TournaPro

Applicazioni e Servizi Web

Samuele Burattini - 927104
samuele.burattini@studio.unibo.it

Francesco Dente - 933506
francesco.dente@studio.unibo.it

Francesca Tonetti - 935947
francesca.tonetti@studio.unibo.it

29 gennaio 2021

1 Introduzione

Il progetto nasce dall'esigenza di realizzare una soluzione software web-based che agevoli l'organizzazione di tornei e la gestione degli stessi da parte dei partecipanti o dai gestori della competizione.

L'obiettivo è quello di consentire la creazione di tornei secondo diversi modelli possibili, come ad esempio ad eliminazione diretta o all'italiana e diverse possibili attività (Calcio, Beach Volley, Scacchi, ecc.) L'ausilio software fornito è sia nella fase di preparazione che nella gestione al momento del torneo permettendo a tutti i partecipanti di rimanere sempre aggiornati.

La web application si compone di una interfaccia attraverso la quale i gestori dei tornei potranno approvare i partecipanti per il torneo e memorizzare il risultato delle partite. Inoltre, sarà data la possibilità ai partecipanti di visualizzare il loro stato aggiornato in classifica relativamente ai tornei dei quali fanno parte.

2 Requisiti

Prima di poter procedere con la realizzazione dell'applicazione è stato necessario ottenere un elenco di requisiti utili che rispecchiassero gli interessi dei potenziali utenti finali.

L'utente rappresenta il punto centrale sul quale è stata basata sia la progettazione che successivamente il design, in accordo con le moderne tecniche che valorizzino la User Experience.

Data la difficoltà nell'intervistare dei potenziali utilizzatori, sono stati definiti, tramite la tecnica di Personas e Scenarios, alcuni modelli che rappresentassero

un possibile gruppo di utenti di riferimento per l'applicazione. A ciascuna personas è stato assegnato uno scenario preciso che ricostruisse in maniera dettagliata una particolare situazione d'uso ipotetica.

Riportiamo di seguito le personas individuate con i loro scenarios.

- **Giovanni**

  Ex giocatore di pallavolo, 32 anni, che nella stagione estiva si diletta nel Beach Volley. In seguito ad una scommessa fatta con il proprio vicino di ombrellone, è nata la richiesta di poter programmare le partite domenicali, tenendo traccia dei punteggi, in modo da decretare il vincitore finale.

- **Maria**

  Giovane amministratrice dei profili social del Centro Giovanile della Parrocchia nella quale fa parte, organizzatrice di eventi ed attività come il periodico torneo di Burraco al quale sono iscritte sempre più persone. Per questo Maria vorrebbe un sistema affidabile per mantenere i punteggi delle partite, senza rischiare di fare errori, assegnando così ai vincitori i premi in palio.

- **Marco**

  Giocatore amatoriale di calcetto, 25 anni, studente di Giurisprudenza che non sempre riesce a stare al passo con i punteggi passati tra i compagni nel gruppo di Whatsapp di cui fa parte. Marco vorrebbe conoscere le date delle partite in modo da sapere quando poter partecipare senza rischiare di perdere degli appuntamenti.

- **Franco**

  Proprietario di un rinomato negozio di giochi da tavolo, appassionato di Magic. Il numero di clienti che partecipano ai tornei organizzati da Franco sono sempre di più, per questo vorrebbe usufruire di una qualche piattaforma che lo aiutasse automatizzando il processo di iscrizione e calcolo dei punti con annessa graduatoria, in modo che non ci siano più limitazioni per il numero di clienti partecipanti.

- **Giorgia**

  Giovane 27'enne, barista di un famoso chiosco che da circa un paio d'anni organizza tornei di beer-pong. Vorrebbe farsi notare dai titolari del chiosco, per questo desidererebbe avere un modo facile e veloce che l'aiuti ad inserire i giocatori di un torneo, gestire l'ordine dei turni dei giocatori, mantenerne i punti e decretare i vincitori finali ai quali offrire una pinta di birra.

