

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

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

Progetto 5: Portami a destinazione

Date	22/12/2019
Deliverable	Deliverable 1
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	4
B. Raffinamento dei Requisiti	5
A.1 Servizi (con prioritizzazione)	5
A.2 Requisiti non Funzionali	7
A.3 Scenari d'uso dettagliati	8
A.4 Excluded Requirements	9
A.5 Assunzioni	10
A.6 Use Case Diagrams	10
C. Architettura Software	11
C.1 The static view of the system: Component Diagram	11
C.2 The dynamic view of the software architecture: Sequence Diagram	14
D. Dati e loro modellazione	16
E. Design Decisions	17
F. Design di Basso Livello	18
G. Explain how the FRs and the NFRs are satisfied by design	20
G. Effort Recording	22
PERT	22
Logging	22
Categorization	25
Summary Statistics	25
Appendix. Prototype	26

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	Abbiamo deciso che il database una volta a settimana dovrà obbligatoriamente effettuare un backup su disco rigido. Così nel caso in cui il nostro database non funziona più sappiamo dove andare a riprendere i dati mancanti.
Segnalazione di nuovi POI	23/11/2019	24/11/2019	Abbiamo deciso che non diamo l'opportunità allo studente di segnalare nuovi POI in quanto non abbiamo personale che può andare a controllare se il POI segnalato è veritiero o fittizio. I POI verranno aggiornati solamente dopo segnalazione da parte dell'Università stessa.
Chat Supporto Studente-Università	23/11/2019	26/11/2019	Abbiamo deciso che non offriamo neanche una chat di supporto studente-supporto applicazione perchè non avendo personale che può seguire la pratica si rischierebbe che i messaggi resterebbero lì non letti. Abbiamo previsto che nel caso in cui lo studente ha problemi deve inviare una mail alla segreteria studenti che poi girerà la segnalazione ai gestori della piattaforma.

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.

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

<http://www.unife.it/studenti/en/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.

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.

B. Raffinamento dei Requisiti

A.1 Servizi (con prioritizzazione)

Descrivete in dettaglio i servizi offerti dal vostro Sistema, insieme a quelli che ritenete siano le soluzioni concettuali necessarie. In questa fase, non fate riferimento ad alcuna tecnologia specifica. Se volete, intervistate stakeholder e collezionate dati dal web o da altre sorgenti. Dovete acquisire una conoscenza avanzata dei problemi associate ai vostri servizi. Assegnate un ID a ciascun servizio. Prioritizzate inoltre i servizi in base a due scale: importanza alta, media, bassa. Complessità alta, media, bassa.

- 1) Mappare tutti i POI di un Ateneo in una mappa. (Priorità **ALTA**)
- 2) 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**)
- 3) Se lo studente è registrato prioritizzare i POI del CdL e Dipartimento; (Priorità **MEDIA**)
- 4) Categorizzare i POI in base a CdL-specifico, Dipartimento-specifico, Università; (Priorità **ALTA**)
- 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. Studente (Cliente Finale dell' Applicazione)
 - 1.1 Deve Poter Scaricare L' Applicazione
 - 1.1.1 Deve Poter Registrarsi
 - 1.1.2 In Registrazione deve Fornire Dati Anagrafici
 - 1.1.3 In Registrazione deve fornire anche dati Universitari Anagrafici (Matricola-Email di ateneo)...
 - 1.2 Deve Poter Loggare
 - 1.2.1 Deve Poter Accedere all'applicazione attraverso Social Network
 - 1.2.2 Deve Poter Accedere all'applicazione attraverso servizi Mail.
 - 1.3 Una volta loggato per la prima può accedere alla mappa con tutti i POI interni alla sua università presente nelle anagrafiche Universitarie.
 - 2.1 Lo studente può selezionare i propri interessi in una sezione dedicata nel proprio profilo personale.
 - 2.1.1 Lo studente in base ai propri interessi può filtrare tali POI.
 - 3.1 Lo studente in fase di registrazione deve selezionare il proprio CDL e dipartimento e la sede di appartenenza
 - 3.1.1 Lo studente in base ai proprio CLD e dipartimento e in base alla sede di appartenenza può prioritizzare nella mappa i propri POI.
 - 4.1 Deve poter visualizzare nella mappa i POI colorati/rappresentati diversamente in base al servizio offerto
 - 5.1 Deve poter usare i comandi vocali
 - 5.1.1 Deve poter usare i comandi vocali per raggiungere un determinato POI
 - 5.1.2 Deve poter usare i comandi vocali per catalogare i POI in base ad una nuova ad una nuova ricerca
 - 5.1.3 Deve poter usare i comandi vocali per registrarsi nell'applicazione (CASO NON VEDENTE)
 - 5.1.4 Deve poter usare i comandi vocali per loggare nell'applicazione (CASO NON VEDENTE)

