



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

*Membri del gruppo:*

Raffaele CASTAGNA

Alberto ROVESTI

Zeno SALETTI

*Numero gruppo:* G17

*Ultima revisione:* 2 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	Attori coinvolti ed esigenze . . . . .	2
<b>2</b>	<b>Requisiti funzionali</b>	<b>3</b>
2.1	Accesso . . . . .	3
2.2	Consultazione dei problemi . . . . .	3
2.3	Esercitazione . . . . .	4
2.4	Gestione profilo e account . . . . .	5
<b>3</b>	<b>Requisiti non funzionali</b>	<b>6</b>
3.1	Caratteristiche delle risorse . . . . .	6
3.2	Caratteristiche di sistema . . . . .	6
3.3	Privacy e sicurezza . . . . .	7
<b>4</b>	<b>Relazioni tra obiettivi e requisiti</b>	<b>8</b>
<b>5</b>	<b>Design front-end</b>	<b>8</b>
<b>6</b>	<b>Design back-end</b>	<b>8</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—utenti finali e utilizzatori del frutto di questo progetto—verranno definiti i requisiti funzionali e non funzionali; verrà presentata una proposta di design di back-end; infine saranno riportati i sistemi 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 Attori coinvolti ed esigenze

Per comprendere meglio i requisiti che verranno descritti in seguito (in particolar modo quelli funzionali), è innanzitutto essenziale specificare il pubblico, insieme alle loro potenziali esigenze, al quale il servizio intende rivolgersi. Tale 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** Il sistema deve mettere a disposizione dei nuovi utenti una pagina nella quale sia possibile effettuare la registrazione alla piattaforma online. Devono essere richiesti i seguenti campi:

- Indirizzo email.
- Nome utente.
- Password: la password deve essere oscurata durante l'inserimento. Al fine di limitare gli errori di digitazione e l'inserimento di una password che non corrisponde a quella scelta dall'utente, deve essere presente un secondo campo di conferma, dove la password deve essere riscritta. La password deve in ogni caso essere conforme a quanto specificato nel RNF 11.

**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. Qualora la password inserita non corrisponda a quella dell'account con email associata, l'utente non può accedere alla piattaforma e deve essere notificato di ciò; è comunque possibile ritentare il login [sicurezza? n massimo tentativi?].

**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 codice di verifica all'indirizzo email specificato dall'utente. L'inserimento dell'email e la richiesta di invio del codice avvengono in una pagina apposita, accessibile da quella di login. Dopo l'invio della richiesta, l'utente deve poter inserire il codice ricevuto.
- L'inserimento di una nuova password che, come in fase di registrazione, deve essere scritta e confermata.
- La notifica di eventuali errori avvenuti durante la procedura di recupero, quali codice errato e incongruenze nell'inserimento e conferma della nuova password (RNF 11).

**RF 4. Logout** L'utente registrato e autenticato deve poter interrompere la sessione di accesso al servizio effettuando un logout.

### 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 in una vista dedicata.
2. Cercare uno o più problemi specifici mediante ricerca filtrata per campi, specificati dai metadati elencati al RNF 2, oppure non filtrata.

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 del documento contenente le informazioni del singolo problema precedentemente selezionato dal catalogo. La visualizzazione deve rendere disponibile alla vista dell'utente i dati relativi al problema e specificati nel RNF 1.

## 2.3 Esercitazione

**RF 7. Avviare una sessione di esercitazione** L'utente deve poter selezionare, attraverso l'apposito catalogo, il problema desiderato avviando una sessione di esercitazione con lo scopo di risolverlo. A tal fine, l'utente deve poter:

1. Attivare, dopo averlo visualizzato, il problema scelto. L'attivazione permette di accedere alle funzionalità descritte nei prossimi punti, oltre a incrementare di una unità il numero di tentativi effettuati dall'utente su quel problema (a meno che il problema non sia già stato risolto in precedenza). L'avvio della sessione di esercitazione deve avvenire previa conferma da parte dell'utente.
2. Essere al corrente di quale linguaggio di programmazione sia attualmente attivo per la scrittura di codice, tramite un menu dedicato dal quale deve altresì essere possibile selezionare uno dei linguaggi disponibili<sup>2</sup>.
3. Scrivere, sotto forma di codice nel linguaggio di programmazione scelto, l'algoritmo risolutivo del problema attualmente attivo. La scrittura deve poter essere effettuata in una vista o finestra apposita.

**RF 8. Correttezza sintattica del codice** L'utente deve poter verificare che il codice scritto sia sintatticamente corretto e pronto per l'esecuzione. Quindi devono essere messe a disposizione le seguenti funzionalità:

1. Compilazione del codice.
2. Visualizzazione di avvisi relativi a eventuali errori di sintassi *oppure* 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 verificare quali e quanti test cases sono andati a buon fine. Per ogni test case, solo le istanze in input potranno essere visualizzate. Inoltre, l'utente deve poter riscrivere e perfezionare l'algoritmo e sottoporre ripetutamente il codice ai test cases.

---

<sup>2</sup>Per approfondimenti sulla disponibilità dei linguaggi, si veda RNF 3.1.

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

3. L'utente deve poter terminare la sessione in un tempo finito. Pertanto: nel caso in cui tutti i test cases sono andati a buon fine la sessione di esercitazione deve terminare e i progressi contenuti nel profilo dell'utente (RF 10) devono essere aggiornati di conseguenza, quindi contrassegnando come *risolto* il problema corrente insieme al numero di tentativi impiegati fino all'istante della risoluzione; in caso contrario, l'utente deve avere la possibilità di rinunciare a fornire soluzioni al problema, terminando la sessione di esercitazione.

## 2.4 Gestione profilo e account

**RF 10. Progressi** L'utente deve poter monitorare i propri progressi, accedendo ai dati seguenti mediante una pagina dedicata:

- Problemi risolti e, per ciascuno di essi, il numero di tentativi richiesti.
- Problemi avviati in passato ma non ancora risolti, con relativo numero di tentativi.
- Score globale: una media campionaria dei tentativi richiesti per problema.

**RF 11. Preferiti:?**

**RF 12. Aggiornamento account** L'utente registrato e autenticato deve poter modificare i dati chiave del proprio account in conformità con i requisiti non funzionali relativi alla 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 nella pagina dedicata a questa particolare operazione. 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 11. L'utente deve essere notificato di tale modifica anche tramite l'indirizzo email attualmente associato all'account [possibilità di disattivare account a causa di azioni sospette?].

### 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 un esempio 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 di cui al RNF 1.
- 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?:

#### 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 i principali browser attualmente disponibili in commercio.

**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.

**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 sarà erogato in lingua italiana. Altrettanto sarà fatto per i testi dei problemi.

**RNF 8. Affidabilità**

**RNF 9. Prestazioni**

### 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](#) per la protezione dei dati.

**RNF 11. Password** 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:

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

**RNF 12. Integrità e operazioni su dati sensibili** L'integrità di dati sensibili dell'account, quali email ma soprattutto password, è garantita da opportune procedure di conferma e notifica all'utente. Data la natura di queste operazioni di sicurezza, i requisiti funzionali relativi alla gestione dei dati sensibili (RF 1, RF 3, RF 12) provvedono a descrivere il comportamento del servizio nel caso di errori e incongruenze.



## **4 Relazioni tra obiettivi e requisiti**

[tabella che associa obiettivi e requisiti?]

## **5 Design front-end**

## **6 Design back-end**