- **Mirco**

  Neo maggiorenne e appassionato degli scacchi. Mirco dopo aver ricevuto un libro per il suo compleanno sui migliori scacchisti, vorrebbe scoprire un'applicazione che lo aiuti a trovare i tornei nella sua zona per potersi

misurare con altri giocatori appassionati come lui del mondo degli scacchi, conoscendo nuove persone sia del mondo amatoriale che professionale.

Dall'analisi dei potenziali utenti così definiti sono stati individuati i seguenti requisiti.

2.1 Funzionali

Il sistema si occupa di gestire gli aspetti di gestione di un torneo sia dal punto di vista di chi partecipa che di chi organizza. Dato che non esiste una separazione rigida tra questi due ruoli abbiamo permesso a ciascun utente registrato di poter essere contemporaneamente sia organizzatore che partecipante a patto che non si partecipi ad un torneo di cui si è anche organizzatori per evitare conflitto di interessi.

Dall'analisi degli scenari d'uso possibile è emerso che spesso l'applicazione può essere utile anche solo per l'organizzatore e richiedere quindi di poter inserire partecipanti non registrati nei propri tornei. Questi ultimi ovviamente non godranno dei benefici che l'app fornisce ma non saranno costretti ad utilizzare uno smartphone per fare un account sul sito.

Utente Gli utenti registrati possono:

- Inserire informazioni personali per essere riconoscibili dagli altri utenti
- Fornire dei contatti per poter essere raggiungibili per comunicazioni esterne all'app
- Visualizzare le informazioni relative ai tornei
- Creare squadre con altri utenti per partecipare a tornei insieme
- Richiedere di iscriversi a dei tornei
- Ritirarsi da un torneo a cui si sono iscritti
- Visualizzare il loro stato di avanzamento nei tornei a cui partecipano
- Creare tornei per permettere ad altri utenti di partecipare
- Cancellare un torneo creato
- Approvare o Rifiutare le richieste di iscrizione ai loro tornei
- Aggiungere altri utenti come organizzatori di un torneo per agevolare le operazioni
- Inserire i risultati delle partite

Sistema Oltre ai requisiti legati alle operazioni degli utenti sono stati individuati alcuni requisiti del sistema, tra i quali:

- Produrre una notifica ogni qualvolta venga svolta una azione significativa come ad esempio partecipare ad un torneo, l'inizio di un match o la sua conclusione, l'aggiunta di un membro ad una squadra ecc.
- Verificare le condizioni in cui un utente deve ricevere un achievement per sbloccarlo in modo da rendere l'esperienza degli utenti più divertente e invogliarli a partecipare a nuovi tornei.

2.2 Non Funzionali

Il requisito non funzionale principale per la nostra web-app è la scelta di progettazione **Mobile First**, secondo cui l'approccio applicato al web design è pensato principalmente per l'ottimizzazione dell'applicazione su dispositivi mobili.

Questo requisito è stato individuato dall'analisi degli scenari d'uso principali. Spesso i tornei si gestiscono "sul campo" anche a volte con match della stessa competizione disputati in luoghi fisici diversi. Abbiamo ipotizzato che organizzatori e partecipanti non avessero facile accesso ad un PC quindi la scelta più ovvia è quella di rendere l'app semplice da navigare con lo smartphone.

Per questo motivo abbiamo scelto di concentrarci, a partire dal design e dai mockups, sulla realizzazione di pagine adatte ai dispositivi mobili e applicando successivamente le modifiche necessarie per gestire la visualizzazione su schermi di maggiori dimensioni come tablet e desktop immaginando che nella fase preliminare al torneo può essere comodo per utenti e organizzatori rivedere la situazione di iscrizione al torneo dal proprio computer o nel caso in cui si disponga di una postazione fissa in sede di torneo per registrare ufficialmente tutti i risultati.