A.2 Requisiti non Funzionali

- E' obbligatorio creare un' account interno all' applicazione.
 - 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.
-

A.3 Scenari d'uso dettagliati

- Registrazione/login utente tramite credenziali terze

Per identificare l'utente in fase di login, permettiamo allo studente di utilizzare le credenziali di ateneo. Questo caso d'uso presuppone che l'ateneo abbia un database di utente da cui prelevare le informazioni sull'utente quali matricola, username, email, password, corso di laurea, ...

- Operazioni su mappa

Il caso d'uso più comune è rappresentato dall'utente generico che interagisce con la mappa, su cui sono mostrati tutti i POI con colore diverso in base alla categoria o servizio offerti.

L'utente sarà in grado di:

- zoommare e muovere la mappa per visualizzare i POI in una determinata zona;
- utilizzare la barra di ricerca per cercare un POI in base al nome;
- selezionare quali dei servizi disponibili visualizzare sulla mappa (ad esempio scegliere di visualizzare solamente i POI che offrono ristorazione).

- Navigazione verso un POI

Un caso d'uso più particolare è rappresentato dall'utente che vuole farsi portare a destinazione verso un POI selezionato. In questo caso, premendo l'apposito pulsante, verrà visualizzata prima una schermata in cui scegliere la posizione di partenza (di default la posizione corrente, ma modificabile) e successivamente saranno visualizzato il percorso passo passo per raggiungere la destinazione. Opzionalmente potrà essere attivato un sistema di navigazione che, attraverso comandi vocali, porterà la persona verso il POI prescelto.

- Prioritizzazione per CdL

Un utente che abbia effettuato la registrazione, e quindi si sia identificato come studente, potrà visualizzare solamente i POI di interesse per il suo Corso di Laurea. Se questa preferenza sarà attivata, i POI non inerenti al suo percorso di studi verranno:

- nascosti sulla mappa;
- elencati dopo i POI del proprio CdL in caso di ricerca testuale.

A.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.

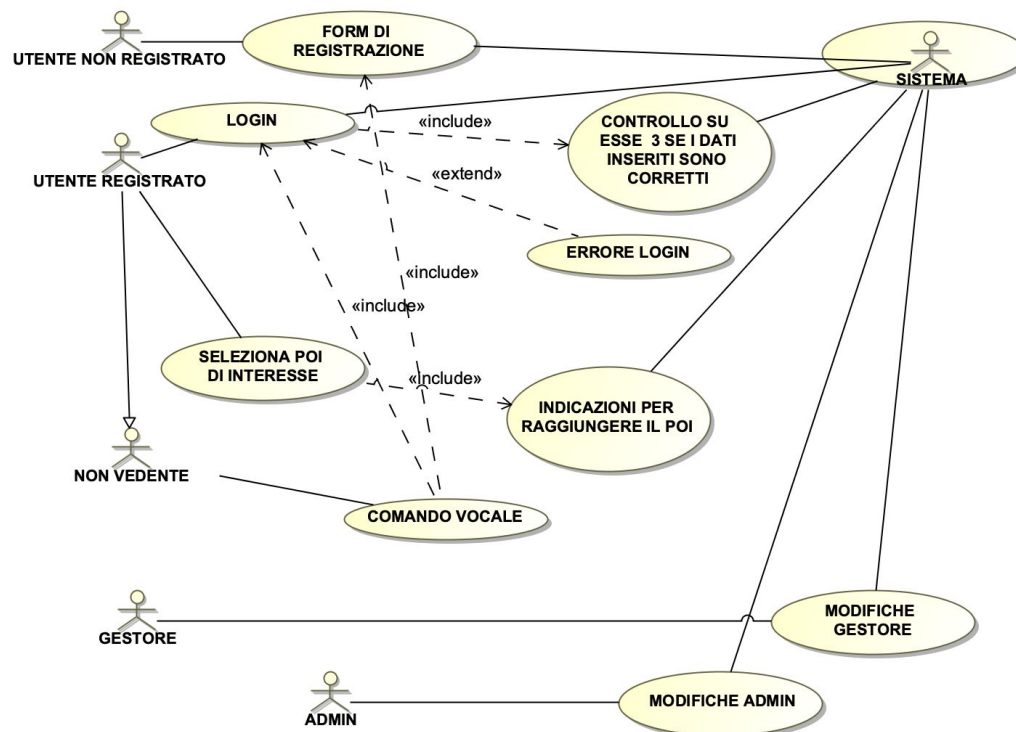
Ad eccezione dei dati di login che per **LEGGE** devono essere nascosti ai proprietari dell' applicazione.

