

Università degli studi di Milano  
Corso di Programmazione Web e Mobile  
A.A. 2018-2019



Riccardo Tomelleri, 922756



# Sommario

<b>1. Introduzione</b>	<b>3</b>
1.1. Descrizione del progetto	3
1.2. Versioni del documento	3
<b>2. Analisi dei requisiti</b>	<b>4</b>
2.1. Destinatari	4
2.2. Modello di valore	4
2.3. Flusso dei dati	4
2.4. Aspetti tecnologici	5
<b>3. Interfacce</b>	<b>6</b>
3.1. Accesso e registrazione	6
3.2. Home page	7
3.3. Creazione timer	8
3.4. Visualizzazione timer	9
3.5. Navigazione	9
<b>4. Architettura</b>	<b>10</b>
4.1. Legenda della notazione utilizzata	10
4.2. Diagramma dell'ordine gerarchico delle risorse	11
4.3. Diagramma generale dei componenti MVC	11
4.4. Descrizione delle risorse	12
4.4.1. Accesso e registrazione	12
4.4.2. Home page	12
4.4.3. Creazione timer	12
4.4.4. Visualizzazione timer	13
4.4.5. Gestione timer	13
4.4.6. Eliminazione timer	13
4.5. Diagramma del database	14
<b>5. Codice</b>	<b>15</b>
5.1. HTML	15
5.2. CSS	15
5.3. API	15
5.4. Node.js	16
<b>6. Sviluppi futuri</b>	<b>17</b>
<b>7. Conclusioni</b>	<b>17</b>
<b>8. Risorse</b>	<b>17</b>
<b>9. Nota bibliografica e sitografica</b>	<b>17</b>

# Clook

Visualizza lo scorrere del tempo

## 1. Introduzione

### 1.1. Descrizione del progetto

Clook è un'applicazione web ispirata al dispositivo TimeTimer<sup>[1]</sup> per la creazione e gestione di timer personalizzati, che permettono di visualizzare intuitivamente lo scorrere del tempo, con lo scopo di migliorare la produttività in ambito scolastico e lavorativo, oppure di fornire uno strumento di supporto alle persone affette da deficit cognitivi.

Ciò che caratterizza questi timer (più avanti citati anche come “clook”, dall'inglese “clock” e “look”), è la loro visualizzazione per mezzo di un anello colorato che si svuota man mano che il tempo passa, che rende immediatamente consapevole l'utente del tempo che è passato e di quello che resta ancora a sua disposizione.

### 1.2. Versioni del documento

Versione	Data	Commento
1.0	4 giugno 2019	Stesura iniziale.
1.1	17 giugno 2019	Completamento della sezione <i>Interfacce</i> al termine dello sviluppo del progetto.
1.2	20 settembre 2019	Revisione finale e correzioni varie.

## 2. Analisi dei requisiti

### 2.1. Destinatari

L'applicazione è fruibile potenzialmente da chiunque: anche se inizialmente pensata per l'utilizzo in ambito scolastico/educativo e come strumento di supporto per le persone affette da disturbi cognitivi legati al pensiero astratto (come ad esempio l'autismo o il deficit di attenzione)<sup>[2]</sup>, si rivela molto utile anche in ambito lavorativo come strumento per ottimizzare la produttività e migliorare la consapevolezza dello scorrere del tempo.

Essendo Clook un'applicazione web sviluppata in un'ottica di design responsivo, l'utilizzo è possibile da qualunque tipo di dispositivo che possieda un web browser (smartphone, tablet, personal computer, ecc.) e l'adattabilità dell'interfaccia alle dimensioni dello schermo permette una più semplice accessibilità ai contenuti.

L'interfaccia utente dell'applicazione è minimale ed autoesplicativa, permettendone quindi un utilizzo semplice ed intuitivo anche da parte di utenti molto giovani e/o tecnologicamente inesperti. L'esperienza utente è resa il più possibile gradevole e fluida per garantire un piacevole utilizzo dei timer, evitando stress e ansia legati al passaggio del tempo e mantenendo focalizzati gli utenti sull'attività che stanno svolgendo.

L'utente, quindi, non deve essere a conoscenza di linguaggi specifici per il corretto utilizzo dell'applicazione, la quale comunica visivamente e intuitivamente lo stato dei vari timer creati dall'utente.

### 2.2. Modello di valore

Uno dei valori più importanti che caratterizzano il progetto è il massiccio bacino d'utenza potenziale, che comprende studenti e lavoratori di ogni età, con capacità tecnologiche più o meno avanzate.

La semplicità e leggerezza dell'applicazione, inoltre, garantiscono sia un maggiore interesse e predisposizione all'utilizzo da parte degli utenti, sia un costo di realizzazione minore da parte dell'investitore.

L'applicazione è facilmente scalabile a seconda del flusso di utenti e in futuro potrebbe prevedere la possibilità di creare account premium per l'utilizzo di funzioni particolari (ad esempio maggiore personalizzazione dei timer e diverse possibilità di condivisione).

