

“Software Engineering”

Course

a.a. 2019-2020

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

<PROGETTO: 5 - PORTAMI A DESTINAZIONE>

Date	<05/12/2019>
Deliverable	D1
Team (Name)	Team Two

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

--	--	--	--

Table of Contents of this deliverable

LIST OF CHALLENGING/RISKY REQUIREMENTS OR TASK.....	3
A. STATO DELL'ARTE.....	4
B. RAFFINAMENTO DEI REQUISITI	
A.1 Servizi (con prioritizzazione).....	5
A.2 Requisiti non Funzionali.....	6
A.3 Scenari d'uso dettagliati.....	7
A.4 Excluded Requirements.....	7
A.5 Assunzioni.....	7
A.6 Use Case Diagrams.....	8
C. ARCHITETTURA SOFTWARE	
C.1 The static view of the system: Component Diagram.....	9
C.2 The dynamic view of the software architecture: Sequence Diagram....	10
D. DATI E LORO MODELLAZIONE.....	11
E. DESIGN DECISIONS.....	12
F. DESIGN DI BASSO LIVELLO.....	13
G. EXPLAIN HOW THE FRs AND THE NFRs ARE SATISFIED BY DESIGN.....	14
G. EFFORT RECORDING.....	15
APPENDIX. PROTOTYPE.....	17

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

A. Stato dell'Arte

Analizzate sistemi esistenti, prendete spunto da ciò che esiste

- *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*

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.

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.

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

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

A.4 Excluded Requirements

Il sistema escluderà l'utilizzo del controllo vocale e supporto vocale, per ragioni di tempo e conoscenza.