A.5 Assunzioni

ASSUNZIONI PRINCIPALI

L'Assunzione Principale nel Progetto commissionatoci dal Prof. Henry Muccini, è quello di scegliere quale mappa usare (**Satellitare oppure non**) inoltre, altra principale assunzione interna al progetto è scegliere se l'applicativo può funzionare **OFFLINE** e quindi usare una mappa **OFFLINE** oppure debba funzionare solo **ONLINE** e quindi è possibile usare le mappe Google (My Maps) utilizzando anche noi **EMBED** oppure **IFRAME** come hanno fatto le altre principali università.

A.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 Social Network oppure tramite credenziali ESSE3 o simili per le università convenzionate vengono processate dal sistema per controllare la veridicità dei dati inseriti nel form.

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:

è quella parte del sistema che si occupa della gestione dell'utente, dalla sua autenticazione al recupero dei dati nel database di uno studente.

4. Visualizzatore Mappa:

Costituisce la componente che visualizza i POI di ogni studente sulla mappa e rappresenta l'interfaccia di un utente generico o di uno studente. In tal caso si occupa di filtrare in base al proprio CDL, alla sede universitaria di appartenenza oppure in base ai propri interessi.

5. Navigatore:

Viene definito Tragitto tutta quella parte software chiamata tramite API a **TOOL ESTERNO** che porta il nostro Cliente (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:

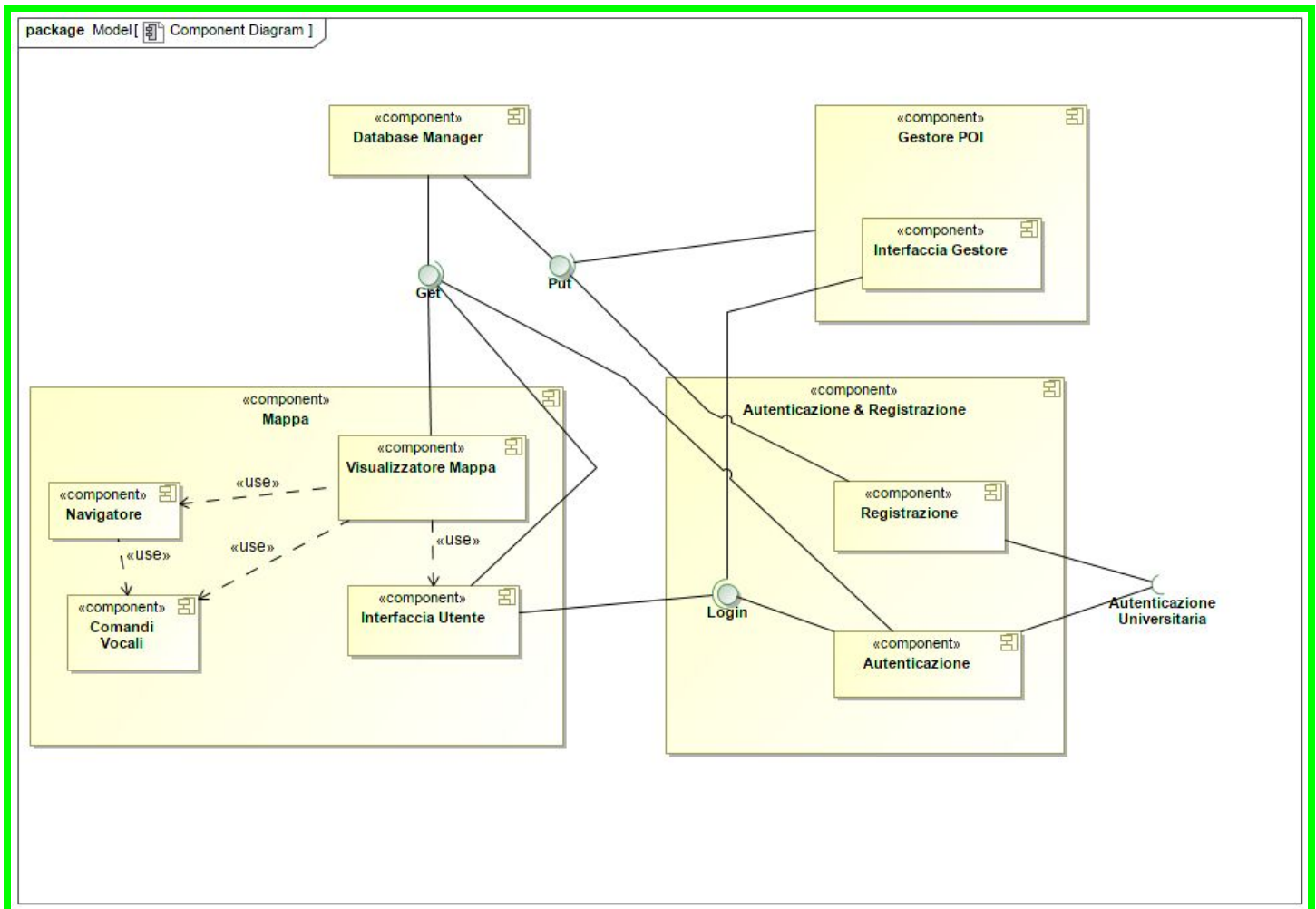
Questo modulo raccoglie tutte le componenti che si occupano della registrazione e autenticazione degli utenti, e della gestione degli stessi.