3 Design

La progettazione dell'intero sistema si focalizza sul modello iterativo basato sullo **User Centered Design** il cui focus riguarda utenti "virtualizzati". Come introdotto nel paragrafo precedente, sono state individuate sei **Personas**, con annessi **Scenarios**, per capire a quali funzionalità l'applicazione dovesse rispondere.

Il team ha affrontato il progetto facendo uso della metodologia **AGILE** sin dalle prime fasi di Design fino a quelle di sviluppo effettivo del software. Si è rivelata una metodologia molto utile per tenere traccia dei progressi anche nonostante la costretta distanza fisica tra i membri del gruppo. Le riunioni periodiche e gli obiettivi settimanali si sposano molto bene con lo sviluppo di un progetto di questa natura.

3.1 Design delle interfacce

Dal punto di vista del design delle interfacce si è cercato di allinearsi il più possibile ai principi **KISS** e **"Less Is More"** con l'obiettivo di realizzare una interfaccia semplice ed intuitiva nell'utilizzo da parte degli utenti. Infine, per

rendere le interfacce utilizzabili da qualsiasi dispositivo e migliorare la User Experience, sono stati applicati i principi che stanno alla base del responsive design.

Inizialmente ci siamo concentrati sullo sviluppo dei mockup attraverso l'uso di **Adobe XD**, software estremamente professionale per la prototipazione, che ci ha permesso di creare agilmente tutti gli elementi delle interfacce, esprimendo i loro flussi, la struttura di navigazione e l'architettura delle informazioni, (**clicca qui** per visualizzare i mockup live).

Terminata la progettazione delle interfacce, abbiamo scelto di condividere, i nostri mockup interattivi con una decina di utenti reali appartenenti a diverse categorie professionali. Questo ci ha permesso di raccogliere una serie di informazioni, commenti e suggerimenti riguardo alla User Experience (UX design). Per la raccolta delle opinioni ci siamo aiutati creando un questionario, (**clicca qui** per accedere al link del questionario).

3.2 Design Architetturale

Il sistema, come ogni applicazione web di base, si compone di un servizio backend che mantiene i dati in modo persistente e gestisce la business logic e un'interfaccia frontend per la visualizzazione e interazione lato utente.

Per la comunicazione e l'accesso ai dati persistenti il backend è stato progettato con un'API REST-ful in modo da godere dei vantaggi tipici di questo approccio:

- Indipendenza tra i dati persistenti e i modelli di comunicazione
- Comportamento stateless della comunicazione client-server
- Semplicità di identificatione e accesso alle risorse
- Indipendenza dalle tecnologie adoperate

Inoltre, dato che tra i requisiti del Sistema è emersa la necessità di fornire in tempo reale agli utenti gli aggiornamenti significativi riguardanti lo stato dei tornei a cui essi partecipano, si è scelto di aggiungere un modulo di comunicazione event-based per gestire questo comportamento dato che REST non si sposa bene con questa caratteristica.

4 Tecnologie

Il sistema è stato realizzato utilizzato il **solution stack MEVN** tra i moderni stack di tecnologie Javascript-based per la realizzazione di Single-Page-Applications.

Il sistema si compone quindi di un servizio di backend basato su un server in **Node.js** con **Express.js** come framework di gestione delle richieste e un layer di persistenza implementato da un database non relazionale con motore

MongoDB. Il sistema è poi reso disponibile agli utenti tramite una interfaccia sviluppata tramite **VueJS**.

La scelta di approfondire Vue tra le diverse tecnologie del corso è stata in parte influenzata dal fatto che un membro del gruppo poteva trarne beneficio per degli impegni lavorativi e in parte perché ci sembrava la tecnologia più interessante per approcciarci al mondo della creazione di interfacce basate su componenti, data la crescente tra gli sviluppatori rispetto ad Angular e React.

