

Relazione Ingegneria del software

Relazione dell'esperienza fatta sviluppando un progetto software con un sistema per lo sviluppo agile in team per il corso di Ingegneria del Software 2019/2020.

Team:

- Filippo Bartolucci 0000838531
- Umberto Case 0000833051
- Matteo Celani 0000834303
- Francesco Cerio 0000832618

Indice

1. Progetto Sviluppato
 - 1.1 Tecnologie Utilizzate
 - 1.2 Software Utilizzati
 - 1.3 Divisione compiti
2. Taiga
 - 2.1 Specifiche del progetto
 - 2.2 Organizzazione User Story
 - 2.3 Organizzazione Kanban
 - 2.4 Definition of Done
 - 2.5 Issue
 - 2.6 Sprint
 - 2.7 Primo Sprint
 - 2.8 Secondo Sprint
 - 2.9 Terzo Sprint
 - 2.10 Quarto Sprint
 - 2.11 Burndown
3. Gitlab
4. Mattermost
5. Sonarqube
 - 5.1 Analisi Primo Sprint
 - 5.2 Analisi Secondo e Terzo Sprint
 - 5.4 Analisi Quarto Sprint
6. Innometrics
 - 6.1 Collector
 - 6.2 Transfer
7. Commenti Finali

1. Progetto Sviluppato

Where MI? L'audioguida turistica del XXI secolo



Where MI è una web app che fornisce un modo semplice e veloce per trovare audio guide sul web.

È pensata per due tipi di utenti:

- Le guide, degli influencer/vlogger che posteranno gratuitamente video guide sul web
- I turisti, in cerca di consigli, cultura e svago.

È composta da una coppia di web application:

- Il browser permette ad un turista di scoprire annotazioni audio o testuali sui punti di interesse presso cui si trova e leggerle a schermo oppure sentire una voce sintetica o umana pronunciarle in un auricolare. Le annotazioni sono divise in tre categorie in base allo scopo per il quale sono state registrate le clip:
 - **What:** l'identità del luogo in questione;
 - **Why:** perché è interessante visitare il punto d'interesse;
 - **How:** clip che forniscono informazioni su come accedere ad un luogo;
- L'editor permette ad una guida di creare annotazioni audio oppure testuali, associarle a locazioni georeferenziate, taggarle secondo numerose caratteristiche e caricarle su un server (nel nostro caso Youtube) che possa essere usato da turisti che visitano quei luoghi di interesse.

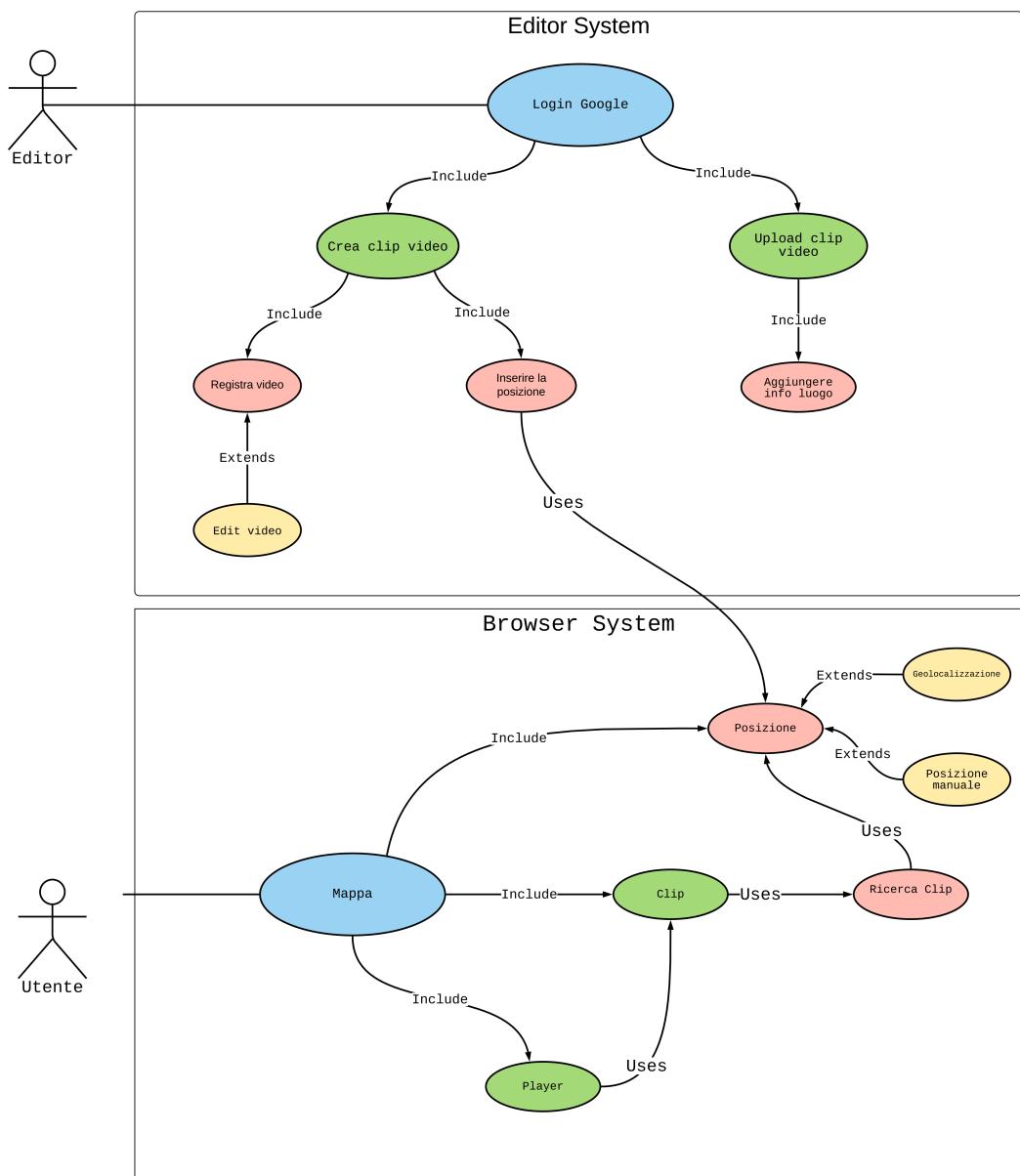


Grafico UML dei casi d'uso della web application.

Diagramma casi d'uso Browser

Caso d'uso	Pre-condizioni	Sequenza di Eventi	Errori	Post-condizioni
GeoLocalizzazione	GPS Attivo	- Cliccare sull'apposito pulsante "La mia posizione" per geolocalizzare la posizione dell'utente.	Se GPS non attivo, non restituirà alcun risultato.	La posizione rilevata viene passata al Set Posizione, comparirà sulla mappa il marker relativo alla posizione dell'Utente.
Posizione Manuale	Nessuna	- Tramite il Drag&Drop si può selezionare una posizione a piacimento sulla mappa. - Tramite la barra di ricerca si potrà definire la posizione tramite la ricerca per indirizzo.		La posizione rilevata viene passata al Set Posizione, comparirà sulla mappa il marker relativo alla posizione dell'Utente.
Set Posizione	GeoLocalizzazione e/o Posizione Manuale deve essere specificata.	Una volta selezionata la modalità con cui rilevare la posizione il marker relativo alla posizione dell'utente genererà l'OLC (Open Location Code) relativo alla posizione, indispensabile per la ricerca delle Clip.		Viene attivata la ricerca delle Clip nelle vicinanze tramite l'OLC generato dal SetPosizione.
Ricerca Clip	Set Posizione deve aver definito la posizione.	- Automaticamente, dopo aver definito la posizione, sulla mappa compariranno tutte le clip esistenti in un certo raggio di distanza dalla posizione rilevata.	- Superato limite d'utilizzo giornaliero YouTube API Key - Nella posizione rilevata non sono presenti Clip - Formato URL Clip diverso dal formato richiesto dalla nostra WebApp.	- Viene ordinata, in base alla distanza, la playlist di clip riproducibile tramite il relativo player Audio. - Viene riempita la sezione Vicino A Me, con le clip nelle strette vicinanze.

Caso d'uso	Pre-condizioni	Sequenza di Eventi	Errori	Post-condizioni
Player	- Ricerca Clip deve aver trovato almeno una clip - L'utente deve attivare il microfono per poter utilizzare i comandi vocali.	- Cliccando sul pulsante Play, oppure tramite il comando vocale "Where Am I" viene riprodotta la clip relativa al Luogo d'interesse più vicino alla posizione dell'utente. - Le clip vengono riprodotte con uno specifico ordine, basato sulla Purpose : la prima clip riprodotta sarà il What, successivamente il Why e infine il How. - E' possibile mettere in pausa la riproduzione tramite l'apposito pulsante Pause; è possibile riprodurre la clip successiva/precedente tramite il pulsante forward/backward.		
Filtraggio Clip	I campi che l'utente vuole filtrare devono essere specificati, tramite le apposite checkBox	- L'utente accede alla sezione nel menu "Filtra Clip" - Seleziona le checkBox relative ai campi che vuole che siano presenti nei risultati, e sono divisi per : Purpose, Audience e Language - Seleziona il raggio di distanza delle clip che vuole siano mostrate.		All'interno della stessa pagina verranno mostrate solamente le Clip che soddisferanno i filtri precedentemente selezionati.
Vicino A Me	Ricerca Clip deve aver trovato almeno una Clip nel raggio dell' OLC individuato dalla Posizione.	Accedere tramite il menu all'apposita sezione "Vicino A Me"		All'interno della sezione verranno visualizzati tutte le Clip relative ai punti d'interesse nelle strette vicinanze, con la possibilità di riprodurre la Clip che si vuole.