### 2.3. Flusso dei dati

I contenuti che caratterizzano l'applicazione sono creati, gestiti e rimossi direttamente dagli utenti. I dati sono memorizzati in un database non relazionale e consistono, quindi, nelle informazioni di accesso degli utenti (username e hash della password) e nei dati relativi ai timer (autore, titolo, durata, ecc.).

In sviluppi futuri, potrebbe essere resa possibile la condivisione di singoli timer tra i vari utenti, o addirittura la creazione di clook pubblici (con lo scopo di permettere, ad esempio, la pubblicizzazione di determinati eventi) o collettivi (utilizzabili ad esempio in ambito

educativo all'interno delle classi o in ambito lavorativo in occasione di scadenze di vario tipo). Potrebbe anche essere resa possibile una funzione che permetta l'integrazione di un timer personalizzato in pagine web e app.

## 2.4. Aspetti tecnologici

L'applicazione ha un'architettura di tipo Client-Server ed è sviluppata seguendo il design pattern MVC: il Model consiste nel database NoSQL MongoDB<sup>[3]</sup>, contenente due collezioni per la memorizzazione di utenti e clock; la View è creata con HTML5 e CSS3, con l'ausilio del framework W3.CSS (creato dal *World Wide Web Consortium* e scritto solamente con l'utilizzo di CSS standard) che semplifica la creazione di interfacce responsive e compatibili con ogni browser e dispositivo<sup>[4]</sup>; il Controller, infine, è costituito dall'applicazione NodeJS, con l'utilizzo del modulo ExpressJS (che semplifica la creazione di applicazioni web e fornisce una serie di funzioni avanzate utili a questo scopo)<sup>[5]</sup>.

I dati trasmessi in modo asincrono tra client e server sono codificati in formato JSON, che ne permette un semplice ed immediato utilizzo da parte di entrambi, in quanto sia sul client che sul server la logica è gestita da programmi scritti in JavaScript, per il quale questo formato è stato specificatamente progettato. Le pagine web sono costruite a partire da un template che viene renderizzato dal view engine Handlebars (che fornisce un modulo appositamente creato per l'utilizzo con ExpressJS)<sup>[6]</sup> in base ai dati ricavati dal Model.

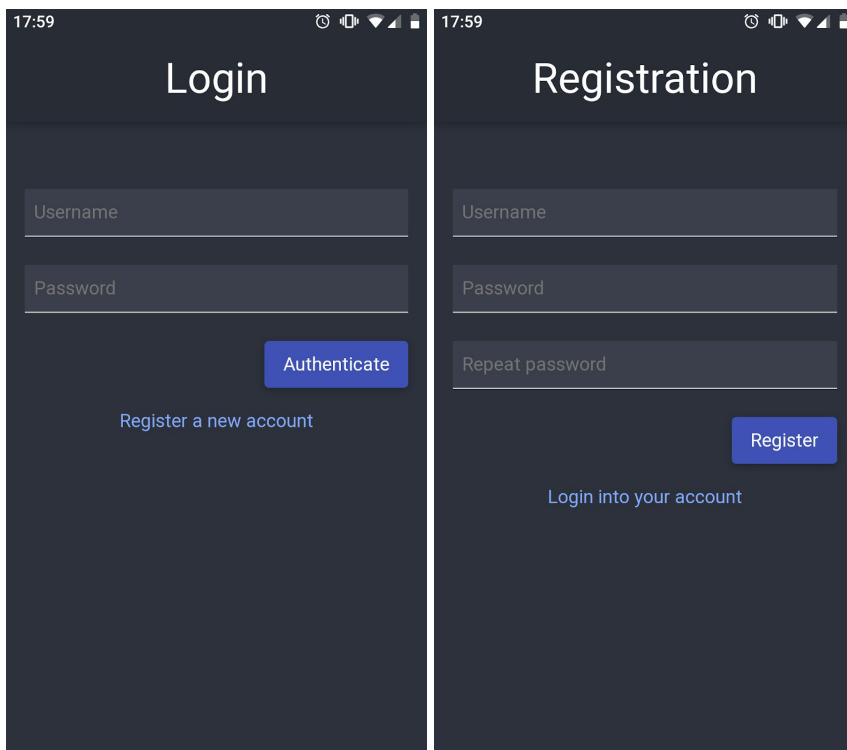
Per migliorare ulteriormente l'esperienza utente, l'applicazione è stata resa una Progressive Web App, il che gli consente di essere installata sui dispositivi e quindi di funzionare parzialmente anche offline e di comportarsi in modo simile ad un'applicazione nativa, senza però precluderne la compatibilità multiplatforma. Per farlo è stato necessario servire l'applicazione tramite protocollo HTTPS (che garantisce quindi anche la sicurezza di canale) e fornirla di un Web Manifest in formato JSON e di un Service Worker (ovvero uno script che il browser esegue in background e che funge da middleware durante le richieste tra client e server, offrendo in particolare la possibilità di memorizzare in cache le risorse statiche dell'applicazione e ridurre la banda necessaria per il caricamento delle pagine).

Infine, sono stati utilizzati i moduli Mongoose (per la configurazione del Model basato su MongoDB) e JSON Web Token (che permette di mantenere la sessione degli utenti una volta effettuato l'accesso, così da verificarne l'autorizzazione ad accedere a determinate risorse)<sup>[7]</sup>.