4.1 Frontend

Oltre a quelle elencate, come supporto alla programmazione lato frontend abbiamo scelto una serie di tecnologie analizzate qui di seguito.

4.1.1 Sviluppo software e testing

Abbiamo scelto di ricorrere all'uso della **Vue CLI** [6] per migliorare la strutturazione e organizzazione del codice lato client, la cui complessità era sempre più elevata. Questo tipo di supporto mira ad essere uno strumento di base tra gli standard appartenenti all'ecosistema di Vue.

La Vue CLI rende estremamente facile l'esecuzione del codice del front-end anche in fase di sviluppo, aggiornando automaticamente il rendering man mano che le pagine vengono costruite e fornendo dei messaggi di errore sufficientemente chiari quando ci sono problemi di compilazione o esecuzione.

Estremamente utile come controparte della Vue CLI è l'estensione per browser Vue DevTools disponibile sia per Firefox che Chrome che permette di riconoscere e navigare l'albero dei componenti aggiungendo una tab alla pagina degli strumenti per sviluppatore nativa dei browser. Questo strumento ci ha aiutato molto in una prima fase a capire come comporre correttamente lo scheletro della visualizzazione nelle pagine e successivamente anche per debuggarne il comportamento dato che permette di visualizzare in modo reattivo anche le proprietà e i dati dei componenti.

4.1.2 Stile

Per quanto riguarda lo stile delle interfacce client, abbiamo scelto di appoggiarci a diversi strumenti tra i quali **BootstrapVue** [1] basato sul toolkit Bootstrap 4 che utilizza **Flexbox**[2] per realizzare il proprio Grid System e che permette la creazione di siti caratterizzati da un layout fluido.

Data la scarsa configurazione dei componenti di BootstrapVue li abbiamo mantenuti solo dove non avevamo necessità di ottenere un particolare stile grafico e abbiamo invece creato molti componenti in modo nativo. Questo ci ha permesso di fare un largo uso di Flexbox direttamente su buona parte delle interfacce, in modo da ottenere un comportamento predicibile e stabile degli elementi al ridimensionamento delle pagine per display con diverse dimensioni, in particolar modo per la modellazione delle card riferite alle interfacce dei tornei e dei team.

Abbiamo avuto modo di apprezzare particolarmente flexbox per la sua facilità d'uso una volta compreso il funzionamento di base e riteniamo che sia un modo molto moderno di scrivere CSS decisamente più pratico rispetto alle soluzioni più "legacy".

Sempre per avvalerci dei benefici delle tecnologie moderne per la creazioni di Applicazioni Web abbiamo scelto utilizzare **SCSS**. Scrivere CSS con questa sintassi risulta incredibilmente più agevole specie se combinato con la caratteristica di Vue che consente di inserire lo stile di un componente all'interno del sorgente del componente stesso e attraverso il modificatore *scoped* fa in modo che le regole scritte per quel componente non valgano nel resto del progetto.

SCSS si è rivelato inoltre fondamentale per gestire la palette di colori attraverso la definizione di variabili dando spazio ad una grande flessibilità e permettendo di modificare lo stile grafico di tutto il sito con facilità.

4.1.3 Gestione dei dati e dello stato

La gestione dello stato sul frontend è stata realizzata utilizzando la libreria axios per realizzare le chiamate di rete dirette verso il server. La libreria è molto semplice da utilizzare, uno standard *de facto* nel campo e ci ha permesso di realizzare un layer di accesso ai dati con facilità tenendo ben separato l'uso delle API dall'interfaccia grafica.

Per alcuni elementi statici e persistenti durante il ciclo di vita dell'applicazione abbiamo deciso di utilizzare **Vuex**[7]. Vuex è una estensione di Vue per lo *State Management*, ovvero un modello per applicazioni a stato centralizzato in grado di mantenere con facilità informazioni tra diversi componenti in una Single Page Application e permette di modificare e accedere ai dati in modo estremamente rigoroso e protetto in modo da non creare situazioni di inconsistenza.