11. Registrazione:

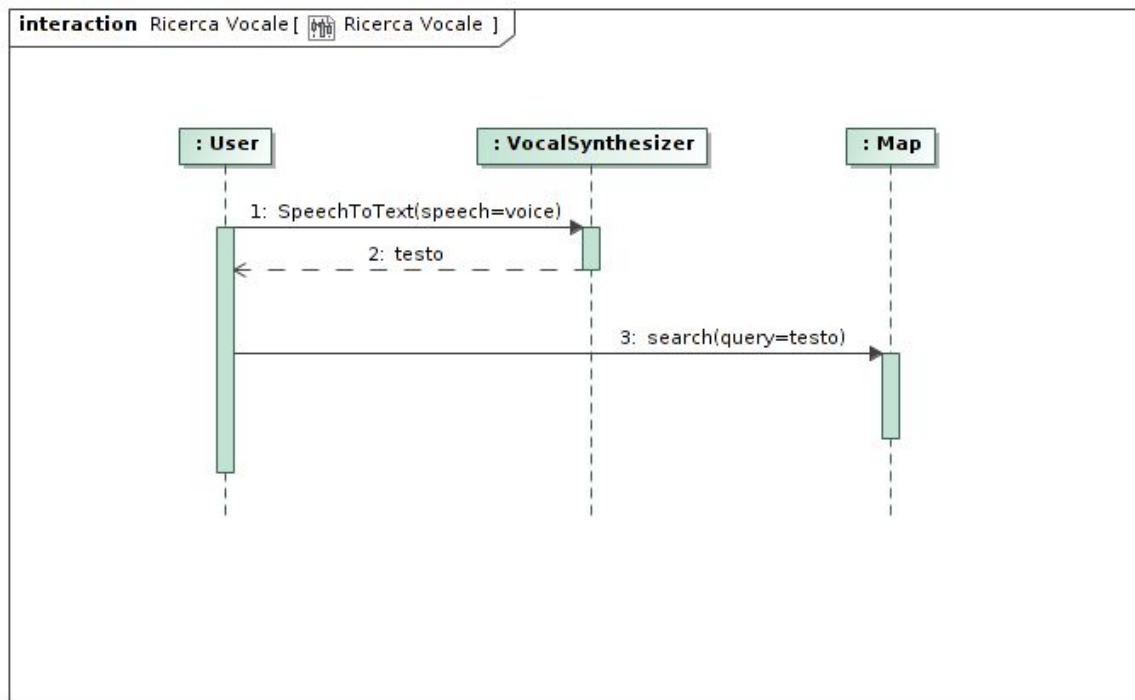
Componente che si occupa della registrazione di un nuovo utente. Nel caso sia un utente locale, esso viene semplicemente inserito nel database di utenti. Nel caso in cui sia un studente che utilizza le credenziali di ateneo, questo modulo si occuperà prima di verificare la correttezza delle credenziali,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

