DOCUMENTO DI SPECIFICA DEI REQUISITI SOFTWARE

**1. Introduzione**

1.1)Propositi

Il progetto “Logos” nasce dalla necessità di mettere in pratica quanto studiato durante il corso IFTS da quattro ragazzi (Giacomo Mattia Franco e Federico), sviluppando un applicativo Andorid/Web con supporto IoT per la creazione e gestione di corsi erogati frontalmente.

1.2)Obbiettivi

Facilitare la gestione dei corsi erogati in aula e seguente controllo presenze.

Si desidera dal punto di visto dello studente avere la possibilità di iscriversi ai corsi, reperire informazioni riguardo alle lezioni, da sostenere e sostenute, e tenere controllo delle proprie assenze e presenze alle lezioni ai corsi alla quale si è iscritti;

dal lato docente l’obbiettivo è di offrire la possibilità di creare, modificare e cancellare corsi e lezioni annesse, con la relativa gestione delle presenze ed assenze degli studenti.

1.3) Panoramica

La restante parte di questo documento contiene una descrizione dettagliata e approfondita delle funzionalità richieste al sistema software “Logos” secondo gli obiettivi espressi al punto 1.2.

**2. Descrizione**

Il sistema software “Logos” è integrata con la piattaforma hardware Arduino, utilizzata per il rilevamento delle presenze e delle assenze degli studenti alle lezioni dei corsi. Vedremo in dettaglio nei punti successivi come avviene.

2.1) Funzionalità del prodotto

Il sistema software “Logos” deve:

* Gestire una dashboard docente che mette a disposizione le seguenti funzionalità:
* login, logout e registrazione dell’account;
* creare un corso alla quale gli studenti possono iscriversi direttamente o essere iscritti dal docente stesso;
* definire lezione ed aggiungere argomenti al corso;
* Modificare e segnare le presenze e le assenze degli studenti.
* Gestire una dashboard studente che permette di:
* login, logout e registrazione dell’account;
* vedere la lista dei corso alla quale è possibile iscriversi;
* iscriversi ad uno o più corsi.

Le informazioni trattate devono poter essere gestite da diverse postazioni (terminali).

2.2) Caratteristiche dell’utente

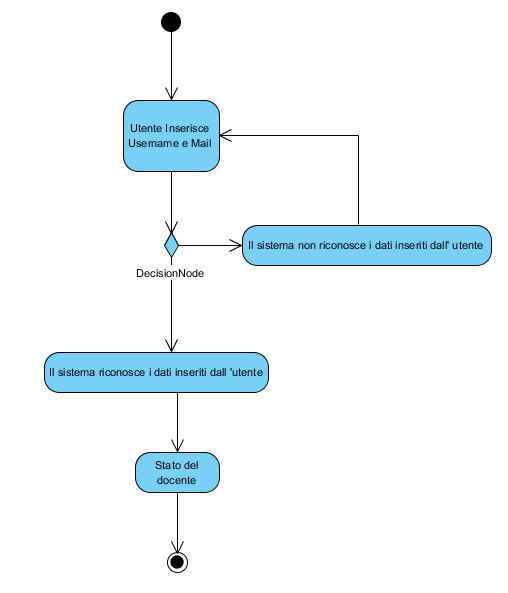
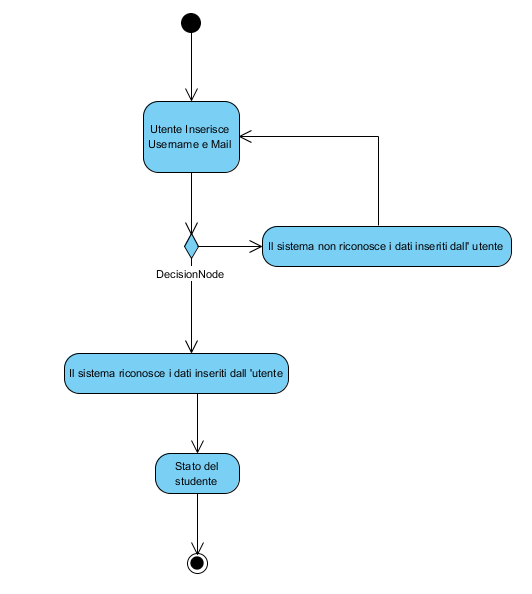
Il sistema software “Logos” è rivolto ad una utenza con discreta conoscenza del dominio applicativo (attività svolte legate all’erogazione di corsi di studio) ma senza particolari conoscenze informatiche. All’utente è richiesta solamente una conoscenza informatica di base (alfabetizzazione informatica).