Ci sarebbe piaciuto riuscire ad utilizzare Vuex in modo più intensivo per migliorare anche le performance della nostra applicazione, ma dato che questo avrebbe allungato di molto i tempi di consegna ci siamo limitati ad utilizzarlo per alcuni elementi che si sposavano bene con questo pattern, uno fra tutti la gestione dei dati recuperati dallo storage di sessione per ricordare l'utente loggato.

4.2 Backend

Come già citato il backend è stato realizzato con Node e Express per realizzare la business logic lato server e MongoDB per il layer di persistenza.

È stata la prima volta che realizzavamo un progetto di grosse dimensioni in javascript e utilizzare queste tecnologie ha reso lo sviluppo del server molto rapido nelle fasi iniziali, ma difficoltoso quando la complessità della logica è aumentata.

4.2.1 Persistenza

La persistenza dei dati è stata ottenuta tramite un database non relazionale. Lavorare con questo tipo di database è stata per noi una nuova esperienza e ci sembra di essere riusciti a sfruttare a sufficienza i vantaggi del non dover mantenere uno schema rigido dei dati. Per interfacciarci al database abbiamo utilizzato la libreria **Mongoose**[4]. Si tratta di una delle librerie più utilizzate nel suo campo e consente di creare in codice una rappresentazione dello schema dei dati che rispecchia quello che deve essere mantenuto nel database. Infatti, una delle feature che abbiamo apprezzato di più è stata la possibilità di definire lo schema dei dati tramite Mongoose in modo da poter creare le collezioni quando necessario senza bisogno di altre operazioni di setup.

4.2.2 Autenticazione e Autorizzazione

Per la trasmissione compatta, autonoma e sicura delle informazioni tra client e server, al login viene rilasciato dal server un **JWT (Json Web Token)**[3] per garantire l'identità dell'utente loggato.

Qualsisi richiesta successiva avrà in allegato lo stesso JWT generato in partenza così che l'utente possa accedere a qualsaisi risorsa dell'applicazione consentita da quel preciso token. I token vengono salvati sul browser del client in modo da poter essere recuperati e garantire di nuovo l'accesso all'utente fino alla scadenza dei token stessi.

4.2.3 Comunicazione Real-Time

Per l'implementazione delle funzionalità real-time quali l'invio delle notifiche al sistema, relative alla creazione di nuove squadre, si è deciso di fare uso della libreria **socket.io**[5] presentata a lezione.

5 Codice

Durante lo sviluppo del software si è rivelata di particolare importanza la scrittura di codice scalabile al fine di supportare semplicemente l'aggiunta di nuove funzionalità.

In particolare, lo sviluppo delle notifiche e degli achievement ci ha portato a sviluppare un sistema di eventi confinato nel backend per gestire separatamente la logica delle richieste RESTful e la comunicazione event-based con i client. Con questo sistema, diversi moduli dell'applicazione possono ascoltare eventi di diversa natura (creazione di squadre, risultati di un match disponibili, ecc) e gestirli autonomamente, lasciando il flusso principale libero di gestire le richieste del frontend senza essere bloccato.

Un altro aspetto rilevante nella scrittura del server tramite express è stata la scelta di gestire operazioni comuni a tutte le richieste sfruttando una **pipeline**, creata tramite i middleware di express. Questa strategia è stata utilizzata per la gestione di autenticazione e autorizzazione delle richieste, che come detto in

precedenza è basata su Json Web Token. Nel nostro caso, è stato creato un middleware che prima della logica core, valida il token se presente tra gli header e rende disponibile quest'informazione agli stadi successivi della pipeline. Uno tra questi è il middleware di autorizzazione, che blocca la richiesta con un codice di stato 403 (Forbidden) ed evita che venga eseguita la logica core se l'utente non è stato autenticato.

