Solve.it

Documento di progetto

Gruppo T053

Ingegneria del Software

Sommario

[Obiettivo del progetto 3](#_Toc116312012)

[Macro-funzionalità del sistema 3](#_Toc116312013)

[Requisiti funzionali 4](#_Toc116312014)

[Requisiti non funzionali 6](#_Toc116312015)

[Front-end 7](#_Toc116312016)

[Back-end 8](#_Toc116312017)

# Obiettivo del progetto

Il progetto Solve.it consiste nella realizzazione di una Web-app accessibile a studenti e docenti dell’università di Trento per mettere a disposizione un servizio automatico di esercitazione e di condivisione di materiale didattico. Il focus dell’applicazione è quello di offrire, agli studenti che accedono al servizio, l’opportunità di esercitarsi con del materiale inerente ai corsi di studio seguiti, nella maniera più efficiente possibile stando allo stesso tempo al passo con le proposte del docente. Il sistema Solve.it si focalizza su esercitazioni di tipologia “problem solving”.

## Macro-funzionalità del sistema

1. Il materiale e le esercitazioni pubblicate su Solve.it sono suddivisi in corsi
2. Per ogni corso creato i docenti possono caricare le esercitazioni e del materiale didattico (video, pdf, documenti, ecc.)
3. Gli studenti che accedono alla piattaforma possono iscriversi a qualunque dei corsi proposti dai docenti e usufruire del materiale caricato e delle esercitazioni pubblicate
4. Solve.it raccoglie delle statistiche sugli esiti delle esercitazioni, che vengono poi mostrate sia agli studenti (statistiche personali) che ai docenti (statistiche sulla singola esercitazione)
5. Gli studenti iscritti ad un corso possono usufruire delle esercitazioni passate per prepararsi all’esame

