



UNIVERSITÀ
DI TRENTO

DIPARTIMENTO DI INGEGNERIA
E SCIENZA DELL'INFORMAZIONE

Sleep Code

PROGETTO PER IL CORSO DI INGEGNERIA DEL SOFTWARE
ANNO ACCADEMICO 2023-2024

Analisi dei Requisiti

Descrizione: documento di analisi dei requisiti funzionali, non funzionali, front-end e back-end.

Numero documento: D1

Versione documento: 2.4

Membri del gruppo:

Raffaele CASTAGNA

Alberto ROVESTI

Zeno SALETTI

Numero gruppo: G17

Ultima revisione: 22 gennaio 2024

Indice

1	Introduzione	2
1.1	Scopo del documento	2
1.2	Obiettivo del progetto	2
1.3	Pubblico di riferimento ed esigenze	2
2	Requisiti funzionali	3
2.1	Accesso e autenticazione	3
2.2	Consultazione dei problemi	4
2.3	Esercitazione	5
2.4	Gestione dei dati personali	7
2.5	Gestione del catalogo dei problemi	8
3	Requisiti non funzionali	9
3.1	Caratteristiche di sistema	9
3.2	Affidabilità	10
3.3	Privacy e sicurezza	10
4	Design front-end	12
5	Design back-end	15

Consigli utili per la consultazione: Se il lettore per file `.pdf` attualmente in uso lo consente, è possibile navigare con più semplicità e velocità all'interno di questo documento cliccando sugli elementi dell'indice. I riferimenti interni, sottolineati e colorati in azzurro, agiscono nello stesso modo.

1 Introduzione

1.1 Scopo del documento

Le informazioni contenute in questo documento concorrono ad esporre l'analisi dei requisiti relativa al progetto *Sleep Code*. In particolare, dopo aver specificato gli obiettivi e il pubblico di riferimento, verranno definiti i requisiti funzionali e non funzionali; verrà presentata una proposta di design di front-end; infine saranno riportati i servizi esterni di back-end coi quali il sistema dovrà interagire.

1.2 Obiettivo del progetto

Il progetto proposto si prefigge, come scopo fondante, di fornire alla comunità di giovani informatici un servizio online di:

- *esercitazione* mirata alla programmazione e alla progettazione di piccoli algoritmi risolutivi, mediante la scrittura di codice;
- *raccolta* di problemi attinenti.

1.3 Pubblico di riferimento ed esigenze

Il servizio vuole rendersi utile soprattutto a coloro che sono coinvolti in percorsi di studio attinenti all'ambito informatico, ma specialmente anche a chiunque desideri cimentarsi nella risoluzione di piccoli problemi di programmazione; pertanto ci si aspetta che chiunque desideri usufruire del servizio possieda almeno le conoscenze basilari della programmazione. Esempi di queste nozioni pregresse, che tuttavia non devono necessariamente essere ampie e approfondite per utilizzare il servizio¹, sono: cosa si intende per algoritmo e linguaggio di programmazione, familiarità nell'uso di qualche linguaggio di programmazione, tipi e strutture di dati più comuni.

Di fatto, il progetto che verrà sviluppato ha come scopo principale di creare una piattaforma accessibile online a singoli utilizzatori che desiderano esercitarsi, valutare e approfondire le personali conoscenze e abilità di *problem solving* legate alla programmazione, insieme all'archiviazione di nuovi problemi da rendere disponibili a coloro che intendono usufruire del servizio. D'ora in avanti, in questo e nei successivi documenti, questo pubblico di individui appena descritti verrà indicato con il termine *utenti*.

¹Gli utenti più esperti possono indubbiamente trarre vantaggio dal loro bagaglio culturale per approcciarsi con maggior facilità al servizio.

2 Requisiti funzionali

Vengono di seguito elencati i requisiti funzionali (RF) del progetto. Ogni sottosezione di questa parte del documento risponde a diversi scopi precedentemente accennati, suddividendo eventuali macro-funzionalità in requisiti minori nel caso di obiettivi di più ampia portata.