12. Autenticazione:

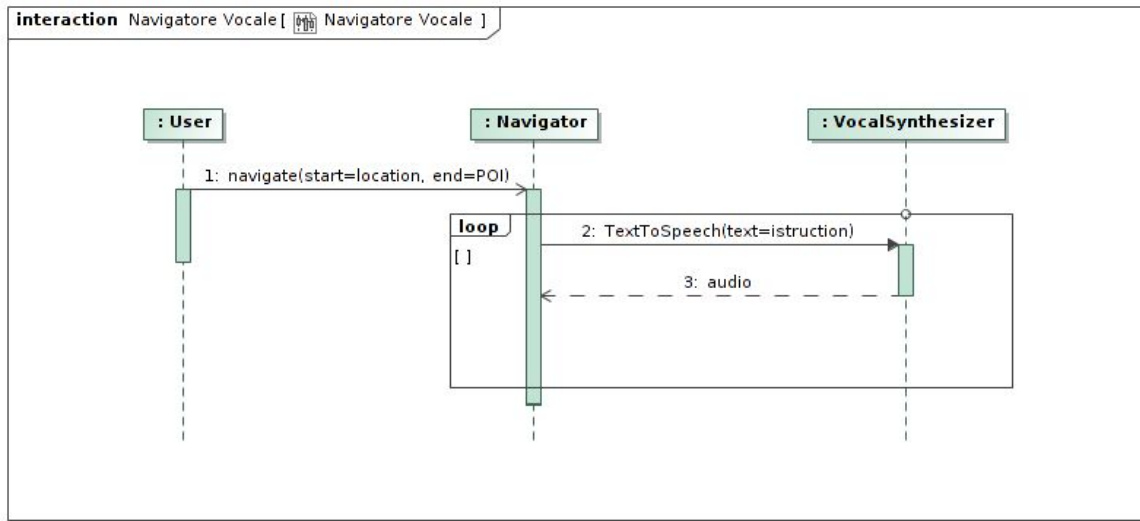
Componente che effettua l'autenticazione di un utente, attraverso l'apposito form di login. L'autenticazione può avvenire o con credenziali salvate sul database o con credenziali esterne salvate sul sistema gestito dall'università.



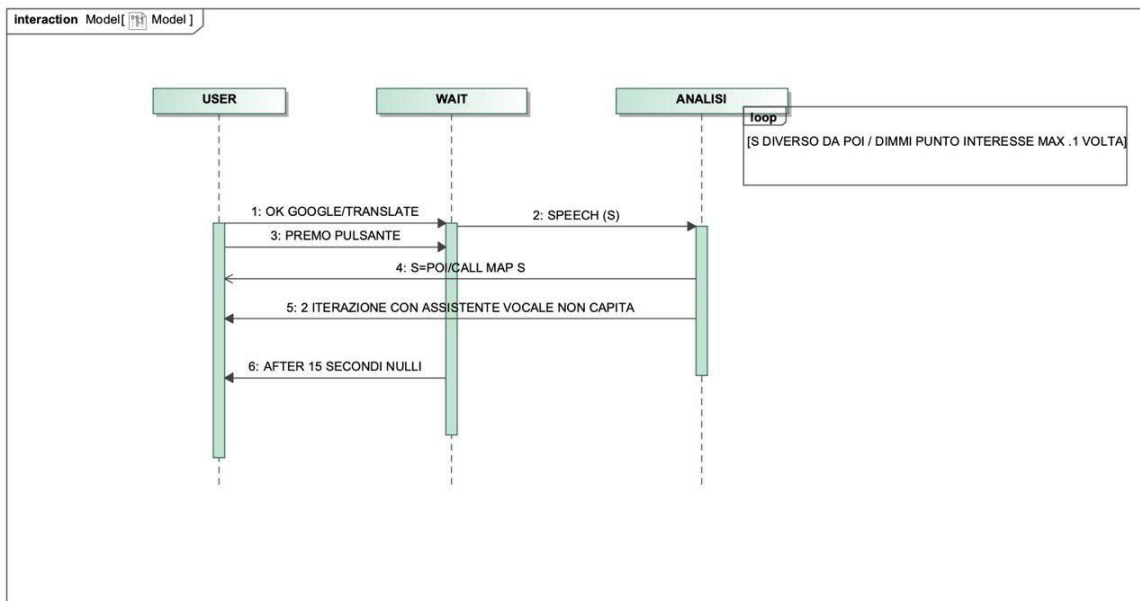
C.2 The dynamic view of the software architecture: Sequence Diagram