Diagramma casi d'uso Editor

Caso d'uso	Pre-condizioni	Sequenza di Eventi	Errori	Post-condizioni
CLogin Google	Disporre di un Account Google	L'utente effettua il login tramite il suo Account Google	Se non si effettua il login non si può accedere alla sezione di Editor.	L'utente può accedere alla sezione di Editor
Inserire la posizione	<ul style="list-style-type: none"> • GPS Attivo • L'utente deve essere loggato 	L'Editor rileva automaticamente la posizione tramite il GPS. Se l'utente vuole cambiare la posizione deve spostare il marker presente sulla mappa nel Browser.		La posizione rilevata serve per generare un codice OLC inerente alla clip registrata per il luogo in questione.
Aggiungere info luogo	L'utente deve essere loggato	<p>L'utente può inserire le informazioni relative al luogo quali:</p> <ul style="list-style-type: none"> • Nome; • Scopo, selezionando uno dei tre valori tra WHAT, WHY, HOW; • Lingua; • Categoria; • Pubblico; • Aggiungere una breve descrizione che comparirà successivamente nella descrizione del video su YouTube. 		Viene generata una stringa utilizzata come titolo del video che contiene tutte le informazioni necessarie per la ricerca dell'OLC.
Upload Clip video	<ul style="list-style-type: none"> • L'utente deve essere loggato • I campi delle informazioni ed il video non devono essere nulli. 	L'utente preme sul pulsante "Carica luogo".	Superato limite d'utilizzo giornaliero YouTube API Key	Il video e le informazioni relative sono caricati su YouTube rispettando i criteri di ricerca dell'OLC.

Caso d'uso	Pre-condizioni	Sequenza di Eventi	Errori	Post-condizioni
Registra video	<ul style="list-style-type: none"> L'utente deve essere loggato L'utente deve concedere i permessi al browser per accedere alla fotocamera e al microfono 	L'utente può avviare la registrazione ed interromperla manualmente oppure aspettare che finisca automaticamente secondo il tempo prestabilito.		Il video registrato viene salvato.
Edit video	<ul style="list-style-type: none"> L'utente deve essere loggato L'utente deve aver registrato un video 	L'utente può eliminare il video registrato e registrare un altro		Il video è salvato localmente in attesa di upload.
Crea clip video	<ul style="list-style-type: none"> Deve essere registrato un video L'utente deve essere loggato L'utente deve aver inserito le informazioni 	L'insieme delle informazioni ed il video vengono raggruppati in un unico file.		Il video è pronto per essere caricato su Youtube.

1.1 Tecnologie utilizzate

Dovendo sviluppare un progetto per il web, i linguaggi da noi usati sono stati HTML, CSS e JavaScript. L'unico vincolo tecnologico dato era che il sito doveva funzionare correttamente su Gocker, una macchina del Dipartimento di Informatica dedicata ai progetti web degli studenti. Gocker offre due servizi: NodeJS per il lato server, obbligatorio, e MongoDB per eventuali database, non obbligatorio.

1.2 Software Utilizzati

Filippo Bartolucci

SO: MacOS Catalina

Come editor per il codice ho scelto di usare [Atom](#) perché molto leggero e allo stesso tempo molto potente grazie alla sua modularità. Installando solo pochi plugin riesco a ottenere un editor con tutte le funzionalità che mi servono e senza distrazioni inutili.

Per testare localmente il sito ho usato NodeJS da terminale, dato che non avevo grandi necessità oltre a quelle offerte dai moduli di node.

Ho usato [Github Desktop](#) per gestire la repository GitLab attraverso un'interfaccia grafica semplice, che preferisco rispetto alla shell.

Umberto Case

SO: MacOs Catalina

Editor utilizzato per lo sviluppo in HTML, JavaScript e CSS: [Atom](#) con le sue estensioni per facilitare la scrittura di codice e debugging.

Software utilizzato per il testing: [MAMP](#) è una collezione di software con cui può essere gestito un server web in locale.

Gestione GitHub: plugin Atom GitHub, per la comodità di utilizzare un solo strumento, facile ed intuitivo, sia per la scrittura di codice sia per le commits nella repository.

Francesco Cerio

SO: MacOs Catalina

Per lo sviluppo ho scelto di utilizzare [Atom](#) in quanto, come già scritto in precedenza, è un software versatile e molto potente.

Ho usato [MAMP](#) come server locale per testare il progetto durante lo sviluppo. MAMP è una raccolta di software libero distribuiti insieme per il funzionamento di un server web per siti web dinamici, sul sistema operativo macOS. È semplice, veloce e molto versatile per testare al meglio un sito web prima di caricarlo sul server.

Per gestire la repository GitLab ho utilizzato un plugin Atom che permettesse di collegare la repository in questione e di effettuare le comuni operazioni di Git direttamente dall'editor. Dato che non è molto stabile e spesso causa delle incoerenze tra le versioni dei codici, ho fatto affidamento a [Github Desktop](#) per risolvere i conflitti.

Matteo Celani

SO: MacOS Mojave

Ho sviluppato il mio codice su [Atom](#) un editor di testo open source molto potente che supporta la maggior parte dei linguaggi di programmazione, installando pochi plugin sono riuscito ad avere tanti strumenti per lo sviluppo di codice.

Come server in locale ho utilizzato [MAMP](#), esso mi ha permesso di crearmi tutte le directory necessarie per avere un lavoro organizzato e pulito. MAMP è il nome dato ad una precisa collezione di software con cui può essere gestito un web server in locale. Con MAMP gli utenti Mac dispongono di una suite di software che si può installare velocemente e usare facilmente. L'anteprima del lavoro è stata lanciata su [Chrome](#) che possiede un pannello di sviluppo molto utile: analizzare il sito mentre veniva costruito si è rivelato fondamentale.

Per gestire le repository di GitLab ho utilizzato [Github Desktop](#). Molto semplice da utilizzare, l'interfaccia grafica ti aiuta facilmente a controllare il codice aggiunto o rimosso, in caso di conflitti segnala le righe dove si sono verificati, inoltre aggiornare le directory o inviare modifiche è davvero semplice ed intuitivo.

1.3 Divisione compiti

Dato che il progetto si può dividere in due componenti ben separabili, browser ed editor, abbiamo formato due sottoteam:

Team Browser:

- Filippo Bartolucci
- Umberto Case

Team Editor:

- Francesco Cerio
- Matteo Celani

2. Taiga

2.1 Specifiche del progetto e user story

Al fine della validità del progetto, abbiamo dovuto rispettare un serie di specifiche date dal docente, senza un formato preciso. Alcune erano facoltative, utili solo ad aumentare il punteggio per progetto.

Sei azioni fondamentali per l'utente:

- **Where am I:** il sistema riproduce la prima clip sul luogo in cui si trova
- **Next:** il sistema sceglie il prossimo luogo da visitare e,
 - se l'utente non si trova vicino, **fornisce indicazioni stradali per raggiungerlo** oppure riproduce un clip che dice "raggiungi il posto indicato nella mappa"
 - se/quando l'utente arriva, riproduce il primo clip sul luogo.
- **Previous:** il sistema ri-seleziona il luogo precedente,
 - se l'utente non si trova vicino, **fornisce indicazioni stradali per raggiungerlo** oppure riproduce un clip che dice "raggiungi il posto indicato nella mappa"
 - se/quando l'utente arriva, riproduce il primo clip sul luogo.
- **More:** il sistema riproduce il prossimo clip in lista sullo stesso luogo da visitare
- **Stop:** il sistema interrompe la riproduzione.
- **Continue:** il sistema riprende la riproduzione.



Esempio di specifiche

Per popolare Taiga siamo partiti da queste specifiche per creare le user story, scritte, nella maggioranza dei casi, nel formato SOGGETTO-VERBO-FINE. I possibili soggetti delle nostre user story sono "utente", chi interagisce con la mappa e visualizza le clip, e "editor", chi crea le nuove clip.

- Accesso ai contenuti di Wikipedia: cercare il contenuto di DBpedia (Wikipedia accessibile via API) relativo al luogo nella lingua richiesta e riprodurre il contenuto della descrizione iniziale con sintesi vocale

WHEREMI USER STORY

#53 Descrizione Wikipedia
This User Story belongs to #29 Browser

wiki facoltativa + Add tag

Created by Filippo Bartolucci
10 Feb 2020 00:20

Closed Done

Related tasks

0 attachments

Drop attachments here!

1 Comments 3 Activities

Filippo Bartolucci 12 Feb 2020 09:47
User story facoltativa

UX
Design
Front
Back
total points

Assigned to
Filippo Bartolucci
+ Add assigned

1 Watchers
Filippo Bartolucci
+ WATCHING + ADD WATCHERS

Taskboard

Trasformazione di una specifica in user story

Inizialmente abbiamo riportato tutte le specifiche come user story, comprese quelle facoltative, che poi nel corso dei 4 sprint abbiamo implementato. Alcune di quelle facoltative, per questioni di tempo, non siamo riusciti a portarle a termine.

