



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:* 2.1

*Membri del gruppo:*

Raffaele CASTAGNA

Alberto ROVESTI

Zeno SALETTI

*Numero gruppo:* G17

*Ultima revisione:* 17 novembre 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	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 . . . . .	5
2.4	Gestione del profilo e dell'account . . . . .	7
2.5	Gestione del catalogo dei problemi . . . . .	7
<b>3</b>	<b>Requisiti non funzionali</b>	<b>9</b>
3.1	Caratteristiche di sistema . . . . .	9
3.2	Affidabilità . . . . .	10
3.3	Privacy e sicurezza . . . . .	10
<b>4</b>	<b>Design front-end</b>	<b>11</b>
<b>5</b>	<b>Design back-end</b>	<b>14</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 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 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, 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*.

---

<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.

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

- *Utente anonimo.*
- *Utente autenticato.*
- *Utente amministratore.*

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 livelli superiori.

### Utente anonimo

#### 2.1 Accesso

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

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

In alternativa, l'utente deve poter effettuare la registrazione collegando un account Google.

Si denoterà con l'aggettivo *registrato* un qualsiasi utente che abbia precedentemente effettuato la registrazione.

**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 e anonimo rientra nella categoria degli *utenti autenticati*.

**RF 3. 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:

- 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.
- L'inserimento di una nuova password nella pagina dedicata. La password, come in fase di registrazione, deve essere confermata e conforme al [RNF 11](#).

## 2.2 Consultazione dei problemi

**RF 4. 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*.
  - *Difficoltà*: vengono definiti tre livelli indicativi della difficoltà: bassa, intermedia, alta.
  - *Categoria*: si tratta della principale area di interesse del problema, come ad esempio le strutture di dati coinvolte (array, grafi, ecc.), oppure l'obiettivo principale (come ricerca e ordinamento).
  - *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.
2. Cercare uno o più problemi specifici mediante ricerca filtrata per campi. Ai fini della consultazione del catalogo, i filtri applicabili riguardano i campi *nome*, *difficoltà*, *categoria*.
3. Selezionare dal catalogo un problema specifico, cosicché la descrizione dell'intero problema possa essere visionata per mezzo delle funzionalità di cui al [RF 5](#).

**RF 5. 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 (*dati strutturali*):

- 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 possono accompagnare i testi.
- Almeno due esempi di input insieme al relativo output corretto, che mostra il risultato atteso<sup>2</sup>.

Qualora l'utente desideri consultare il problema, vengono contemporaneamente messe a disposizione le funzionalità di cui al [RF 6](#).

## 2.3 Esercitazione

**RF 6. Avviare un'esercitazione** L'utente deve poter scegliere dal catalogo il problema desiderato e cominciare a risolverlo. L'esercitazione deve avvenire in una vista apposita, 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 5](#)).
2. Visualizzare i test cases ([RF 7.2](#)).
3. Avviare e fermare a proprio piacimento un timer per mantenere traccia del 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>3</sup>.
5. Scrivere, sotto forma di codice nel linguaggio di programmazione scelto, l'algoritmo risolutivo del problema selezionato.
6. Visualizzare il suggerimento ([RF 4.1](#)).

---

<sup>2</sup>Non si devono confondere questi *esempi* con i *test cases* ([RF 7.2](#))

<sup>3</sup>Per approfondimenti sulla disponibilità dei linguaggi, si veda [RNF 1.2](#).

**RF 7. 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 (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 suoi risultati di output.)
2. L'utente deve poter conoscere l'esito dei test case—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 riguardanti la correttezza risolutiva dell'algoritmo.

3. Nel caso in cui l'utente sia in grado di superare tutti i test case disponibili, deve essere mostrato il tempo impiegato, registrato dal cronometro.

## Utente autenticato

Nei seguenti requisiti funzionali, si suppone che l'utente appartenga al livello di accesso *autenticato*, qualora tale informazione dovesse essere omessa. Si sottolinea nuovamente che l'utente autenticato eredita le stesse funzionalità accessibili all'utente anonimo.

**RF 8. Metadati aggiuntivi** Oltre ai campi descrittivi specificati nel [RF 4.1](#), l'utente autenticato deve poter visualizzare e influenzare i seguenti dati aggiuntivi di ogni problema:

1. *Stato*: i problemi risolti dall'utente devono essere contrassegnati, grazie alla disponibilità di questo ulteriore campo.
2. *Preferito*: l'utente che desidera tener traccia dei problemi per lui rilevanti deve poter contrassegnare tali problemi come *preferiti*.