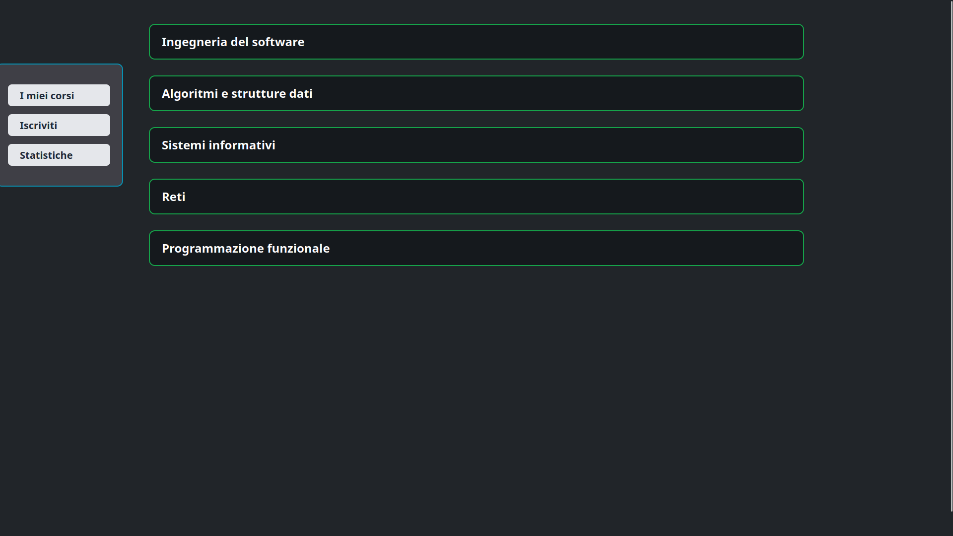
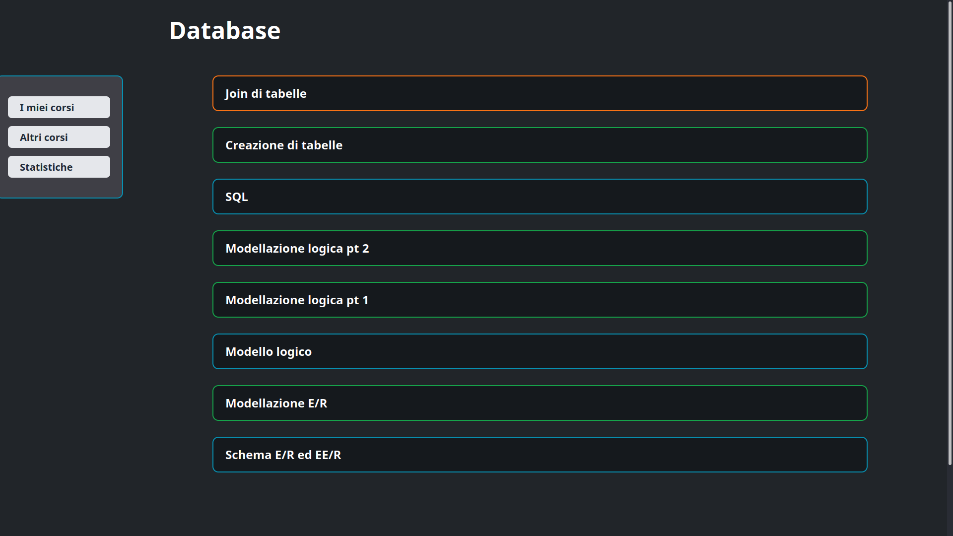
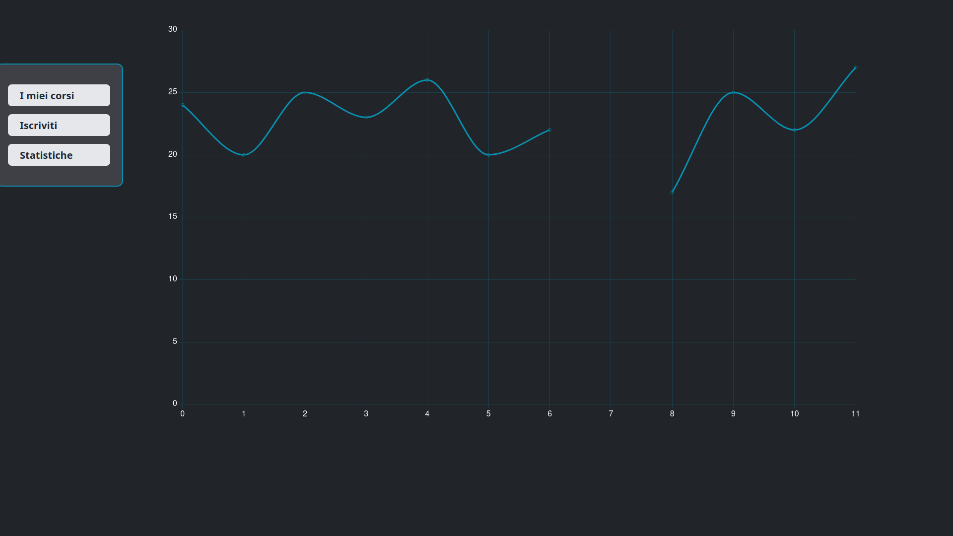
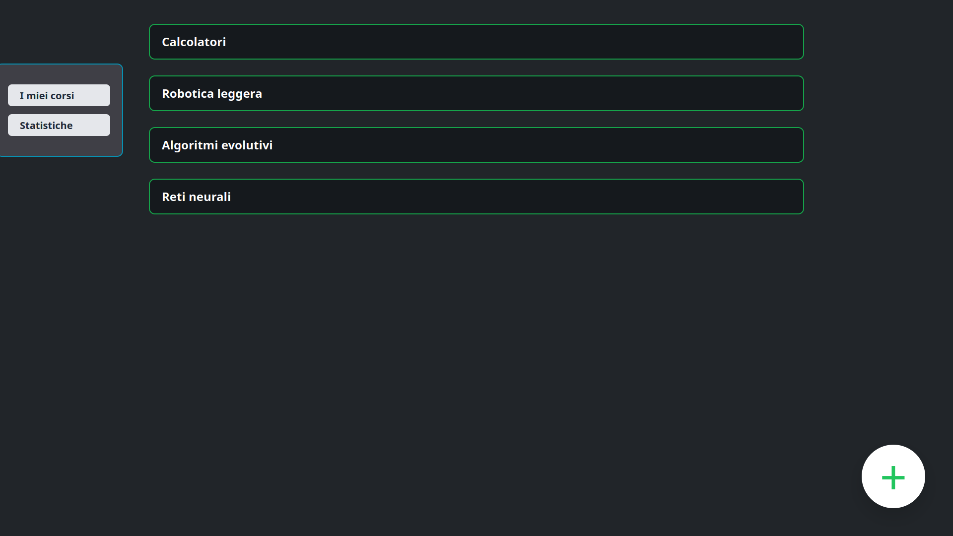
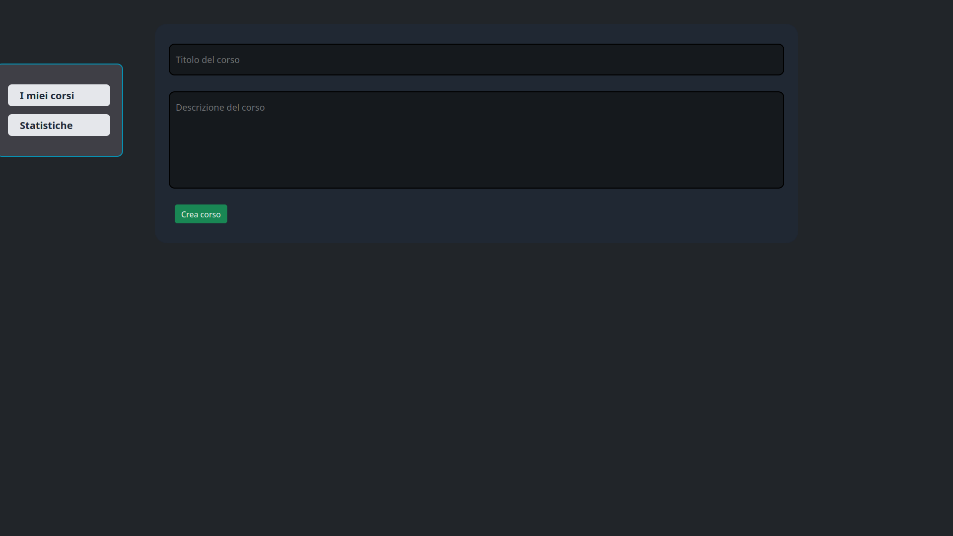
# Requisiti funzionali