Il meccanismo di creazione delle story di Taiga è intuitivo e allo stesso tempo funzionale: ci ha permesso di assegnare ad ogni story informazioni secondarie come tag, commenti, related task e status della story.

2.2 Organizzazione User Story

Ad ogni user story abbiamo associato dei tag per facilitare la loro organizzazione. Per esempio, sono stati usati dei tag per differenziare le story facoltative da quelle obbligatorie e tag per le story dell'utente e dall'editor. Sono stati assegnati i punteggi UX/Design/Front/Back e abbiamo utilizzato i commenti per mostrare i progressi nel loro sviluppo.

The screenshot shows a detailed view of a user story titled '#30 Aggiungere informazioni su un luogo'. The story is categorized under 'UX' and 'Design' with a total point value of 93. It was created by 'francesco cerio' on '30 Jan 2020 21:24'. The story description is: 'Come utente voglio poter aggiungere informazioni inerenti ad un luogo sulla mappa'. Below the description, there are sections for 'Related tasks' and 'Attachments'. The 'Comments' tab is active, showing two entries from 'Filippo Bartolucci' dated '12 Feb 2020 09:58'. The first comment shows an action being assigned to 'francesco cerio', and the second shows a tag being added: 'facoltativa'. The 'Activities' tab is also present. On the right side of the screen, there is a sidebar with sections for 'CLOSED' (with a 'DONE' button), 'UX', 'Design', 'Front', 'Back', 'total points' (93), 'Assigned to' (francesco cerio), '0 Watchers', and a list of icons for watching and sharing.

Dettaglio di una user story

Per separare meglio le story del browser da quelle dell'editor abbiamo creato una epic user story, che contiene tutte le story più piccole collegate, per entrambe le parti del progetto.

The screenshot shows a board view of an epic user story. The board has columns for 'Votes', 'Name', 'Project', 'Sprint', 'Assigned', 'Status', and 'Progress'. There are two items on the board: '#13 Editor' and '#29 Browser'. Both items are in the 'in progress' status, indicated by green progress bars. The 'Assigned' column shows 'francesco cerio' for both items. The 'Progress' column shows a green bar for each item, with a grey bar indicating the remaining work. A 'View options' dropdown is visible at the top right of the board.

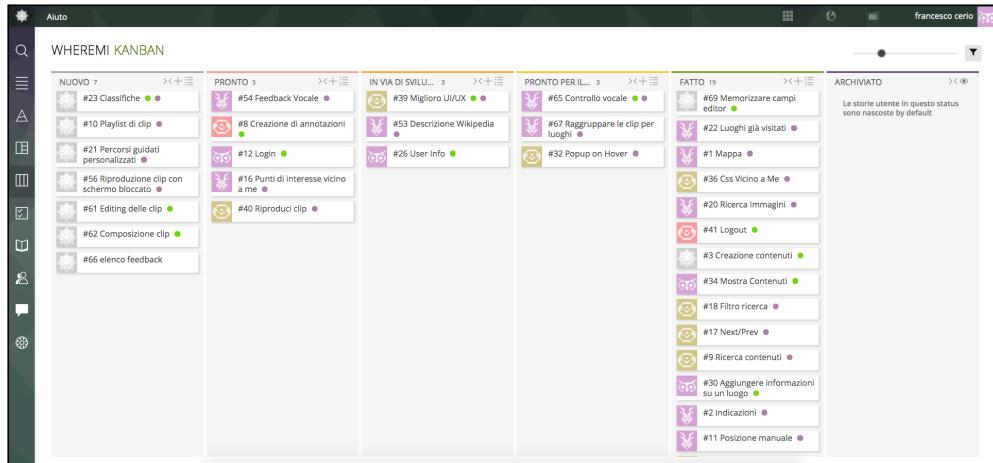
Epic user story

Avere le user story ben organizzate sul kanban ci ha aiutato molto nello sviluppo. Avendo iniziato a lavorare al progetto durante il periodo delle lezioni, il tempo dedicato allo sviluppo non è stato molto regolare. Tuttavia utilizzando Taiga è stato molto più facile riprendere in breve tempo lo sviluppo dopo periodi di inattività e fare il punto della situazione.

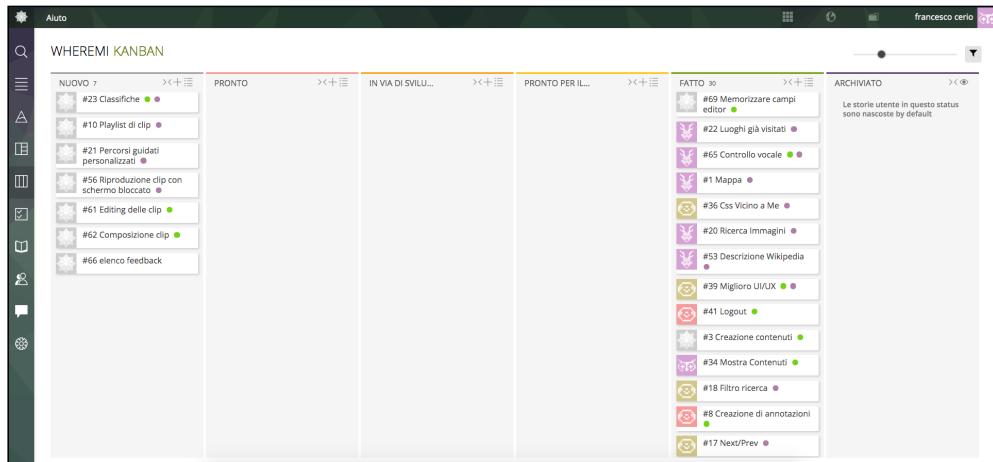
Per la gestione degli issue ci siamo affidati sempre a Taiga perché lo abbiamo trovato più che sufficiente per i nostri scopi, dal momento che per ogni issue permette di assegnare tipo, gravità, priorità e status.

2.3 Organizzazione Kanban

Di seguito sono riportati degli snapshot effettuati a campione durante lo sviluppo del progetto.



Screenshot kanban durante il processo di sviluppo



Screenshot kanban alla fine del processo di sviluppo

Organizzare il Kanban ci ha aiutato ad avere una visione globale dell'avanzamento dello sviluppo del progetto. Visualizzare graficamente i task ci ha permesso di avere un'organizzazione pulita e chiara di tutte le parti da sviluppare e di quelle già completate.

2.4 Definition of done

Il sito per essere funzionante doveva essere compatibile con le clip create dagli altri progetti. Ci siamo basati su questo per la nostra definition of done. Il lato browser doveva trovare e visualizzare senza problemi le clip altrui, d'altra parte il lato editor doveva creare nuove clip e far sì che queste comparissero insieme alle altre.

2.5 Issue

Anche per la gestione degli issue abbiamo utilizzato le funzionalità che ci offre Taiga. Non ha una gestione particolarmente avanzata, però per il nostro progetto è stato più che sufficiente.

Type	Severity	Priority	Votes	Subject	Status	Modified	Assigned to
bug	critical	critical	▲ 0	#68 Chiamate Asincrone	Closed ▾	05 Mar 2020	Filippo Ba...
bug	major	medium	▲ 0	#64 Alert wikipedia	Closed ▾	05 Mar 2020	Filippo Ba...
bug	critical	medium	▲ 0	#63 Chiamate a Wikipedia sincrone	Closed ▾	17 Feb 2020	Filippo Ba...
bug	critical	critical	▲ 0	#55 Visualizzazione mappa rottta	Closed ▾	12 Feb 2020	Filippo Ba...
bug	major	medium	▲ 0	#52 Visualizzazione rotta con display 2560*1440 e maggiori	Closed ▾	05 Mar 2020	Matteo C...
bug	major	medium	▲ 0	#51 Login non sempre funzionante	Closed ▾	05 Mar 2020	francesco...
bug	major	critical	▲ 0	#38 Dettagli in nearMe	Closed ▾	12 Feb 2020	umberto ...
bug	major	medium	▲ 0	#37 NearMe pulsanti	Closed ▾	06 Feb 2020	umberto ...
bug	major	medium	▲ 0	#27 Zoom mobile	Closed ▾	09 Feb 2020	Filippo Ba...
bug	major	medium	▲ 0	#25 Indicazioni non vanno con posiziona manuale	Closed ▾	11 Dec 2019	Filippo Ba...
bug	major	medium	▲ 0	#24 Posizione attuale doppia	Closed ▾	11 Dec 2019	Filippo Ba...
bug	major	medium	▲ 0	#15 Redirect dopo login	Closed ▾	09 Feb 2020	francesco...
bug	major	medium	▲ 0	#14 Sovraposizione elementi da mobile	Closed ▾	09 Feb 2020	Filippo Ba...
bug	major	medium	▲ 0	#7 Icone navbar sovrapposte in landscape	New ▾	26 Nov 2019	Not assig...
bug	major	medium	▲ 0	#6 Route di navigazione non si cancella	Closed ▾	11 Dec 2019	Filippo Ba...
bug	major	medium	▲ 0	#5 Indicazioni in lingua italiana	Closed ▾	11 Dec 2019	Filippo Ba...
bug	major	medium	▲ 0	#4 Indicazioni solo per auto	Closed ▾	11 Dec 2019	Filippo Ba...