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

## 2.4 Gestione del profilo e dell'account

### RF 9. Progressi

1. Il servizio deve prevedere il tracciamento dei progressi dell'utente autenticato, incrementando il numero di problemi risolti ad ogni sessione di esercitazione andata a buon fine: con riferimento al [RF 7.3](#), dopo la risoluzione, il problema deve inoltre essere contrassegnato come *risolto* (nel rispettivo campo *stato* visibile nel catalogo e nella pagina di consultazione ed esercitazione di quel problema).
2. L'utente 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 10. Aggiornamento account** L'utente registrato e autenticato deve poter modificare i dati chiave del proprio account in conformità con alcune procedure di sicurezza opportune:

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 11](#). L'utente deve essere notificato di tale modifica anche tramite l'indirizzo email attualmente associato all'account.

**RF 11. 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.

## Utente amministratore

### 2.5 Gestione del catalogo dei problemi

**RF 12. 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 5](#) e, all'atto dell'inserimento nel catalogo, devono essere specificati i campi elencati nel [RF 4.1](#).



Inoltre, devono essere forniti almeno 3 test cases (questi saranno impiegati come descritto dal [RF 7.1](#)); ogni test case consiste in un dato in input e il rispettivo output corretto.

**RF 13. Modificare un problema** L'utente amministratore deve poter modificare sia i campi descrittivi del problema, sia i dati strutturali (il contenuto).

**RF 14. 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 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 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 imparare almeno il 90% delle funzionalità del servizio in meno di 30 minuti.

**RNF 4. 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 5. 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 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 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.

**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 richiesto l'inserimento di una password, devono essere rispettate le seguenti caratteristiche:

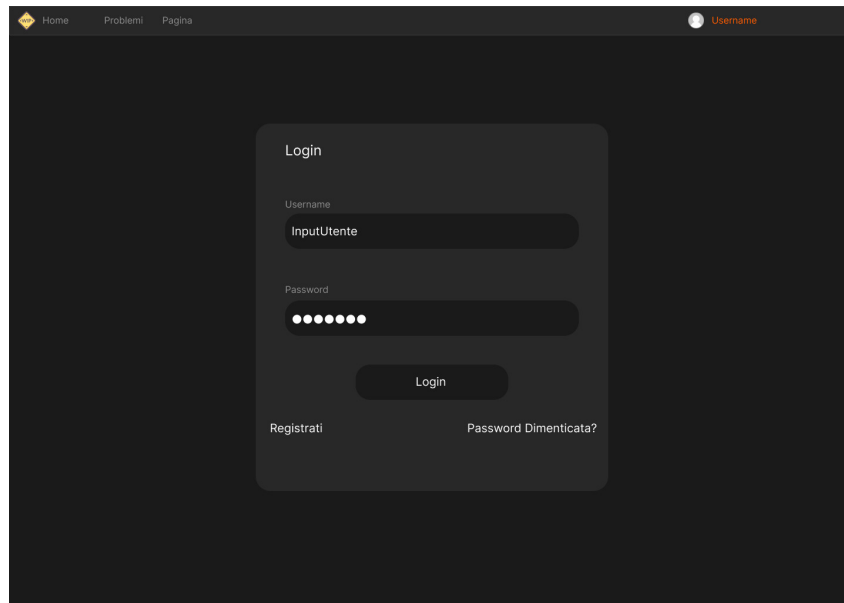
- Lunghezza compresa tra 8 e 64 caratteri.
- Contenere almeno una lettera maiuscola.
- Contenere almeno una lettera minuscola.
- Contenere almeno un numero.
- Contenere almeno un carattere speciale scelto tra i seguenti:

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

## 4 Design front-end

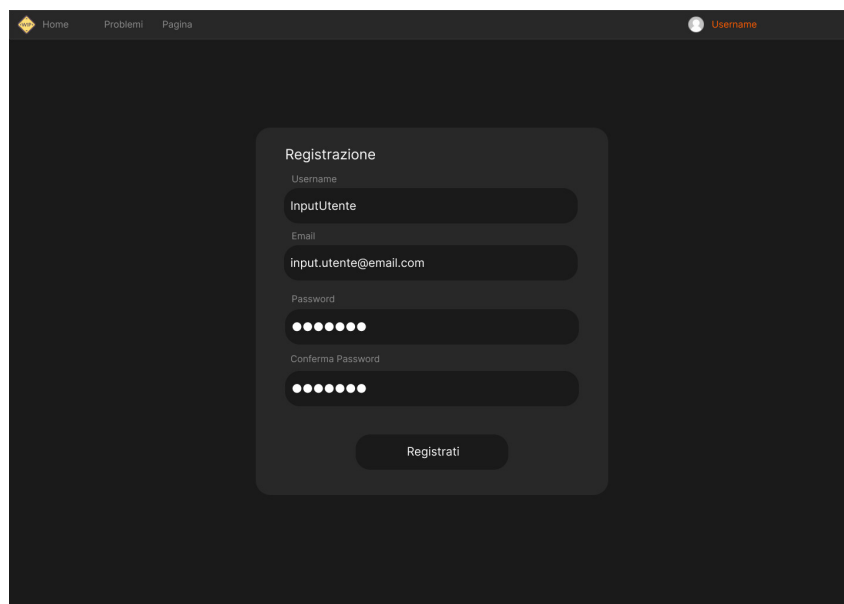
In questa sezione vengono presentati alcuni mock-up dell'interfaccia che la piattaforma online espone all'utente.