Le funzionalità sono inoltre suddivise secondo alcuni livelli di accesso degli utenti (di seguito in ordine "crescente"):

- *Utente anonimo* (Sezioni 2.1, 2.2, 2.3).
- *Utente autenticato* (Sezione 2.4).
- *Utente amministratore* (Sezione 2.5).

Si precisa che un utente presente ad un dato livello di accesso può usufruire delle funzionalità disponibili a quel livello *più* le funzionalità presenti al livello inferiore. Non è invece possibile utilizzare le funzionalità messe a disposizione a eventuali livelli superiori.

Utente anonimo

2.1 Accesso e autenticazione

RF 1. Registrazione² I nuovi utenti devono poter registrarsi al servizio, ovvero creare un *account* che richiede la compilazione dei seguenti campi:

- Nome utente (*username*).
- Indirizzo di posta elettronica (*email*).
- *Password*: data la sensibilità di questo campo, è importante che siano rispettati i vincoli riportati in [RNF 11](#); inoltre, l'utente deve poter riscrivere la password per confermarla, in modo tale da rilevare eventuali errori di digitazione.

RF 2. Login Il sistema deve permettere all'utente registrato di autenticarsi, accedendo al proprio account, mediante l'inserimento dell'indirizzo email e la password impostate in fase di registrazione. L'utente deve essere notificato qualora non fosse possibile concludere correttamente l'operazione di login, quindi se l'email inserita non è associata ad alcun account registrato in precedenza oppure se la password è errata; è comunque possibile ritentare il login un numero illimitato di volte. Effettuato il login, l'utente accede alle funzionalità del livello *autenticato*.

²Si denoterà con l'aggettivo *registrato* un qualsiasi utente che abbia precedentemente effettuato la registrazione e pertanto dotato di un account.

RF 3. Autenticazione Google L'utente deve poter registrarsi ed effettuare il login sulla piattaforma utilizzando un account Google, qualora non si intenda ricorrere al sistema di autenticazione interno descritto precedentemente ([RF 1](#), [RF 2](#)). In caso di registrazione, oltre alle operazioni previste dal servizio Google (come la scelta dell'indirizzo email), l'utente deve solo specificare un username.

RF 4. Recupero password L'utente registrato e anonimo deve poter recuperare la password del proprio account qualora tale dato dovesse essere dimenticato in fase di login. La procedura di recupero deve prevedere:

1. L'invio di un messaggio all'indirizzo email attualmente associato all'account dell'utente—l'email contiene un collegamento ipertestuale che reindirizza l'utente ad una pagina dedicata, sulla piattaforma, nella quale completare l'operazione di recupero.
2. L'inserimento di una nuova password nella pagina dedicata. La password, come in fase di registrazione, deve essere confermata e conforme al [RNF 11](#).

Il recupero deve essere accessibile a tutti gli utenti registrati mediante sistema di credenziali interno ([RF 1](#), [RF 2](#)), e non riguarda gli utenti registrati mediante autenticazione Google ([RF 3](#)).

2.2 Consultazione dei problemi

RF 5. Consultazione del catalogo dei problemi Il servizio deve mettere a disposizione un insieme di problemi sui quali l'utente possa esercitarsi. L'utente deve poter consultare un catalogo, atto a raccogliere i quesiti, e navigare al suo interno. Deve quindi essere possibile:

1. Visualizzare tale catalogo, che mostra i campi *descrittivi* di ogni problema elencato:
 - *Nome*: un breve nome del problema.
 - *Difficoltà*: i problemi sono categorizzati in base alla loro difficoltà. Vengono definiti tre livelli indicativi: bassa, intermedia, alta.
 - *Tags*: un insieme di parole chiave che individuano una o più aree di interesse affrontate nel problema (ad esempio ricerca, grafi, ordinamento, programmazione dinamica, ecc.)
 - *Suggerimento*: viene messo a disposizione, per ogni problema nel catalogo, un video in cui vengono trattate nozioni che possono aiutare ad affrontare il problema, suggerendo quindi all'utente un possibile approccio all'elaborazione della soluzione specifica del problema. Il suggerimento è disponibile solo dal catalogo e non dall'area di esercitazione ([RF 7](#)).