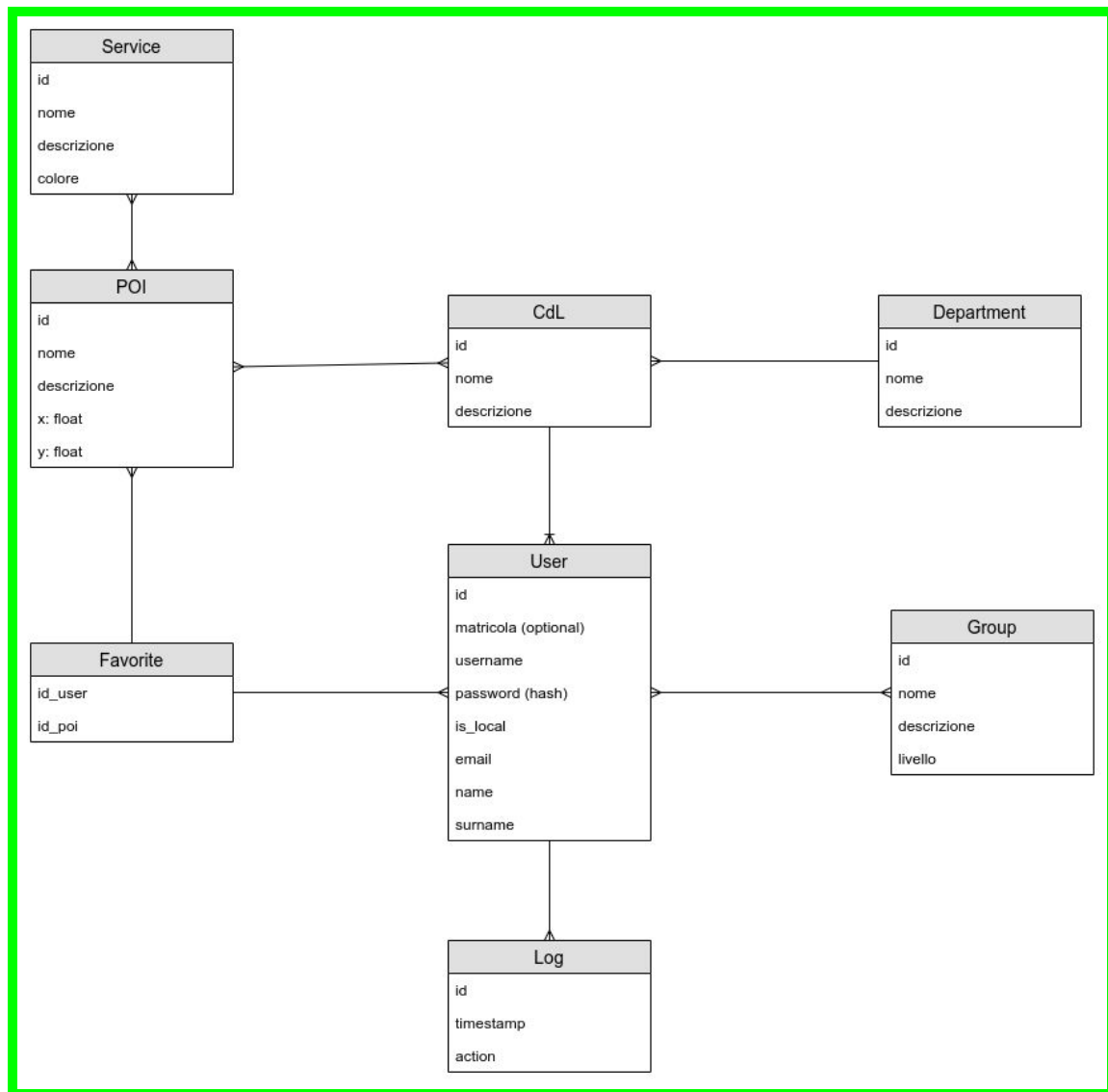
Ricerca di POI tramite input vocale: l'utente richiede al sintetizzatore vocale di convertire la voce in testo, il quale sarà utilizzato come query di ricerca sulla mappa.



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).