Issue riportati in Taiga

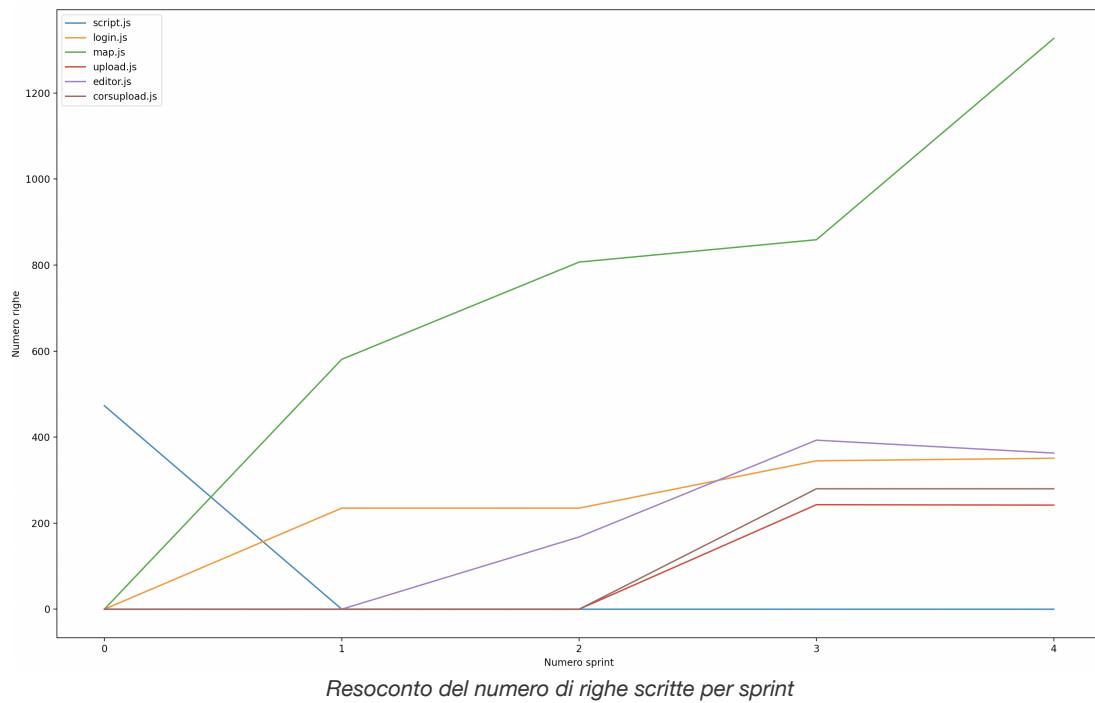
Per ogni issue abbiamo riportato tipo, gravità e priorità, inoltre abbiamo usato il campo descrizione per spiegare il problema. Invece, per aggiungere nuove informazioni sullo stato di un issue abbiamo sfruttato i commenti specifici per issue.

2.6 Sprint

All'inizio dello sviluppo abbiamo deciso che il nostro sito doveva essere sviluppato seguendo lo stile di un'applicazione mobile, quindi doveva essere mobile-first, basato su navbar e reattivo. Per fare ciò abbiamo deciso di usare un footer fisso come navbar e di utilizzare Ajax per il caricamento delle pagine dell'applicazione.

Inizialmente avevamo pianificato 3 sprint, che avrebbero dovuto coprire i 30 giorni prima della consegna. Durante la presentazione del progetto ci è stato chiesto di fare delle modifiche per migliorarlo e riconsegnarlo, per questo abbiamo deciso di organizzare un quarto sprint finale con le modifiche da fare.

Prima di iniziare gli sprint, le uniche righe di codice scritte erano contenute nel file "script.js" e servivano per gestire la navbar e le chiamate ajax.



2.7 Primo sprint

Durata: 20/01/2020 - 03/02/2020

Per il primo sprint abbiamo dato priorità assoluta alle user story obbligatorie. Abbiamo inserito tutte le story che ritenevano base per lo sviluppo delle altre. Per esempio sono state inserite le user story che riguardano il login che è una funzionalità necessaria per lo sviluppo delle le user story che riguardano la creazione e il caricamento dei video.

Le uniche cose sviluppate, prima di iniziare gli sprint, erano il tema grafico, la navbar e le relative chiamate ajax per la navigazione delle pagine.

Risultato sprint

File	Righe inizio	Righe fine	Differenza
script.js	473	0	-473
login.js	0	235	+235
map.js	0	581	+581
Totale	473	581	+108

Alla fine del primo sprint avevamo una versione molto base dell'applicazione. Il file "script.js" è stato diviso in "login.js" e "map.js" per tenere ordinati il codice delle due parti del progetto. Le user story completate con successo sono state le seguenti:

#1 Mappa	●	120
#12 Login	●	73
#26 User Info	●	66
#28 Migliorare interfaccia Login	●	58
#16 Punti di interesse vicino a me	●	48
#31 Css Clip	●	61
#11 Posizione manuale	●	41
#2 Indicazioni	●	65

Story realizzate alla fine dello sprint

Il sito è ancora molto semplice: c'è una mappa con qualche pulsante per interagire, le indicazioni stradali per raggiungere i luoghi ed è possibile fare login e vedere le proprie info dalla pagina profilo. I punti di interesse sulla mappa sono stati creati manualmente, in quanto ancora non sviluppato un sistema per la ricerca delle clip. I bug rilevati sono tracciati nella sezione issues Taiga, ma non tutti sono stati risolti: per alcuni c'è un fix da testare, mentre altri non hanno ancora soluzione. Le sezioni "Vicino a me", "Cerca" e "Categorie" sono solo mockup grafici e non sono ancora state sviluppate.

The image displays four screenshots of a mobile application interface, likely a city guide or local information app, showing different sections of the platform.

- Top Left:** A map view of a city area, specifically Bologna, showing streets like Bolognina, Europa, Via delle Carraia, and Teatro Comunale di Bologna. Several location markers are visible, including green dots and a red dot. The interface includes navigation controls (+, -, ▲, ▼) and a search bar at the top.
- Top Right:** A user profile screen for "Mario Rossi". It features a green circular profile picture, the user's name, and two buttons: "Carica File" (Upload File) with a cloud icon and "Apri Fotocamera" (Open Camera) with a camera icon. Below these buttons is the text "I Miei Contributi" (My Contributions). At the bottom are five navigation icons: WhereAMI, VicinoMe, Cerca, Categorie, and Profilo.
- Bottom Left:** A section titled "Vicino a me" (Nearby). It shows two cards: one for "Piazza Maggiore" featuring a photo of the square with people and a building, and another for "Le Due Torri" featuring a photo of the two towers. Each card has a "Vedi Dettagli" (See Details) button and a "Distanza: 100m" (Distance: 100m) indicator. The bottom navigation bar includes icons for WhereAMI, VicinoMe, Cerca, Categorie, and Profilo.
- Bottom Right:** A section titled "Categorie" (Categories). It lists categories on the left: Piazze, Università, Parchi e Giardini, and on the right: Basiliche, Centro Storico, and Categorie. Below this is a grid of three dark cards labeled "Card". The bottom navigation bar includes icons for WhereAMI, VicinoMe, Cerca, Categorie, and Profilo.

Risultato primo sprint.

2.8 Secondo sprint

Durata: 03/02/2020 - 09/02/2020

Anche nel secondo sprint le story obbligatorie hanno avuto la priorità, ma abbiamo iniziato a lavorare anche su qualcuna di quelle facoltative.

#3 Creazione contenuti	63
#36 Css Vicino a Me	85
#32 Popup on Hover	33
#19 Clip Luogo Corrente	37
#17 Next/Prev	83
#9 Ricerca contenuti	63
#39 Miglioro UI/UX	83
#34 Mostra Contenuti	63
#41 Logout	55

Secondo sprint

Al termine di questo sprint abbiamo implementato delle feature ulteriori in modo da rendere il sito più completo, esaurendo l'elenco di tutte le richieste obbligatorie. Inoltre, abbiamo iniziato anche a migliorare l'aspetto grafico, rendendolo meno macchinoso e più intuitivo per utente.

Risultato sprint

File	Righe inizio	Righe fine	Differenza
editor.js	0	168	+168
login.js	235	292	+57
map.js	581	807	+226
Totale	581	1325	+744

Nella mappa è stata aggiunta la ricerca delle clip, le quali vengono ora visualizzate come luoghi di interesse, nonostante ci sia ancora qualche problema nella lettura dei metadati associati. Sempre sulla mappa è stato aggiunto un player per il controllo della riproduzione delle clip. Per la sezione "Vicino a me" le funzionalità sono rispettate, ma ci sono ancora dei problemi grafici da risolvere. Dal lato Editor è ora possibile registrare video, aggiungere informazioni sul tipo di clip e caricare il tutto su YouTube.

Map View: Shows a map of Bologna with a blue circle centered on Piazza Maggiore. A callout box displays "Distanza: 287m" and "Piazza Maggiore". Buttons at the bottom include "Play per iniziare" and navigation arrows.

Login Screen: Displays the title "WhereMI" and a button "Login with Google". Below it, text reads "Effettua il login entra nell'Editor" and "Oppure torna indietro Non potrai aggiungere o modificare luoghi".

