

# **“Software Engineering”**

## **Course**

### **a.a. 2019-2020**

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

## **<PROGETTO: 5 - PORTAMI A DESTINAZIONE>**

<b>Date</b>	<05/12/2019>
<b>Deliverable</b>	D3
<b>Team (Name)</b>	Team Two

<b>Team Members</b>			
<b>Name &amp; Surname</b>	<b>Matriculation Number</b>		<b>E-mail address</b>
Davide Benvenuti	253236		david-b98@live.it
Andrea Pulcini	252006		andripulc@hotmail.it

--	--	--	--

# Table of Contents of this deliverable

<b>LIST OF CHALLENGING/RISKY REQUIREMENTS OR TASK.....</b>	<b>3</b>
<b>A. STATO DELL'ARTE.....</b>	<b>4</b>
<b>B. RAFFINAMENTO DEI REQUISITI</b>	
A.1 Servizi (con prioritizzazione).....	5
A.2 Requisiti non Funzionali.....	6
A.3 Scenari d'uso dettagliati.....	7
A.4 Excluded Requirements.....	8
A.5 Assunzioni.....	8
A.6 Use Case Diagrams.....	9
<b>C. ARCHITETTURA SOFTWARE</b>	
C.1 The static view of the system: Component Diagram.....	10
C.2 The dynamic view of the software architecture: Sequence Diagram....	12
<b>D. DATI E LORO MODELLAZIONE.....</b>	<b>13</b>
<b>E. DESIGN DECISIONS.....</b>	<b>14</b>
<b>F. DESIGN DI BASSO LIVELLO.....</b>	<b>15</b>
<b>G. EXPLAIN HOW THE FRs AND THE NFRs ARE SATISFIED BY DESIGN.....</b>	<b>16</b>
<b>G. EFFORT RECORDING.....</b>	<b>16</b>
<b>APPENDIX. PROTOTYPE.....</b>	<b>18</b>

## List of Challenging/Risky Requirements or Tasks

### Creazione challenging task Deliverable D2

Challenging Task	Date the task is identified	Date the challenge is resolved	Explanation on how the challenge has been managed
Visualizzazione mappa	25/11	26/11	Abbiamo deciso di importare la mappa di google my maps
Controllo credenziali tramite Database Ateneo	26/11	27/11	Abbiamo deciso di non dare la possibilità all'utente di registrarsi sul sito bensì di autenticarsi esclusivamente tramite credenziali già fornite agli studenti dall'Ateneo, utilizzando direttamente il database dell'Ateneo o importando i dati da esso
Gestione P.O.I. (Modifica, aggiunta)	02/12	03/12	Abbiamo deciso di implementare un admin per la gestione dei POI, non verrà data la possibilità a gli utenti di aggiungere o modificare i POI
Target utilizzo utenti	26/11	27/11	Abbiamo deciso che il sistema sarà utilizzabile sia da parte di utenti registrati che non, il sistema prevederà due tipologie di utenti: studente o docente
Come progettare il prototipo	4/12	10/12	
Backup Database	5/12	7/12	Data la scarsa frequenza di aggiunta o rimozione POI abbiamo deciso che il backup del database verrà effettuato una volta al mese
Visualizzazione P.O.I.	2/01	3/01	Per evitare una visualizzazione disordinata e confusionaria dei P.O.I. le aule e gli uffici (posti nello stesso edificio) verranno raggruppare con due icone differenti (uno per gli uffici e uno per le aule)
Implementazione controllo vocale	4/01	5/01	Abbiamo deciso che per l'implementazione del controllo vocale useremo supporti esterni

## A. Stato dell'Arte

- *Università La Sapienza Virtual Tour*

<https://virtualtour.uniroma1.it>

*Il sistema sopra citato permette di cercare i P.O.I. dell'università "LA SAPIENZA " distribuiti invarrei zone di roma e offre i seguenti servizi:*

- *ricerca di un aula o edificio dato un nome con relativa visualizzazione in mappa*
- *scelta della categoria di POI da visualizzare sulla mappa (es. solo mense, solo fermate autobus)*
- *dettagli di ogni POI*
- *Calcolo percorso per arrivare a un POI, aula o edificio dalla posizione attuale*

*A differenza del nostro sistema il Virtual Tour della Sapienza non permette di cercare in modo specifico:*

- *una fermata dell'autobus*
- *una mensa*
- *un ufficio*

*La mappa è stata creata attraverso **GeoCMS** il quale è un **Content Management System** basato sulla tecnologia **Google Maps Platform** e sviluppato da Noovle per la gestione di informazioni geolocalizzate destinate alla fruizione su web, dispositivi mobili e altre piattaforme.*

## B. Raffinamento dei Requisiti

### *A.1 Servizi (con prioritizzazione)*

- **Autenticazione**

Il sistema deve fornire un'area di accesso tramite credenziali universitarie già fornite dall'Ateneo, ma sarà accessibile anche ad un utente non autenticato.

Il sistema prevede due tipologie di utenza:

- STUDENTE**: Se lo studente è autenticato il software dovrà acquisire dal database Dipartimento e Corso di Laurea dello specifico studente.

- PROFESSORE**: Se il professore è autenticato il software dovrà acquisire dal database il nome del professore e i Corsi di Laurea nel quale è docente.

