nel repository GitHub:

https://github.com/realgabriele/Software_Engineering_19-20/blob/master/test_performance/testResult.pdf

Abbiamo osservato che tutte le richieste sono state soddisfatte in un tempo compreso tra 0.2 e 0.9 secondi, soddisfacendo così il nostro requisito non funzionale.

Documentazione Postman

La documentazione delle API realizzate tramite Postman:

<https://documenter.getpostman.com/view/10176889/SWTK5EMU>