



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:* 1.0

*Membri del gruppo:*

Raffaele CASTAGNA

Alberto ROVESTI

Zeno SALETTI

*Numero gruppo:* G17

*Ultima revisione:* 5 ottobre 2023

## Indice

<b>1</b>	<b>Introduzione e scopi</b>	<b>2</b>
1.1	Scopo del documento . . . . .	2
1.2	Obiettivo del progetto . . . . .	2
1.3	Pubblico di riferimento ed esigenze . . . . .	2
<b>2</b>	<b>Requisiti funzionali</b>	<b>3</b>
2.1	Accesso . . . . .	3
2.2	Consultazione dei problemi . . . . .	4
2.3	Esercitazione . . . . .	4
2.4	Gestione profilo e account . . . . .	5
<b>3</b>	<b>Requisiti non funzionali</b>	<b>7</b>
3.1	Caratteristiche delle risorse . . . . .	7
3.2	Caratteristiche di sistema . . . . .	8
3.3	Privacy e sicurezza . . . . .	9
<b>4</b>	<b>Design front-end</b>	<b>10</b>
<b>5</b>	<b>Design back-end</b>	<b>13</b>

# 1 Introduzione e scopi

## 1.1 Scopo del documento

Le informazioni contenute in questo documento concorrono ad esporre l'analisi dei requisiti relativa al progetto *SleepCode*. In particolare, dopo aver specificato gli obiettivi e gli attori coinvolti, verranno definiti i requisiti funzionali e non funzionali; verrà presentata una proposta di design di back-end; infine saranno riportati i sistemi esterni di back-end coi quali il servizio dovrà interfacciarsi.

## 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.

## 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 servizio<sup>1</sup>, 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. D'ora in avanti, in questo e nei successivi documenti, questo pubblico di individui appena descritti verrà indicato con il termine *utenti*.

---

<sup>1</sup>Gli 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 principali requisiti funzionali (RF) del progetto. Ogni sottosezione di questa parte del documento risponde a diversi scopi precedentemente accennati, suddividendo eventuali macro-funzioni in requisiti minori nel caso di obiettivi di più ampia portata.

### 2.1 Accesso

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

- Indirizzo email.
- Nome utente.
- Password: data la sensibilità di questo campo, è importante che siano rispettati i vincoli del [RNF 12](#); inoltre, l'utente deve poter riscrivere la password per confermarla, in modo tale da rilevare eventuali errori di digitazione.

L'utente dotato di un account sarà considerato dal servizio come *utente registrato*.

**RF 2. Login** Il sistema deve permettere all'utente già registrato al servizio di accedere alla piattaforma mediante l'inserimento dell'indirizzo email e la password, impostate in fase di registrazione. L'utente deve essere notificato qualora non sia possibile concludere correttamente l'operazione di login, quindi se l'email inserita non sia registrata o la password non corrisponda; è comunque possibile ritentare il login un numero illimitato di volte.

Effettuato il login, l'utente registrato rientra nella categoria degli *utenti autenticati*.

**RF 3. Recupero password** L'utente registrato deve poter recuperare la password del proprio account qualora tale dato dovesse essere dimenticato in fase di login. La procedura di recupero deve prevedere:

- L'invio di un messaggio all'indirizzo email attualmente associato all'account dell'utente; l'email contiene un link che guida l'utente nella creazione di una nuova password.
- L'inserimento (sulla piattaforma) di una nuova password che, come in fase di registrazione, deve essere scritta, confermata ed essere conforme al [RNF 12](#).

**RF 4. 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.

## 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 di ogni problema elencato (descritti nel [RNF 2](#)).
2. Cercare uno o più problemi specifici mediante ricerca filtrata per campi.
3. Selezionare dal catalogo un problema specifico, cosicché la descrizione dell'intero problema possa essere visionata per mezzo delle funzionalità di cui al [RF 6](#).