- ADMIN**: Avrà una sezione dedicata da cui ci si arriverà tramite link.

Id01, Importanza: Media, Complessità: Bassa

- **Mappa**

Il sistema deve fornire una visualizzazione chiara ed intuitiva della posizione dei POI, mappandoli con icone diverse a seconda del servizio offerto.

Id02, Importanza: Alta, Complessità: Media

- **Filtri di ricerca**

Un utente deve avere la possibilità di visualizzare determinate categorie di POI tramite filtri di ricerca (Esempio: visualizzazione di solo aule, fermate autobus, ecc), oppure tramite Dipartimento e/o Corso di Laurea.

Id03, Importanza: Alta, Complessità: Bassa

- **Funzione di ricerca**

Il sistema prevedere una barra di ricerca per trovare facilmente qualsiasi aula o edificio di interesse tramite nome o dipartimento.

Inoltre tutte le ricerche verranno ordinate in base alla distanza posizione utente-POI, se l'utente fornirà il consenso alla posizione

Id04, Importanza: Alta, Complessità: Media

- **Prioritizzazione POI (studente/docente)**

Se lo studente è autenticato il sistema deve prioritizzare i POI in base al Dipartimento e Corso di Laurea di cui fa parte lo studente.

Se il docente è autenticato il sistema deve prioritizzare i POI in base al Corso nel quale insegna ed ai Dipartimenti nel quale fa parte.

Id05, Importanza: Alta, Complessità: Media

- **Calcolo Percorso**

Il sistema deve avere la possibilità di calcolare il percorso da fare dalla posizione attuale dell'utente verso il POI desiderato.

Id06, Importanza: Alta, Complessità: Alta

- **Gestione P.O.I. (Da parte dell'amministratore)**

L'amministratore dopo aver effettuato il login avrà la possibilità di aggiungere e/o rimuovere i P.O.I. e modificare quelli esistenti

Id07, Importanza: Alta, Complessità: Bassa

- **Comandi Vocali**

Il sistema avrà la possibilità di usare i comandi vocali per potersi loggare, per cercare un determinato P.O.I. oppure per il percorso da effettuare.

Id08, Importanza: Alta, Complessità: Alta

## ***A.2 Requisiti non Funzionali***

- **USABILITY**

Il sistema deve essere facile da utilizzare anche per gli utenti meno esperti per questo deve avere un'interfaccia semplice ed intuitiva.

- **EFFICIENCY (PERFORMANCE)**

Il sistema deve essere in grado di gestire una molteplicità importante di P.O.I. potenzialmente distribuiti su una vasta zona. Inoltre deve anche garantire l'accesso in parallelo ad almeno 500 studenti che utilizzano il servizio. Si cerca di soddisfare il requisito limitando il numero di query e ottimizzando l'invio delle query al database.

- **AVAILABILITY**

Il sistema deve essere operativo e funzionante al 99%, con un margine di errore che potrebbe presentarsi in caso di un temporaneo rallentamento della rete.

### *A.3 Scenari d'uso dettagliati*

- **AUTENTICAZIONE**

Un utente accede alla pagina di login cliccando sull'apposito bottone nell'homepage, dopo aver inserito le credenziali di accesso fornite dall'università, il sistema verifica la correttezza delle stesse. Se sono errate il sistema mostra un messaggio di errore, se le credenziali sono corrette l'utente viene reindirizzato nell'homepage.

- **PRIORITAZIONE P.O.I.**

Se l'utente (docente o studente) ha eseguito l'accesso, nell'homepage viene mostrata la mappa dell'ateneo filtrando i P.O.I. in base a dipartimento e corso di laurea dell'utente che ha eseguito l'accesso. Inoltre questi P.O.I. verranno raggruppati in una lista a sinistra della mappa.

- **RICERCA DI UN P.O.I**

Un utente cerca un P.O.I (aula, edificio, fermata autobus, mensa) tramite la barra di ricerca e viene visualizzata la posizione di quel P.O.I sulla mappa con i relativi dettagli

- **AGGIUNTA O MODIFICA P.O.I.**

Un amministratore dopo aver effettuato l'accesso su una pagina dedicata avrà una lista con tutti i poi da poter modificare e un apposito bottone con relativo form per poter inserire un nuovo P.O.I.

- **VISUALIZZAZIONE MAPPA**

Un utente (anche senza aver effettuato il login) potrà accedere alla dashboard principale visualizzando i dipartimenti sulla mappa; l'utente potrà inoltre spaziare nella mappa, utilizzare la funzione di ricerca e selezionare la tipologia di P.O.I. *La mappa verrà visualizzata sempre all'interno della dashboard utente.*

- **SELEZIONARE FILTRO**

*Un utente seleziona una o più categorie di P.O.I. tramite una lista presente ad un lato della mappa ed il sistema visualizzerà sulla mappa solamente le categorie di P.O.I. che interessano all'utente.*

#### *A.4 Excluded Requirements*

- Un normale utente non potrà aggiungere/modificare/rimuovere POI.
- *Un utente non avrà la possibilità di registrarsi ma soltanto di loggare con credenziali già fornite dall'Ateneo.*
- *Un utente non potrà recensire un P.O.I*