## 3. Interfacce

### 3.1. Accesso e registrazione

Queste due pagine permettono rispettivamente l'accesso e la registrazione all'applicazione. Nella prima è necessario inserire username e password e premere il tasto *Authenticate* per fare l'accesso ed essere reindirizzati alla home page. Nella seconda occorre inserire un nuovo username (che non può essere uguale a quello di utente già registrato) e una nuova password, che va poi ripetuta; a questo punto premendo il tasto *Register* si completa la registrazione e si viene reindirizzati alla home page.



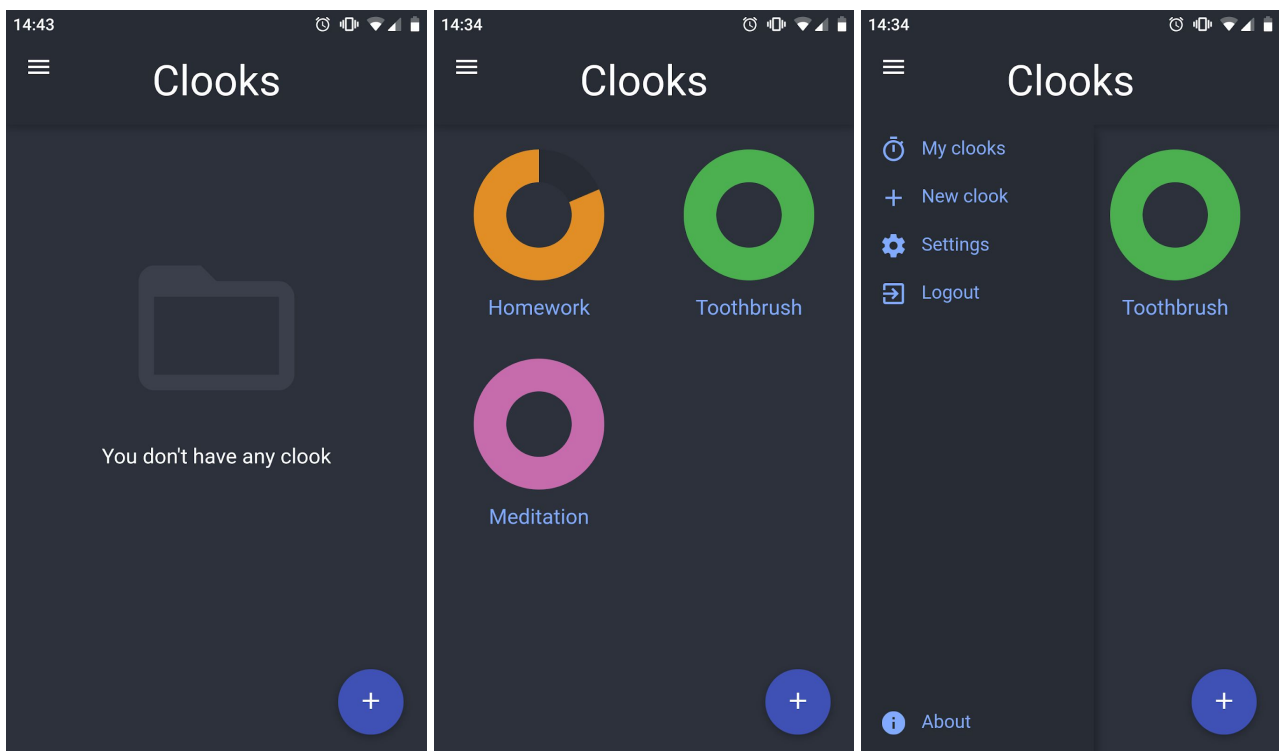
The image displays two side-by-side mobile application screens. The left screen, titled 'Login', features a dark blue background with white text. It has two input fields for 'Username' and 'Password', followed by a blue 'Authenticate' button. Below the button is a link that says 'Register a new account'. The right screen, titled 'Registration', also has a dark blue background with white text. It includes input fields for 'Username', 'Password', and 'Repeat password', followed by a blue 'Register' button. Below the button is a link that says 'Login into your account'. Both screens show a status bar at the top with the time 17:59 and various icons.

Schermata di accesso

Schermata di registrazione

### 3.2. Home page

Nella home page sono visualizzati i vari timer creati dall'utente, compreso il loro stato di avanzamento; cliccando su ogni clook, si viene indirizzati alla pagina di visualizzazione di quello specifico timer. Sulla sinistra è presente un menù di navigazione che indirizza alle varie pagine dell'applicazione; questa può essere mostrata o nascosta premendo sull'icona in alto a sinistra. Inoltre, in basso a destra, è presente un cosiddetto bottone *fab* (Favourite Action Button) che permette di navigare velocemente alla pagina dedicata alla creazione di un nuovo clook.