Per quanto riguarda il frontend, la gestione dello stato globale è stata affidata allo standard de facto per lo state management per VueJs: **Vuex**. Tramite questo tool, sono state memorizzate informazioni da rendere disponibili tra svariati componenti e a diversi livelli di nesting, come ad esempio i dati di autenticazione (user ID e token di autenticazione dell'utente) oppure le attività e gli achievement supportati dal sistema. Questo ci ha permesso di semplificare notevolmente lo scambio tra i componenti e ha reso le loro interfacce di comunicazione più snelle.

6 Test

Le funzionalità del sistema sono state testate dai componenti del team sia su browser **Chrome** che su browser **Firefox** con l'obiettivo di garantire portabilità. Inoltre, oltre all'uso del device fisico è stato opportunamente testato anche su dispositivo mobile in quanto applicazione mobile first. In ogni caso i test hanno voluto verificare che le funzionalità e i task principali funzionassero allo stesso modo in tutti i browser e dispositivi e che la visualizzazione fosse corretta anche su piattaforme differenti.

Le API sviluppate lato server sono state opportunamente testate utilizzando lo strumento **Postman** che ha permesso di verificare che il loro comportamento fosse quello desiderato, prima di procedere con l'integrazione con il front-end e quindi con il test finale relativamente alla funzionalità completa dell'applicazione.

6.0.1 User Experience

Dal punto di vista dell'usabilità dell'interfaccia, come espresso nel paragrafo relativo al design di progetto, il sistema è stato preventivamente sottoposto al testing realizzato attraverso utenti reali e la compilazione di un questionario che, basato sull'**Euristica di Nielsen**, permettesse di verificare e mettere in luce gli aspetti salienti del progetto. Di seguito si riportano alcune considerazioni emerse.

- **Corrispondenza tra sistema e mondo reale**: termini e icone utilizzate nel sistema sono in linea con quelli che caratterizzano i social network.
- **Controllo e libertà**: il numero di operazioni necessarie per portare l'utente al compimento di un task sono ridotte al minimo indispendabile.
- **Consistenza e standard nel sistema**: la definizione iniziale di una gamma di colori da utilizzare è resa uniforme all'interno di tutto il sistema

in modo da rivolgere l'attenzione dell'utente agli elementi principali delle varie interfacce.

- **Flessibilità ed efficienza d'uso**: le funzionalità offerte dal sistema sono semplici per cui non necessitano di scorciatoie particolari o modalità di utilizzo alternative.
- **Design ed estetica minimalista**: il principio chiave del progetto si basa sulle regole KISS che permettono di comporre interfacce chiare e semplici all'uso.
- **Prevenzione dell'errore**: vengono evitate situazioni ambigue per l'utente durante la navigazione nel sistema, che possano portarlo a commettere errori.

7 Usability Test

Una volta terminato, il sistema è stato sottoposto all'attenzione di entrambe le tipologie di utente conivolte in modo da valutare la correttezza dell'uso del sistema.

In questa ultima fase tutti i componenti del team si sono immedesimati nei target user individuati in fase di analisi, testando opportunamente il sistema.

8 Deployment

Il progetto può essere avviato sfruttando il compose file presente nella root del progetto. La configurazione mette in esecuzione le tre componenti principali del progetto (il database, il backend e il frontend). Non avendo a disposizione la versione desktop di docker sulle nostre versioni di windows, si è fatto ricorso a **Docker toolbox** per emulare una macchina docker su Virtual Box. Questo richiede configurazioni leggermente diverse per i due ambienti che sono fornite tramite due diversi compose file da utilizzare in aggiunta a un compose file principale. Il compose file principale è `docker-compose.yml`, mentre i due file specifici per ambiente sono `docker-compose.standalone.yml` e `docker-compose.toolbox.yml`.