2. Selezionare dal catalogo un problema specifico, cosicché il contenuto dell'intero problema possa essere visionato per mezzo delle funzionalità di cui al [RF 6](#).

RF 6. Consultazione di un problema Deve essere fornito un visualizzatore che mostra le informazioni del problema precedentemente selezionato dal catalogo. La visualizzazione deve rendere disponibile alla vista dell'utente i contenuti del problema (*dati strutturali*):

- Un titolo.
- Un testo che descrive uno scenario che richiede di essere risolto per mezzo di un algoritmo. Eventuali immagini e proposizioni matematiche possono accompagnare i testi.
- Almeno due esempi³ di esecuzione, ognuno costituito da un input e il relativo output atteso.

Qualora l'utente desideri consultare il problema, vengono contemporaneamente messe a disposizione le funzionalità di cui al [RF 7](#), che nel complesso costituiscono l'*area di esercitazione*.

2.3 Esercitazione

RF 7. Effettuare un'esercitazione L'utente deve poter scegliere dal catalogo il problema desiderato e cominciare a risolverlo. L'esercitazione deve avvenire in una vista apposita (l'area di esercitazione), dove l'utente deve poter disporre di tutti gli strumenti necessari alla risoluzione del problema, accedendo alle seguenti funzionalità:

1. Visualizzare il contenuto dell'esercizio ([RF 6](#)).
2. Essere al corrente di quale linguaggio di programmazione sia attualmente attivo per la scrittura di codice. Deve altresì essere possibile selezionare uno dei linguaggi messi a disposizione dalla piattaforma⁴.
3. Scrivere, sotto forma di codice nel linguaggio di programmazione scelto, l'algoritmo risolutivo del problema selezionato. Accedendo all'area di esercitazione, viene fornito un frammento di *driver code* (un esempio è visibile in [FE 2](#)). Il codice scritto dall'utente anonimo non viene conservato dopo aver abbandonato l'area di esercitazione e viene sovrascritto dal driver code.
4. Richiedere la sottoposizione del codice scritto e visualizzare i test cases insieme al loro esito ([RF 8](#)).

³Non si devono confondere questi *esempi* con i *test cases* ([RF 8](#))

⁴Per approfondimenti sulla disponibilità dei linguaggi, si veda [RNF 1.2](#).

5. Visualizzare ed utilizzare un cronometro ([RF 9](#)).
6. Modificare le dimensioni delle aree dedicate alla visualizzazione del contenuto del problema, alla scrittura di codice e consultazione dei test. Ciò avviene grazie ad un *layout dinamico*, con il quale l'utente può adattare la visibilità di alcuni elementi dell'area di esercitazione a propria discrezione⁵.

RF 8. Verifica della correttezza dell'algoritmo L'utente deve poter verificare la correttezza del codice scritto eseguendolo e testandolo:

1. Il codice deve essere eseguito sottoponendolo ad un certo numero di test cases, cioè fornendo opportune istanze di input, previste dallo specifico problema selezionato, e controllando l'output atteso con quello restituito dal programma dell'utente. Il numero minimo di test cases per ogni problema corrisponde a 3.
2. L'utente deve poter conoscere l'esito dei test cases—per ognuno di questi, l'input e l'output atteso sono visibili. Inoltre, l'utente deve poter riscrivere e perfezionare l'algoritmo e sottoporre ripetutamente il codice ai test cases.