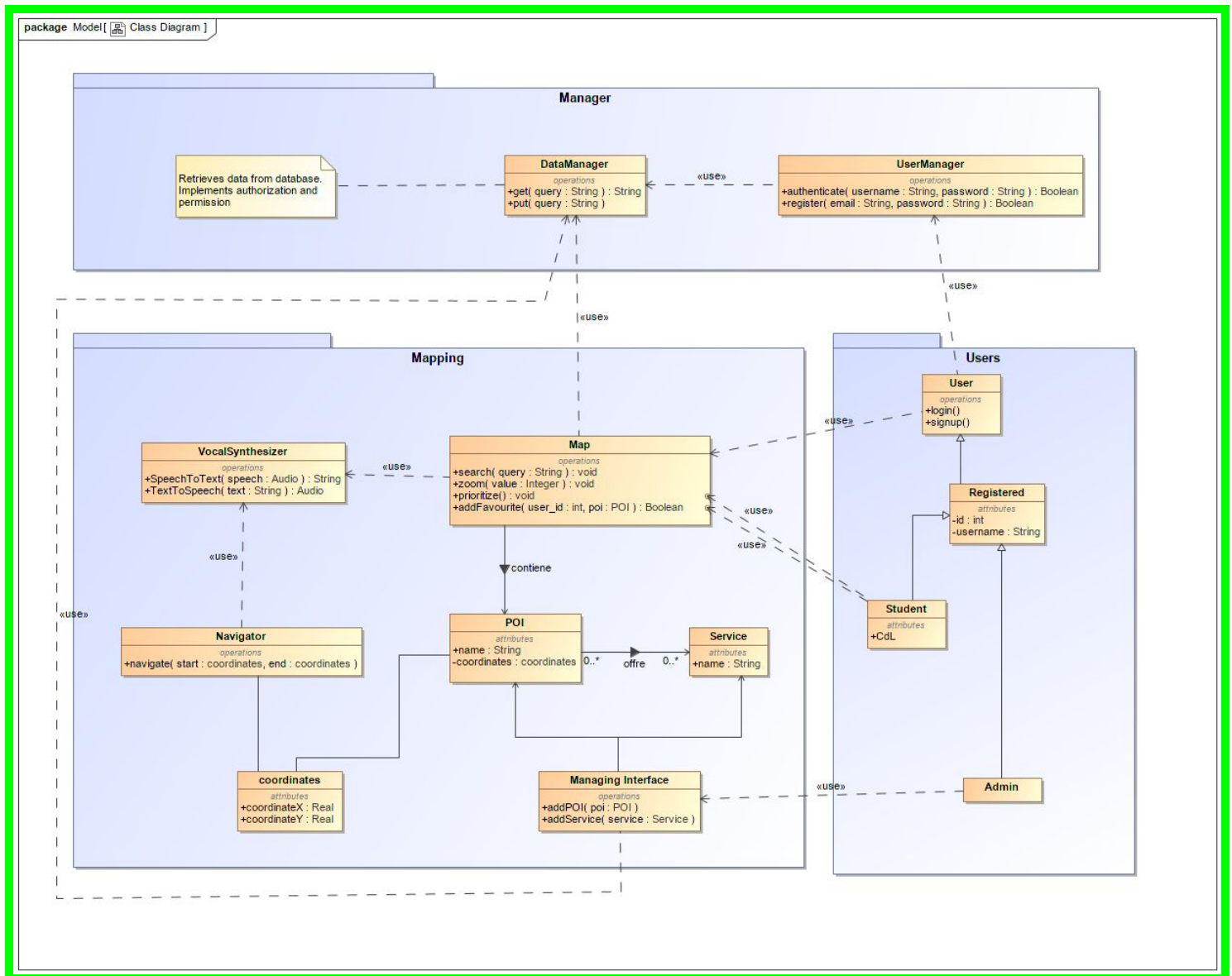
Se l'università non dispone di tale sistema dovrà essere disposto un data entry delle informazioni sul corso di laurea frequentato direttamente dall'utente utilizzatore.

E. Design Decisions

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

1. Un'interfaccia amichevole, giovanile e con caratteri ben leggibili, così da poter essere utilizzata da tutti gli utenti senza fatica. Abbiamo Pensato di Posizionare all'apertura dell'applicazione una schermata di Login che presenta 3 opzioni (Login By Facebook, Login By Google, Login By Esse3)
2. Nella fase della registrazione/login saranno richiesti pochi elementi in quanto deve essere un'applicazione di utilizzo veloce, cliccando nelle login dei social network si verrà reindirizzati alle classiche schermate di login.
3. La schermata principale (dopo l'accesso) presenterà il logo dell'università nel momento in cui si sta caricando la mappa. (BUFFERING)
4. Una volta caricata la Mappa L'app presenterà subito la mappa con i POI
5. A sinistra della mappa sarà presente un menù ad hamburger che permette la disconnessione dell'account, la ricerca dei POI e la possibilità di inoltrare una richiesta di assistenza (presente anche nella schermata di Login)

F. Design di Basso Livello



Il diagramma sopra descritto è diviso in tre package:

1. **Mapping:** contiene le classi che implementeranno la mappa e tutte le funzioni ad essa collegate;
2. **Users:** contiene le classi che definiscono le diverse situazioni di utente che viene ad interagire con il sistema;
3. **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 registrato) è in grado di accedere alla mappa e ad alcune sue funzionalità. 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). 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 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.