#### *A.5 Assunzioni*

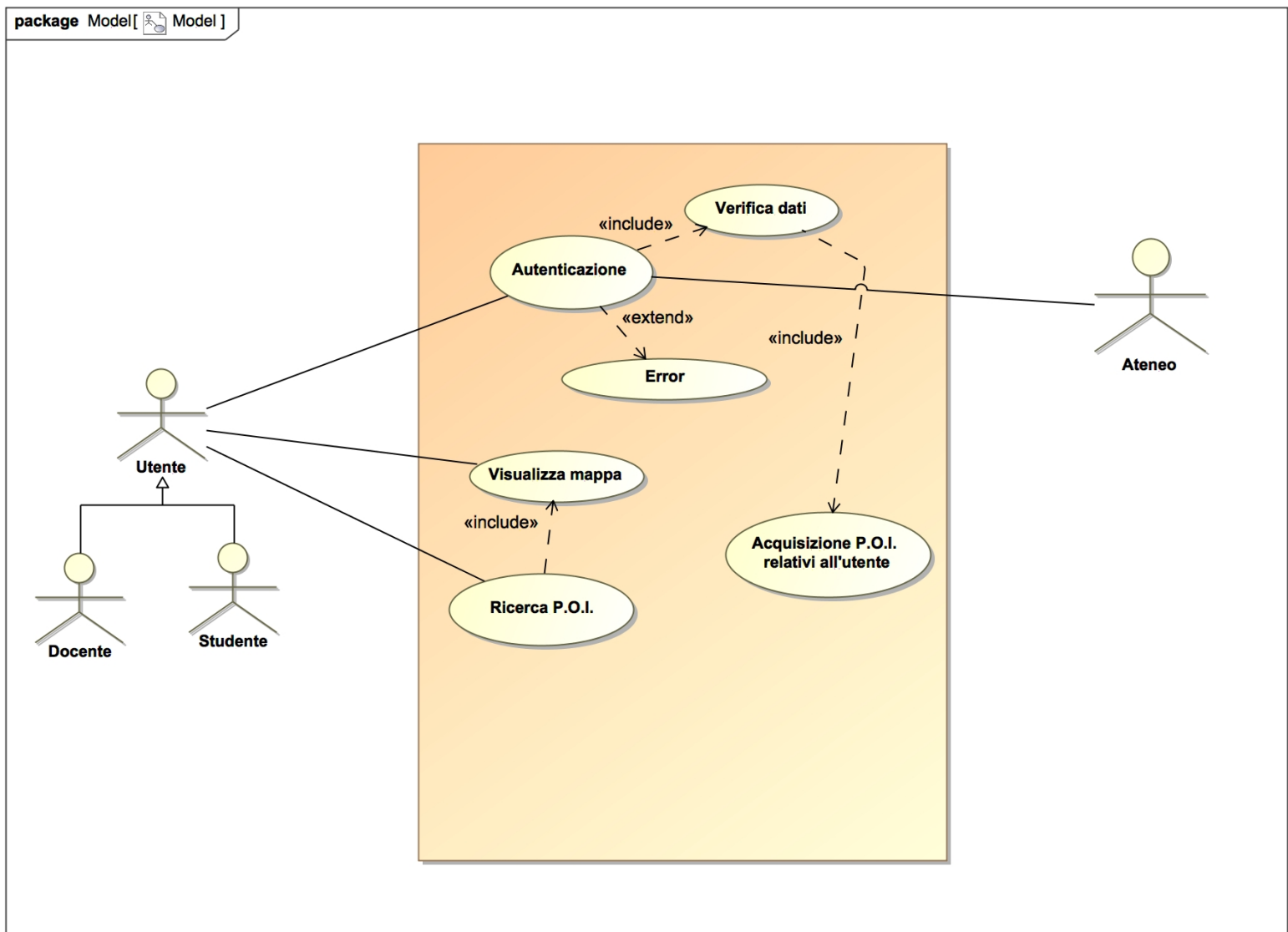
Assumiamo che il sistema prevede l'autenticazione solamente tramite credenziali già fornite in precedenza dall'Ateneo e utilizzate già da altri sistemi (ESSE3); Non sarà presente un form di registrazione.



Assumiamo di conoscere tutte le informazioni riguardo i POI, importandole da software esistenti (es. linfcop).

Assumiamo che la mappa dell'Ateneo e il servizio del calcolo percorso siano già forniti da software esistenti (google mymaps) .