Nel caso in cui l'esito dei test cases dovesse risultare negativo, l'utente viene avvisato senza fare riferimento agli errori commessi, siano essi relativi alla correttezza sintattica del codice oppure alla correttezza risolutiva dell'algoritmo—casi nei quali l'output fornito non corrisponde a quello atteso.

RF 9. Cronometro L'utente deve poter tenere traccia del tempo impiegato per risolvere un problema. Pertanto, deve essere possibile avviare, interrompere e reimpostare a 0 secondi un cronometro. Il cronometro è disponibile nell'area di esercitazione, dopo aver scelto il problema da risolvere, insieme a tutti gli altri strumenti descritti in [RF 7](#).

⁵Ciò può avvenire entro alcuni limiti, in modo da non ledere la visibilità degli elementi grafici: per esempio, non è consentito modificare il layout al punto da nascondere completamente il testo del problema.

Utente autenticato

Nei seguenti requisiti funzionali, si suppone che l'utente appartenga al livello di accesso autenticato *autenticato*, qualora tale informazione dovesse essere omessa. Si sottolinea nuovamente che l'utente autenticato eredita le stesse funzionalità accessibili all'utente anonimo e descritte precedentemente. Il livello autenticato aggiunge un certo grado di personalizzazione all'esperienza fornita dal servizio, consentendo la gestione di dati personali come l'*account* ([RF 1](#)) e i *progressi* ([RF 12](#)).

2.4 Gestione dei dati personali

RF 10. Metadati aggiuntivi Oltre ai campi descrittivi specificati nel [RF 5.1](#), l'utente autenticato deve poter visualizzare e influenzare i seguenti dati aggiuntivi in ogni problema:

1. *Stato*: lo stato permette di indicare all'utente se un certo problema è stato risolto o meno. I problemi risolti con successo dall'utente devono essere contrassegnati in questo apposito campo.
2. *Preferito*: l'utente che desidera tener traccia dei problemi per lui rilevanti deve poter contrassegnare tali problemi come *preferiti*. Nel catalogo, l'utente autenticato può distinguere i problemi contrassegnati personalmente, ma entrando nell'area di esercitazione è possibile visualizzare anche il numero di utenti che hanno contrassegnato quello stesso problema come preferito.

Le informazioni riguardanti questi campi sono visibili all'utente autenticato, sia durante la consultazione del catalogo che del singolo problema, nell'area di esercitazione.

RF 11. Codice Il codice scritto dall'utente autenticato, al contrario di quanto accade per gli utenti anonimi in [RF 7.3](#), non viene perso al momento dell'uscita dall'area del problema, cosicché quanto scritto possa essere ripreso in momenti successivi.

RF 12. Progressi risolvendo i problemi, l'utente autenticato può migliorare i propri *progressi*.

1. Il servizio deve prevedere il tracciamento dei progressi di ogni utente autenticato, incrementando il numero di problemi risolti ad ogni sessione di esercitazione andata a buon fine: con riferimento al [RF 8](#), dopo la risoluzione, il problema deve essere contrassegnato come *risolto* (ciò modifica il campo *stato* visibile nel catalogo e nell'area di consultazione ed esercitazione di quel problema).
2. L'utente deve poter monitorare i propri progressi, visualizzando il numero totale di problemi risolti nel catalogo.

RF 13. Aggiornamento password L'utente registrato e autenticato mediante il sistema di credenziali interno deve poter modificare la password del proprio account. Essa può essere cambiata previo inserimento di quella attualmente associata all'account. La nuova password deve essere digitata due volte per conferma e deve rispettare quanto specificato dal [RNF 11](#).

RF 14. Logout L'utente autenticato deve poter interrompere la sessione di accesso al servizio. Questa procedura di *logout* realizza il passaggio dell'utente dallo stato autenticato a quello anonimo.

RF 15. Eliminazione account L'utente registrato e autenticato deve poter eliminare il proprio account, insieme ai dati ad esso collegati (preferiti, progressi ecc.).

