**Classroom Scanner AR**

**Titolo progetto:** Classroom Scanner AR

**Alunno/i:** Julian Sprugasci

Lorenzo Piazza,

Pierpaolo Casati e

Claudio Engeler

**Classe:**  Info 3

**Anno scolastico:** 2019/2020

**Responsabile:** Geo Petrini

Sommario

**1 Introduzione**

* 1. **Informazioni sul proggetto**
  2. **Abstract**
  3. **Scopo**

1. **Analisi**
   1. **Analisi del dominio**
   2. **Analisi e specifica dei requisiti**
   3. **Use case**
   4. **Pianificazione**
   5. **Analisi dei mezzi**
      1. **Software**
      2. **Hardware**
2. **Progettazione e implementazione**
   1. **Location**
   2. **Pattern**
      1. **Come creare un QR Code**
      2. **Come creare un Pattern**
      3. **QR Code Floor**
   3. **Database**
   4. **Creazione del database**
      1. **Introduzione**
      2. **Codice**
      3. **Entità Classroom**
         1. **Introduzione**
         2. **Progettazione**
         3. **Codice**
      4. **Entità School\_Hour**
         1. **Introduzione**
         2. **Progettazione**
         3. **Codice**
   5. **Selenium**
      1. **Estrapolazione dati (Scraper)**
      2. **Assegnazione del browser Firefox**
      3. **Mascheramento del browser**
      4. **Ricerca e recupero dati dell’orario scolastico**
      5. **Aggiunta di un sito web**
      6. **Verifica caricamento degli elementi**
      7. **Ricavare elementi HTML**
      8. **Scrittura nome della classe**
      9. **Cancellazione barra di ricerca**
      10. **Scrittura all’interno della barra**
      11. **Avvio della ricerca**
      12. **Parsing dei dati**
      13. **Struttura del nostro JSON**
      14. **Parte codice importante**
   6. **Flask**
   7. **Sito Web**
      1. **Interfaccia admin**
         1. **Accesso alla interfaccia admin**
         2. **Spiegazione interfaccia**
            1. **Sezione Classrooms**
            2. **Sezione School hours**
      2. **Guida utente ufficiale**
         1. **Introduzione**
         2. **Passaggi**
            1. **Passo 1**
            2. **Passo 2**
            3. **Passo 3**
            4. **Passo 4**
            5. **Passo 5**
   8. **Installazione server**
      1. **Requisiti**
      2. **Come avviare lo script**
      3. **Spiegazione script install\_ubuntu.sh**
      4. **Spiegazione script uninstall\_ubuntu.sh**
   9. **Struttura generale**
   10. **Web server**
   11. **Reverse Proxy**
3. **Test dei requisiti**
4. **Consuntivo**
5. **Conclusioni**
6. **Sviluppi futuri**
7. **Allegati**
8. **Indice delle figure**
9. **Introduzione**
   1. **Informazioni sul progetto**

Allievi coinvolti: Julian Sprugasci, Lorenzo Piazza, Pierpaolo Casati e Claudio Engeler

Classe: Informatica 3AC presso la SAM Trevano

Docente responsabile: Geo Petrini

Data inizio: 2019/01/17

Data fine: 08.05.2020

* 1. **Abstract**

*A school needs a system capable of managing lesson times and various classrooms. This project aims to create an augmented reality system, which through the QR codes present in the classrooms will make the time of day of that single classroom appear to improve research and facilitate people. It will also be possible to search for a classroom through a GPS function. To do this we will use the notions learned in our school using the following languages:*

*sql, php, css, js and html.*

* 1. **Scopo**

Questo progetto ha lo scopo didattico principale di imparare a gestire in team un progetto IT e di prepararci per l’esame di fine tirocinio. Per fare ciò dobbiamo ricorrere a tutte nozioni apprese durante la nostra formazione, utilizzando molti aspetti visti in varie materie, come per esempio la creazione e gestione di un Gantt oppure l’utilizzo di vari linguaggi di programmazione. L'altro scopo di questo progetto sarà quello di creare un applicativo che permetta di gestire l’orario scolastico della nostra sezione attraverso la realtà aumentata. Tutto il lavoro verrà suddiviso equamente all’interno del nostro team di lavoro per spartirci bene i compiti e svolgere al meglio il progetto.

1. **Analisi**
   1. **Analisi del dominio**

Il docente supervisore che ci ha stipulato le specifiche del progetto è dalla parte del cliente e ci ha assegnato un diario dei compiti. Il risultato finale, come spiegato nelle specifiche dovrà essere un applicativo web che dovrà gestire gli orari della nostra sezione e dare la possibilità alle persone di guardare l’orario in maniera immediata e più semplice attraverso la loro telecamera del telefono. L’applicativo sarà implementato in html, css, js e python. Il tutto sarà caricato su un web server di Github e quindi sarà facilmente accessibile da tutti coloro che hanno un browser web adeguato (Firefox, Chrome e Safari).

Questo prodotto non esiste ancora sul mercato ma non verrà progettato per scopi lucrativi ma solamente a scopo didattico per consolidare le nostre nozioni. All’inizio del piano della nostra sezione scolastica ci sarà un codice QR che corrisponderà al piano corrente. Poi, in ogni aula del quarto piano sarà presente un codice QR con un pattern univoco per distinguerle, che una volta scannerizzato mostrerà a schermo l’orario della lezione in corso. Mentre se si volessero avere più informazioni riguardanti quell’aula si potrà premere sul popup virtuale a schermo che porterà direttamente ad una pagina web con tutte le informazioni necessarie. Tutte le specifiche e la guida su come utilizzare il prodotto saranno presenti sul nostro sito.

* 1. **Analisi e specifica dei requisiti**

|  |  |
| --- | --- |
| **ID: REQ-001** | |
| **Nome** | Realizzare un sito web per l’applicativo |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Il sito deve essere adatto per qualsiasi dispositivo |
| **002** | Deve esserci una guida utente |
| **003** | Il sito dovrà essere indipendente |
| **004** | L’interfaccia e i colori di sfondo dovranno essere adatti a qualsiasi utente |

|  |  |
| --- | --- |
| **ID: REQ-002** | |
| **Nome** | Dovrà essere presente un QR Code per il quarto piano |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Categoria** | Sistema |
| **Sotto Requisiti** | |
| **001** | Il QR Code avrà un colore di sfondo particolare per essere distinto |
| **002** | Il QR Code dovrà portare direttamente al nostro sito web |

|  |  |
| --- | --- |
| **ID: REQ-003** | |
| **Nome** | Ogni aula dovrà avere un pattern univoco per essere riconosciuta |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Categoria** | Sistema |
| **Sotto Requisiti** | |
| **001** | Il pattern non dovrà avere delle forme complesse |

|  |  |
| --- | --- |
| **ID: REQ-004** | |
| **Nome** | L’applicativo dovrà avere la funzione di lettura dei pattern |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Sistema e Linguaggio |
| **Sotto Requisiti** | |
| **001** | Lo scanner dovrà essere eseguito attraverso la fotocamera del dispositivo |
| **002** | Dovrà essere in grado di riconoscere tutti i pattern assegnati alle aule |
| **003** | Una volta proiettato il risultato dovrà essere possibile cliccare sul popup per mostrare delle informazioni in più riguardanti l’orario |
| **004** | L’interfaccia e i colori di sfondo dovranno essere adatti a qualsiasi utente |

|  |  |
| --- | --- |
| **ID: REQ-005** | |
| **Nome** | L’applicativo dovrà mostrare gli orari in AR |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Bisogna creare un finestra popup uguale per tutte le aule |
| **002** | Dovrà essere possibile leggere il database e estrarre l’orario corrente |
| **003** | Dovrà esserci la funzione “Maggiori informazioni” |
| **004** | Bisogna mostrare gli orari da qualsiasi angolazione si legge il QR Code |

|  |  |
| --- | --- |
| **ID: REQ-006** | |
| **Nome** | L’applicativo dovrà avere un database con tutti gli orari |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio e Sistema |
| **Sotto Requisiti** | |
| **001** | Il database dovrà essere aggiornato una volta ogni anno |
| **002** | Il database deve contenere tutti i dati riguardanti gli orari |
| **003** | Il database dovrà essere sempre attivo durante i giorni lavorati |
| **004** | Il database sarà sviluppato il MySQL e caricato su un server |

|  |  |
| --- | --- |
| **ID: REQ-007** | |
| **Nome** | Bisogna creare uno script in grado di estrarre i dati dal sito dell’orario |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Lo script verrà sviluppato il Python e Flask |
| **002** | Dovrà essere in grado di estrarre i dati e inserirli nel database |
| **003** | Dovrà essere eseguito una volta all’anno per aggiornare i dati |

|  |  |
| --- | --- |
| **ID: REQ-008** | |
| **Nome** | L’applicativo dovrà avere un sistema per ricercare le aule |
| **Priorità** | 3 |
| **Versione** | 1.0 |
| **Categoria** | Linguaggio |
| **Sotto Requisiti** | |
| **001** | Dovrà avere la possibilità di avere un filtro per la ricerca (docente, aula o classe) |
| **002** | La funzione dovrà tenere conto della posizione corrente dell’utente |
| **003** | La funzione dovrà mostrare a schermo o indicativamente il percorso da svolgere per arrivare alla destinazione |
| **004** | La funzione dovrà funzionare tramite il gps integrato del dispositivo senza dover installare niente di esterno. |