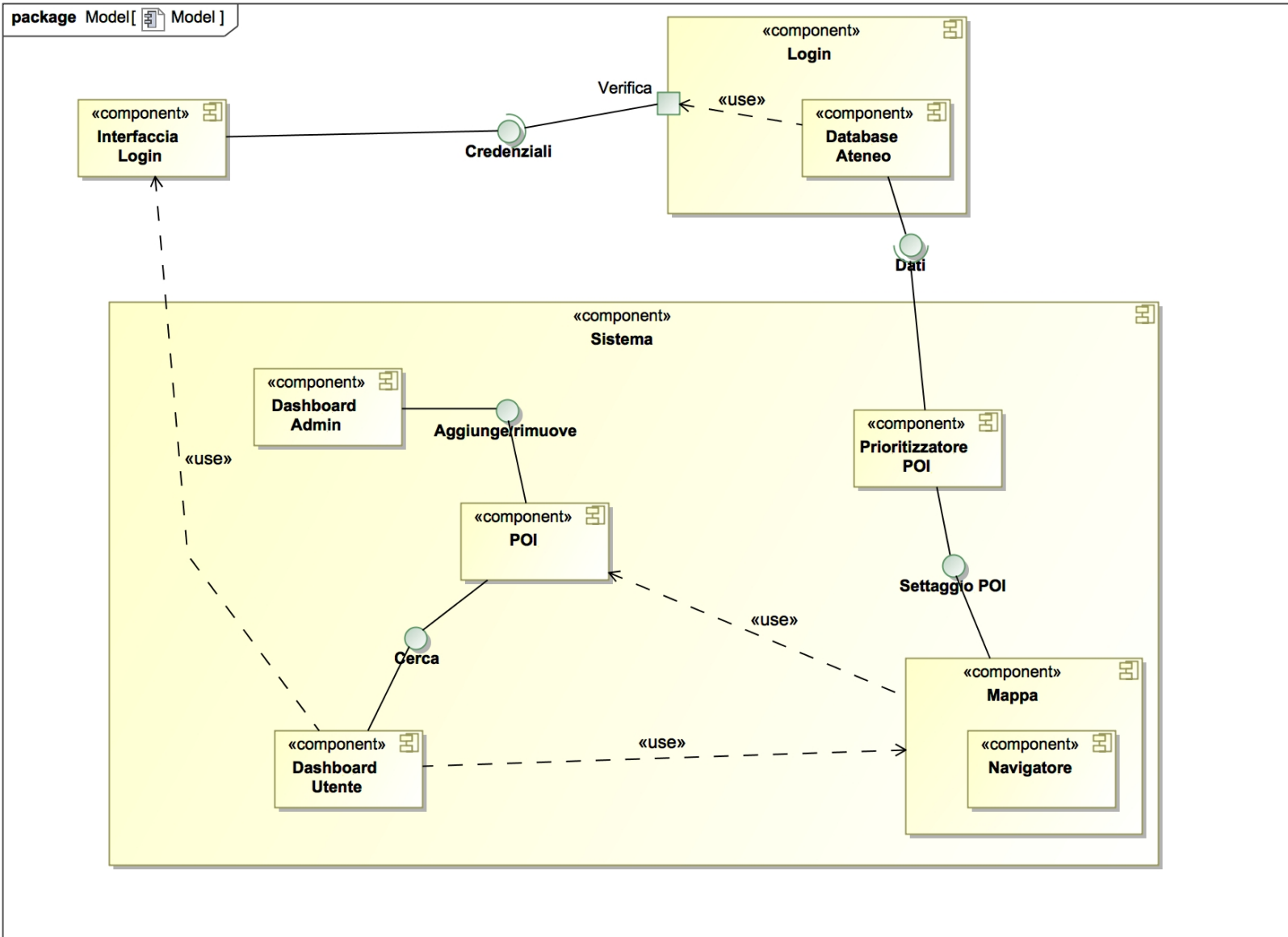
## A.6 Use Case Diagrams



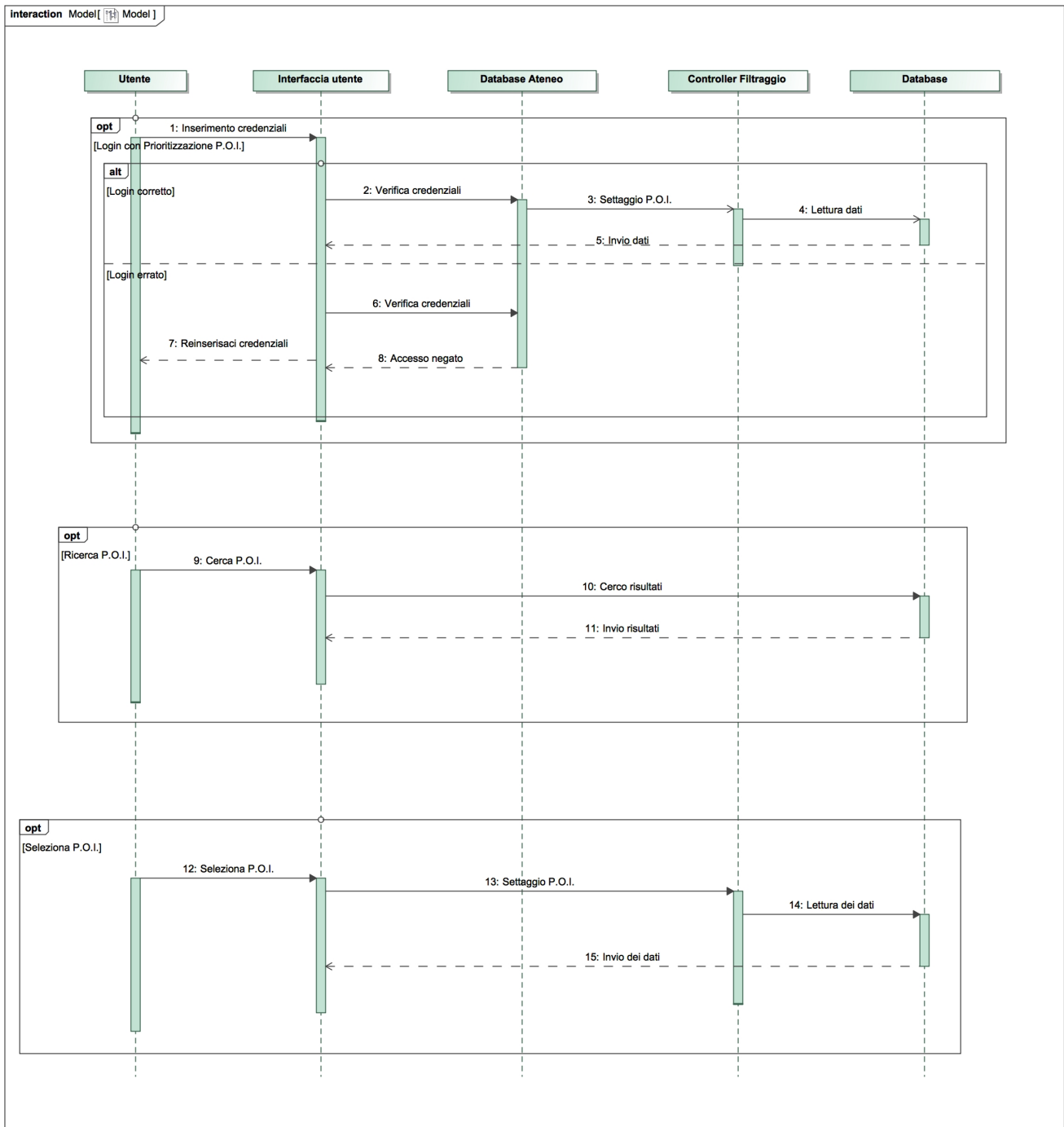
# C. Architettura Software

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

