

PERCORSO ANALITICO NELLO SVILUPPO SOFTWARE

ANALISI DEI REQUISITI

Introduzione:

Scopo del Sistema:

- L'applicazione mira a fornire una piattaforma digitale che simulerebbe il funzionamento di un normale registro scolastico ma elettronico, facilitando la gestione delle attività didattiche e semplificando la comunicazione tra docenti, studenti e genitori.

Destinatari:

- L'applicazione è destinata agli studenti, ai professori e ai genitori.

REQUISITI FUNZIONALI:

Gestione Utenti:

- Differenza degli utenti con ruoli specifici (docente, studente, genitore, amministratore).
- Possibilità di creare, modificare e cancellare account utente.

Gestione Classi e Corsi:

- Creazione e gestione delle classi con informazioni dettagliate.
- Assegnazione di docenti e studenti ai corsi specifici.

Gestione delle Lezioni:

- Inserimento e visualizzazione delle lezioni pianificate con dettagli sugli orari e le aule.
- Presenze ed assenze degli studenti.
- Possibilità di inserire e valutare argomenti trattati durante la lezione.

Gestione Compiti e Valutazioni:

- Assegnazione di compiti agli studenti.
- Registrazione e visualizzazione delle valutazioni inserite.
- Calcolo automatico delle medie e mostrata.

Gestioni degli eventi

- Il sistema potrebbe includere un calendario degli eventi scolastici, come riunioni dei genitori, gite scolastiche, esami, festività, ecc.
- Registrazione delle note: Implementare un sistema per registrare e visualizzare note disciplinari, richiami o avvertenze per gli studenti.

- Bacheca di notizie: Creare una bacheca virtuale che consenta agli studenti di visualizzare notizie e comunicazioni inerenti alla scuola.
- Visualizzazione della pagella dell'anno precedente: Fornire agli studenti la possibilità di visualizzare la pagella dell'anno precedente con dettagli sui voti e la media.

Archiviazione dei documenti

- Possibilità di visualizzazione della pagella dell'anno precedente, voti e tutto ciò che era registrato.

ANALISI FUNZIONALE:

Accesso:

- L'amministratore ha la possibilità di poter aggiungere e eliminare gli utenti.
- Gli utenti devono accedere inserendo le proprie credenziali di accesso (username e password), se le credenziali errate, il sistema deve fornire un feedback e consentire un nuovo tentativo, e dare la possibilità di eseguire un logout.

Visualizzazione delle Lezioni:

- Gli utenti devono poter visualizzare l'elenco delle lezioni pianificate, con dettagli sugli orari, le aule, gli argomenti trattati e i vari compiti assegnati, le valutazioni, le presenze e le assenze di ogni lezione.
- I Professori hanno la possibilità di aggiungere le lezioni svolte, compiti, le valutazioni, le presenze e le assenze di ogni lezione.

Visualizzazione degli eventi:

- Mostrare gli eventi scolastici, gli appuntamenti e riunioni, visualizzazione delle note o annotazioni, nuovi pagamenti da effettuare, visualizzare gli scrutini e i debiti da effettuare.

ANALISI TECNICA:

JAVA:

UTENTE (CLASSE GENERALE):

- insertUsername(): metodo che serve per inserire il nome utente del user nella fase di accesso, di creazione o di modifica del proprio account.
- insertPassword(): metodo per inserire la password di ogni user nella fase di creazione dell'account, di modifica, o di accesso al proprio account.
- viewAccountSettings(): metodo per modificare le impostazioni dell'account compresa la password e il nome utente.

STUDENTE (EXTEND UTENTE):

- viewGrades(): metodo che consente all'utente di tipo studente di visualizzare le valutazioni.
- viewLessons(): metodo che consente allo studente di visualizzare le lezioni.
- viewNoticeBoard(): metodo per visualizzare la bacheca.
- viewEvents(): metodo per visualizzare gli eventi in programma.
- viewAgenda(): metodo per visualizzare l'agenda.
- viewDocuments(): metodo che permette allo studente di scaricare documenti, come scrutini e certificati vari.
- viewSchoolReport(): metodo che permette allo studente di visualizzare le note disciplinari e le annotazioni.
- viewFiles(): metodo che permette di visualizzare i file inviati dalla scuola (es: circolari)

INSEGNANTE (EXTEND UTENTE):

- setGrade(): metodo che permette all'utente di tipo insegnante di inserire le valutazioni di uno più studenti della stessa classe.
- viewGrade(): metodo che permette all'insegnante di visualizzare i voti di ogni studente in parte.
- insertLessons(): metodo che permette all'insegnante di inserire sul registro gli argomenti svolti a lezione.
- viewLessons(): metodo che permette di visualizzare quello che ha inserito nella sezione "lezioni" del registro
- viewNoticeBoard(): metodo che permette all'insegnante di visualizzare la bacheca.
- viewEvents(): metodo che consente all'utente di visualizzare gli eventi
- viewAgenda(): metodo che permette di visualizzare il contenuto dell'agenda
- setAgendaNote(): metodo che permette di inserire compiti in agenda
- viewDocuments(): metodo che permette allo studente di scaricare documenti, come scrutini e certificati vari.
- viewSchoolReport(): metodo che permette allo studente di visualizzare le note disciplinari e le annotazioni.
- viewFiles(): metodo che permette di visualizzare i file inviati dalla scuola (es: circolari).
- viewJustifyAbsences(): Possibilità di visualizzare se il genitore ha giustificato l'assenza. (questo metodo vale sia per le assenze, che per i ritardi, che per gli anticipi).
- confirmJustify(): conferma l'assenza dello studente.

GENITORE (EXTEND STUDENTE):

- setMeeting(): Possibilità di prenotare un incontro con i professori.
- Amministrazione assenze/presenze.
- justifyAbsences(): Possibilità di giustificare le assenze.

AMMINISTRATORE (EXTEND UTENTE):

- viewClass(): visualizza elenco degli studenti di una classe
- viewTeacher(): visualizza elenco degli insegnanti
- setGrade(): permette di inserire i voti a un singolo studente.
- viewGrade(): permette di vedere i voti di uno studente.
- insertLessons(): permette di inserire gli argomenti delle lezioni.
- viewLessons(): permette di visualizzare le lezioni svolte.
- viewNoticeBoard(): metodo che consente di visualizzare la bacheca.
- viewEvents(): metodo che consente di visualizzare gli eventi in programma.
- viewAgenda(): permette di visualizzare i contenuti dell'agenda
- setAgendaNote(): permette di inserire compiti, verifiche.
- viewDocuments(): metodo che consente di visualizzare i documenti.
- viewSchoolReport(): permette di visualizzare le pagelle
- viewFiles(): Visualizzazione dei file all'interno della bacheca.