* 1. **Use case**

Questa sezione è dedicata al nostro Use case. Essendo che il nostro progetto non è molto complesso con le interazioni lo schema non risulta complicato ma bensì molto semplice:



Figura 1 - Use case

* 1. **Pianificazione**

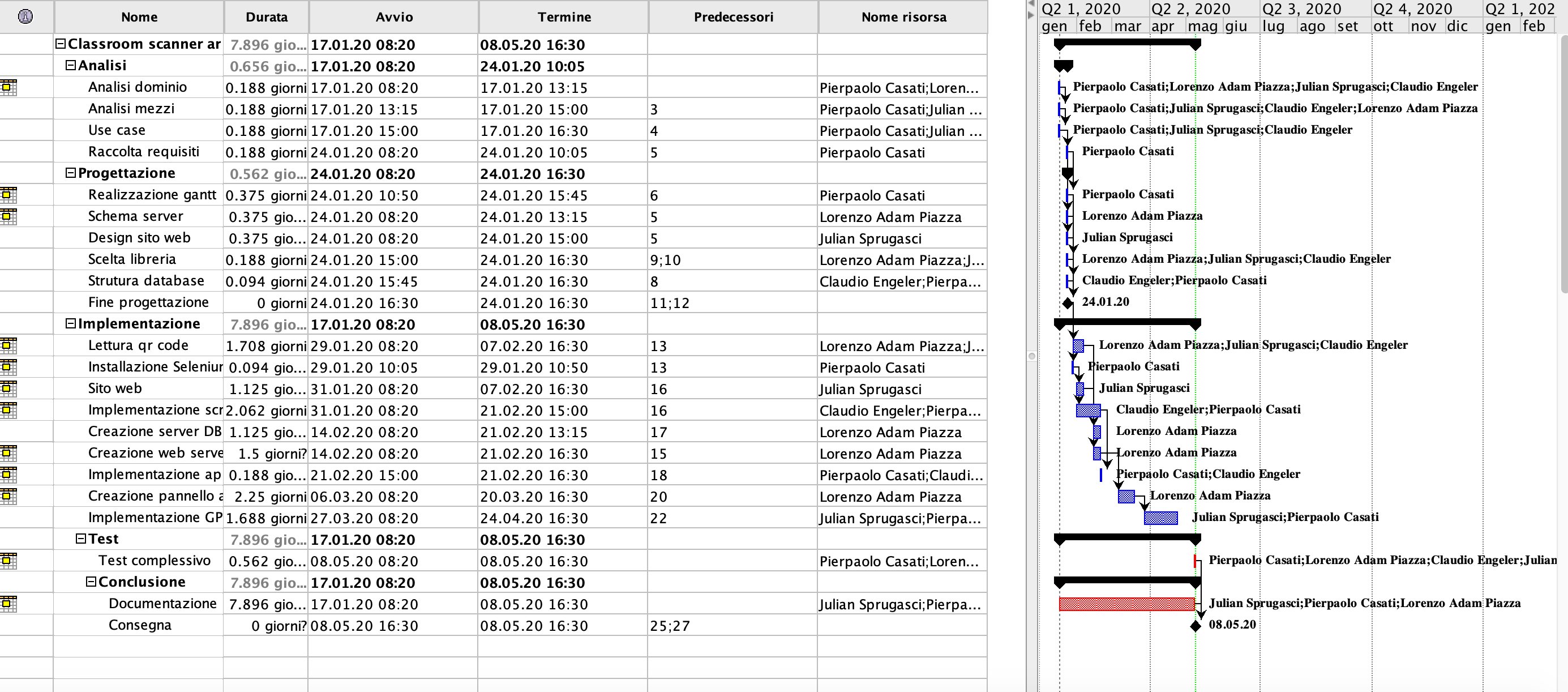


Figura 2 - Gantt

* 1. **Analisi dei mezzi**
     1. **Software**

Per la realizzazione del progetto abbiamo utilizzato le seguenti librerie:

|  |  |
| --- | --- |
| **AR.js – 2.2.2** | Questa libreria l’abbiamo utilizzata per la realtà aumentata, precisamente per leggere i codici QR in modo da visualizzare gli orari attraverso degli oggetti virtuali. |
| **Plugin Firefox Selenium IDE – 3.16.1** | Questo programma l’abbiamo utilizzato per ricavare i dati dell’orario scolastico attraverso il sito della scuola. Selenium permette di registrare delle interazioni con dei siti web per aiutare a generare e mantenere l’automazione del sito, facendo dei test. |
| **Bootstrap - 4.4.1** | Questa strumento è stato utilizzato per tutto quello che concerne la parte grafica del nostro sito web, dando a disposizione degli strumenti veramente performanti e puliti con un’ottima documentazione intuitiva. |
| **Selenium webdriver - 3.141.59** | Permette di definire quale driver browser dovrà essere utilizzato per eseguire i test e le varie operazioni sul sito in modo automatizzato. Il web browser che utilizziamo è Chrome. |
| **BeautifulSoup – 4.4.0** | Permette di ricavare i dati dal sito. Precisamente è un “parser” di linguaggi xml e html. |
| **Flask – 1.1.1** | È un micro web framework. |
| **Re** | Libreria per le espressioni regolari |
| **Time** | Libreria che contiene dei metodi per il tempo. |
| **Datetime** | Libreria che contiene dei metodi per le date. |
| **Selenium** | Package di selenium che contiene diverse librerie. |
| **Beautiful soup** | Libreria che permette di parsare del codice HTML. |

* + 1. **Hardware**

Visto che il nostro progetto è interamente lato web, non abbiamo dovuto utilizzare delle macchine particolari per lo sviluppo. Abbiamo utilizzato semplicemente i nostri laptop (Windows, Linux, Mac).

1. **Progettazione e Implementazione**
   1. **Location**

L’utente entrerà dalla porta di entrata della scuola. All’inizio del quarto piano sarà presente un codice QR per entrare nel nostro sito web. Nel menu principale del nostro sito ci sarà una breve guida utente per guidarlo nell’utilizzo dell’applicativo. Una volta capito il funzionamento potrà scegliere quale funzione utilizzare.

* 1. **Pattern**

Il codice QR è il marchio di fabbrica di un tipo di codice a barre a matrice (o codice a barre bidimensionale). Un codice a barre è un'etichetta ottica leggibile dalla macchina che contiene informazioni sull'elemento a cui è allegato. In pratica, un codice QR spesso contiene dati per un localizzatore, identificatore o tracker che puntano a un sito Web o un'applicazione. Un codice QR è costituito da quadrati neri disposti su una griglia quadrata su uno sfondo bianco, che possono essere letti da un dispositivo di imaging come una fotocamera ed elaborati utilizzando la correzione dell'errore Reed-Solomon fino a quando l'immagine non può essere interpretata in modo appropriato. I dati richiesti vengono quindi estratti dai motivi presenti in entrambi i componenti orizzontali e verticali dell'immagine.



Figura 3 - QR Code Example

Tutti i codici QR hanno una forma quadrata e includono tre contorni quadrati negli angoli in basso a sinistra, in alto a sinistra e in alto a destra. Questi contorni quadrati definiscono l'orientamento del codice. I punti all'interno del codice QR contengono informazioni sul formato e sulla versione, nonché il contenuto stesso. I codici QR includono anche un certo livello di correzione degli errori, definito come L, M, Q o H. Una bassa quantità di correzione degli errori (L) consente al codice QR di contenere più contenuti, mentre una maggiore correzione degli errori (H) rende il codice più facile da scansionare. I codici QR hanno due vantaggi significativi rispetto agli UPC tradizionali: i codici a barre comunemente utilizzati negli imballaggi al dettaglio. Innanzitutto, poiché i codici QR sono bidimensionali, possono contenere significativamente più dati di un UPC monodimensionale. Mentre un UPC può includere fino a 25 caratteri diversi, un codice QR 33x33 (versione 4), può contenere 640 bit o 114 caratteri alfanumerici. Un codice QR 177x177 (versione 40) può contenere fino a 23.648 bit o 4.296 caratteri. Un altro vantaggio dei codici QR è che possono essere scansionati da uno schermo.