### Deliverable D2



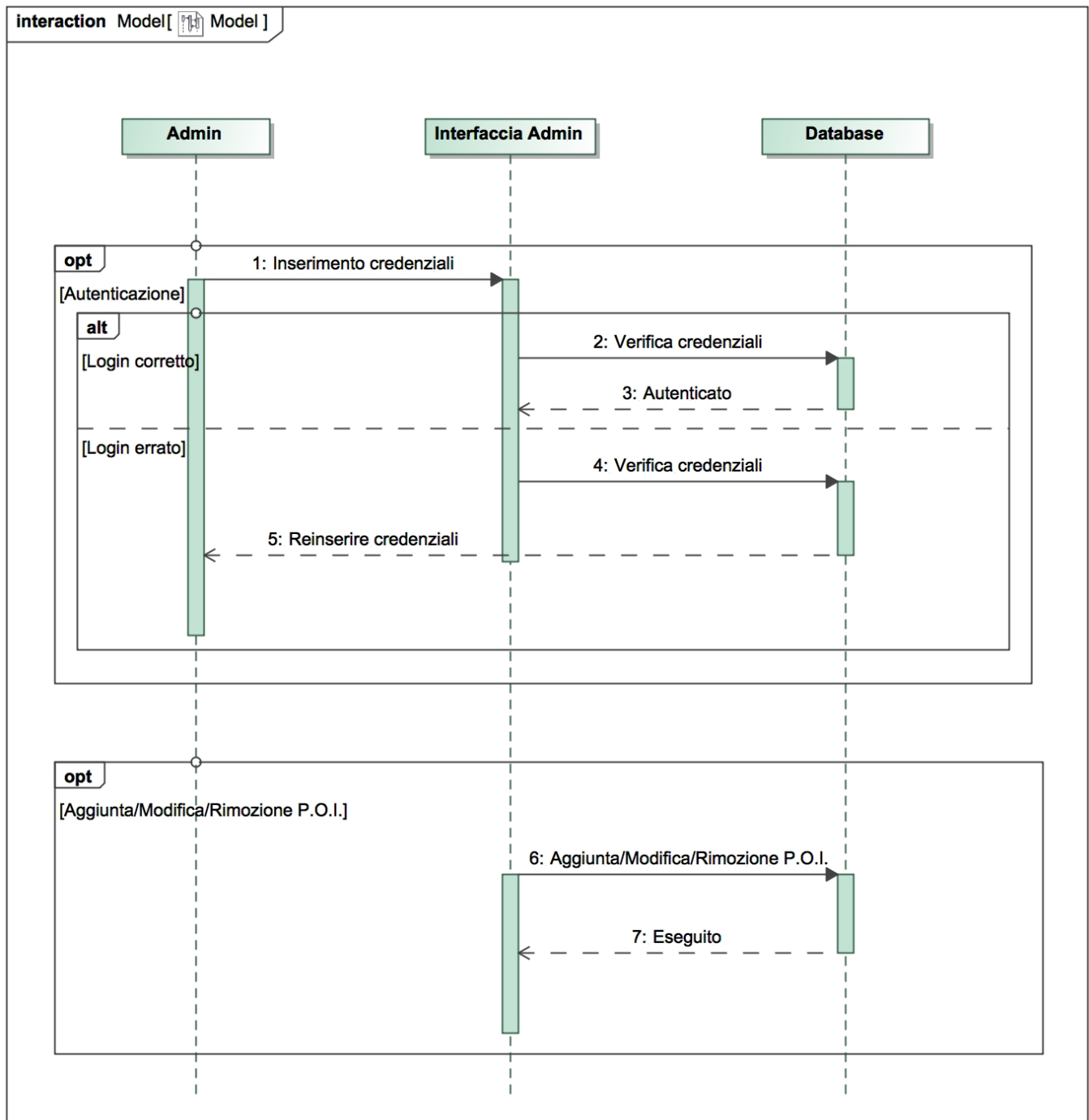
## C.2 The dynamic view of the software architecture: Sequence Diagram