1. Le esercitazioni che il sistema deve distribuire agli studenti sono composte da:
   1. Un testo (in cui viene descritto il problema da risolvere)
   2. Una tipologia di risposta
   3. Una soluzione
   4. Una spiegazione della risoluzione del problema
2. Il sistema deve supportare come formato di risposta per un’esercitazione uno tra i seguenti:
   1. Risposta numerica
   2. Crocette a risposta singola
   3. Crocette a risposta multipla
3. Il sistema deve temporizzare la pubblicazione delle esercitazioni come segue:
   1. Ogni esercitazione deve avere una data di pubblicazione
   2. Ogni esercitazione deve avere una data di terminazione (alla quale viene pubblicata la spiegazione della procedura di risoluzione)
4. La descrizione della procedura di risoluzione di una data esercitazione può essere espressa in formato testuale o allegando un relativo documento
5. Il docente deve essere vincolato dal sistema a caricare per ogni esercitazione la spiegazione della procedura di risoluzione
6. Il sistema valuta in modo automatico le risposte alle esercitazioni fornite dagli studenti:
   1. Ogni risposta deve essere valutata in centesimi
   2. Agli studenti deve essere concesso sottomettere un numero arbitrario di soluzioni, ma il sistema deve considerare valida esclusivamente l’ultima risposta data
7. Il sistema deve permettere ai docenti di creare nuovi corsi
   1. Il docente può specificare un abstract per il corso
   2. Il docente deve specificare il titolo del corso
8. Il sistema deve permettere ai docenti di visualizzare i propri corsi
9. Il sistema deve permettere al docente di caricare in un corso delle risorse, quali:
   1. Documenti
   2. Immagini
   3. Video
   4. Audio
10. Tutti gli utenti devono registrarsi al sistema tramite le proprie credenziali dell’università di Trento
11. Il sistema deve mostrare ai docenti una schermata per visualizzare tutti i propri corsi
12. Il sistema, a sign-in dello studente effettuato, deve mostrare:
    1. Tutti i corsi a cui lo studente è iscritto
    2. L’elenco di tutti i corsi disponibili a cui lo studente non è iscritto attualmente
13. Quando lo studente seleziona un corso a cui è iscritto dal relativo elenco, la piattaforma deve mostrare:
    1. L’elenco dei materiali didattici del corso e l’elenco delle esercitazioni suddivise in esercitazioni in corso ed esercitazioni già terminate
14. Il sistema deve fornire ad ogni studente una schermata per visualizzare tutti i corsi a cui è iscritto
15. Quando lo studente seleziona dal relativo elenco un corso a cui non è attualmente iscritto, la piattaforma deve mostrare i dettagli associati al corso (docente e abstract del corso)
16. La piattaforma deve permettere agli studenti di effettuare una ricerca dagli elenchi di corsi in base ai seguenti criteri:
    1. Nome del docente
    2. Nome del corso
17. La piattaforma deve raccogliere dati statistici sui risultati conseguiti dagli studenti nello svolgimento delle esercitazioni
    1. Per una data esercitazione la piattaforma deve raccogliere i dati esclusivamente sull’ultima consegna effettuata dallo studente
18. La piattaforma deve permettere ai docenti di visualizzare le statistiche raccolte per una data esercitazione
19. La piattaforma deve permettere agli studenti di visualizzare per ogni corso le statistiche di svolgimento di tutti gli esercizi già consegnati
20. Il sistema deve permettere ai docenti di vedere il numero di studenti che hanno consegnato almeno una risposta ad una certa esercitazione, e in forma anonima di visualizzare i punteggi di ogni singolo studente che ha effettuato la consegna
21. Ogni esercitazione deve poter essere marcata dal docente che la crea come “esercitazione d’esame”
22. Il sistema deve permettere agli studenti di effettuare simulazioni d’esame
    1. Il sistema deve selezionare a random tra le esercitazioni marcate come “d’esame” in un numero specificato dall’utente
    2. La durata limite della simulazione deve essere specificata dall’utente prima dell’avvio della simulazione
    3. Il sistema al termine dell’esercitazione deve calcolare in automatico il punteggio e mostrarlo allo studente