### FE 1. Pagina di login



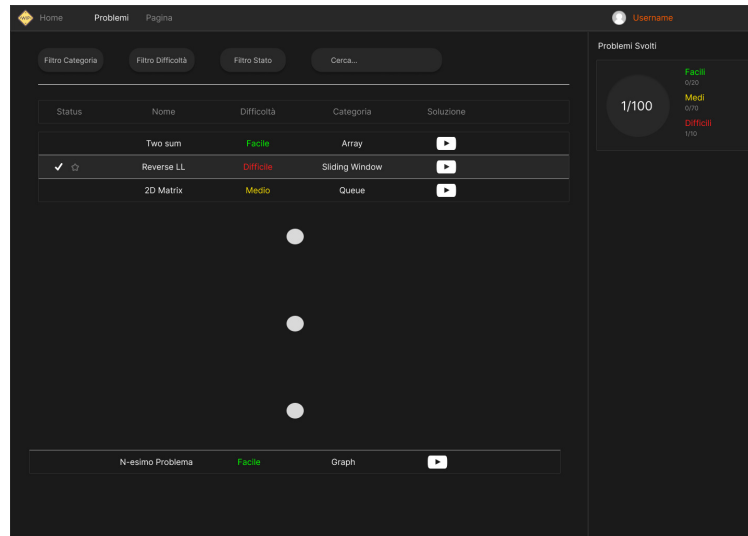
The login page features a dark background with a central light gray login form. The form includes a 'Login' title, a 'Username' label with an input field containing 'InputUtente', a 'Password' label with a masked input field (dots), a 'Login' button, and links for 'Registrati' and 'Password Dimenticata?'.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

### FE 2. Pagina di registrazione

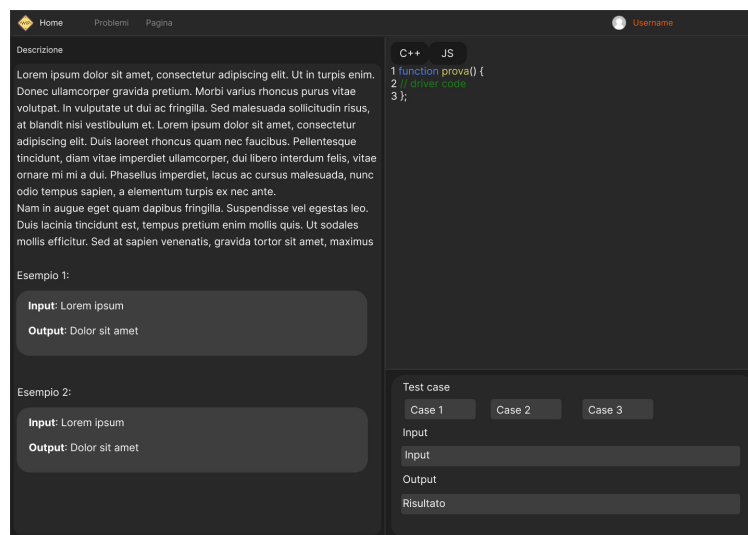


The registration page features a dark background with a central light gray registration form. The form includes a 'Registrazione' title, a 'Username' label with an input field containing 'InputUtente', an 'Email' label with an input field containing 'input.utente@email.com', a 'Password' label with a 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 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. 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*).

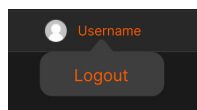


**FE 4. Pagina di esercitazione** Una pagina dedicata raccoglie le funzionalità di consultazione dettagliata del problema e risoluzione di quest'ultimo.



**FE 5. Editor del catalogo** La seguente figura mostra l'interfaccia che si presenta ad un utente amministratore nel momento in cui desidera modificare il catalogo dei problemi.

**FE 6. 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 e realizzare in tal modo le 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 contenuti 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 sarà gestito da servizi di posta elettronica di terze parti.