2.3) Vincoli

Le informazioni trattate sono strettamente riservate e devono essere accessibili solamente al personale autorizzato. A ciascun studente è consentito di accedere solo ed esclusivamente ai dati della propria dashboard utente.

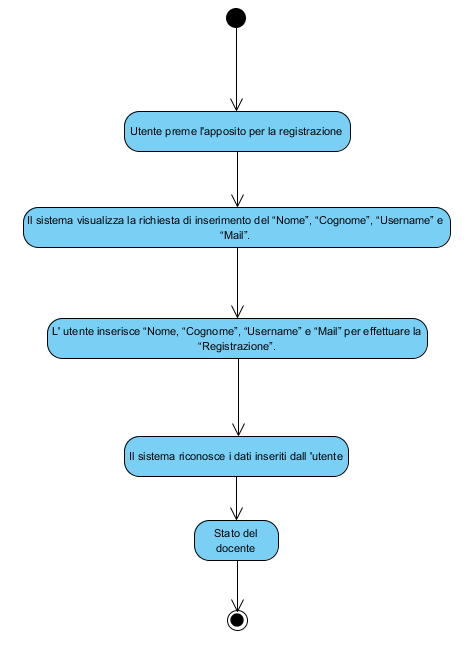
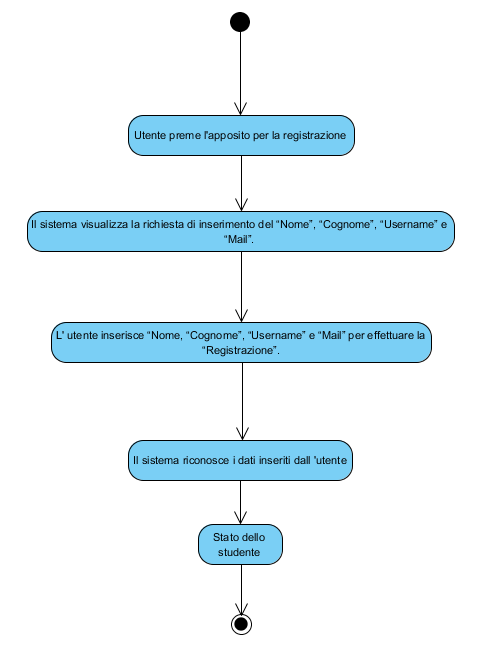
I vincoli da rispettare per effettuare il **login dell’utente**, prima citati nel punto 1.2, possono essere rappresentati mediante i diagrammi:

**lato studente: lato docente:**

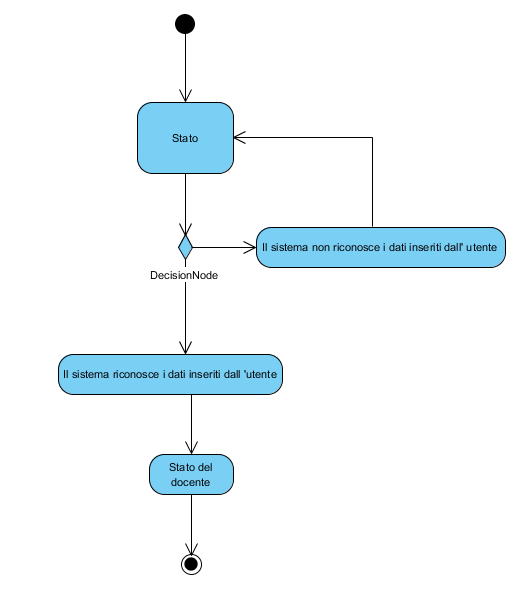
****

Per poter effettuare la **registrazione dell’utente**, i vincoli da rispettare possono essere rappresentati mediante i diagrammi:

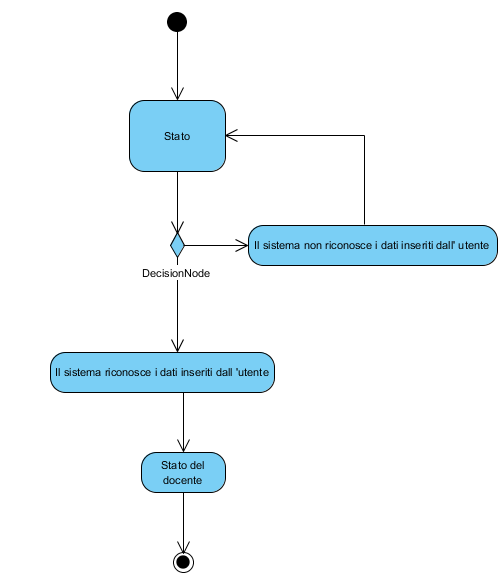
**lato studente: lato docente:**



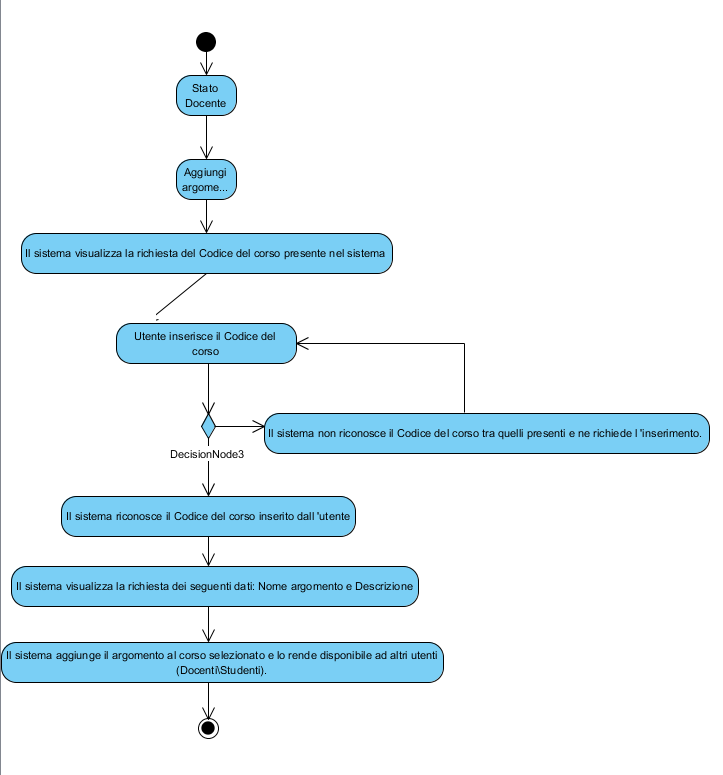
I vincoli da rispettare per **iscriversi ad un corso, da lato studente**, espressi nel punto 1.2, possono essere rappresentati mediante un diagramma:



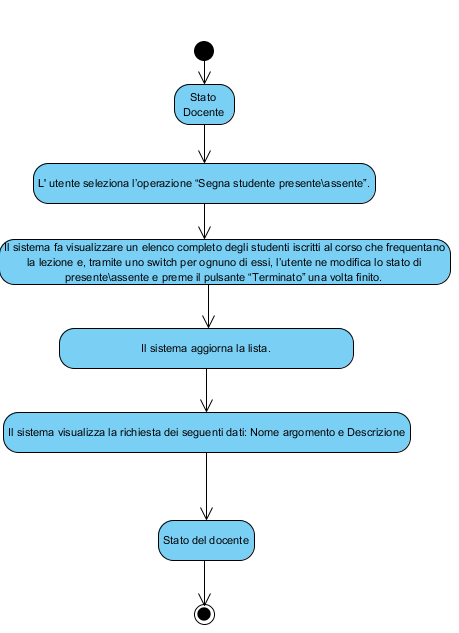
I vincoli da rispettare per **creare un corso, da lato docente**, espressi nel punto 1.2 possono essere rappresentati mediante un diagramma:



Per consentire all’utente **docente di poter aggiungere argomenti al corso**, i vincoli da rispettare sono rappresentati nel seguente diagramma:



La funzione di **segnare l’assenza di uno studente, lato docente,** espressa nel punto 1.2, è rappresentabile mediante diagrama:



2.4) Assunzioni e dipendenze

Il sistema software “Logos” dovrà essere utilizzato su dispositivi dotati di sistema operativo Android 4.4 APls® e superiori.

**3. Requisiti specifici**

3.1) Requisiti interfaccia esterna

L’ **interfaccia utente** nel sistema software “Logos” deve essere dotato di un’interfaccia amichevole, con menu a tendina, finestre e pulsanti.