# Requisiti non funzionali

1. L’accesso al sistema (meccanismi di sign-in e sign-up) deve avvenire in maniera sicura
   1. Tutte le transazioni tra front-end e back-end devono essere trasmesse tramite protocollo https
   2. Se un utente front-end non è attivo (ossia non invia richieste al server) la sessione scade dopo 20 minuti
   3. A sign-in effettuato una sessione di attività del front-end deve essere identificata da un token temporaneo riconosciuto dal back-end, che dunque lo deve conservare durante tutto il suo arco di attivazione
   4. Tutte le richieste https che il client effettua verso il server devono essere marcate con il token della sessione
   5. Il back-end può fornire risposte alle richieste https dei client solo se il token fornito nella richiesta è valido
   6. Un token è valido se è presente tra i token attivi noti al back-end ed il tempo trascorso dall’ultima richiesta ricevuta con lo stesso token è inferiore al tempo limite stabilito al requisito non funzionale 1.2.
   7. Se il token di sessione non è valido il back-end informa il front-end che la sessione non è attiva ed è necessario effettuare il sign-in
2. Al meccanismo di specifica della risposta di un esercizio da parte di un utente loggato vengono applicati i seguenti vincoli:
   1. La verifica del punteggio della risposta è effettuata a back-end
   2. Non si può effettuare submitting di risposte a meno di 30 secondi una dall’altra
   3. Ogni volta che viene effettuato un submit nel database il punteggio viene sovrascritto
3. Il sistema deve essere accessibile ventiquattr’ore su ventiquattro a meno di malfunzionamenti o manutenzione
   1. Sono previsti 5 giorni all’anno per la manutenzione del back-end
4. Usabilità: l’utente è in grado di usare il sistema senza la necessità di essere formato sul suo funzionamento
   1. Il numero di elementi dell’interfaccia utente front-end di tipologia differente (due voci cliccabili di un elenco di corsi non sono considerati tali) presenti a schermo sono limitati a 10
5. Privacy:
   1. Non è possibile visualizzare dati di altri account
   2. Le statistiche visibili ai docenti sono anonime
   3. L’unica informazione sull’utente a cui il sistema accede in fase di sign-in o sign-up è il suo ruolo nell’ateneo
6. Compatibilità:
   1. il sistema supporta l’utilizzo da computer, smartphone e tablet, purché siano dotati di uno tra i seguenti browser: Chrome (versione 105 o successiva), Firefox (versione 104 o successiva) e Microsoft Edge (versione 105 o successiva)

# Front-end

1. Schermata home della piattaforma
2. Schermate dell’interfaccia studente
   1. Visualizzazione dell’elenco dei corsi
   2. Visualizzazione del materiale di un corso
   3. Statistiche sulle esercitazioni completate
3. Schermate dell’interfaccia docente
   1. Visualizzazione dell’elenco dei propri corsi
   2. Fill-in delle informazioni essenziali di un corso alla sua creazione (come specificato dai requisiti funzionali 7.1 e 7.2) 

# Back-end

Solve.it necessita delle seguenti API esterne per il suo funzionamento

* UNITN: le API di Unitn saranno utilizzate per la gestione dell'accesso al sito
* WOLFRAM: le API di Wolfram saranno utilizzate per interpretare risposte esplicitate in forma di espressioni matematiche
* MONGODB: useremo Mongodb per salvare tutti i dati relativi agli utenti, le sessioni attive, gli accessi, e i dati necessari all’applicazione per funzionare