* + 1. **Come creare un QR Code**

Per creare un codice QR non è difficile. Non lo creeremo manualmente ma utilizzeremo strumenti disponibili su Internet. Il sito Web per la creazione del codice QR scelto da noi è qr-code-monkey.com. Un semplice sito Web che consente di creare codici QR modificandone lo stile per renderlo più bello esteticamente.

* + 1. **Come creare un Pattern**

Per creare un pattern abbiamo utilizzato il seguente sito web:

<https://au.gmented.com/app/marker/marker.php>

Una volta entrati per generare un pattern corretto basta inserire le seguenti impostazioni e una volta fatto sarà possibile salvarlo.

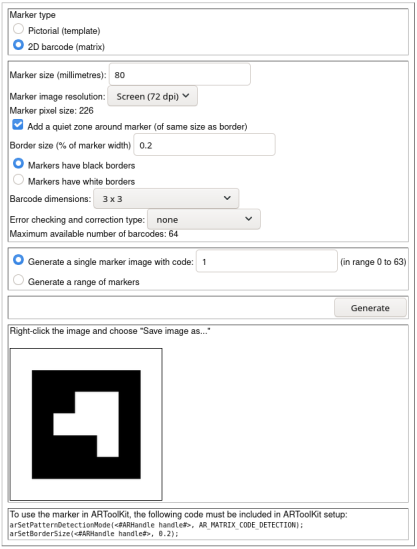


Figura 4 - Pattern Creator

* + 1. **QR Code Floor**

Noi avremo un QR Code all’inizio del quarto piano che porterà direttamente al nostro sito, simile al seguente:

****

Figura 5 - QR Code 4 Floor

* 1. **Database**

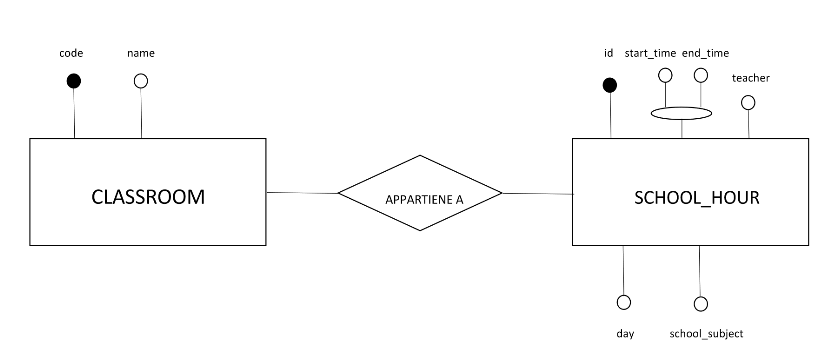


Figura 6 - Database

* 1. **Creazione del database**
     1. **Introduzione**

Come prima cosa, bisogna ovviamente creare il database **School\_Hours** per riuscire ad inserire tutti gli orari dell’anno scolastico. Per farlo è molto semplice.

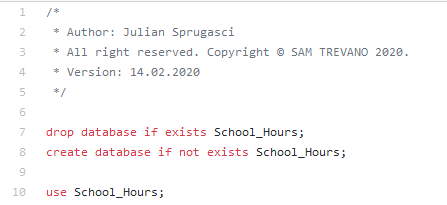
* + - 1. **Codice**

Figura 7 - Create Database Code

* + 1. **Entità Classroom**
       1. **Introduzione**

Questa entità ha il compito di racchiudere tutte le classi del quarto piano attraverso il nome dell’aula e il codice. Successivamente questa entità verrà poi collegata a **School\_Hour** per assegnare all’aula corrispondente il proprio orario scolastico.

* + - 1. **Progettazione**

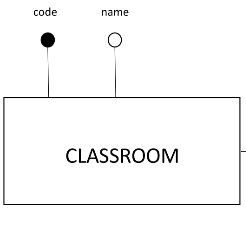


Figura 8 - Classroom Entity

**Attributi**

|  |  |
| --- | --- |
| **Code:** | Codice univoco che rappresenta e distingue ogni aula dalle altre. |
| **Name:** | Nome dell’aula in questione. |

* + - 1. **Codice**

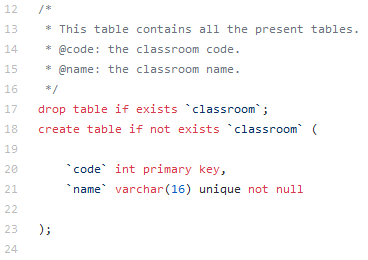


Figura 9 - Classroom entity code

* + 1. **Entità School\_Hour**
       1. **Introduzione**

Questa entità ha il compito di racchiudere tutti gli orari del quarto piano attraverso vari parametri. Successivamente questa entità assegnerà alla entità **Classroom** l’orario corrispondente.

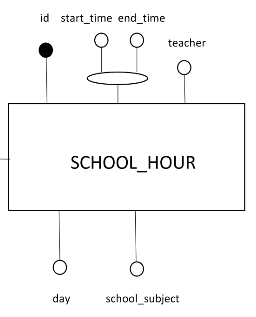
* + - 1. **Progettazione**

Figura 10 - School\_Hour Entity

**Attributi**

|  |  |
| --- | --- |
| **Id:** | Id univoco per identificare l’orario scolastico. |
| **Start\_time:** | Orario di inizio della lezione in questione. |
| **End\_time:** | Orario di fine della lezione in questione. |
| **Teacher:** | Docente assegnato. |
| **Day:** | Giorno della lezione. |
| **School\_subject:** | Materia scolastica. |

* + - 1. **Codice**

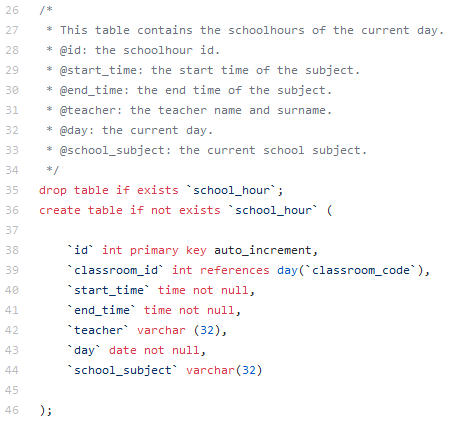


Figura 11 - School\_Hour entity code

* 1. **Selenium**

Selenium è un Framework di test sviluppato in Java che permette di interagire con diversi browser Web. In effetti permette di realizzare dei test dinamici che facilitano dei test funzionali e quindi non richiedono l’esecuzione di un software. Per realizzare i nostri test abbiamo utilizzato l’estensione che si chiama Selenium IDE che può essere installata su qualsiasi browser.

* + 1. **Estrapolazione dati (Scraper)**

Per estrapolare i dati dal sito abbiamo creato uno script chiamato Scraper. Questo è uno script molto potente che permette in generale di estrapolare qualsiasi dato desiderato da qualsiasi sito possibile immaginabile. Per realizzarlo ci siamo basati su un esempio fornito dal nostro professore, però modificandone alcune parti.

* + 1. **Assegnazione del browser Firefox**

Il browser che abbiamo utilizzato per effettuare i vari test è **Mozilla Firefox**, quindi nello script abbiamo dovuto specificare quale driver utilizzare. Per farlo abbiamo solo dovuto scrivere la seguente riga di codice:



Figura 12 - Scraper.py

* + 1. **Mascheramento del browser**

Ogni volta che avviamo il nostro Scraper questo aprirà il browser Google Chrome, andrà sul sito dell'orario della scuola ed eseguirà i vari test. Per nascondere la finestra del browser e evitare di

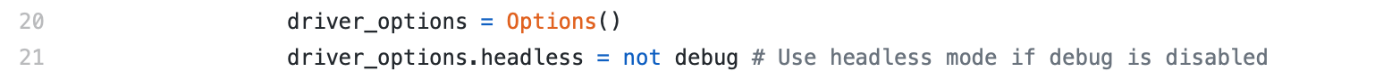
mostrare all’utente le procedure eseguite dallo Scraper, è possibile mascherare il tutto aggiungendo i seguenti parametri:

Figura 13 - Scraper.py

Il metodo **Options** permette di aggiungere delle opzioni aggiuntive al Web driver Selenium. Successivamente l’attributo **headless** permette di nascondere la finestra del browser.