**RF 6. Consultazione di un problema** Deve essere fornito un visualizzatore che mostri le informazioni del problema precedentemente selezionato dal catalogo. La visualizzazione deve rendere disponibile alla vista dell'utente i contenuti del problema, specificati nel [RNF 1](#).

## 2.3 Esercitazione

**RF 7. Avviare una sessione di esercitazione** L'utente deve poter scegliere dal catalogo il problema desiderato e cominciare a risolverlo. La sessione di esercitazione deve avvenire in una vista apposita, dove l'utente deve poter disporre di tutti gli strumenti necessari alla realizzazione delle seguenti funzionalità:

1. Visualizzare il contenuto dell'esercizio ([RNF 1](#)).
2. Visualizzare i test cases ([RNF 9.2](#)).
3. Avviare e fermare a proprio piacimento un timer per cronometrare il tempo speso per risolvere il quesito scelto.
4. 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 piattaforma<sup>2</sup>.
5. Scrivere, sotto forma di codice nel linguaggio di programmazione scelto, l'algoritmo risolutivo del problema attualmente attivo.

**RF 8. Correttezza sintattica del codice** L'utente deve poter verificare che il codice scritto sia corretto e in grado di essere eseguito in maniera appropriata. Quindi devono essere messe a disposizione le seguenti funzionalità:

---

<sup>2</sup>Per approfondimenti sulla disponibilità dei linguaggi, si veda [RNF 3.2](#).

1. Compilazione del codice.
2. Visualizzazione di avvisi relativi a eventuali errori di compilazione *op-pure* di compilazione andata a buon fine. In caso di errori di scrittura, l'utente deve poter correggere tali errori riscrivendo nell'area destinata al codice.

**RF 9. Verifica della correttezza dell'algoritmo**<sup>3</sup> 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 (al minimo 3), cioè fornendo opportune istanze di input (Per esempio, se un problema richiede di sommare due numeri interi, il codice risolutivo proposto dall'utente verrà eseguito fornendo ad esso coppie di interi e registrando i risultati in output.)
2. L'utente deve poter conoscere l'esito dei test case e, 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.
3. Nel caso in cui l'utente autenticato sia in grado di superare tutti i test case disponibili, la sessione di esercitazione deve terminare: deve essere visualizzato il tempo impiegato (se il cronometro è attivo) e, qualora l'utente sia autenticato, il problema deve essere contrassegnato come *risolto* nel campo *stato* del problema e i progressi ([RF 11](#)) devono essere aggiornati.

**RF 10. Soluzione** Per ogni problema viene messo a disposizione un video che mostra una possibile soluzione. Il video è accessibile sia durante la consultazione del catalogo sia in una sessione di esercitazione.

## 2.4 Gestione profilo e account

**RF 11. Progressi** L'utente autenticato deve poter monitorare i propri progressi, accedendo ai dati seguenti:

- Il numero totale di problemi risolti.
- Il numero di problemi risolti suddivisi per difficoltà.

**RF 12. Preferiti** L'utente deve poter contrassegnare un qualsiasi problema come *preferito*. Tale informazione deve essere visibile nel catalogo, accanto ad ogni problema contrassegnato in questo modo.

---

<sup>3</sup>La *correttezza* di cui si parla in questo caso riguarda solo l'efficacia risolutiva dell'algoritmo.

**RF 13. Aggiornamento account** L'utente registrato e autenticato deve poter modificare i dati chiave del proprio account in conformità con le opportune procedure di sicurezza:

1. L'utente deve poter migrare ad un indirizzo email differente. A tale indirizzo deve essere inviato un messaggio di conferma, previo inserimento della password. L'email deve cambiare effettivamente solo nel momento in cui l'utente conferma la ricezione del messaggio presso il nuovo indirizzo.
2. L'utente deve poter modificare la password del proprio account. La password 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 12](#). L'utente deve essere notificato di tale modifica anche tramite l'indirizzo email attualmente associato all'account.