Place Details: Shows "Piazza Maggiore" in large text, "Language: English", and sections for "Informazioni Clip" (Purpose: Come, Content: Storia, Audience: Medie) and "Piazza Maggiore" (Italiano, Distanza: 287m, Audience: Agen, Purpose: Perché).

Clip Information: Shows two clips: "SFPHF800+" (Italiano, Distanza: 771m, Audience: Agen, Purpose: Perché) and "PalazzoPepoli" (Italiano, Distanza: 489m, Audience: Elementare, Purpose: Che cosa).

Risultato secondo sprint.

2.9 Terzo sprint

Durata: 09/02/2020 - 13/02/2020

Inizialmente questo era stato programmato per essere lo sprint finale, dato che eravamo giunti a ridosso della data di scadenza per la presentazione del progetto. Ci siamo, quindi concentrati nell'aggiungere alcune delle feature facoltative, nel sistemare i bug e ripulire il codice secondo i report di SonarQube.

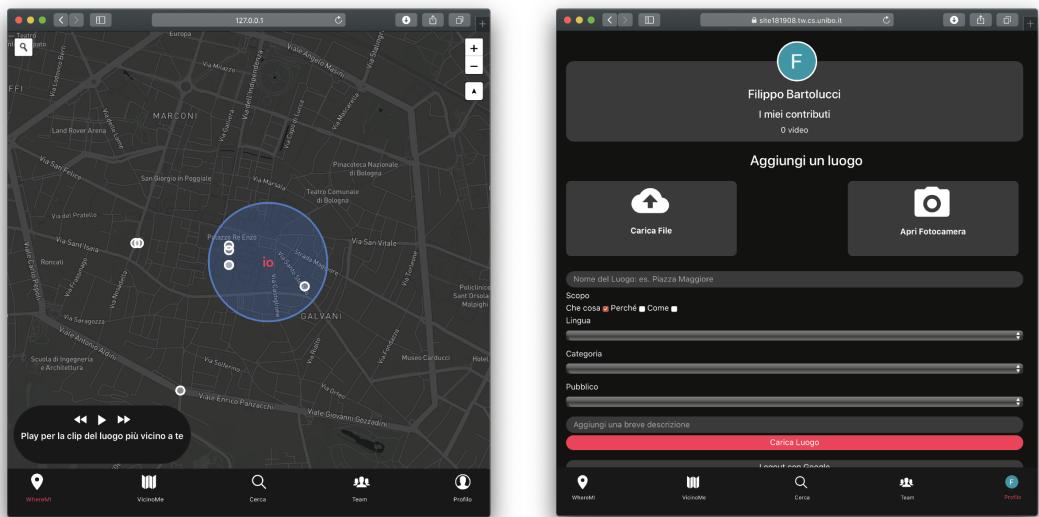
#53 Descrizione Wikipedia	●	13
#54 Feedback Vocale	●	5
#40 Riproduci clip	●	19
#18 Filtro ricerca	●	53.5
#8 Creazione di annotazioni	●	48
#20 Ricerca Immagini	●	43
#30 Aggiungere informazioni su un luogo	●	93

Terzo sprint

Risultato sprint

File	Righe inizio	Righe fine	Differenza
corsupload.js	0	280	+280
editor.js	168	393	+225
login.js	292	345	+53
map.js	807	859	+52
upload.js	0	243	+243
Totale	1325	2120	+795

Tutte le sezioni dell'applicazione sono state riviste graficamente. Nella mappa sono stati aggiunti: la ricerca delle descrizioni dei luoghi su Wikipedia; la possibilità di far leggere i metadati della clip e un meccanismo per filtrare quest'ultime. Insieme alla clip, l'editor può ora allegare anche una descrizione testuale.



Risultato terzo sprint.

2.10 Quarto sprint

Durata: 15/02/2020 - 04/03/2020

Questo è stato per noi uno sprint non programmato, ma necessario per soddisfare le richieste del professore Vitali. Ci è stato chiesto di: cambiare il modo di visualizzare le clip sulla mappa, queste non devono comparire come singoli punti di interesse, ma vanno raggruppate per luoghi; trasformare le chiamate ajax sincrone in asincrone; aggiungere il controllo vocale; e, infine, implementare parti facoltative minori.



Quarto sprint

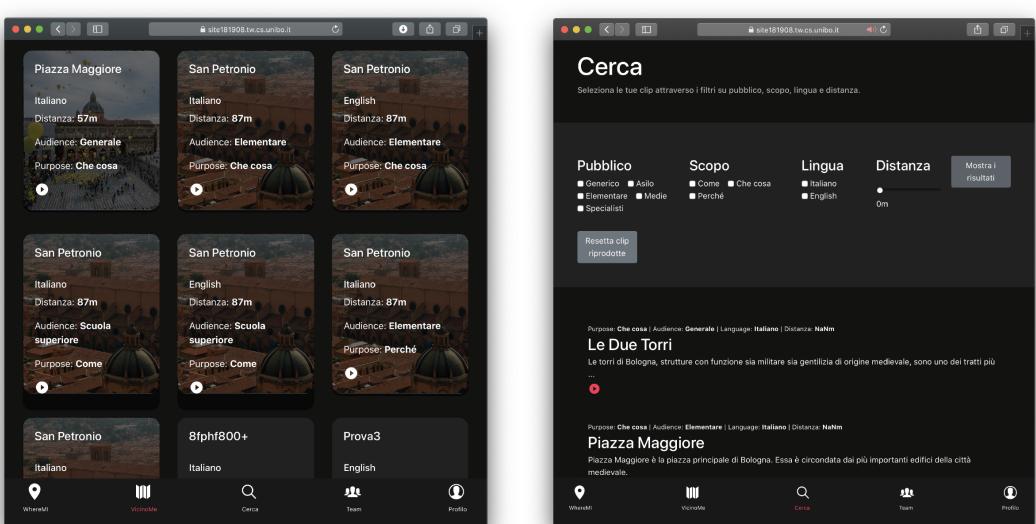
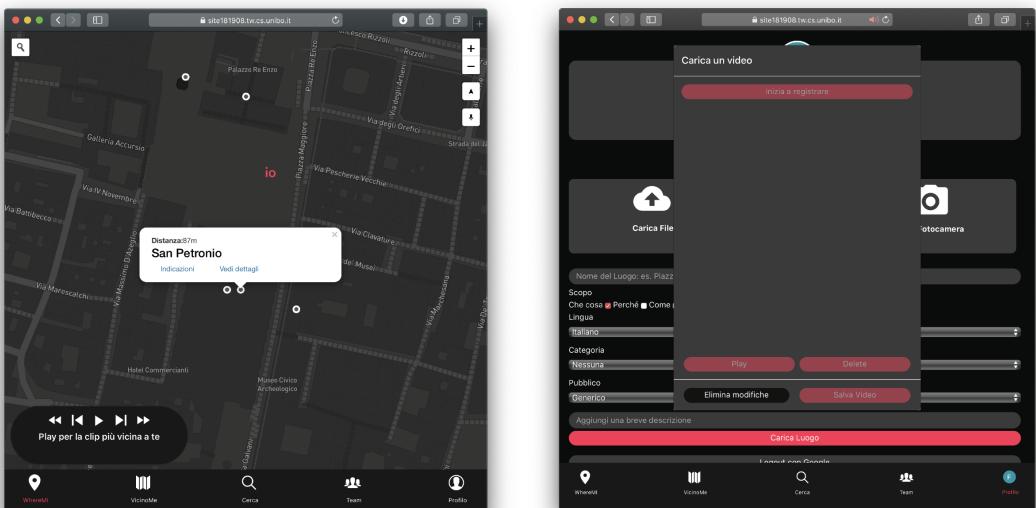
Abbiamo sfruttato il tempo extra a disposizione per aggiungere anche qualche story facoltativa, sistemare e riorganizzare ulteriormente il codice per rendere il progetto più chiaro e facile da leggere.

Risultato sprint

File	Righe inizio	Righe fine	Differenza
corsupload.js	280	280	0
editor.js	393	363	-30
login.js	345	351	+6
map.js	859	1327	+468
upload.js	243	242	-1
Totale	2120	2563	+443

L'applicazione è ormai terminata ed ogni parte sviluppata rispetta la definition of done. È stata aggiunta la possibilità di utilizzare i controlli vocali per interagire con il sistema, come, ad esempio, navigare nella mappa, riprodurre/interrompere un file audio, eccetera. Inoltre sono state migliorate alcune funzioni segnalate dalle analisi di SonarQube e sistemati gli ultimi bug.

Alla consegna del progetto il professore Vitali ci ha assegnato 26/30.



Risultato quarto sprint.

2.11 Burndown

Un burn down chart è una rappresentazione grafica del lavoro da fare su un progetto nel tempo. Il diagramma rappresenta una serie storica del lavoro da noi svolto. È utile per prevedere quando avverrà il completamento del lavoro. È spesso utilizzato in metodologie agili di sviluppo software come Scrum (appunto da noi utilizzata).