Utente amministratore

2.5 Gestione del catalogo dei problemi

RF 16. Aggiungere un problema L'utente amministratore deve poter aggiungere un nuovo problema al catalogo. La struttura del problema deve essere conforme a quanto descritto nel [RF 6](#) e, all'atto dell'inserimento nel catalogo, devono essere specificati i campi elencati nel [RF 5.1](#).

Inoltre, devono essere forniti almeno 3 test cases (questi saranno impiegati come descritto dal [RF 8.1](#)); ogni test case consiste in un dato in input e il rispettivo output corretto.

RF 17. Modificare un problema L'utente amministratore deve poter modificare sia i campi descrittivi del problema, sia i dati strutturali (il contenuto), sia i relativi test cases.

RF 18. Eliminare un problema L'utente amministratore deve poter eliminare i problemi dal catalogo.

3 Requisiti non funzionali

Vengono ora elencati i requisiti non funzionali (RNF) del servizio.

3.1 Caratteristiche di sistema

RNF 1. Scalabilità L'infrastruttura del servizio deve essere scalabile e aperta alle esigenze derivanti dall'aumento di nuovi utenti. Questo requisito è motivato dalla disponibilità online del servizio che verrà sviluppato. In particolare:

1. L'infrastruttura del servizio deve essere adattabile a eventuali crescita nel numero di utenti, in modo da prevenire possibili cali di prestazioni eccessivi.
2. Data l'eterogeneità di linguaggi di programmazione esistenti al momento della stesura di questo documento, è importante che il servizio sia in grado di accogliere con l'avanzare del tempo codici scritti in linguaggi differenti.

RNF 2. Compatibilità La piattaforma del servizio deve essere accessibile mediante le versioni più recenti dei principali browser attualmente disponibili in commercio:

- Chrome: 117.0.5938.150
- Firefox: 118.0.1
- Edge: 117.0.2045.60

RNF 3. Usabilità La piattaforma del servizio deve permettere all'utente di sfruttare le funzionalità disponibili al proprio livello di accesso senza l'ausilio di istruzioni scritte e verbose. L'intuitività dell'interfaccia deve essere sufficiente a guidare l'utente nella realizzazione dei suoi scopi, permettendo di venire a conoscenza di almeno il 90%⁶ delle funzionalità del servizio in meno di 30 minuti.

RNF 4. Aspetto L'interfaccia deve presentarsi gradevole alla vista dell'utente. Devono essere curate le seguenti caratteristiche visive:

1. Colore: si preferiscono gradazioni cromatiche scure e un contrasto sufficientemente equilibrato, al fine di garantire la leggibilità e contribuire alla riduzione dell'affaticamento della vista. Una motivazione a supporto di questo requisito è il potenziale rischio di sessioni eccessivamente prolungate sulla piattaforma.

⁶La percentuale esprime il rapporto tra il numero di funzionalità alle quali l'utente di un dato livello di accesso viene a conoscenza entro l'intervallo di tempo specificato e il numero totale di funzionalità, utilizzando come riferimento il numero di requisiti funzionali elencati precedentemente.

2. Durante eventuali attese di risposta della piattaforma che in qualche modo coinvolgono transazioni sul lato back-end (ad esempio richieste effettuate tra client e server), l'utente deve poter essere segnalato riguardo lo stato di questi momenti di attesa, per mezzo di opportune segnalazioni grafiche come *skeleton screens*⁷ o *popup*. Alcuni esempi di queste situazioni di attesa sono l'apertura di un problema o la conferma di registrazione dell'account.

RNF 5. Lingua di sistema Il servizio viene erogato in lingua italiana. I video di suggerimento ([RF 5](#)) potrebbero essere distribuiti in lingua inglese.

RNF 6. Prestazioni L'esperienza di utilizzo deve poter essere soddisfacente in relazione ai livelli di prestazioni dei moderni siti web, purché l'utente utilizzi il servizio in condizioni di connettività sufficienti. In particolare, i tempi di reazione relativi a caricamenti e transizioni tra pagine non deve eccedere, indicativamente, i 2 secondi.