8.1 Avvio tramite Docker compose

- Clonare il repository
  git clone https://github.com/francescodente/tourna-pro.git
- Avviare il progetto con docker compose
  docker-compose -f docker-compose.yml -f docker-compose.[environment].yml up
  (eventualmente aggiungere l'opzione –build se è necessario ricompilare le immagini)

9 Conclusioni

Il team si ritiene soddisfatto del lavoro svolto, avendo portato a termine la stragrande maggioranza degli obiettivi che erano stati proposti come requisiti del sistema.

Il prodotto finale è semplice da utilizzare, svolge tutti i compiti che erano richiesti e garantisce una piacevole esperienza d'uso agli utenti sia come engagement che come qualità visiva generale.

Per quanto riguarda le funzionalità che interessano gli utenti, un'evoluzione futura del sistema potrà sicuramente concentrarsi su possibili estensioni quali:

- la possibilità di instaurare delle amicizie virtuali tra utenti registrati che utilizzano il sistema;
- introdurre un sistema di messaggistica tra gli utenti per avere la possibilità di comunicare direttamente tramite l'app;
- introdurre la visualizzazione del valore di ranking per il singolo giocatore o per l'intera squadra in modo da incentivare la competizione

Inoltre, si potrebbe introdurre un'estensione delle funzionalità anche dal punto di vista dei tornei ovvero articolando gli stessi su più fasi, ciascuna delle quali aventi modalità diverse come ad esempio l'organizzazione dei mondiali di calcio.

Il gruppo aveva valutato anche come extra di aumentare la condivisibilità dei tornei tramite l'utilizzo di QR code e migliorare la ricerca dei tornei in base alla geolocalizzazione, purtroppo queste features sono state scartate a causa del tempo necessario allo sviluppo ma arricchirebbero sicuramente il sistema in modo interessante.

9.0.1 Commenti finali

Complessivamente lo sviluppo di questo progetto ha permesso a ciascun membro del team di arricchire il proprio bagaglio di conoscenze tecniche, dando la possibilità a ciascuno di mettersi alla prova con diverse tecnologie proprie dello sviluppo web moderno.

Durante lo sviluppo è stato di grande aiuto la collaborazione continua tra i componenti del gruppo, sempre disponibili nel dare una mano e pazienti nel comprendere eventuali limiti di comprensione da parte dei membri del team.

Non solo, questo progetto è stato utile anche per migliorare le proprie abilità dal punto di vista della programmazione JavaScript pura, in quanto unico linguaggio interpretato fino ad ora visto approfonditamente nella carriera universitaria.

Per quanto riguarda il testing al termine dello sviluppo, siamo consapevoli che sarebbe stato più efficiace conivolgere utenti "reali" come gli stessi chiamati in fase di validazione per il design della User Experience, ma tenendo fissi dei task ben definiti abbiamo trovato modo di testare l'applicazione astraendoci dalle figure di sviluppatori del team ma di comuni utenti futuri che potrebbero voler usare una web app per la gestioni di tornei.

L'idea di questo progetto di fatto è nata da una stessa volontà da parte dei componenti del team in quanto appassionati di sport o di giochi da tavolo e affascinati all'idea di poter realizzare un progetto che fosse utile per loro stessi. Questo ci ha dato modo di sperimentare in prima persona l'utilità e le funzionalità da mettere in campo permettendoci anche di evolvere le funzionalità individuate inizialmente anche in fase di sviluppo specie per quel che riguarda l'esperienza d'uso finale.

Riferimenti bibliografici

[1] Bootstrap-vue. `https://bootstrap-vue.org/`.

[2] Flexbox. `https://css-tricks.com/snippets/css/a-guide-to-flexbox/`.

[3] Jasonwebtoken. `https://github.com/auth0/node-jsonwebtoken`.

[4] Mongoose. `https://mongoosejs.com/`.

[5] Socket.io. `https://socket.io/`.

[6] Vue cli. `https://cli.vuejs.org/`.

[7] Vuex. `https://vuex.vuejs.org/`.