L’ **interfaccia hardware** del sistema software “Logos” deve interfacciarsi con la piattaforma hardware Arduino, utilizzata per il rilevamento dei presenti alla lezione mediante rilevamento Bluetooth.

L’ **interfaccia software** del sistema software “Logos” non deve interfacciarsi con nessun sistema software.

L’ **interfaccia di comunicazione** del sistema software “Logos” non richiede l’uso di una particolare interfaccia di comunicazione.

3.2) Requisiti funzionali

**Login:**

Introduzione: effettua il login di un utente.

Input: si forniscono la mail e la password dell’utente.

Elaborazione: verrà controllato che i dati inseriti corrispondono alle specifiche di accesso dell’utente presenti su database.

Output: in caso di corretto inserimento dei dati il sistema mostrerà la dashboard relativa all’utente ed i suoi relativi menu, altrimenti stampa un messaggio di errore e la richiesta di reinserimento dei dati o registrazione.

**Registrazione:**

Introduzione: registrazione di un utente.

Input: si forniscono il nome, il cognome, la mail, la password dell’utente e la selezione di uno dei due bottoni per selezionare il tipo di utente (studente o docente).

Elaborazione: verrà controllato che i dati inseriti non corrispondono alle specifiche di accesso di un utente già presente nel database e quindi aggiunti in quest’ultimo.

Output: in caso di corretto inserimento dei dati il sistema mostrerà la dashboard relativa all’utente ed i suoi relativi menu, altrimenti stampa un messaggio di errore e la richiesta di reinserimento dei dati di registrazione.

**Cambia Password**

Introduzione: permette agli utenti di cambiare la propria password.

Input: richiede la vecchia password e l’inserimento della nuova password.

Elaborazione: verrà controllato che i dati inseriti siano corretti ed infine li aggiornerà.

Output: in caso di corretto inserimento, il sistema stamperà un messaggio di successo ed il relativo menu del corso, altrimenti stampa un messaggio di errore e la richiesta di reinserimento dei dati.

**Lato docente: cancella studente**

Introduzione: consente di rimuovere uno studente dal corso.

Input: chiede la mail del docente, la mail dello studente e l’ID del corso.

Elaborazione: si assicura che gli attributi passati siano corretti.

Output: stamperà un messaggio di successo. In caso di errore nell’inserimento dei dati stampa un messaggio di errore e la richiesta di reinserimento dei dati.

**Lato docente: crea nuovo corso**

Introduzione: consente al docente di creare un nuovo corso.

Input: si dovrà fornire il titolo del corso, il numero di lezioni, ore per lezione, numero di lezioni per giorno, la descrizione del corso, il numero massimo di studenti, la sede dove si terrà il corso, requisiti minimi degli studenti.

Elaborazione: verrà generato un oggetto JSON contenente le informazioni che

verranno inviate tramite Servlet, creando un oggetto contenente la tabella del database.

Output: registrazione dei dati relativi al corso creato nell’archivio destinato a contenerli. Sarà stampato un messaggio che conferma la creazione.

**Lato docente: elimina corso**

Introduzione: il docente elimina un corso.

Input: ha bisogno della mail del docente, l’ID del corso

Elaborazione : controlla che i dati passati siano giusti ed elimina il corso tra quelli disponibili.

Output: manda un messaggio di successo o , in caso di errore, richiede l’inserimento dei precedenti dati.

**Lato docente: modifica corso**

Introduzione: consente al docente di modificare le caratteristiche del corso riguardo alla data di inizio, l’ora di inizio lezione, la descrizione, l’immagine, la lezione corrente, le lezioni effettuate, il numero di lezioni, numero studenti iscritti, numero massimo di studenti, la durata delle lezioni, le ore totali, le ore trascorse, la sede, lezione per giorno, il titolo.

Input: richiede la mail del docente e l’ID del corso.

Elaborazione : controlla che i dati passati siano giusti e modifica l’attributo interessato del corso tra quelli disponibili.

Output: stampa un messaggio di successo o , in caso di errore, richiede l’inserimento dei precedenti dati.

**Lato docente: iscrivi studente**

Introduzione: il docente iscrive uno studente

Input: richiede la mail del docente, l’ID del corso e la mail dello studente.

Elaborazione: controlla che i dati siano corretti e aggiunge lo studente al corso.