3.2 Affidabilità

RNF 7. Downtime Il downtime annuo della piattaforma non deve eccedere gli intervalli di tempo indispensabili a eventuali opere di manutenzione e aggiornamento *sui componenti interni al servizio*, evitando ove possibile interruzioni di servizio inutilmente prolungate. Si prevede di non superare un downtime del 2,7% nel primo anno (10 giorni su 365, per un totale di 240 ore) dopo il lancio della piattaforma, per poi mantenere il rapporto al di sotto del 0,85% (circa 72 ore).

RNF 8. Disponibilità Il progetto sviluppato ha il 98% di probabilità di non guastarsi entro le prime 8.000 ore di funzionamento. La disponibilità dei servizi terzi, sui quali il progetto si affida (database, contenuti multimediali), permette di trascurare eventuali variazioni di tempo aggiuntive riguardanti le interruzioni di servizio inattese.

3.3 Privacy e sicurezza

RNF 9. Privacy e trattamento dei dati Il servizio deve essere progettato e realizzato in ottemperanza delle vigenti disposizioni di legge in materia di tutela della privacy e trattamento dei dati:

1. L'applicazione fornita dal servizio deve essere conforme al regolamento [UE n.2016/679](#) (GDPR) per la protezione dei dati.
2. Il codice scritto dagli utenti autenticati ([RF 11](#)) viene memorizzato da parte del client e non al di fuori di esso.

⁷Una breve descrizione degli *skeleton screens* è reperibile [qui](#).

RNF 10. Connessione sicura La comunicazione con il client deve essere protetta da protocolli di sicurezza, come **https**, che consentano di preservare la riservatezza dei dati scambiati tra piattaforma e client utente.

RNF 11. Password strength In tutti gli scenari nei quali è richiesta la creazione di una password, la stringa inserita deve rispettare le seguenti caratteristiche:

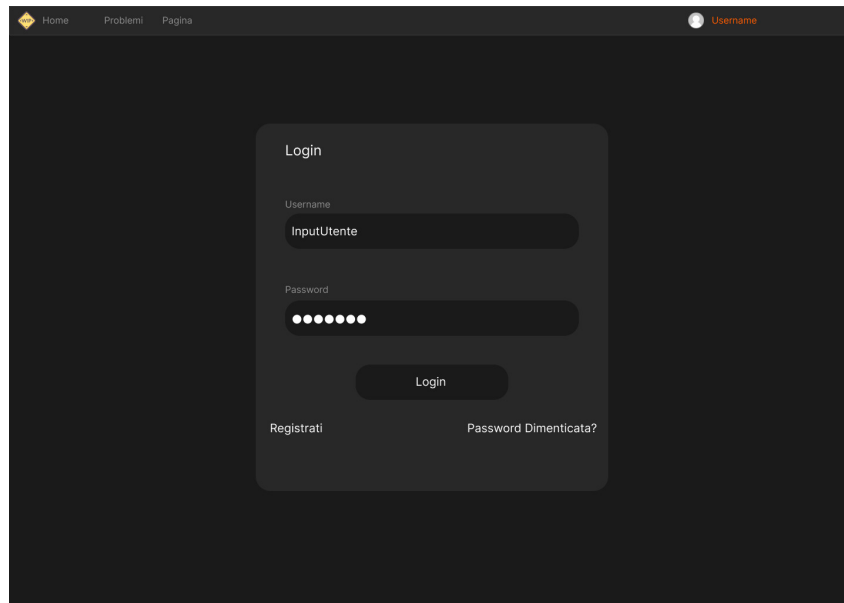
- Lunghezza compresa tra 8 e 64 caratteri.
- Contenere almeno una lettera maiuscola.
- Contenere almeno una lettera minuscola.
- Contenere almeno un numero.
- Contenere almeno un carattere speciale non alfanumerico scelto tra i seguenti:

! ? # \$ % & @ * + . , ; : / \ - = _ () [] { }

4 Design front-end

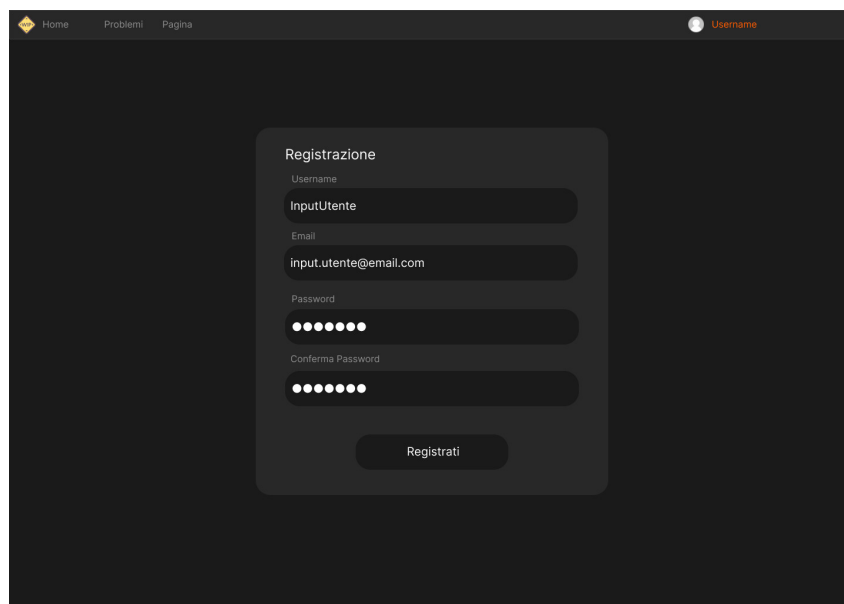
In questa sezione vengono presentati alcuni mock-up dell'interfaccia che la piattaforma online espone all'utente.

FE 1. Pagina di login



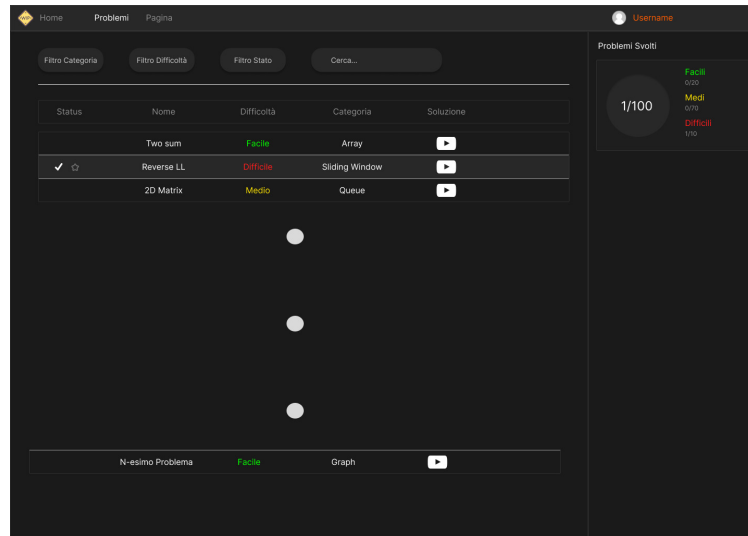
The login page features a dark background with a central light gray login form. The form includes a 'Login' title, a 'Username' label with an input field containing 'InputUtente', a 'Password' label with a masked input field (dots), a 'Login' button, and links for 'Registrati' and 'Password Dimenticata?'. The top navigation bar contains 'Home', 'Problemi', 'Pagina', and a user profile icon labeled 'Username'.