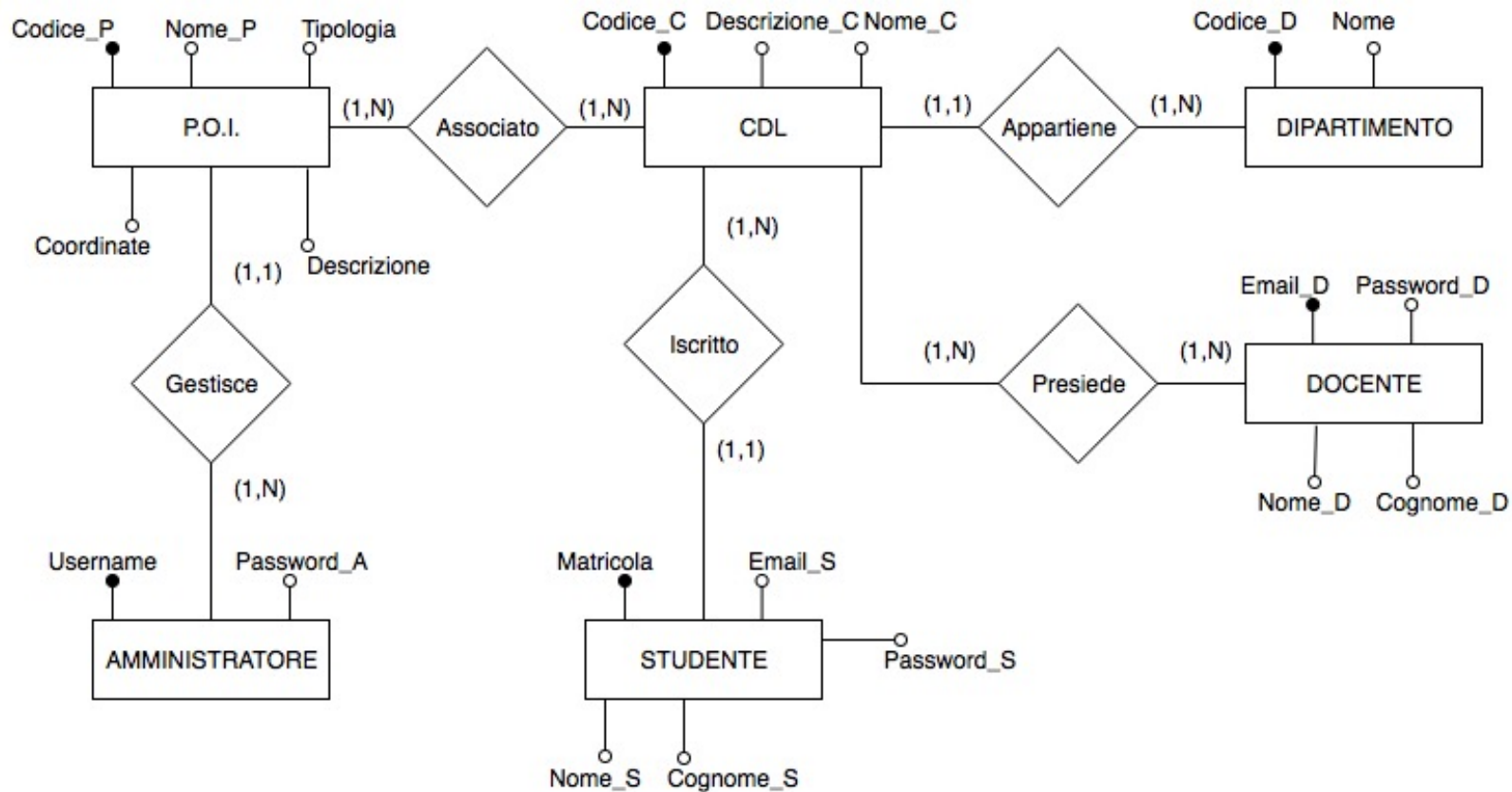
In questo esempio di Sequence Diagram mostriamo come viene gestito l'accesso di un utente, i dati vengono inviati al Database dell'Ateneo il quale li verificherà, successivamente tramite il controller filtraggio vengono settati tutti i

P.O.I. (in base all'utente loggato) presi dal Database ed infine viene rimandata l'interfaccia utente.

Viene mostrato anche come viene gestita la richiesta di ricerca P.O.I. tramite interfaccia, il P.O.I. viene cercato dall'utente e successivamente il Database verifica l'esistenza di esso e viene rimandato il risultato nell'interfaccia.



