



UNIVERSITY  
OF 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:* 28 settembre 2023

# Indice

<b>1</b>	<b>Introduzione</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>
<b>3</b>	<b>Requisiti non funzionali</b>	<b>4</b>
<b>4</b>	<b>Design front-end</b>	<b>4</b>
<b>5</b>	<b>Design back-end</b>	<b>4</b>

# 1 Introduzione

## 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 servizi di back-end.

## 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* e di *raccolta* di problemi mirati 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 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 del progetto. Essi sono organizzati secondo uno schema che segue gli obiettivi elencati nei paragrafi precedenti. Più in particolare, ogni sottosezione di questa parte del documento risponde a diversi scopi precedentemente accennati, suddividendo eventuali macro-funzioni in requisiti minori nel caso di obiettivi più ampi.

### Accesso al servizio

**RF 1.** Altro requisito funzionale

### Consultazione dei problemi

Il servizio deve mettere a disposizione, tra le altre risorse, un insieme di problemi sui quali l'utente possa esercitarsi. [come sono fatti i problemi, come sono classificati, come vengono ordinati e categorizzati.]

**RF 2.** L'utente deve poter consultare un catalogo di problemi e navigare al suo interno. In particolare, l'utente deve poter:

1. Visualizzare tale catalogo in una vista dedicata (si veda la sezione riguardante il design di front-end per ulteriori dettagli).
2. Cercare uno o più problemi specifici mediante ricerca per campi (nome, difficoltà).

### Esercitazione e risoluzione dei problemi

**RF 3.** 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 il problema scelto, cioè visualizzarlo ed eseguire le funzionalità mostrate nei prossimi punti.
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 4.** 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.

**RF 5.** L'utente deve poter verificare la correttezza del codice scritto eseguendolo, ovvero:

1. .

---

<sup>2</sup>Per approfondimenti sulla disponibilità dei linguaggi, si veda RNF 1

Valutazione della propria carriera

### **3 Requisiti non funzionali**

RNF 1. Scalabilità:

**4 Design front-end**

**5 Design back-end**