* + 1. **Ricerca e recupero dati dell’orario scolastico**

Il metodo **get\_timetable** permette di ricercare e recuperare dall’orario scolastico le lezioni di una determinata classe.

Figura 14 - Scraper.py

Il parametro **classroom\_name** è una stringa che corrisponde al nome della classe.

* + 1. **Aggiunta di un sito web**

Per raggiungere un sito web dobbiamo fare una richiesta GET con webdriver di Selenium.

Figura 15 - Scraper.py

Il parametro **self.url** corrisponde al URL del orario scolastico. Esso viene passato quando si istanzia la classe Scraper.

* + 1. **Verifica caricamento degli elementi**

Questo comando permette di verificare che un determinato elemento sia caricato correttamente. All’interno del metodo **loaded\_check** bisogna mettere l’id dell’elemento che disideriamo caricare. In questo caso l’id corrisponde al link interno “corsi”.

Figura 16 - Scraper.py

* + 1. **Ricavare elementi HTML**

Per potere ricavare un elemento HTML con Selenium bisogna utilizzare il metodo: **find\_element\_by\_id** che permette di trovare un elemento con un determinato id.

Per cliccare diversi elementi di un sito con Selenium bisogna utilizzare il metodo **click**.

Ad esempio questa linea di codice permette di cliccare il link interno “corsi”.

Figura 17 - Scraper.py

* + 1. **Scrittura nome della classe**

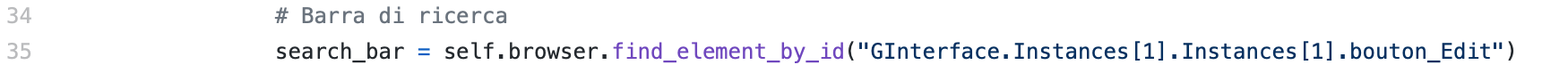
Per potere scrivere il nome della classe all’interno della barra della ricerca abbiamo utilizzato il seguente codice. Esso permette di selezionare la barra di ricerca e di cercare l’orario di una determinata classe.

Figura 18 – Scraper.py

* + 1. **Cancellazione barra di ricerca**

Il metodo **clear** permette di cancellare il contenuto presente all’interno della barra di ricerca (ricerche precedenti rimaste nella barra).

Figura 19 - Scraper.py

* + 1. **Scrittura all’interno della barra**

Il metodo **send\_keys** permette di scrivere all’interno della barra di ricerca.

Figura 20 - Scraper.py

Il parametro **classroom\_name** corrisponde al nome della classe che si vuole cercare. Questo parametro viene passato dal metodo **get\_timetable.**

* + 1. **Avvio della ricerca**

Questa linea di codice permette allo Scraper di eseguire la ricerca attraverso i parametri inseriti nella barra di navigazione del sito degli orari scolastici.

Figura 21 - Scraper.py

* + 1. **Parsing dei dati**

Una volta che abbiamo scelto la classe dovevamo selezionare la tabella dell’orario e iniziare a “parsare” i dati. Per farlo abbiamo aggiunto queste linee di codice dove utilizziamo la libreria **beautiful soup**.

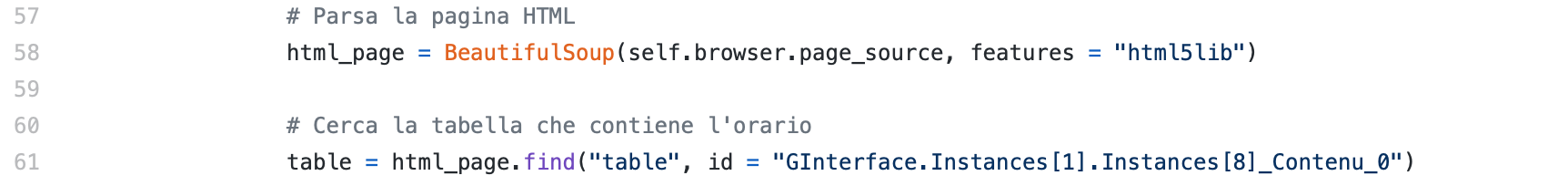


Figura 22 - Scraper.py

All’interno del metodo **BautifulSoup** possiamo aggiungere i seguenti parametri:

|  |  |
| --- | --- |
| **page\_source** | Attributo che permette di acquisire il codice HTML di qualsiasi pagina web. |
| **features** | Parametro che definisce quale libreria viene utilizzata per catturare i dati all’interno della pagina web. |
| **html5lib** | Libreria per l’analisi del HTML. Utilizziamo questa libreria per catturare i dati all’interno della pagina web. |

Con il metodo **find** possiamo andare a cercare nella pagina web un determinato tag con un determinato id.

Prima di “parsare” i dati bisogna controllare se la tabella cercato contiene dei dati, questo perché potrebbe essere una settimana di vacanza dove non sono presenti lezioni. Inseguito per potere selezionare gli elementi della tabella dobbiamo cercare tutti tag **tr** che contengono dei tag **td** o th. Per farlo abbiamo aggiunto la seguente linea di codice.

Figura 23 - Scraper.py

Con il metodo **find\_all** possiamo catturare un tag che presente più volte all’interno della pagina web. Inseguito abbiamo pensato anche in quale modo salvare le varie informazioni all’interno di un file JSON.

* + 1. **Struttura del nostro JSON**

Immagine che contiene screenshot

Descrizione generata automaticamentePer creare la seguente struttura abbiamo creato e utilizzato le seguenti variabili:

Figura 24 - JSON Structure

|  |  |
| --- | --- |
| **result** | Dizionario che contiene gli orari scolastici di una classe. Al suo interno ci sono i seguenti elementi:   * + **is\_holiday** serve per definire se non ci sono lezioni.   + **timetable** contiene i dati dell’orario scolastico. |
| **day** | Dizionario che contiene il giorno in forma testuale e le materie del giorno Al suo interno ci sono i seguenti elementi:   * + **day** corrisponde alla data del giorno.   + **subjects** contiene le materie del giorno. |
| **subjects** | Array che contiene le materie del giorno. |
| **lesson** | Dizionario che contiene i dati della lezione. Al suo interno ci sono:   * + **time** corrisponde all’ora della lezione   + **lesson** corrisponde al nome della lezione   + **teacher** corrisponde al nome dell’insegnante   + **classes** corrisponde al nome della classe |

* + 1. Immagine che contiene screenshot

       Descrizione generata automaticamente**Parte codice importante**

Figura 25 - Scraper.py

|  |  |
| --- | --- |
| **Riga 93** | Il rimo ciclo corrisponde ad ogni elemento **tr** della tabella. |
| **Riga 95** | Il secondo ciclo corrisponde ad ogni elemento **td** della tabella. |
| **Riga 97 - 103** | La variabile **first\_time** permette al primo ciclo di inserire all’interno del dizionario **result** all’elemento **timetable** il giorno della settimana. Per i cicli successivi fino al giorno successivo si deve inserire i dati catturati all’interno del dizionario **lesson**. |
| **Riga 104-126** | Viene inserito per ogni elemento del dizionario **lesson** i dati dell’orario scolastico. |

**Alcuni metodi utilizzati per questo codice sono i seguenti:**

|  |  |
| --- | --- |
| **has\_attr** | Permette selezionare un tag con un determinato attributo (class, id, style, …) |
| **text.Strip** | Permette di selezionare il contenuto di un tag. |
| **append** | Permette aggiungere degli elementi ad un array. |
| **parse\_classe** | Metodo creato da noi per potere formattare nel modo corretto la classe. All’interno di questo codice abbiamo realizzato un’espressione regolare che corrisponde al testo che vogliamo scrivere all’interno del dizionario classe. |

* 1. **Flask**

Flask è un mini-framework in Python che viene utilizzato per lo sviluppo web e permette di creare dei propri siti web in modo dinamico e interattivo.

Abbiamo creato un file che si chiama app.py che sarà l’applicazione **flask**. All’interno per definire che si tratta di un **flask** dobbiamo importare la libreria **flask** e aggiungere la seguente linea di codice.

Figura 26 - App.py

Inseguito bisogna definire la app routing che viene utilizzato per mappare l’URL specifico con la funzione associata che intende svolgere alcune attività (struttura simile al MVC).

Nel nostro caso l’URL è associato alla funzione aule che permette di stampare l’orario di una determinata classe. Sì può anche aggiungere come parametro il tipo di richiesta che si vuole fare. In questo caso utilizziamo il metodo GET che consiste nell’accordare all’indirizzo della pagina web i diversi parametri contenenti i dati che si vogliono trasmettere. In questo caso vogliamo trasmettere se mostrare i dati della settimana corrente o quella successiva.