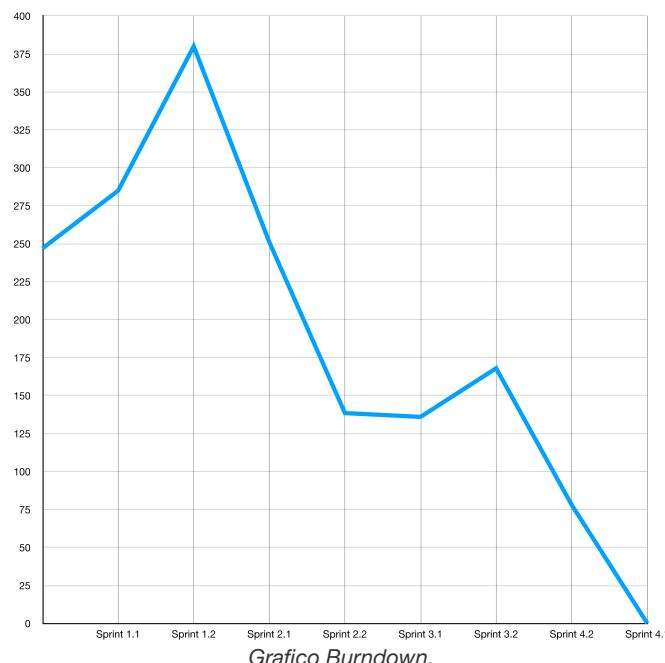
In particolare per ottenere un grafico più accurato, noi abbiamo diviso gli sprint in 2 sprint minori e creato un grafico con il punteggio relativo.

Divisione Sprint

1. Costruzione scheletro del sito
 - 1.1 WhereMI, VicinoMe, Cerca, Team, Profilo
 - 1.2 Login, posizione, punti di interesse
2. Sviluppo tecnologie e UI/UX
 - 2.1 Contenuti, ricerca, popup, logout
 - 2.2 Next Prev, UI/UX, css
3. Incrementato il numero di funzionalità
 - 3.1 Wikipedia, feedback, clip, filtri, annotazioni
 - 3.2 Ricerca immagini, informazioni luogo
4. Implementazioni aggiuntive e correzioni finali
 - 4.1 Controllo vocale, raggruppato clip per luogo
 - 4.2 Cookie

Tabella relativa ai punteggi per sprint

	Sprint 1.1	Sprint 1.2	Sprint 2.1	Sprint 2.2	Sprint 3.1	Sprint 3.2	Sprint 4.1	Sprint 4.2
Punteggio	247	285	380	251	138,5	136	168	78



Analizzando il grafico possiamo notare come il nostro si tratti di un Burndown Scope Creep: questo è accaduto perchè nuove attività sono state pianificate durante l'iterazione, priorità cambiate, attività urgenti da sviluppare il giorno dopo indispensabili per strumenti appena aggiunti, operazioni preliminari pianificate improvvisamente prima di eseguire le attività definite, insomma

nessuno è stato più in grado di dire in modo preciso date di scadenza, perché ogni giorno abbiamo stravolto decisamente le scadenze prese il giorno prima, dando spazio alla fantasia e imparando come funzionasse sviluppare una WebApp.

3. GitLab

GitLab è lo strumento di version control e repository utilizzato per il progetto. Non lo avevamo mai utilizzato, ma avendo usato GitHub in passato, conoscevamo già le sue potenzialità ed è stato facile familiarizzare con l'ambiente.

Sapevamo che l'utilizzo di Git sarebbe stato fondamentale nello sviluppo perché ci ha permesso di scrivere codice e di evitare conflitti tra le varie versioni dei sorgenti.

Anche le funzionalità di repository sono state importantissime per noi: l'utilizzo dei branch ci ha permesso di lavorare alle due parti del progetto in modo autonomo.

Il setup iniziale è stato immediato. Dopo aver creato la pagina del progetto e aver caricato il codice, eravamo subito pronti a iniziare.

Lo sviluppo è proseguito su due branch separati, uno per editor e uno per utente. Dopo la prima consegna del progetto abbiamo creato un terzo branch per integrare le correzioni del codice richieste. Una feature che ci è tornata utile durante le fasi di sviluppo è la history dei branch, perché ci è capitato di dover tornare indietro nelle commit per ripristinare dei cambiamenti fatti.

Quando facevamo il merge delle due parti del progetto ci è capitato ogni tanto di avere dei conflitti, ma non abbiamo mai avuto grossi problemi nel risolverli.

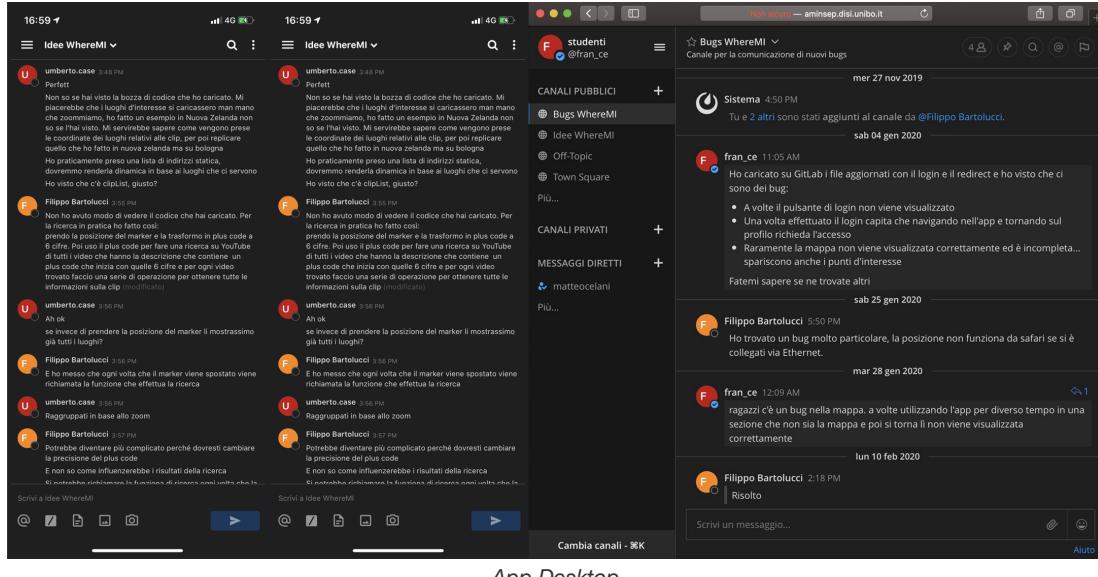
The screenshot shows the 'Active branches' interface in GitLab. It lists three branches:

- master**: Default, protected. Last commit: 6e39c69c - Relazione - 1 hour ago. Status: 55 | 0. Actions: Merge request, Compare, More options (trash bin).
- correzioni-vitali**: Merged. Last commit: 8b9e6f94 - Merge branch 'master' into 'correzioni-vitali' - 1 month ago. Status: 55 | 0. Actions: Merge request, Compare, More options (trash bin).
- EditorSviluppo**: Merged. Last commit: c546f23c - Merge branch 'master' into 'EditorSviluppo' - 1 month ago. Status: 170 | 0. Actions: Merge request, Compare, More options (trash bin).

Divisione dei branches.

4. Mattermost

Per la comunicazione del team abbiamo usato Mattermost, applicazione che offre tutte le funzionalità di una chat moderna (citazioni, risposta diretta, messaggi pinnati...) ed è disponibile come WebApp, app desktop e app mobile. La configurazione è stata immediata, per prima cosa abbiamo inserito l'url del server e poi eseguito l'accesso.



App Desktop

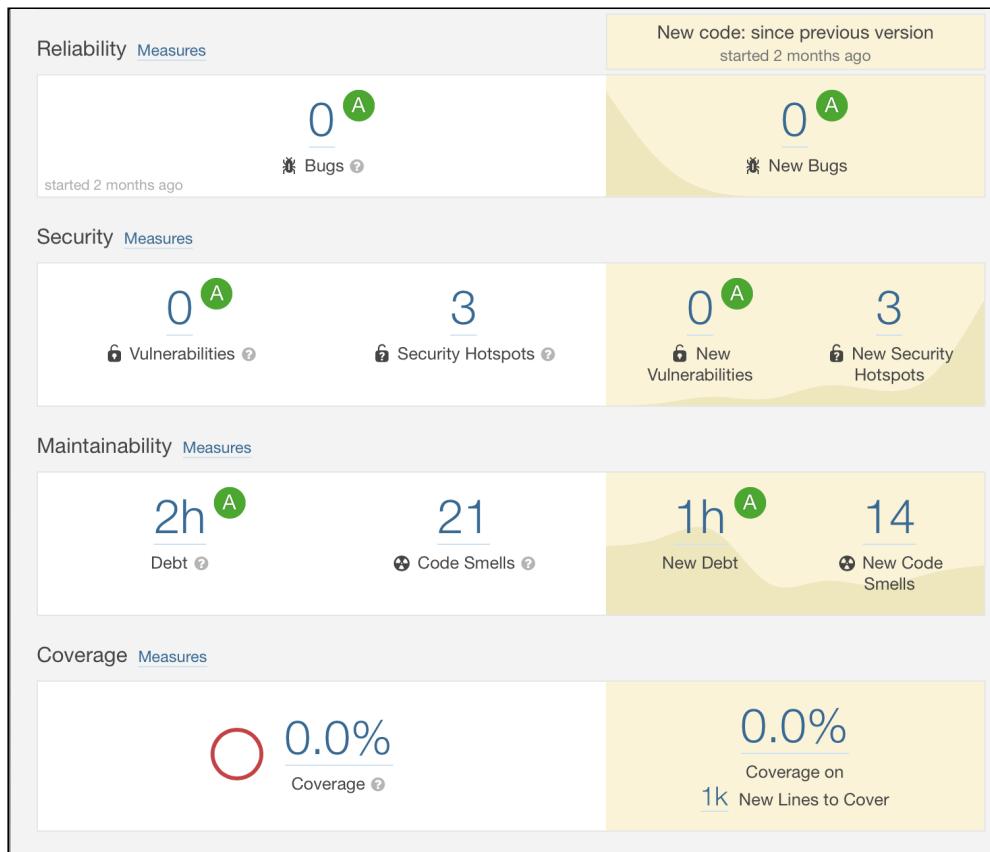
L'App mobile si è presentata molto intuitiva e semplice nell'utilizzo, come ogni applicazione di messaggistica. Sul sistema operativo Android non abbiamo riscontrato problemi di utilizzo.