Home page quando non è presente alcun timer

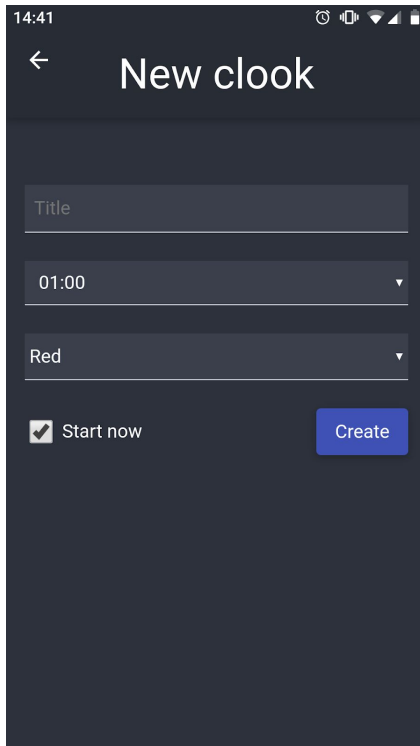
Home page con alcuni clook visualizzati (il primo attivo)

Home page con menù di navigazione visualizzato

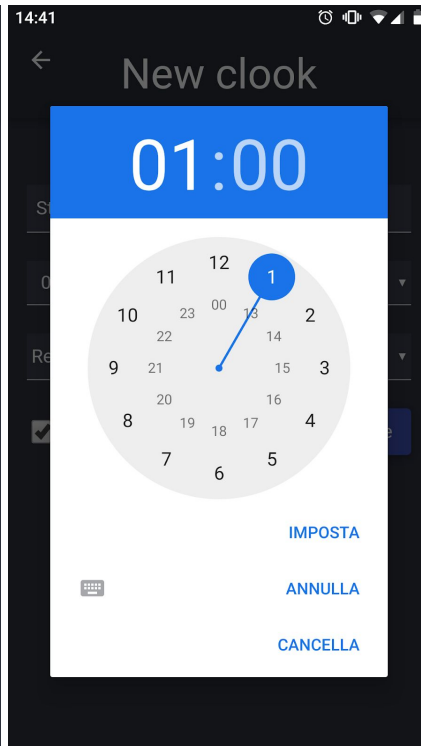


### 3.3. Creazione timer

In questa pagina è possibile creare un nuovo clock: per farlo occorre inserire il nome del nuovo timer, la sua durata e il colore con il quale si vuole che venga visualizzato. Inoltre è possibile selezionare una casella che permette di indicare se si vuole che il timer inizi a scorrere al momento della creazione, oppure no. Una volta inseriti tutti i parametri per la creazione del nuovo Clock, premendo il pulsante *Create* si viene reindirizzati alla pagina di visualizzazione del nuovo timer appena creato.



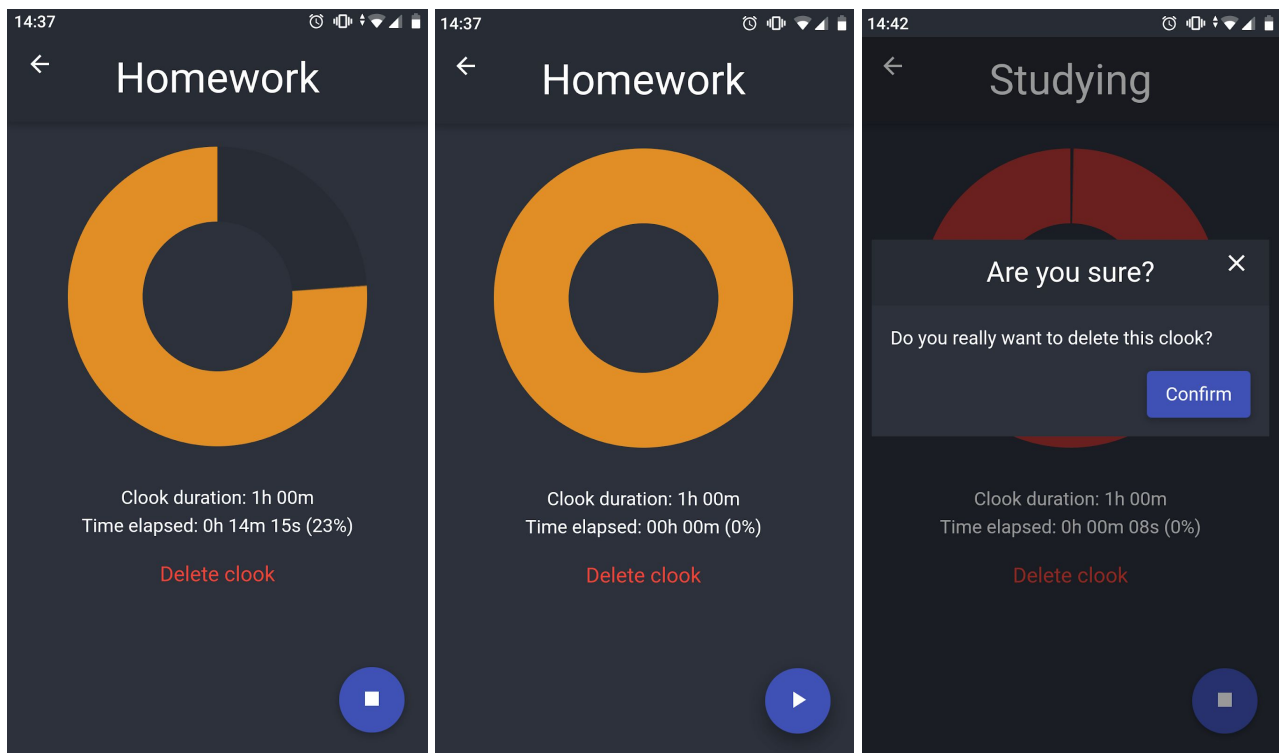
Interfaccia per la creazione di un nuovo clock