## 3 Requisiti non funzionali

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

### 3.1 Caratteristiche delle risorse

Alcuni requisiti funzionali sono dedicati alla descrizione delle principali risorse fornite dalla piattaforma, di cui le principali sono i problemi.

**RNF 1. Struttura di un problema** Tutti i problemi forniti possiedono:

- Un titolo.
- Un testo, scritto prevalentemente in linguaggio naturale, che descrive uno scenario che richiede di essere risolto per mezzo di un algoritmo. Eventuali immagini e proposizioni matematiche formali possono accompagnare i testi.
- Almeno 3 esempi di input insieme al relativo output che mostra il risultato atteso.

**RNF 2. Metadati dei problemi** Ogni problema è essenzialmente un'entità caratterizzata dai seguenti dati descrittivi:

- Stato: i problemi già risolti dall'utente sono contrassegnati in un campo *stato* apposito.
- Nome identificativo: si tratta di una stringa alfanumerica, che riprende una o più parole chiave del titolo del problema.
- Difficoltà: ogni problema possiede un'etichetta (*tag*) associata che ne valuta indicativamente la difficoltà: bassa, intermedia e alta.
- Categoria: la categoria indica la principale area di interesse del problema, come ad esempio le strutture di dati di cui tratta (array, grafi, code, ecc.).
- Soluzione: un video che mostra come il problema potrebbe essere risolto.
- Preferito: la marcatura *preferito* è visibile sul problema quando esso compare nel catalogo e quando viene consultato.

Ai fini della consultazione del catalogo, i filtri applicabili riguardano i campi *stato*, *difficoltà*, *categoria*, *preferito* e *nome*.



### 3.2 Caratteristiche di sistema

**RNF 3. 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 crescite 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 4. Compatibilità** La piattaforma del servizio deve essere accessibile mediante le versioni più recenti dei principali browser attualmente disponibili in commercio:

- Chrome:
- Firefox:
- Edge:
- 

**RNF 5. 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 imparare almeno il 90% delle funzionalità del servizio in meno di 20 minuti.

**RNF 6. Aspetto** L'interfaccia deve presentarsi gradevole alla vista dell'utente, preferendo gradazioni cromatiche scure e un contrasto sufficientemente equilibrato, al fine di garantire la leggibilità e contribuire alla riduzione dell'affaticamento della vista.

**RNF 7. Lingua di sistema** Il servizio viene erogato in lingua italiana. Altrettanto viene fatto per i testi dei problemi. I video relativi alla soluzione dei problemi possono essere in italiano oppure in inglese.

**RNF 8. Disponibilità** Il downtime annuo della piattaforma non deve eccedere gli intervalli di tempo indispensabili a eventuali opere di manutenzione, evitando ove possibile interruzioni di servizio inutilmente prolungate.

**RNF 9. 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 i 2 secondi.

### 3.3 Privacy e sicurezza

**RNF 10. 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.

**RNF 11. 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 12. Password strength** In tutti gli scenari nei quali è richiesto l'inserimento di una password, devono essere rispettate le seguenti caratteristiche:

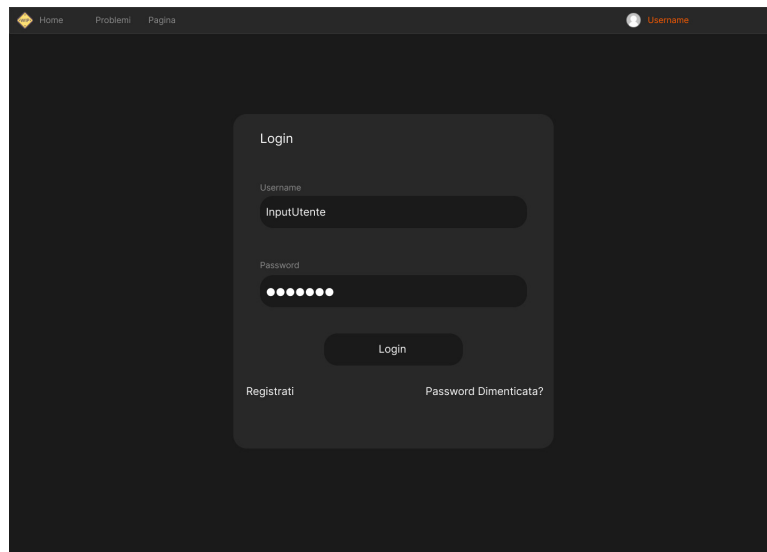
- Lunghezza compresa tra 8 e 64 caratteri.
- Contenere almeno una lettera maiuscola.
- Contenere almeno una lettera minuscola.
- Contenere almeno un numero.
- Contenere almeno un carattere speciale scelto tra i seguenti:

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

## 4 Design front-end

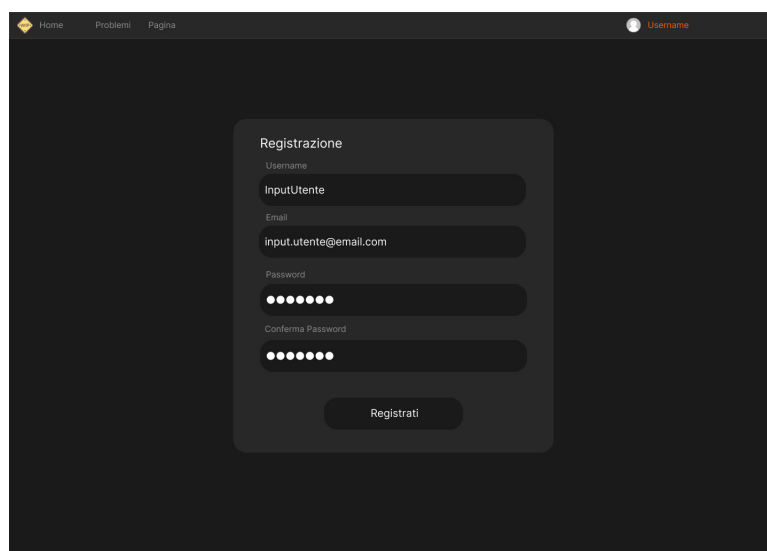
In questa sezione vengono presentati alcuni mock-up dell'interfaccia che la piattaforma online espone all'utente.

### FE 1. Pagina di login



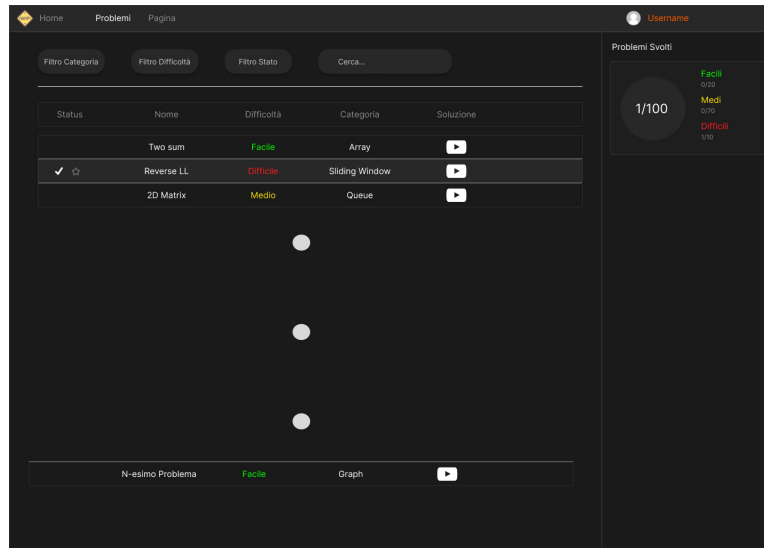
The mock-up shows a dark-themed login page. At the top, there is a navigation bar with a yellow diamond icon, the text 'Home', 'Problemi', and 'Pagina', and a user profile icon with the text 'Username'. The main content area features a central 'Login' form. The form has a title 'Login', followed by a 'Username' label and an 'InputUtente' text input field. Below this is a 'Password' label and a password input field with eight dots. A 'Login' button is positioned below the password field. At the bottom of the form, there are two links: 'Registrati' on the left and 'Password Dimenticata?' on the right.