Output: manda un messaggio di successo o , in caso di errore, richiede l’inserimento dei dati.

**Lato docente: crea lezione**

Introduzione: permette al docente di creare una lezione al corso.

Input: si passano gli argomenti della lezione, l’aula dove si terrà la lezione, la data, la la durata e l’ora di inizio.

Elaborazione: aggiunge gli argomenti passati e crea la lezione nel corso indicato.

Output: visualizza un messaggio di successo di creazione della lezione.

**Lato docente: modifica lezione**

Introduzione: consente al docente di modificare una lezione creata nel corso.

Input:

Elaborazione:

Output:----------------------------------------------------------------------------------------

**Lato docente: elimna lezione**

Introduzione: consente al docente di modificare una lezione creata nel corso.

Input:

Elaborazione:

Output:---------------------------------------------------------------------------------------

**Lato docente: segna studente assente\presente**

Introduzione: consente al docente di segnare le assenze e le presenze degli studenti alle lezioni.

Input: è rappresentato da uno switch button per ogni studente che ne regola le assenze e le presenze per ogni lezione.

Elaborazione: modifica lo stato di assente e presente dello studente a seconda dell’utilizzo dello switch button.

Output: fa visualizzare a monitor una lista completa degli studenti iscritti al corso che sono assenti e presenti nella lezione specifica.

**Lato studente: iscriviti ad un corso**

Introduzione: permette allo studente di iscriversi ad un corso

Input: si fornisce il codice del corso alla quale si vuole iscriversi.

Elaborazione: verrà controllato che il codice del corso inserito corrispondono al codice di un corso presente nel database.

Output: in caso di corretto inserimento, il sistema stamperà un messaggio di successo ed il relativo menu del corso, altrimenti stampa un messaggio di errore e la richiesta di reinserimento dei dati.

**Lato studente: visualizza lista corsi disponibili**

Introduzione: consente allo studente di visualizzare una lista completa dei corsi disponibili per lo studente in questione

Input:

Elaborazione:

Output:---------------------------------------------------------------------------------------

3.3)Requisiti di prestazione

Sono richiesti tempi di risposta ragionevolmente contenuti sia per le operazioni sugli archivi che per l’inizio di una stampa.

3.4) Vincoli di progetto: Nessuno.

3.5) Attributi del sistema software:

**Multiutenza:** Le informazioni trattate dal sistema software “Logos” devono poter essere gestite da diverse postazioni (terminali). Ciascun utente dispone di una dashboard con diverse funzionalità, limitatamente a seconda se si è dal lato docente o dal lato studente.

**Sicurezza:** Il sistema software “Logos” tratta informazioni strettamente riservate che devono essere accessibili solamente al personale autorizzato. A ciascun utente è consentito di accedere solo ed esclusivamente ai dati della propria dashboard. E’ richiesto l’uso di una password e l’e-mail(attributo univoca) per accedere ai servizi forniti dal sistema.

3.6) Altri requisiti:

**Database:**

il sistema software “Logos” non deve utilizzare archivi preesistenti. E’ richiesto l’uso di un RDBMS (Relational DataBase Management System).

**4. Requisiti di sistema, progetto web**

4.1) Descrizione :

Il progetto web del sistema software “Logos” ha un funzionamento del tutto simile a quello precedentemente descritto nella SRS dell’applicazione.

Le **funzionalità** hanno lo stesso comportamento di quelle elencate al punto 2.1) “Funzionalità del prodotto” permettendo la gestione dell’erogazione di corsi di studio, delle loro lezioni e delle presenze ed assenze degli studenti. Le caratteristiche legate ai **vincoli** ed ai **requisiti** sono riconducibili e del tutto analoghi ai punti 2.3)”Vincoli” e 3)”Requisiti”.

5) Funzionalità legate alla comunicazione:

Lo sviluppo web è legato all’utilizzo della funzione hash crittografica MD5, la cui la cui funzione genera un codice che identifica univocamente un file.

Nel web project “Logos” l’utilizzo della MD5 è stato utile per il trasferimento dell’oggetto persona, legato ad ogni utente, utilizzando un codice che è ottenuto attraverso la funzione MD5 della mail (che rappresenta la chiave primaria di ogni utente), garantendo un metodo di trasferimento sicuro e più facile e che ogni utente abbia un suo codice univoco.