**@app.route('/aule', methods=['GET'])**

Per ricavare i parametri che sono presenti nel URL bisogna usare il metodo **request.args.get**.

* 1. **Sito Web**

Questa sezione è dedicata alla spiegazione del nostro sito web. Il nostro sito web è diviso in due parti:

* L’interfaccia dedicata all’utente dove potrà utilizzare lo scanner oppure leggere la nostra guida ufficiale.
* L’interfaccia dedicata agli amministratori del sistema.
  + 1. **Interfaccia admin**
       1. **Accesso alla interfaccia admin**

Per accedere alla nostra pagina admin bisogna scrivere nel tab url del browser il seguente indirizzo:



Figura 27 - Nav bar

* + - 1. **Spiegazione interfaccia**

Una volta entrati apparirà la seguente schermata:

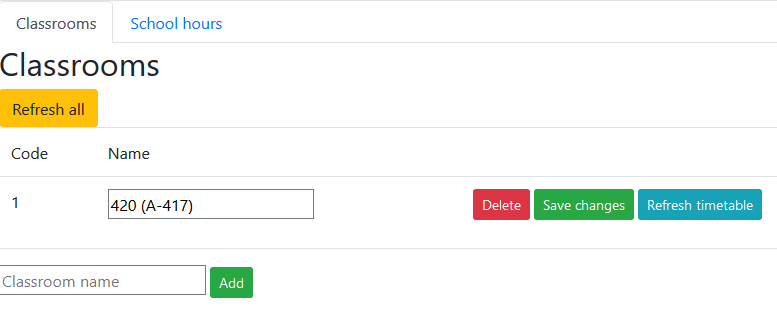


Figura 28 - admin interface

* + - * 1. **Sezione Classrooms**

La sezione Classrooms serve per gestire le varie classi.

In questo caso è presente una singola aula (420 (A-417)). Ne noi volessimo aggiungerne altre bisogna semplicemente inserire il nome della classe nel seguente formato:

**nome (nome\_specifico)**

Una volta aggiunto il nome cliccare il bottone **Add** per aggiungerla.

Dopo aver seguito i seguenti passaggi per ogni aula comparirà un piccolo menu personale che permette di gestire la classe stessa.



Figura 29 - Classroom buttons

|  |  |
| --- | --- |
| **Delete:** | Permette di rimuove l’aula in questione. |
| **Save changes:** | Una volta modificato il nome dell’aula permette di salvare le varie modifiche avvenute su essa. |
| **Refresh timetable:** | Aggiorna gli orari delle lezioni della settimana attuale. |
| **Refresh all:** | Stessa cosa di **Refresh timetable** ma di tutte le aule presenti nella lista. |

* + - * 1. **Sezione School hours**

La sezione School hours permette di gestire tutti gli orari scolastici di tutte le aule presenti nella lista della sezione Classrooms.

Una volta inserita l’aula desiderata e premuto refresh della pagina dopo circa un secondo usciranno gli orari della settimana della aula desiderata. Nel nostro caso abbiamo scelto l’aula **420** e questo è stato il risultato:

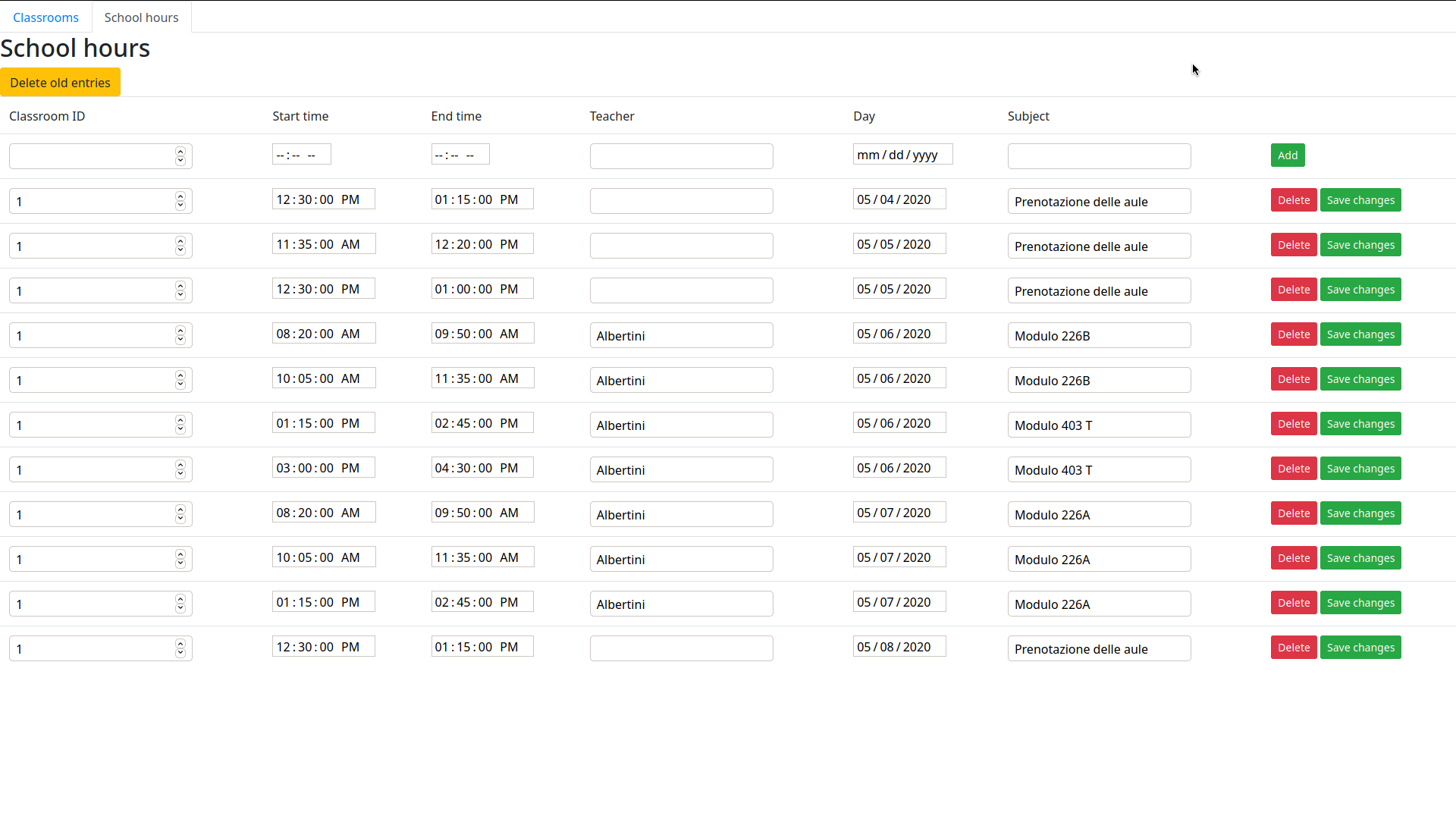


Figura 30 - School hours interface

* + 1. **Guida utente ufficiale**

Nella interfaccia utente è possibile raggiungerla attraverso il seguente link:



Figura 31 - Nav bar 2

* + - 1. **Introduzione**

In questa breve guida verrà spiegato in breve come utilizzare la nostra applicazione in modo semplice e veloce.

* + - 1. **Passaggi**
         1. **Passo 1**

Come prima cosa assicurarsi che siamo sulla homepage del sito ufficiale.

È possibile accedersi nei seguenti modi:

* Andando sul nostro sito [**<indirizzo-sito>**](https://45.63.101.245/)
* Tramite il QR Code appeso al quarto piano
  + - * 1. **Passo 2**

Ora che ci troviamo all'interno del menu bisogna andare in fondo e cliccare **Prova ora** per testare il QR Code Scanner.



Figura 32 - Test now

* + - * 1. **Passo 3**

Nel passo tre bisogna semplicemente accettare al browser il permesso di accedere alla vostra webcam del pc/tablet/telefono.