### FE 2. Pagina di registrazione

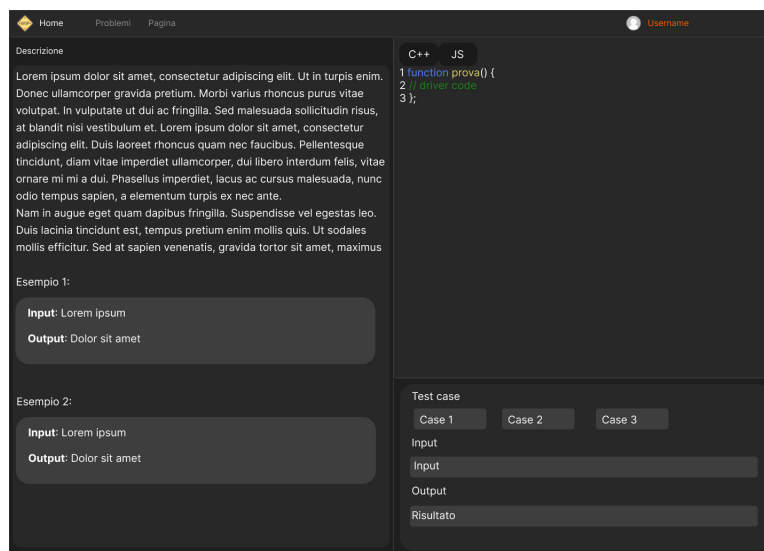


The mock-up shows a dark-themed registration page. At the top, there is a navigation bar with a yellow diamond icon, the text 'Home', 'Problemi', and 'Pagina', and a user profile icon with the text 'Username'. The main content area features a central 'Registrazione' form. The form has a title 'Registrazione', followed by a 'Username' label and an 'InputUtente' text input field. Below this is an 'Email' label and a text input field containing 'input.utente@email.com'. Then, there is a 'Password' label and a password input field with eight dots, followed by a 'Conferma Password' label and another password input field with eight dots. A 'Registrati' button is positioned below the confirmation password field.

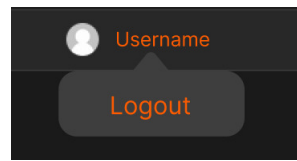
**FE 3. Home page** Il catalogo e il profilo utente, insieme a tutti i collegamenti che consentono di usufruire della maggior parte delle funzionalità del servizio, sono contenute in una pagina principale.



#### FE 4. Pagina di esercitazione



**FE 5. Logout** Un pulsante di logout è raggiungibile in ogni momento da parte dell'utente autenticato.



## 5 Design back-end

Nella sezione seguente vengono elencati i servizi con i quali la piattaforma interagisce per ottenere supporto alle sue funzionalità.

**BE 1. Servizio di autenticazione Google** L'utente potrà scegliere di autenticarsi mediante un account Google.

**BE 2. Contenuti multimediali** Per reperire e riprodurre i video contenenti le soluzioni dei problemi, il servizio si affiderà alla piattaforma YouTube.

**BE 3. Database** Verrà impiegato il servizio di database Firebase per la memorizzazione degli utenti, dei problemi del catalogo e di tutti i dati associati.

**BE 4. Notifica via posta elettronica** L'invio dei messaggi email (da parte della piattaforma) sarà gestito da servizi di terze parti.