## D. Dati e loro modellazione



ENTITA'	ATTRIBUTI
P.O.I.	<b>Codice_P</b> , Nome_P, Tipologia, Descrizione, Coordinate, <u>Username</u>
CDL	<b>Codice_C</b> , Descrizione_C, Nome_C, <u>Codice_D</u>
ASSOCIATO	<b>Codice_P</b> , <b>Codice_C</b>
DIPARTIMENTO	<b>Codice_D</b> , Nome
STUDENTE	<b>Matricola</b> , Email_S, Password_S, Nome_S, Cognome_S, <u>Codice_C</u>
DOCENTE	<b>Email_D</b> , Password_D, Nome_D, Cognome_D
PRESIEDE	<b>Codice_C</b> , <b>Email_D</b>
AMMINISTRATORE	<b>Username</b> , Password_A

-Nella tabella P.O.I. saranno salvati tutti i punti di interesse, differenziati dall'attributo Codice\_P univocamente, per la tipologia di servizio che offrono e per il CDL a cui sono associati (tramite relazione con la tabella Associato).

-Nella tabella CDL saranno salvati tutti i corsi di Laurea con relativo Dipartimento (tramite relazione con la tabella Dipartimento).

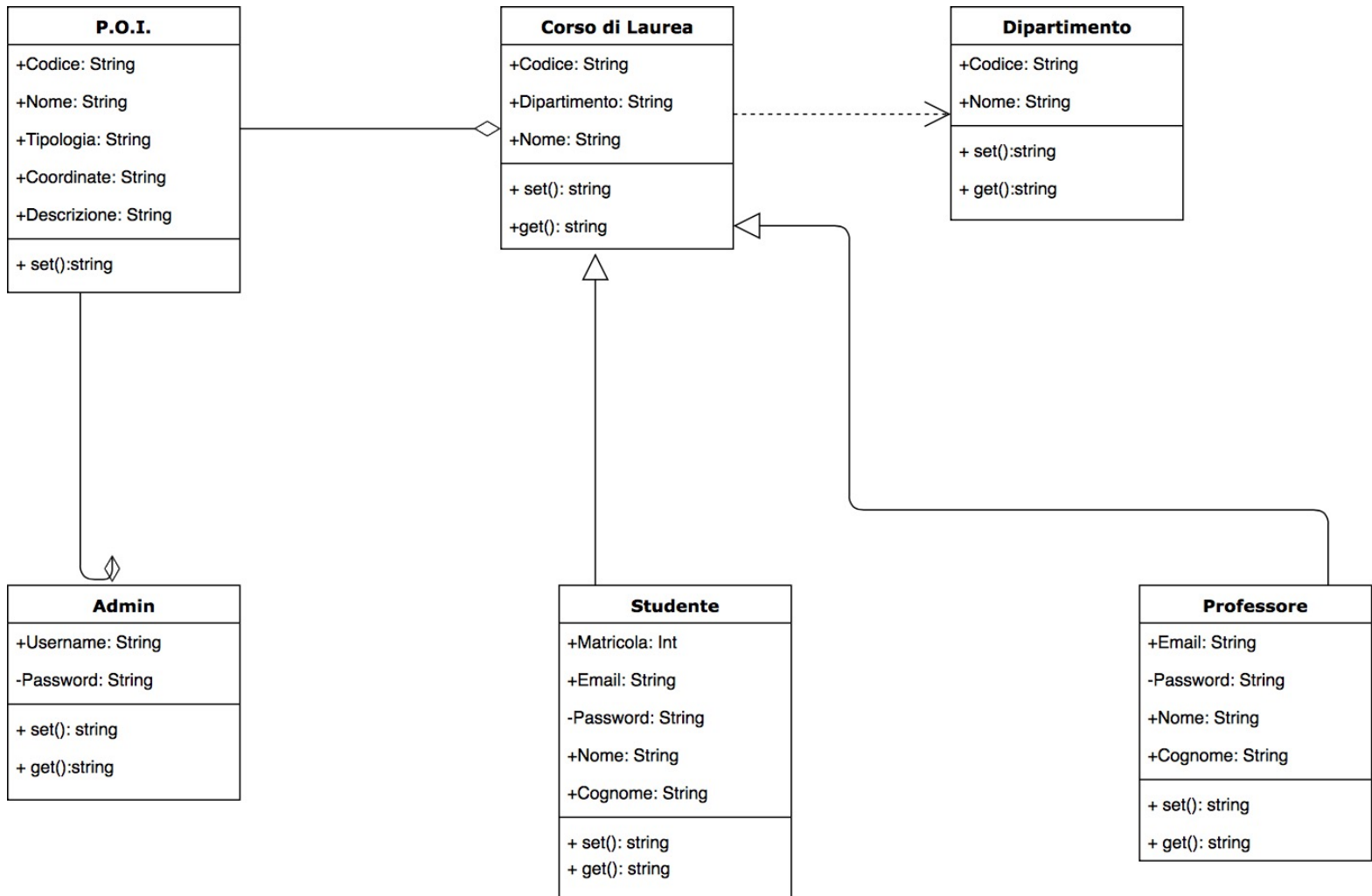
-Nella tabella Studente saranno salvati tutti gli studenti registrati tramite Matricola, Nome, Cognome, Email e Password e sarà associato il Corso di Laurea a cui sono iscritti (tramite la relazione con la tabella CDL)

-Nella tabella Docente saranno salvati tutti i docenti registrati tramite Email, Password, Nome e Cognome ed anch'essi saranno associati al/ai corsi di Laura cui fanno insegnamento (tramite la relazione con la tabella Insegna)