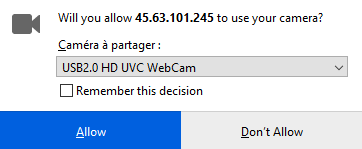


Figura 33 - Webcam Permission

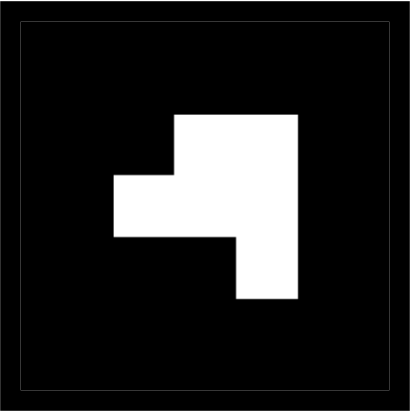
* + - * 1. **Passo 4**

Figura 34 - Pattern Examble

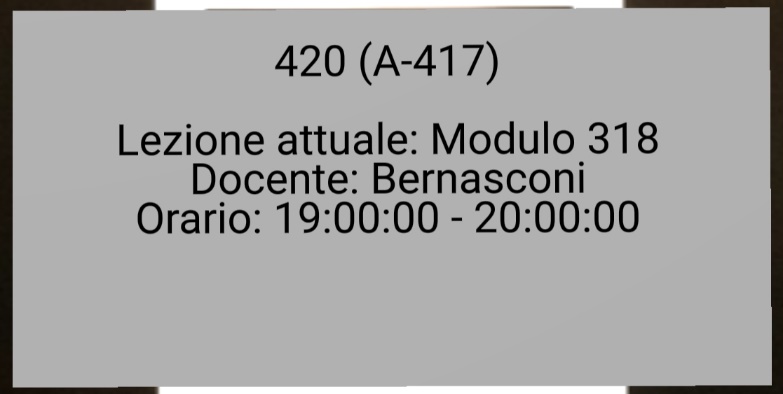
Infine puntando al pattern dell’aula desiderata apparirà un popup con al suo interno l’orario attuale della materia che si sta svolgendo nell’aula in quel momento.

Figura 35 - AR Popup

* + - * 1. **Passo 5 (facoltativo)**
  1. **Installazione server**

Questa parte si occupa di installare il server e tutto il necessario per far funzionare l’applicazione.

* + 1. **Requisiti**

Come prima cosa bisogna dare un’occhiata ai requisiti e controllare che si hanno tutte le cose elencate sotto:

* **Sistema operativo:** Ubuntu o derivate con sistema grafico.
* **Connessione internet stabile.**
  + 1. **Come avviare lo script**

Per avviare lo script bisogna semplicemente inserire il seguente comando nel terminale:

https://cdn.discordapp.com/attachments/688032896077791273/708249237082603570/Screenshot_20200508_112301.png

Figura 36 – Terminal

* + 1. **Spiegazione script install\_ubuntu.sh**

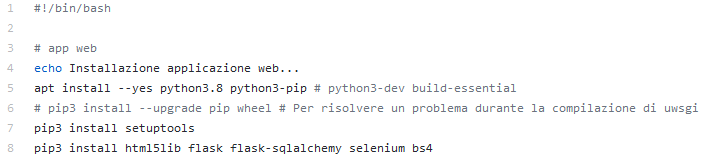


Figura 37 - script.sh

La prima parte installa il necessario per far funzionare l’applicazione web. Precisamente installa **python3.8** e **pip**(gestore librerie python). Una volta fatto ciò installa **setuptools** che si tratta di una libreria base utilizzata dalle altre librerie. Infine installa le librerie necessarie di python.

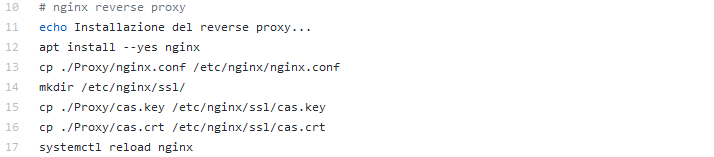


Figura 38 - script.sh

La seconda parte permette di installare il **reverse proxy**. Una volta completata tutta l’istallazione, il server è pronto per essere utilizzato correttamente.

* + 1. **Spiegazione script uninstall\_ubuntu.sh**

Questo script permette semplicemente di fare le operazioni inverse di quello precedente ovvero disinstallare il server.

* 1. **Struttura generale**

Questo capitolo spiega la struttura generale dell’applicazione. La struttura è composta nel seguente modo:

* **Web server:** Permette di fornire il sito web agli utenti
* **Reverse proxy:** Permette di fornire un accesso sicuro attraverso https.
  1. **Web server**

L’applicazione web è stata sviluppata in python utilizzando il micro web framework Flask. Come web server utilizziamo quello di sviluppo integrato con Flask invece di uno apposito (ad esempio uWSGI), siccome abbiamo avuto problemi a farlo funzionare. Il web server integrato regge un traffico piccolo, ma per ambienti di produzione è meglio usarne uno più performante.

* + 1. **Lista file**
* **app.py:** Contiene l’inizializzazione dell’applicazione di Flask.
* **bgtasks.py:** Contiene dei metodi per eseguire della azioni in background.
* **classroom.db:** File del database di SqlLite.
* **db.py:** Contiene le definizioni dei modelli del database.
* **flask\_config.py:** Contiene la configurazione di Flask.
* **geckodriver:** Driver che permette all’applicativo Selenium di interagire con il browser.
* **routes.py:** Definisce i percorsi URL del sito(controller).
* **run.py:** Punto di entrata dell’applicazione web.
* **scraper.py:** Estrae gli orari dal sito della scuola.
* **static:** Questa cartella contiene le risorse come: script, css,js,… per il sito web.
* **templates:** Contiene le pagine html del nostro sito web.
  1. **Reverse Proxy**

Il reverse proxy fornisce una connessione sicura HTTPS con l'applicazione web. Questo è stato necessario perchè AR.js richiede una connessione sicura tra il server e il client. Come software abbiamo usato Nginx. Il file di configurazione di Nginx è situato nella cartella /Proxy/nginx.conf. Nella stessa cartella sono presenti il certificato TLS e la chiave privata necessari all'uso di HTTPS. Nella repository è già presente un certificato di prova (**Attenzione!** Questo certificato è solo di prova, sostituirlo con uno ufficiale!).

1. **Test dei requisiti**

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-001  REQ-001 | **Nome:** | Realizzare un sito web per l’applicativo |
| **Descrizione:** | Viene controllato se il sito web è accessibile all’utente e che tutte le pagine web presenti e i link funziona correttamente. | | |
| **Prerequisiti:** | Bisogna avere accesso a internet e avere un dispositivo in grado di collegarsi a internet. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>** e vedere se il sito è accessibile. 2. Controllare che l’applicazione funzione scanner funzioni. 3. Controllare che i vari link del footer funzionano correttamente. | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il sito funzioni correttamente. 2. L’applicazione scanner funzioni correttamente. 3. Ogni link del sito funzioni correttamente e che sia robusto e efficacie. | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-002  REQ-001 | **Nome:** | Il sito deve essere adatto per qualsiasi dispositivo |
| **Descrizione:** | Viene controllato se il sito web è adattato per tutti i dispositivi possibili(laptop/tablet/smartphone). | | |
| **Prerequisiti:** | Bisogna avere accesso a internet e avere il sito web caricato. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>.** 2. Controllare che il sito è adattato correttamente. 3. Provare l’adattamento con tutti i dispositivi. | | |

|  |  |
| --- | --- |
| **Risultati attesi:** | Mi aspetto che:  **Desktop**    **Tablet**    **Smartphone** |