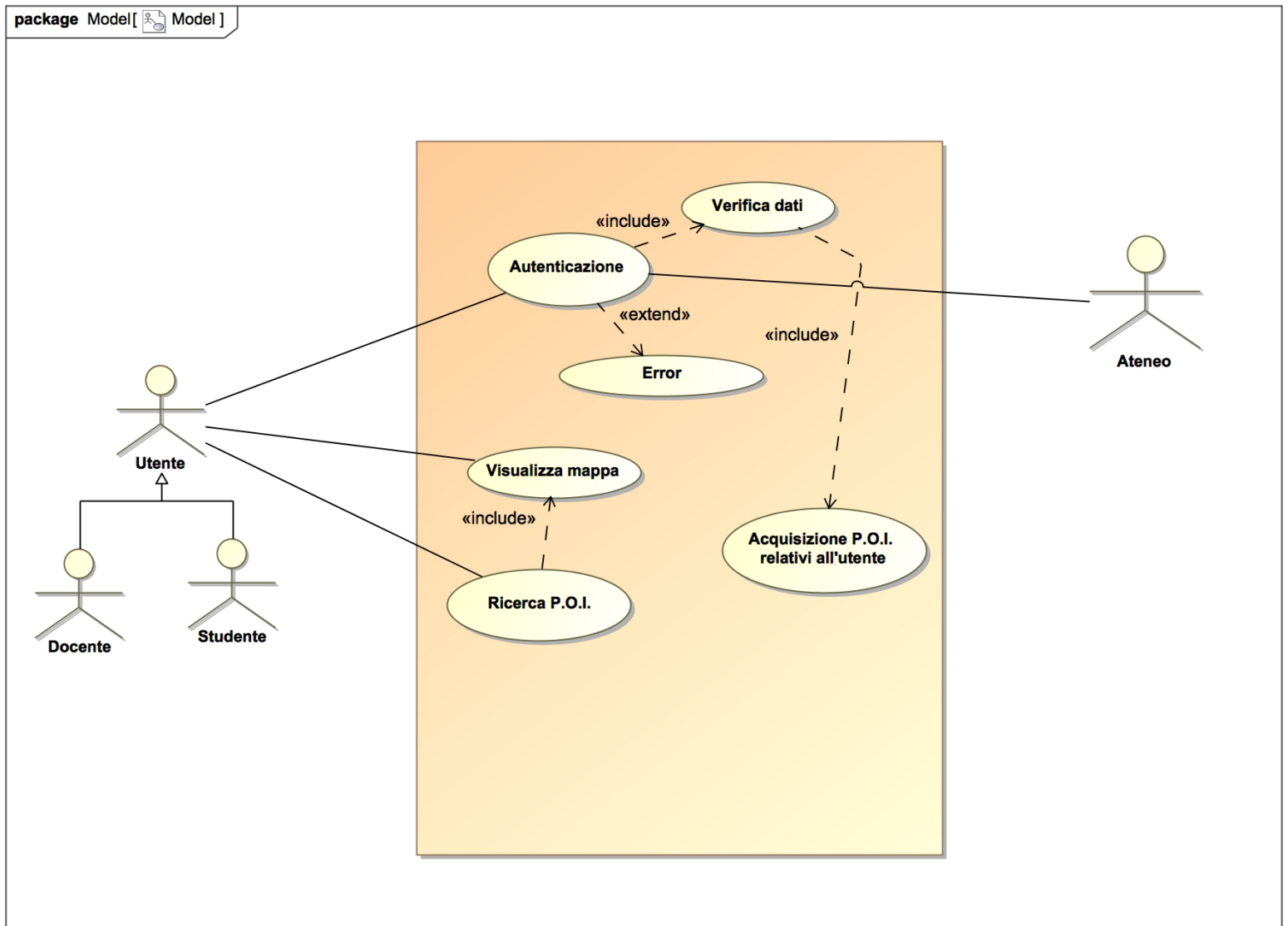
A.5 Assunzioni

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) .