Selezione della durata del timer con il time-picker nativo del dispositivo

### 3.4. Visualizzazione timer

Qui è possibile visualizzare il nome e la durata del clook, e l'avanzamento è espresso con la visualizzazione ad anello, ma anche in minuti e in percentuale. In basso a destra è presente un tasto che permette di avviare il timer o di reimpostarlo (se questo è stato avviato) mentre in fondo alla pagina c'è il tasto che permette eliminare il timer, che chiederà una conferma per evitare errori da parte dell'utente.



Visualizzazione del clook  
(avviato)

Visualizzazione del clook  
(fermato)

Pop-up per la conferma  
dell'eliminazione del timer


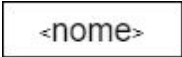

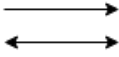

### 3.5. Navigazione

Quando ci si collega all'indirizzo dell'applicazione, si viene reindirizzati alla home page o alla pagina di accesso a seconda che l'utente abbia effettuato l'accesso o meno.

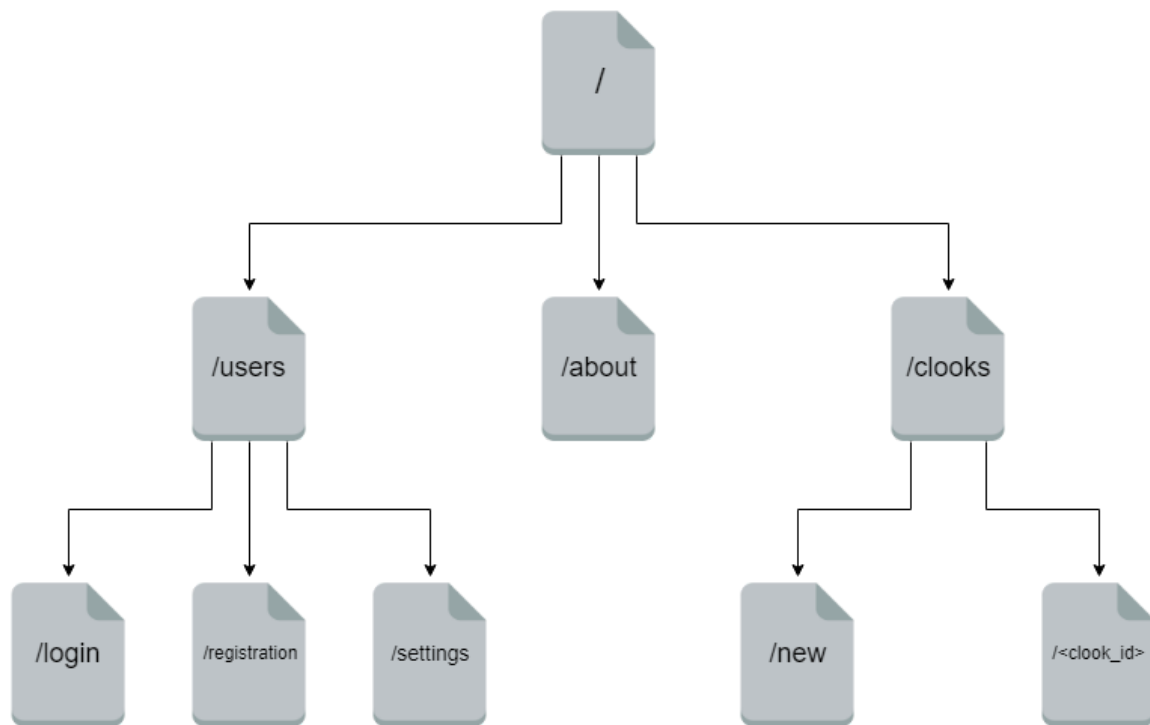
Mentre nella home page è presente un menù di navigazione completo, in tutte le altre pagine (ad eccezione di quelle di accesso e di registrazione) è presente in alto a destra solamente una freccia che porta alla home page.