|  |  |
| --- | --- |
| **Funzionamento:** | Funziona correttamente |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-003  REQ-001 | **Nome:** | Deve esserci una guida utente |
| **Descrizione:** | Viene controllato se il sito web comprende una guida in grado di spiegare all’utente come utilizzare il sito web e la funzione di scanner. | | |
| **Prerequisiti:** | Bisogna avere accesso a internet e avere un dispositivo in grado di collegarsi al sito. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>**. 2. Controllare nel link **<indirizzo-sito>/guide** che sia presente una guida. | | |
| **Risultati attesi:** | Mi aspetto che:   1. La guida appare correttamente. | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-004  REQ-001 | **Nome:** | Il sito dovrà essere indipendente. |
| **Descrizione:** | Viene controllato se il sito web è indipendente e funzionante. | | |
| **Prerequisiti:** | Bisogna avere accesso a internet e avere un dispositivo in grado di collegarsi al sito. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>**. 2. Controllare il funzionamento di ogni link. | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il sito sia indipendente. | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-005  REQ-001 | **Nome:** | L’interfaccia e i colori di sfondo dovranno essere adatti a qualsiasi utente |
| **Descrizione:** | Viene controllato se l’interfaccia grafica del sito web è adatta per qualsiasi utente. Dunque non sarà possibile utilizzare contrasti di colori troppo elevati oppure colori non adatti. | | |
| **Prerequisiti:** | Bisogna avere accesso a internet e avere un dispositivo in grado di collegarsi al sito. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>**. 2. Controllare i colori del sito. | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il sito sia adatto per tutti. Questo perché abbiamo utilizzato solamente uno sfondo bianco e giocato con una sfumatura di grigi. | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-006  REQ-002 | **Nome:** | Dovrà essere presente un QR Code per il quarto piano. |
| **Descrizione:** | All’inizio del quarto piano della nostra sezione informatica viene appeso un codice QR che una volta scannerizzato porterà alla nostra homepage ufficiale del sito web. Visto il noto problema del COVID-19 questo test non è possibile effettuarlo ma abbiamo fatto comunque delle simulazioni. | | |
| **Prerequisiti:** | Bisogna avere accesso a internet, avere un dispositivo in grado di collegarsi al sito e aver stampato il codice QR del piano. | | |
| **Procedura:** | 1. Puntare il codice QR con la fotocamera. 2. Scannerizzare il codice e entrare nella homepage del sito. | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il codice QR funzioni correttamente e che porti direttamente al nostro sito senza errori. | | |
| **Funzionamento:** | Non testabile | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-007  REQ-002 | **Nome:** | Il QR Code avrà un colore di sfondo particolare per essere distinto |
| **Descrizione:** | Il codice QR deve avere un colore di sfondo, in questo caso abbiamo scelto il colore verde, per distinguerlo dagli altri pattern creati.  **Premessa: Il codice QR non porta da nessuna parte perché il link del web server è variabile. Come prova abbiamo aggiunto quello della VPN di Claudio.** | | |
| **Prerequisiti:** | Bisogna avere il codice QR del piano. | | |
| **Procedura:** | 1. Creare il codice QR attraverso il sito <qr-code-monkey.com> 2. Creare il codice QR con al suo interno il link del sito impostato. 3. Personalizzarlo con dei colori a propria scelta | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il codice QR sia distinguibile dagli altri. | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-008  REQ-003 | **Nome:** | Ogni aula dovrà avere un pattern univoco per essere riconosciuta |
| **Descrizione:** | Viene testato ogni singolo pattern assegnato ad un aula e viene controllato se mostra correttamente l’orario di scuola attuale dell’aula selezionata. | | |
| **Prerequisiti:** | Bisogna avere creato tutti i pattern per le varie aule | | |
| **Procedura:** | 1. Puntare un pattern di un aula con la fotocamera 2. Visualizzare gli orari attraverso il popup virtuale | | |
| **Risultati attesi:** | Mi aspetto che:   1. Una volta puntato un pattern riesca a mostrare gli orari correttamente   **Test sull’aula 420(A-417)**  https://cdn.discordapp.com/attachments/688032896077791273/708373065599287336/Screenshot_20200508-193950_Firefox_Nightly.jpg **Pattern Risultato** | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-009  REQ-003 | **Nome:** | Il pattern non dovrà avere delle forme complesse |
| **Descrizione:** | Ogni pattern creato per le aule non dovrà avere delle forme complesse. Per farlo abbiamo utilizzato il sito <https://au.gmented.com/app/marker/marker.php> e abbiamo creato i pattern attraverso una matrice 3x3 | | |
| **Prerequisiti:** | Bisogna avere accesso al sito elencato sopra | | |
| **Procedura:** | 1. Accedere al sito <https://au.gmented.com/app/marker/marker.php>. 2. Create un pattern con le impostazioni elencate sopra. 3. Controllare la forma finale. | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il pattern sia facile da capire e semplice come forma finale.   **Risultato** | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-010  REQ-004 | **Nome:** | L’applicativo dovrà avere la funzione di lettura dei pattern |
| **Descrizione:** | La nostra applicazione deve disporre di una funzione di lettura dei pattern delle aule. | | |
| **Prerequisiti:** | Bisogna aver creato lo scanner dei pattern | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/scan**. 2. Puntare i pattern attraverso la telecamera | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il sito riesca a leggere il codice QR correttamene | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-011  REQ-004 | **Nome:** | Lo scanner dovrà essere eseguito attraverso la fotocamera del dispositivo |
| **Descrizione:** | La nostra applicazione deve disporre di una funzione di lettura dei pattern delle aule e tramite la fotocamera del dispositivo leggere il pattern | | |
| **Prerequisiti:** | Bisogna aver creato lo scanner dei pattern e avere un pattern a disposizione | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/scan**. 2. Accettare i permessi del browser per accedere alla fotocamera del dispositivo. 3. Puntare e scannerizzare. | | |
| **Risultati attesi:** | Mi aspetto che:   1. Riesca a visualizzare la seguente schermata:   https://cdn.discordapp.com/attachments/688032896077791273/708372783075426344/Screenshot_20200508-193950_Firefox_Nightly.jpg | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-012  REQ-004 | **Nome:** | Una volta proiettato il risultato dovrà essere possibile cliccare sul popup per mostrare delle informazioni in più riguardanti l’orario |
| **Descrizione:** | Una volta puntato su un pattern l’applicazione se si preme sul popup sarà possibile accedere a più informazioni riguardanti quell’aula. | | |
| **Prerequisiti:** | Bisogna aver creato lo scanner dei pattern | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/scan**. 2. Puntare i pattern attraverso la telecamera 3. Cliccare il popup virtuale | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il popup una volta cliccato mostri delle informazioni aggiuntive per quanto riguarda l’orario legato a quell’aula.   https://cdn.discordapp.com/attachments/607343958477111326/708375747970203758/Screenshot_20200508-195206_Firefox_Nightly.jpg | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-013  REQ-005 | **Nome:** | L’applicativo dovrà mostrare gli orari in AR |
| **Descrizione:** | La nostra applicazione deve riuscire a mostrare gli orari in AR | | |
| **Prerequisiti:** | Bisogna aver creato lo scanner dei pattern | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/scan**. 2. Puntare i pattern attraverso la telecamera | | |
| **Risultati attesi:** | Mi aspetto che:   1. Il sito riesca a mostrare il popup con gli orari | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-014  REQ-005 | **Nome:** | Dovrà essere possibile leggere il database e estrarre l’orario corrente |
| **Descrizione:** | La nostra applicazione deve riuscire a estrapolare gli orari dal sito e leggerli. Per farlo abbiamo utilizzato lo Scraper. | | |
| **Prerequisiti:** | Bisogna aver creato lo Scraper per leggere gli orari | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/admin**. 2. Inserire l’aula desiderata e fare **refresh**.   https://cdn.discordapp.com/attachments/688032896077791273/708379472126279710/unknown.png   1. Andare su School Hours e aggiornare la pagina. Una volta fatto ciò attendere il processo di estrazione degli orari | | |
| **Risultati attesi:** | Mi aspetto che:   1. https://cdn.discordapp.com/attachments/688032896077791273/708379727328575607/unknown.pngIl sito mostri gli orari di quella determinata aula. | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-015  REQ-005 | **Nome:** | Bisogna mostrare gli orari da qualsiasi angolazione si legge il QR Code |
| **Descrizione:** | Bisogna controllare se la lettura del pattern non funzioni solamente in modo frontale ma anche nelle varie angolazioni | | |
| **Prerequisiti:** | Bisogna avere accesso al sito web. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/scan**. 2. Puntare il pattern dalle varie angolazioni | | |
| **Risultati attesi:** | Mi aspetto che:  https://cdn.discordapp.com/attachments/688032896077791273/708383931455504394/Screenshot_20200508-202248_Firefox_Nightly.jpg https://cdn.discordapp.com/attachments/688032896077791273/708383938116321290/Screenshot_20200508-202304_Firefox_Nightly.jpg https://cdn.discordapp.com/attachments/688032896077791273/708383940649418813/Screenshot_20200508-202328_Firefox_Nightly.jpg | | |
| **Funzionamento:** | Funziona correttamente | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test case:**  **Riferimento:** | TC-016  REQ-008 | **Nome:** | L’applicativo dovrà avere un sistema per ricercare le aule |
| **Descrizione:** | Bisogna controllare la funzione gps che permette di trovare il percorso di un aula. Purtroppo a causa del COVID-19 non abbiamo potuto continuare con questa funzione e alla fine non l’abbiamo conclusa | | |
| **Prerequisiti:** | Bisogna avere accesso al sito web. | | |
| **Procedura:** | 1. Accedere al sito **<indirizzo-sito>/gps**. 2. Provare a ricercare un aula | | |
| **Risultati attesi:** | Mi aspetto che:  Purtroppo la funzione non funziona e non è completa | | |
| **Funzionamento:** | Non funziona | | |