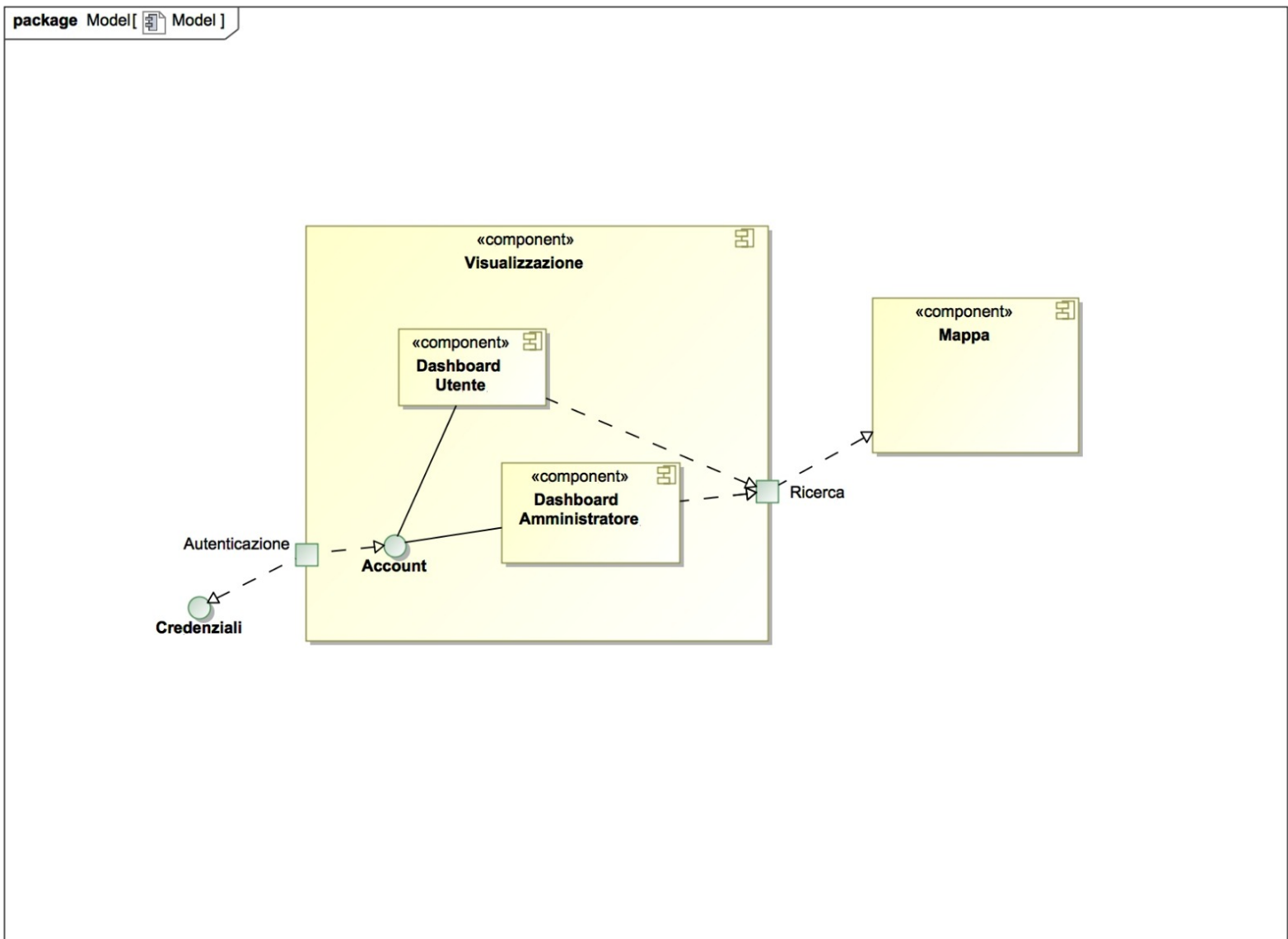
A.6 Use Case Diagrams



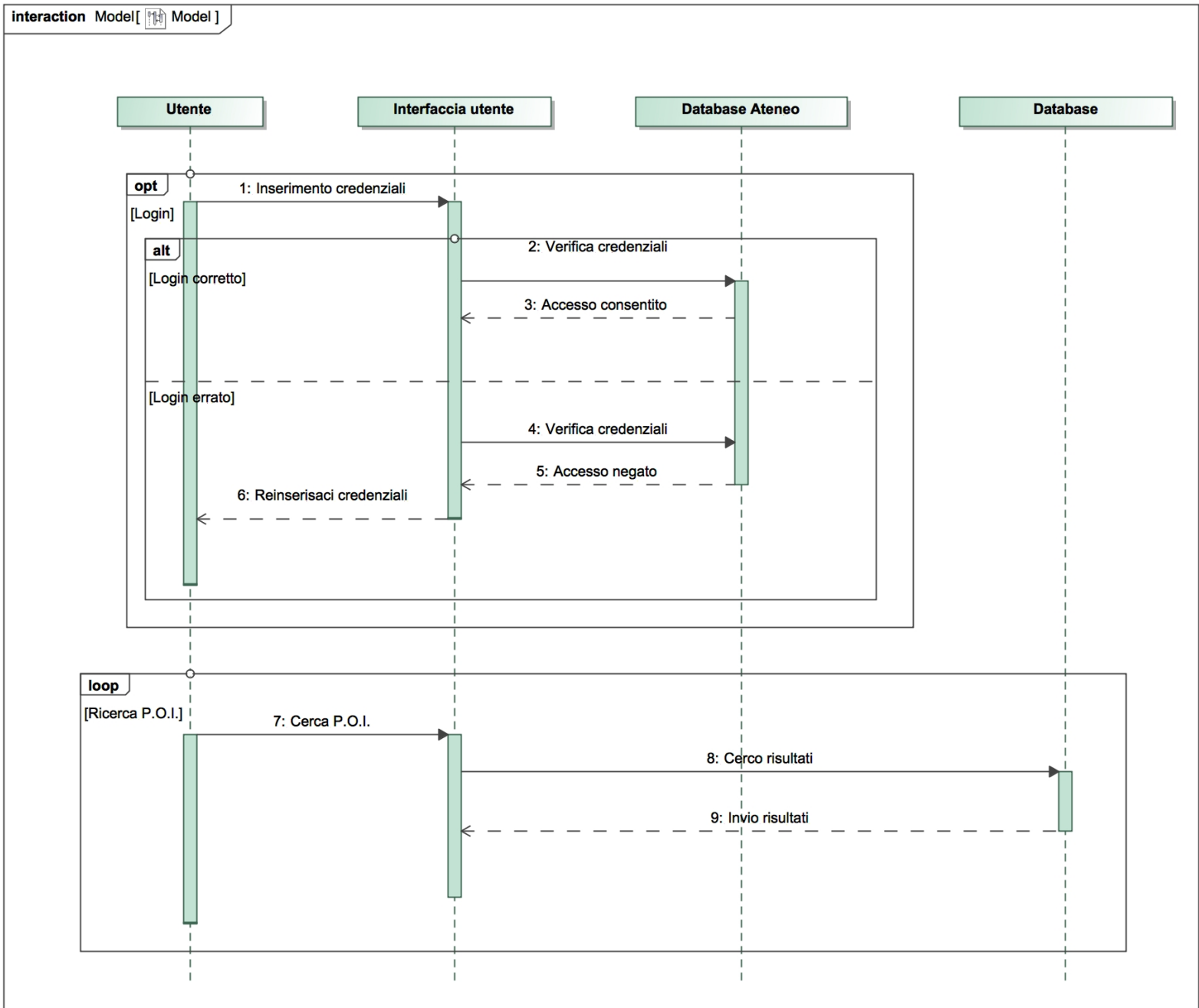
C. Architettura Software

<IF RELEVANT, Report here both the static and the dynamic view of your system design, in terms of a Component Diagram, and their related Sequence Diagrams >

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



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



D. Dati e loro modellazione

*Definite le sorgenti di dati a voi necessarie per realizzare I servizi di cui sopra.
Modellat tali dati tramite un ER o similari. Specificate se e quali di tali dati sono già
forniti da applicativi esistenti.*

E. Design Decisions

<Document here the **5** most important design decisions you had to take. You can use both a textual or a diagrammatic specification.>

F. Design di Basso Livello

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

<Report in this section how the architectural and low level design you produced satisfies the FRs and the NFRs>

- 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 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 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)
- 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)

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

Personal Journal						
Team (number and name):TEAM TWO						
Student name:BENVENUTI DAVIDE						
Student number:253236						
Email: davide-b98@live.it						
When (Month/Day)	Time spent	Partners <i>(please report how many people have been working)</i>	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		

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	

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

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>