Per evitare di creare confusione con i messaggi, abbiamo creato due chat (Mattermost li chiama canali di comunicazione) per il progetto: Idee WhereMI, usato per discutere di nuove idee per lo sviluppo, e Bugs WhereMI, per segnalare nuovi bug.

L'utilizzo di una chat, adibita esclusivamente al progetto, ci ha aiutato nel rimanere concentrati nella conversazione e ad evitare di andare off topic. In questo modo è stato anche più facile ritrovare vecchi messaggi e conversazioni quando sono serviti.

5. SonarQube

È uno strumento molto potente, che permette di effettuare un'analisi dettagliata dell'intero progetto dal punto di vista di ridondanza di codice, errori, bug e code smells.



Schermata di SonarQube dopo aver effettuato un'analisi del codice

È risultato molto utile nello sviluppo del progetto, in quanto ci ha permesso di tenere costantemente sott'occhio l'evoluzione dello stesso. Grazie alle analisi effettuate siamo riusciti ad individuare tutti i problemi presenti nel progetto, dai più gravi a quelli minori e consultando la documentazione disponibile abbiamo potuto capire come risolverli, migliorando le porzioni di codice. Spesso si è trattato solamente di sostituire dei tag HTML o di cambiare delle funzioni javascript oramai deprecate.

This screenshot shows a detailed view of a security hotspot found in the code. The title is "Writing cookies is security-sensitive". It is categorized as a "Security Hotspot" (severity: Minor, source: Main sources, cert: cert, cwe: owasp-a3, sans-top25-porous) and was available since November 26, 2019, by SonarAnalyzer (JavaScript). The constant issue time is 5 minutes.

The description states: "Using cookies is security-sensitive. It has led in the past to the following vulnerabilities:"

- CVE-2018-11639
- CVE-2016-6537

Attackers can use widely-available tools to read cookies, sensitive information written by the server will be exposed.

This rule flags code that writes cookies.

Ask Yourself Whether

- sensitive information is stored inside the cookie.

Esempio di consultazione della documentazione per la risoluzione di un bug

The screenshot shows a SonarQube code review interface. At the top, there's a navigation bar with 'sonarqube' logo, 'Projects', 'Issues', 'Rules', 'Quality Profiles', and 'Quality Gates'. Below the navigation is a search bar and a date stamp 'February 12, 2020, 11:41 PM Version not provided'. The main area shows a file named 'src/cors_upload.js' under the 'WhereMI181908' project. The code editor displays several lines of JavaScript. Three specific lines are highlighted with red boxes and labeled as 'Security Hotspot':

- Line 112: `var cookie = key + "=" + value + ";expires=" + expiration;` with the note 'Make sure that cookie is written safely here.'
- Line 113: `//Aggiungo il valore per il numero di video caricati impostato a 0 di default` with the note 'Make sure that cookie is written safely here.'
- Line 114: `document.cookie = cookie;` with the note 'Make sure that cookie is written safely here.'

Below the code editor, there's a message '3 of 3 shown'.

Schermata in cui viene presentato un bug

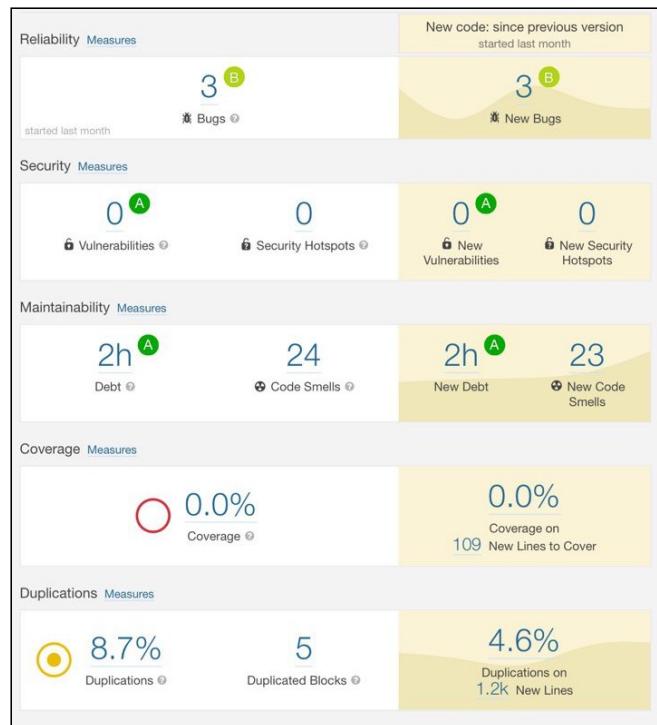
Di seguito sono riportate tre schermate effettuate al termine degli sprint: si può notare come i risultati ottenuti cambino in base al codice sviluppato. Abbiamo studiato le analisi ogni volta per capire quali fossero i nostri errori e cercare di risolverli gradualmente per mantenere un codice pulito e leggibile.

Tra i parametri riportati nell'analisi, i più utili a nostro parere sono:

- i **bug**, scrivendo molto codice è facile che una svolta porti ad incongruenze e possibili falle che ad occhio risultano difficili da trovare. Con l'aiuto di SonarQube siamo stati in grado di trovarne senza problemi e capire anche cosa possa generarne uno, dato che viene evidenziata la porzione di codice che lo contiene con annessa documentazione per la risoluzione dello stesso;
- il **debito tecnico**, ci ha aiutato a capire il tempo effettivamente necessario per poter risolvere i code smells presenti nel codice;
- il **numero di linee duplicate**, ci ha permesso di trovare le porzioni di codice superflue ed inutili.

Un esempio: avendo deciso di avere un div html principale nel quale caricare dinamicamente il contenuto di ciascuna pagina ogni volta che venisse richiesta, non avevamo notato che l'inclusione degli script nelle header di ogni pagina risultava superfluo e ridondante, in quanto bastava includere tutto solamente nel file index.html che contiene il div principale. Grazie allo strumento di analisi siamo riusciti a tenere pulito il nostro codice ed evitare ridondanza.

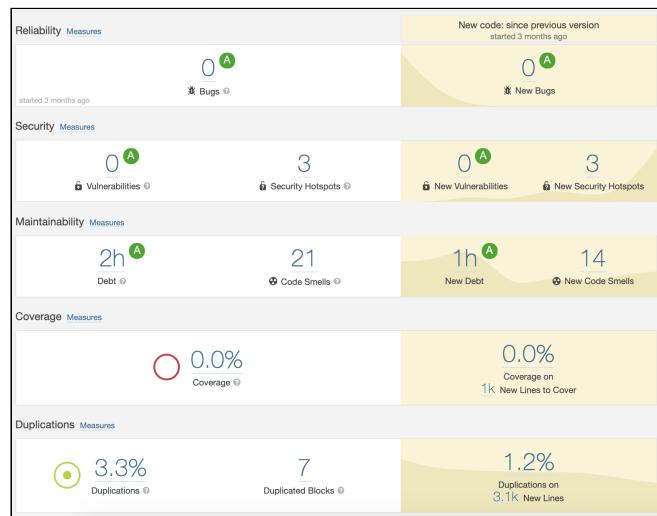
5.1 Analisi primo sprint



Analisi del codice dopo il primo sprint

Abbiamo notato che dopo il primo sprint dall'analisi risultano presenti tre bug, due ore di debito tecnico e l'8.7% di codice duplicato. Quindi abbiamo deciso di apportare le modifiche necessarie prima di continuare con lo sprint successivo in modo da sistemare i problemi man mano in previsione di future correzioni da apportare.

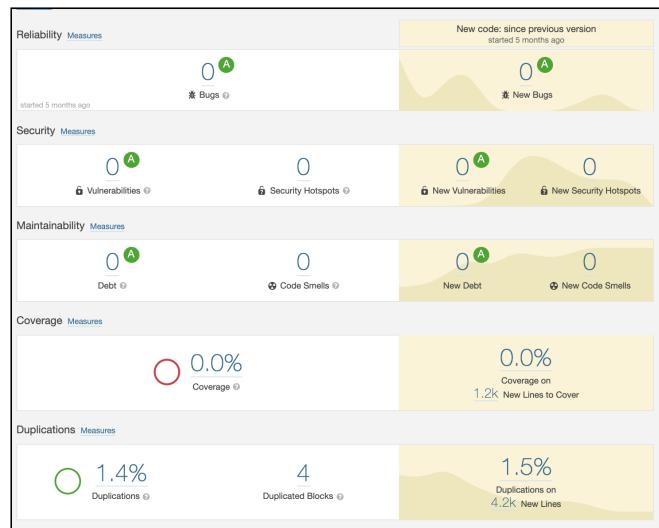
5.2 Analisi secondo e terzo sprint



Analisi del codice dopo il secondo e terzo sprint

Questa schermata raccoglie i risultati delle analisi dei successivi due sprint. Abbiamo visto come rispetto alla precedente analisi i bug sono stati eliminati e non ce ne sono di nuovi, mentre i nuovi code smells sono diminuiti ed anche il debito tecnico nuovo è decrementato, seppur il debito totale è rimasto invariato, ma questo è dovuto al fatto che la schermata include sia l'analisi del secondo che del terzo sprint. L'indice della duplicazione di codice si è abbassato ma rimangono comunque delle porzioni da eliminare per poter portare l'indice a zero.