## 4. Architettura

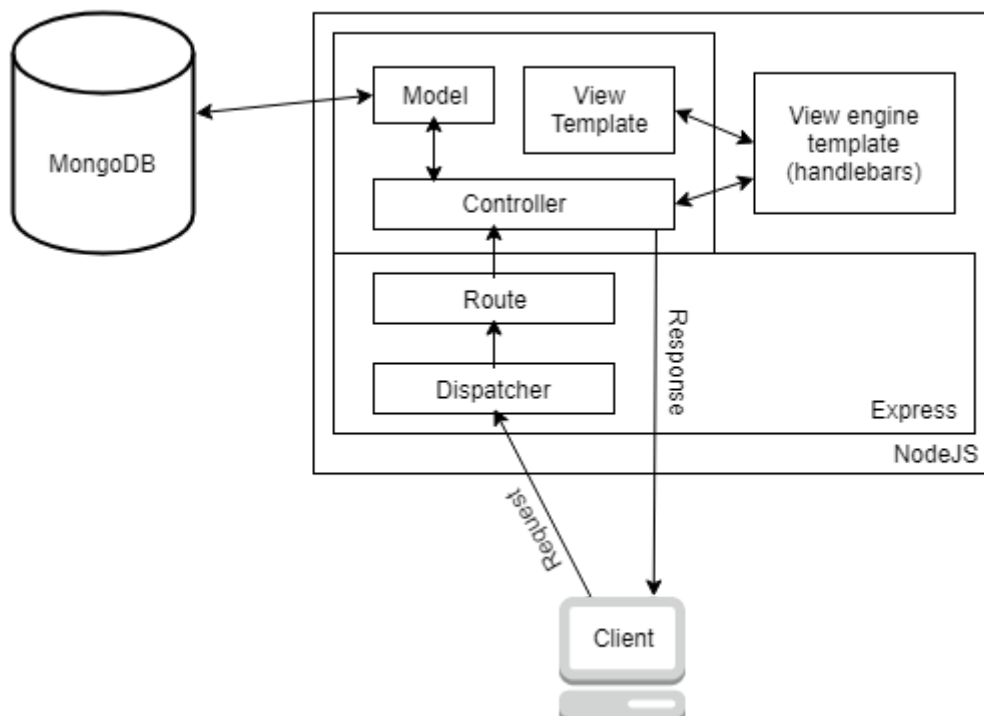
### 4.1. Legenda della notazione utilizzata

Notazione	Descrizione
	Pagina web
	Eseguibile o script
	Database
	Transazione GET o POST
	Ordine di esecuzione della transazione