1. **Limitazioni**

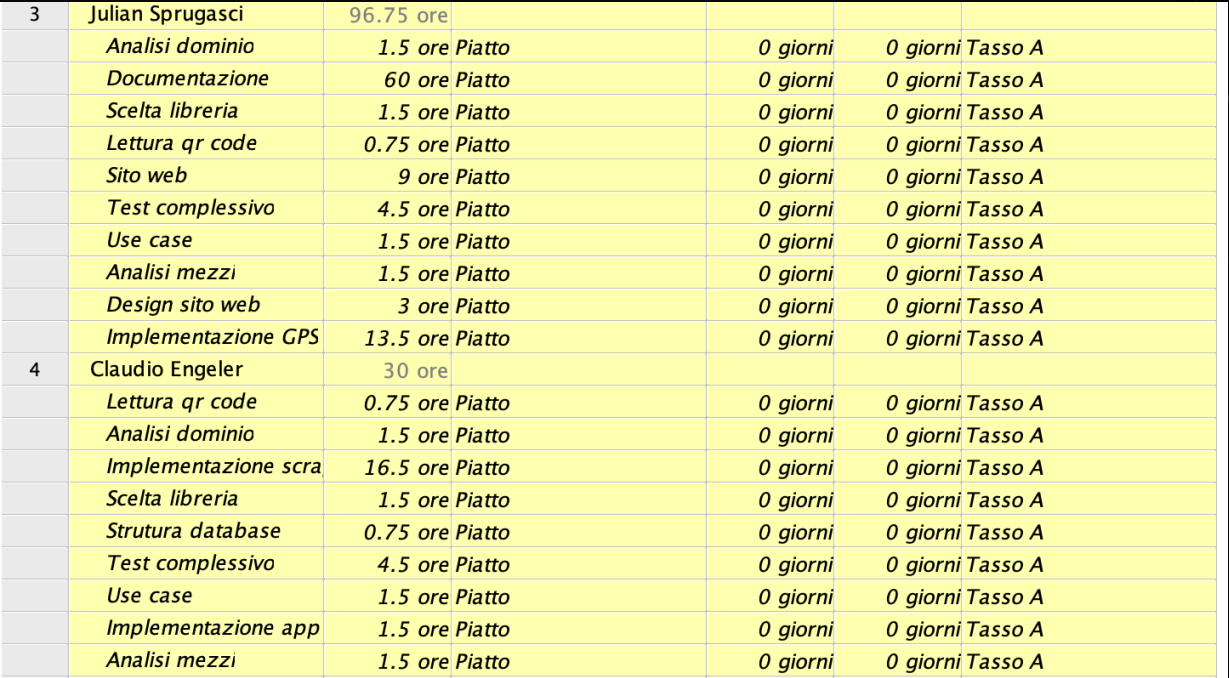
Per quanto riguarda le limitazioni quelle che abbiamo riscontrato sono le seguenti:

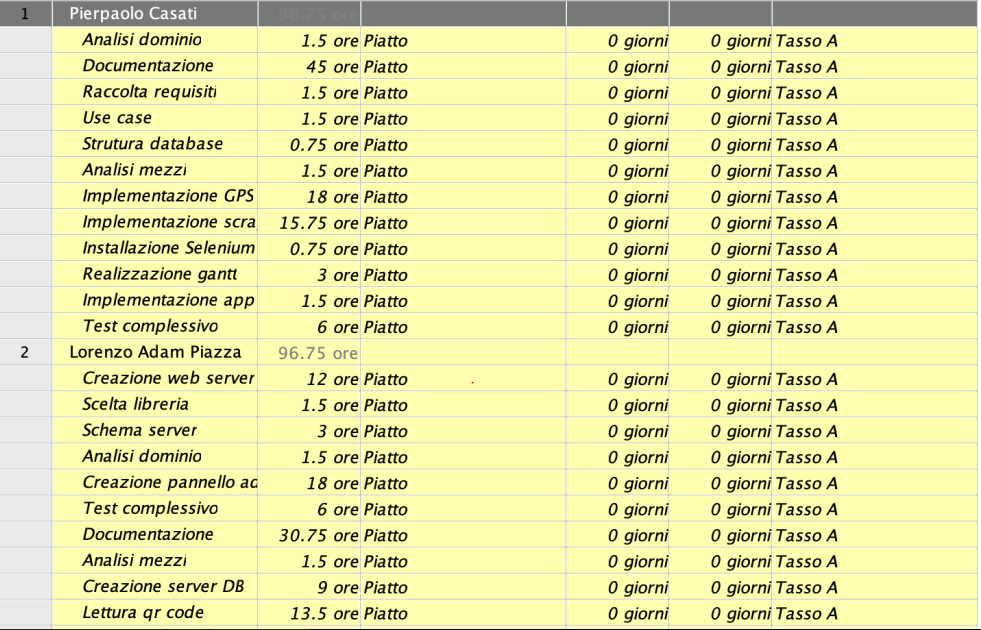
* **Il sito web non è moltosicuroPerPwPerdbjfibhdfbikhjhkvjhv**
* **La pagina admin è facilmente accessibile**
* **Si possono creare solamente max 64 markers**

1. **Consuntivo**

Lo scraping del sito della scuola si è rivelato più complesso del previsto, a causa della complessità del codice HTML. Abbiamo inoltre perso molto tempo

La parte di realtà aumentata, invece, è stata abbastanza veloce. Negli ultimi momenti sono sorti grossi problemi con Selenium, il quale si rifiutava di funzionare in assenza di un'interfaccia grafica. A causa di ciò abbiamo dovuto scartare interamente l'idea di utilizzare Docker per facilitare l'installazione e abbiamo dovuto optare per un approccio tramite script d'installazione.





1. **Conclusioni**

Questo sito offre tutto il necessario per una fantastica esperienza in realtà aumentata direttamente in un ambito scolastico. Siamo felici che abbiamo raggiunto il nostro obbiettivo, nonostante le problematiche causate dal COVID-19. CI è piaciuto molto svolgere questo progetto e abbiamo imparato molto, sia in ambito di sviluppo ma soprattutto come teamwork. Il risultato è migliore di quello che ci aspettavamo.

1. **Sviluppi futuri**

Non abbiamo pensato di effettuare delle modifiche in futuro. Ci è dispiaciuto molto non essere riusciti a svolgere la funzione GPS a causa dell’emergenza e speriamo che in un futuro riusciamo a completarlo.

1. **Allegati**

Elenco degli allegati

* Diari di lavoro
* Codici sorgente
* Prodotto

1. **Indice delle figure**

[Figura 1 - Use case……… ………………………………………………………………………………………………………………………8](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867817)

[Figura 2 - Gantt 9](#_Toc39867818)

[Figura 3 - QR Code Example 11](#_Toc39867819)

[Figura 4 - Pattern Creator 12](#_Toc39867820)

[Figura 5 - QR Code 4 Floor 13](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867821)

[Figura 6 - Database 13](#_Toc39867822)

[Figura 7 - Create Database Code 14](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867823)

[Figura 8 - Classroom Entity 14](#_Toc39867824)

[Figura 9 - Classroom entity code 15](#_Toc39867825)

[Figura 10 - School\_Hour Entity 16](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867826)

[Figura 11 - School\_Hour entity code 17](#_Toc39867827)

[Figura 12 - Scraper.py 17](#_Toc39867828)

[Figura 13 - Scraper.py 18](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867829)

[Figura 14 - Scraper.py 18](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867830)

[Figura 15 - Scraper.py 18](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867831)

[Figura 16 - Scraper.py 18](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867832)

[Figura 17 - Scraper.py 19](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867833)

[Figura 18 – Scraper.py 19](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867834)

[Figura 19 - Scraper.py 19](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867835)

[Figura 20 - Scraper.py 19](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867836)

[Figura 21 - Scraper.py 19](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867837)

[Figura 22 - Scraper.py 20](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867838)

[Figura 23 - Scraper.py 20](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867839)

[Figura 24 - JSON Structure 21](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867840)

[Figura 25 - Scraper.py 22](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867841)

[Figura 26 - App.py 23](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867842)

[Figura 27 - Nav bar 24](#_Toc39867843)

[Figura 28 - admin interface 24](#_Toc39867844)

[Figura 29 - Classroom buttons 25](#_Toc39867845)

[Figura 30 - School hours interface 26](#_Toc39867846)

[Figura 31 - Nav bar 2 26](#_Toc39867847)

[Figura 32 - Test now 27](#_Toc39867848)

[Figura 33 - Webcam Permission 27](#_Toc39867849)

[Figura 34 - Pattern Examble 28](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867850)

[Figura 35 - AR Popup 28](file:///C:\Users\Julian\Documents\GitHub\classroom_ar_scanner\Materiale\Doc\Documentazione_Classroom_Scanner_AR.docx#_Toc39867851)

[Figura 36 – Terminal 28](#_Toc39867852)

[Figura 37 - script.sh 29](#_Toc39867853)

[Figura 38 - script.sh 29](#_Toc39867854)