FE 2. Pagina di registrazione

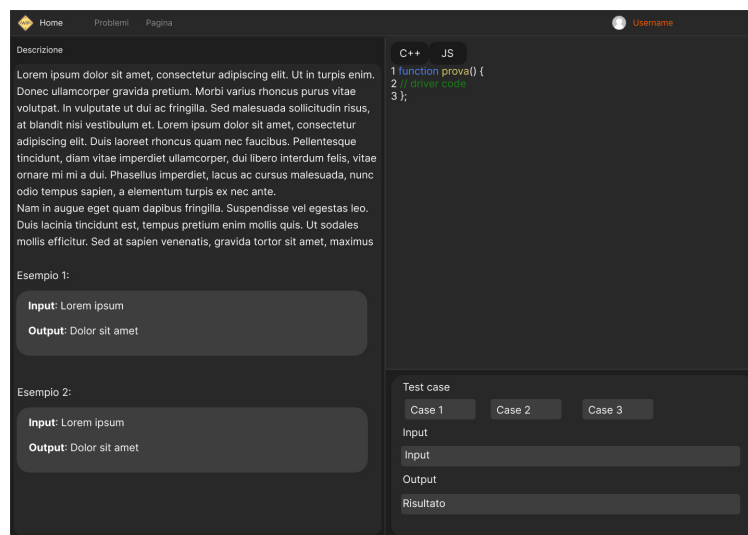


The registration page features a dark background with a central light gray registration form. The form includes a 'Registrazione' title, a 'Username' label with an input field containing 'InputUtente', an 'Email' label with an input field containing 'input.utente@email.com', a 'Password' label with a masked input field (dots), a 'Conferma Password' label with a masked input field (dots), and a 'Registrati' button. The top navigation bar contains 'Home', 'Problemi', 'Pagina', and a user profile icon labeled 'Username'.

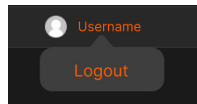
FE 3. Home page Il catalogo, i progressi dell'utente autenticato, insieme a tutti i collegamenti che consentono di usufruire della maggior parte delle funzionalità del servizio, sono contenute in una pagina principale. Il seguente mock-up mostra la vista che si presenta ad un utente autenticato (è presente il campo *stato*, oltre all'icona per contrassegnare i *preferiti*).



FE 4. Pagina di esercitazione Una pagina dedicata raccoglie le funzionalità di consultazione dettagliata del problema e risoluzione di quest'ultimo.



FE 5. Logout Un pulsante di logout è raggiungibile in ogni momento da parte dell'utente autenticato, all'interno di una barra di navigazione che mostra anche lo *username* scelto in fase di registrazione.



5 Design back-end

Nella sezione seguente vengono elencati i servizi esterni con i quali la piattaforma *Sleep Code* interagisce per funzionare correttamente, o per ricevere supporto mirato a soddisfare i propri requisiti. In Figura 1 viene riassunto l'insieme di questi servizi esterni e il ruolo che essi svolgono nell'interazione, in ambito back-end (BE), con il progetto da realizzare.

BE 1. Servizio di autenticazione alternativo L'utente può scegliere di registrarsi e accedere all'account sulla piattaforma mediante un account Google. Questo servizio offre all'utente un meccanismo di autenticazione alternativo al sistema di credenziali interno, realizzando quanto descritto in [RF 3](#).

BE 2. Contenuti multimediali Per reperire e riprodurre i video contenenti i suggerimenti dei problemi ([RF 5.1](#)), il servizio ricorre alla piattaforma YouTube.

BE 3. Database La memorizzazione degli account degli utenti, dei problemi disponibili nel catalogo e dei dati annessi si affida al servizio di database offerto da Firebase.

BE 4. Notifica via posta elettronica L'invio dei messaggi email, ai fini del recupero dell'account ([RF 4](#)), viene gestito da servizi di posta elettronica di terze parti.



Figura 1: Diagramma di back-end con i servizi esterni su cui il progetto si appoggia