## 4.2. Diagramma dell'ordine gerarchico delle risorse

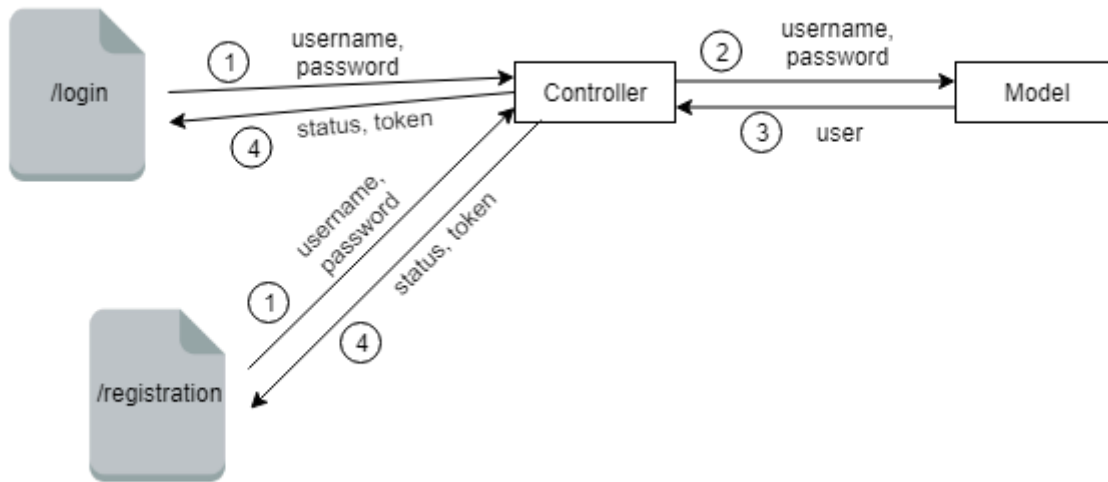


## 4.3. Diagramma generale dei componenti MVC

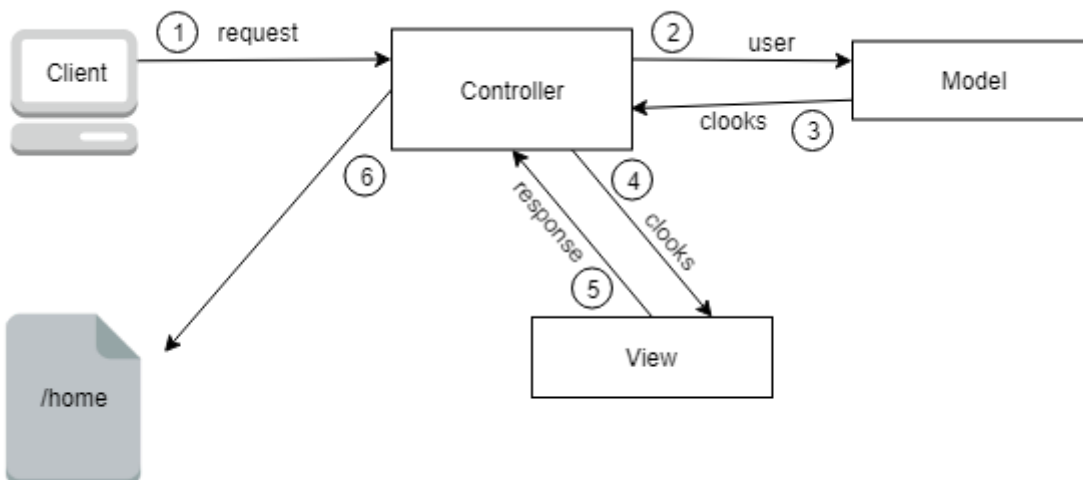


## 4.4. Descrizione delle risorse

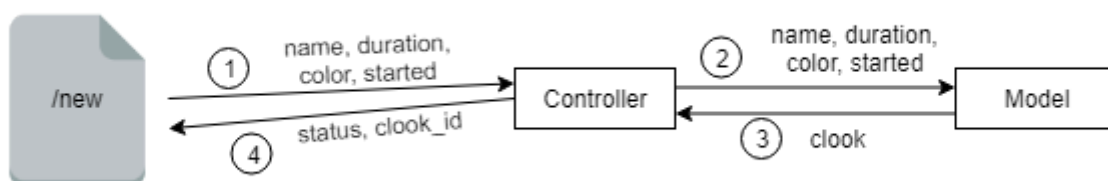
### 4.4.1. Accesso e registrazione



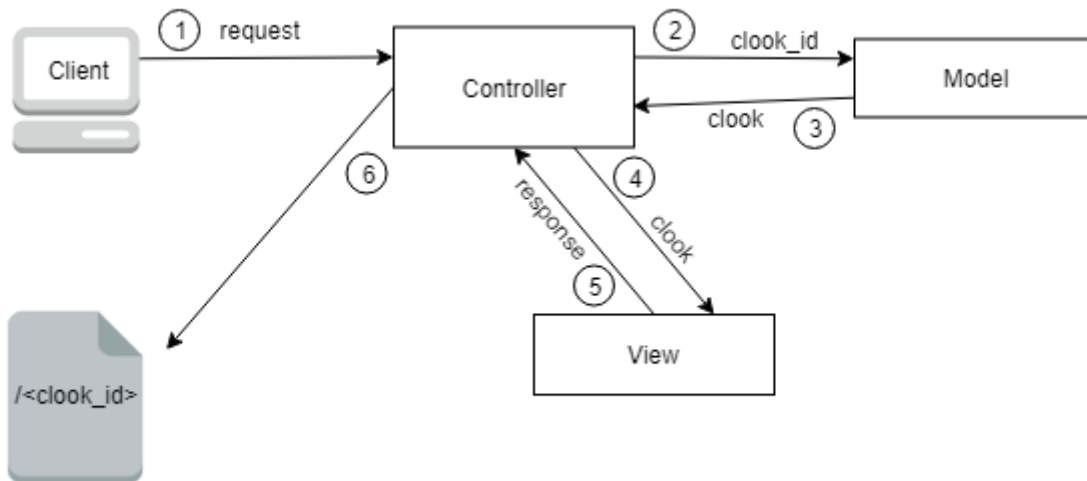
### 4.4.2. Home page



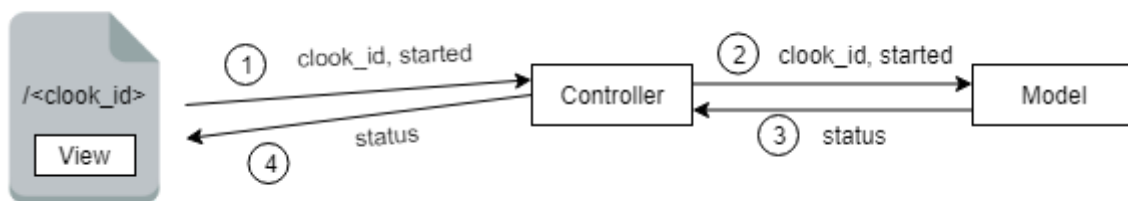
### 4.4.3. Creazione timer



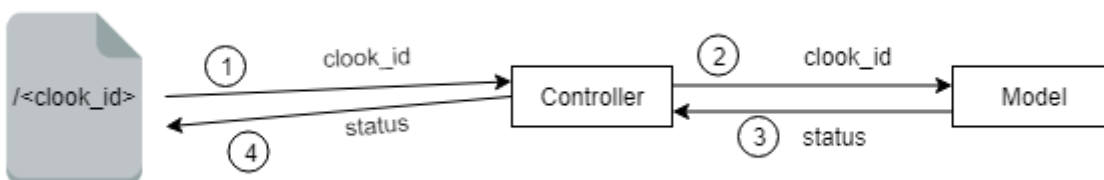
#### 4.4.4. Visualizzazione timer



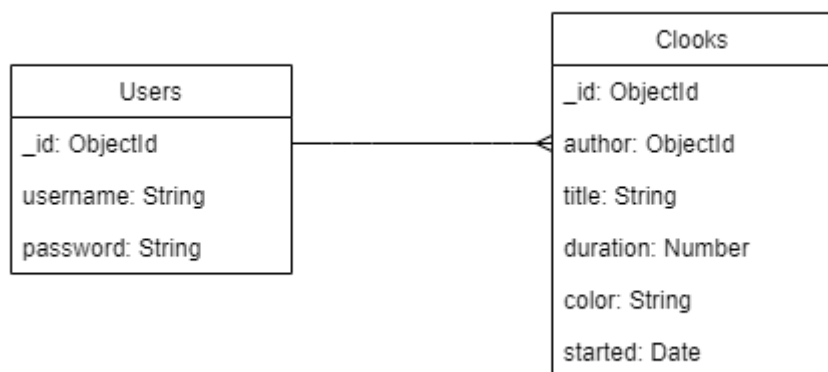
#### 4.4.5. Gestione timer



#### 4.4.6. Eliminazione timer



## 4.5. Diagramma del database



## 5. Codice

Frammenti più significativi del codice dei vari componenti dell'applicazione.

### 5.1. HTML

```
<!-- CLOOK -->
<div class="w3-container clook-container">
  <div class="clook">
    <div class="timer"></div>
    <div class="middle"></div>
  </div>
</div>

<!-- INFO & OPERATIONS -->
<div class="w3-container w3-margin-top w3-center">
  <p>Clook duration: <span id="duration"></span>
  <br>Time elapsed: <span id="elapsed"></span> (<span
id="sector"></span>%)</p>
  <a href="#" id="delete" class="w3-large w3-text-red">Delete clook</a>
</div>

<!-- START & STOP BUTTON -->
<button class="w3-btn w3-card w3-xlarge w3-circle fab start-and-stop">
  <i class="material-icons">play_arrow</i>
</button>
```

### 5.2. CSS

```
/* CLOOK ELEMENT */
.clook {
  position: relative;
  left: 10%;
  width: 80%;
}

/* TIMER ELEMENT */
.timer {
  width: 100%;
  padding-top: 100%;
  border-radius: 50%;
  background-color: var(--primary-color);
}
```



## 5.3. API

```
// Delete button initialization
function init_delete() {
    document.querySelector('#delete').addEventListener('click', function() {
        ajax('DELETE', '/clocks', {
            clock_id: clock_info.id
        }, function(res) {
            window.location.href = '/clocks';
        });
    });
}

// Ajax request
function ajax(method, url, data, callback) {
    var req = new XMLHttpRequest();
    req.open(method, url, true);
    req.setRequestHeader('Content-Type', 'application/json');
    req.onreadystatechange = function() {
        if (req.readyState === 4 && req.status === 200)
            callback(JSON.parse(req.responseText));
    };
    req.send(JSON.stringify(data));
}
```

## 5.4. Node.js

```
// Required modules
const jwt = require('jsonwebtoken');