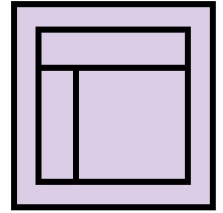
- Portabilità

Presenta un'alta portabilità, in quanto può essere usata su diverse piattaforme come ad esempio IOS e Android.

QUALITA' IN USO

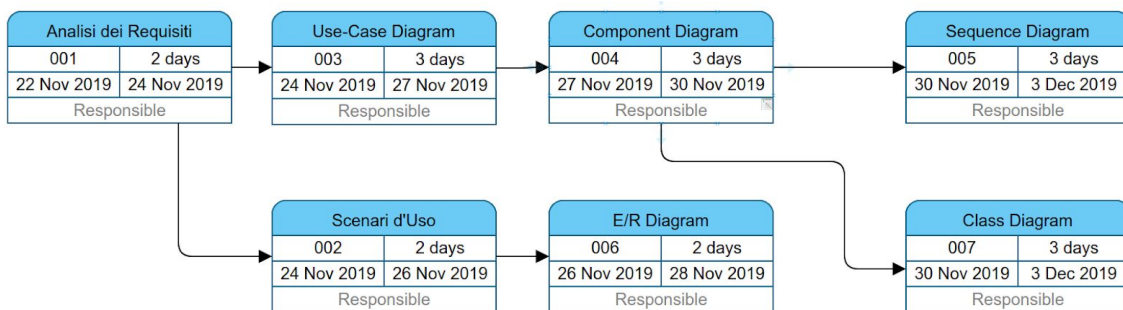
- Efficacia ed efficienza
Vogliamo avere il massimo risultato con il minimo sforzo, per questo vogliamo offrire un servizio che sia di gradimento all'utilizzatore finale. Deve avere quantità e qualità.
- Soddisfazione
Altro punto fondamentale, lo studente deve essere soddisfatto del servizio offerto, 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.

G. Effort Recording



PERT

Make a PERT documenting the tasks and timing you expect to spend on the deliverable. Try to be as precise as possible. Check, after the deliverable deadline, if and how you satisfied (or not) the deadlines.



Logging

As you are working on the assignment, record what you are doing and how long you spent. As a rule of thumb, you should add a log entry every time you switch tasks. For example, if you do something for two hours straight, that can be one log entry. However, if you do two or three things in half an hour, you must have a log entry for each of them. You do not need to include time for logging, but should include the time spent answering the other parts of this question.

For this purpose, please use the **LogTemplate.xls** file.

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		

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

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	
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	
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

Categorization

When logging the time spent on the project, please create different sub- categories. Specifically, it is important to clearly distinguish between two main categories: the time spent for “**learning**” (the modeling languages, the tools, etc.) from the time needed for “**doing**” (creating the models, taking the decisions, ...). Learning tasks are in fact costs to be paid only once, while doing costs are those that will be repeated through the project.

For each category, please define sub-categories. Examples follow. You may add other sub-categories you find useful.

Learning <ul style="list-style-type: none"> · Requirements Engineering · Non functional Requirements · Use Case Diagrams · Tool study 	Doing: <ul style="list-style-type: none"> · Requirements discovery · Requirements Modeling (UC diagrams)
--	---

Summary Statistics

Based on the attributes defined above, calculate the summary statistics of the time spent for “learning”, the time spent for “doing”, and the total time.

Note: this Deliverable report shall document only the Summary Statistics for the different deliverables (D1, D2, and Final). Detailed information shall be reported in the Excel file.

COPY HERE (computed from the spreadsheet): i) the total number of hours spent by the group (that is, hours per task X number of people working on that task), ii) the time spent for LEARNING and for DOING

Total time spent: 66h 30m
 Total time Learning: 19h 25m
 Total time Doing: 47h 05m

Appendix. Prototype

<Provide a brief report on your prototype, and especially: information on what you have implemented, how the implementation covers the FR and NFR, how the prototypes demonstrates your project correctness with respect to the FR and NFR. You may add some screenshots to describe what required above. Be ready to show your prototype during the oral examination>

URL DOCUMENTAZIONE POSTMAN:

<https://documenter.getpostman.com/view/9815348/SWECWFAW>