## E. Design Decisions

- Per la parte di autenticazione nell'homepage ci sarà una sezione dedicata dove poter fare il login con email e password già in possesso. Per l'autenticazione dell'admin ci sarà una pagina dedicata accessibile solo tramite un link non presente nell'interfaccia utente.
- Per la parte della mappa, sempre nella homepage, ci sarà la mappa completa con tutti i POI presenti con relativo logo a seconda del servizio; per evitare la visualizzazione confusionaria dei P.O.I. tutte le aule e gli uffici presenti nello stesso edificio verranno raggruppate in due unici punti. Quando un utente effettua la ricerca di un'aula specifica verranno visualizzate tutte le aule che rispettano i criteri di ricerca con i relativi dettagli riguardo la sua posizione (edificio e piano).
- Per la parte del filtro di ricerca ci sarà una lista a fianco della mappa dove si può decidere che POI visualizzare a seconda del filtro (aule, uffici, mense). Di default tutti i filtri saranno deselezionati.
- Per la funzione di ricerca ci sarà una barra di ricerca dove si può cercare per nome o per servizio il POI necessario.
- Per la prioritizzazione dei POI se l'utente è loggato ( docente o studente ) verranno visualizzati prima i POI più compatibili con l'utente ( in base a dipartimento, corso di laurea, anno di studio, oppure per i docenti corso di insegnamento). L'utente potrà comunque visualizzare la mappa senza nessuna prioritizzazione deselezionando l'opzione "Visualizza in base al corso di laurea".

## F. Design di Basso Livello



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

## G. Effort Recording

**PERT**

**Logging**

Personal Journal					
Team (number and name):TEAM TWO					
Student name:PULCINI ANDREA					
Student number:252006					
Email:andripulc@hotmail.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 23-24	4:00	2	Analisi generale del sistema da realizzare	learning	
11 27	2:00	2	Ricerca dei requisiti funzionali	learning/doing	
11 28	2:00	2	Elenco dei requisiti funzionali	doing	
11 29	1:00	1	Scenari D'uso	learning/doing	
11 29	1:00	1	Assunzioni	doing	
11 30	1:00	2	USE case Diagram	doing	
12 2	1:00	2	Analisi architettura software	learning	
12 3	2:00	1	Sequence diagram	doing	
12 4	1:30	2	Design decision	doing	
12 5	1:00	2	Revisione documentazione	doing	
12 11	2:00	1	Challenging Task	doing	
12 12	3:00	2	Comprensione Laravel	learning	
12 13	2:30	2	Analisi generale Prototipo	learning	
12 13	2:30	1	Comprensione e sviluppo Modello ER	learning/doing	
12 16	2:00	2	Modifiche template	doing	
12 18	3:00	2	Continuo del lavoro	doing	
12 20	2:30	2	Revisione	doing	
1 4	2:30	2	Comprensione Eloquent	learning	
1 6	2:30	2	Revisione documentazione	learning/doing	
1 7	2:30	1	Modifiche architettura software	doing	
1 8	1:30	1	Continuo del lavoro	doing	
1 9	3:00	2	Creazione modelli	learning/doing	
1 14	3:00	2	Aggiornamento Prototipo	doing	
1 16	2:30	2	Modifica documento	doing	
1 17-18	5:30	2	Sviluppo api	learning/doing	
Ore	Totali	57			



Personal Journal						
Team (number and name):TEAM TWO						
Student name:BENVENUTI DAVIDE						
Student number:253236						
Email: <a href="mailto:davide-b98@live.it">davide-b98@live.it</a>						
When (Month/Day)	Time spent	Partners (please report how many people have been working)	Brief Description of the performed task	Category	Sub-Category	
11	23-24	4:00	2	Analisi generale del sistema da realizzare	learning	
11	26	2:00	1	Ricerca di applicativi esistenti	learning	
11	27	2:00	2	Ricerca dei requisiti funzionali	learning/doing	
11	28	2:00	2	Elenco dei requisiti funzionali	doing	
11	29	1:00	1	Requisiti esclusi	learning/doing	
11	30	1:00	2	USE case Diagram	doing	
12	2	1:00	2	Analisi architettura software	learning	
12	3	2:00	1	Component diagram	doing	
12	4	1:30	2	Design decision	doing	
12	5	1:00	2	Revisione documentazione	doing	
12	12	3:00	2	Comprensione Laravel	learning	
12	13	2:00	2	Analisi generale Prototipo	learning	
12	16	1:00	1	Ristrutturazione Component Diagram	learning/doing	
12	16	2:00	2	Modifiche template	doing	
12	17	3:00	1	Sviluppo database Laravel	doing	
12	18	3:00	2	Continuo del lavoro	doing	
12	20	2:30	2	Revisione	doing	
1	4	2:30	2	Comprensione Eloquent	learning	
1	6	2:30	2	Revisione documentazione	learning/doing	
1	7-8	4:00	1	Comprensione Postman	learning	
1	9	3:00	2	Creazione modelli	learning/doing	
1	14	3:00	2	Aggiornamento prototipo	doing	
1	16	2:30	2	Modifica documento	doing	
1	17-18	5:30	2	Sviluppo api	learning/doing	
Ore	Totali	57				

### 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.

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

### **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**

## **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>*