// Check if user is authenticated
function validate_user(req, res, next) {
    var token = req.cookies['token'];
    var secret = req.app.get('secret');
    jwt.verify(token, secret, function(err, decoded) {
        if (err) res.redirect('/users/login');
        else {
            req.body.user_id = decoded.id;
            next();
        }
    });
}
```

## 6. Sviluppi futuri

Come accennato in precedenza, in futuro potrebbero essere aggiunte nuove funzionalità che arricchirebbero notevolmente l'applicazione, alcune delle più interessanti potrebbero essere:

- possibilità di creare timer di durata maggiore (ovvero di più giorni);
- possibilità di inserire qualsiasi colore tramite un color-picker dedicato;
- possibilità di creare widget sul dispositivo o incorporare i timer nei siti web;
- possibilità di condividere i timer ai singoli utenti, ma anche a gruppi di utenti, oppure di renderli pubblici;
- possibilità di programmare i timer in modo che partano automaticamente in un preciso momento.

## 7. Conclusioni

Clook si propone come un'applicazione web semplice e di facile utilizzo, che può essere usata su una vasta gamma di dispositivi e trattata come un'applicazione native-like. La sua particolare capacità di comunicare visivamente tempi e scadenze la rende un dispositivo di supporto educativo dal grande potenziale e uno strumento in grado di ottimizzare il lavoro. Ciò le permette di rivolgersi ad una moltitudine eterogenea di utenti, il che (insieme alla prospettiva di aggiornamenti futuri volti all'arricchimento dell'esperienza utente per mezzo dell'introduzione di funzioni avanzate e possibili piani a pagamento) fa dell'applicazione un prodotto di grande valore, nonché una proposta interessante per eventuali investitori.

## 8. Risorse

- Repository: [github.com/tomellericcardo/clook](https://github.com/tomellericcardo/clook)
- Living example: [clook.herokuapp.com](https://clook.herokuapp.com)

## 9. Nota bibliografica e sitografica

1. [www.timetimer.com](https://www.timetimer.com)
2. [www.timetimer.com/pages/time-timer-in-special-needs](https://www.timetimer.com/pages/time-timer-in-special-needs)
3. [www.mongodb.com](https://www.mongodb.com)
4. [www.w3schools.com/w3css](https://www.w3schools.com/w3css)
5. [expressjs.com](https://expressjs.com)
6. [www.npmjs.com/package/express-handlebars](https://www.npmjs.com/package/express-handlebars)
7. [jwt.io](https://jwt.io)