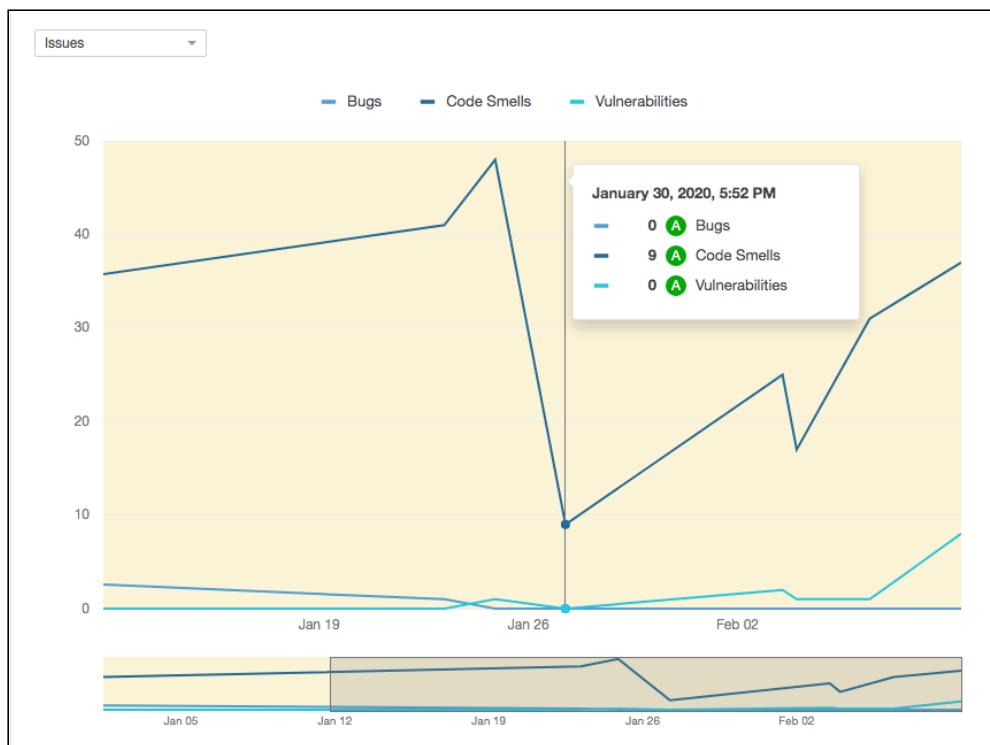
5.2 Analisi quarto sprint



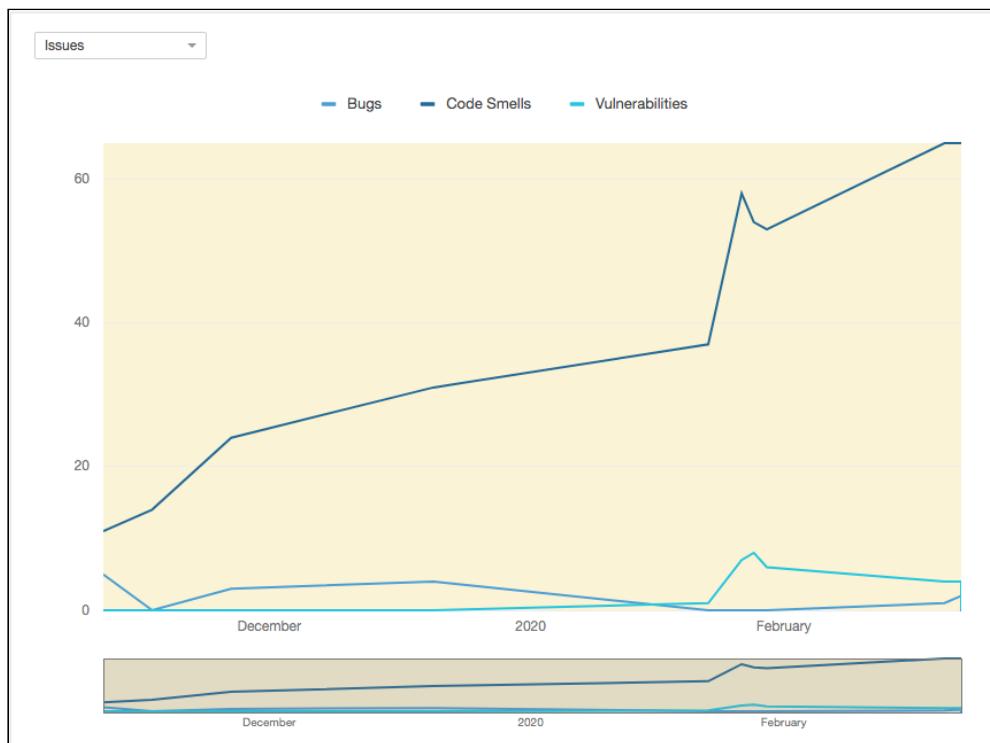
Analisi del codice dopo il quarto sprint

Dall'ultima analisi abbiamo riscontrato dei risultati positivi rispetto alle precedenti, in quanto tutti i parametri sono stati azzerati. Rimane solo una parte di codice duplicato, ma questo è dovuto alla presenza di uno switch nel codice JavaScript che analizza dei casi che possono avere dei nomi diversi ma gestione del codice uguale e quindi ridondanza dello stesso.

Ci siamo avvalsi dei grafici statistici sull'andamento del codice per capire dove stavamo sbagliando, se il lavoro svolto fino a quel momento andasse bene e in cosa migliorarlo.



Screen del grafico durante lo sviluppo del progetto



Screen del grafico al termine dello sviluppo del progetto

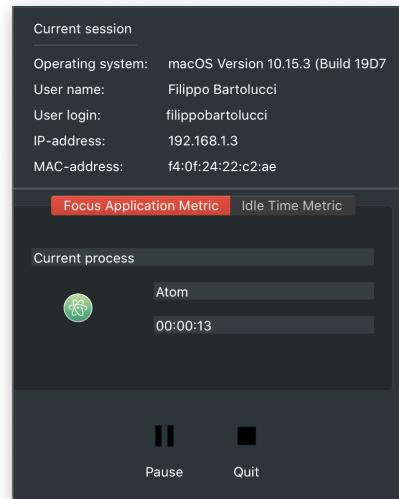
Abbiamo notato come è cambiato il grafico nel corso del tempo e di come, al crescere del numero di righe di codice, la presenza di code smell sia aumentata notevolmente. Molti di quelli riportati dallo strumento di analisi erano banali e abbiamo notato che se non rimossi o corretti, avrebbero compromesso il normale funzionamento del sito.

6. Innometrics

Innometrics si compone di due applicazione: Collector e Transfer.

6.1 Collector

Si occupa di registrare tutte le applicazioni in attività sul computer su cui gira. Una volta lanciato non si nota nulla se non la presenza della sua icona (un occhio che guarda) nella menu bar di MacOS. Finché rimane in esecuzione, Collector monitora quanto tempo passiamo su ogni applicazioni o sito web.



Innometrics Collector

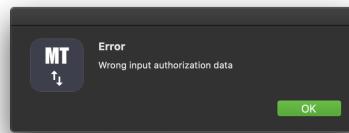
6.2 Transfer

Innometrics Transfer permette di vedere i dati raccolti da Collector. Ha una sola schermata dove vengono mostrati i tempi di utilizzo delle applicazioni. È possibile filtrare e/o caricare i dati su un server.

Refresh table		Metrics Preview (Last update: 18:45 - 29 - 05 - 2020)							Clear Database	
Metric Type	Name	Bundle	Bundle Path	Timestamp Start	Timestamp End	Duration	Browser Tab Title	Browser Tab Uri		
Exclusion Filters:										
Keywords										
Apps	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:32-02...	2020-02-13 23:32-37...	00:00:15	tutu bologna - Ric...	https://www.google.co...	
	idle	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:31-37...	2020-02-13 23:31-35...	00:00:16	Le due Torri - Garis...	https://www.bolognaw...	
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:31-38...	2020-02-13 23:32-02...	00:00:15	Le due Torri - Garis...	https://www.bolognaw...	
	idle	Atom	com.github.atom	file:///Applications/Ato...	2020-02-13 23:27-48...	2020-02-13 23:31-46...	00:03:58			
	App Focus	Atom	com.github.atom	file:///Applications/Ato...	2020-02-13 23:27-48...	2020-02-13 23:31-46...	00:03:58			
	App Focus	InnometricsTransfer	com.denzap.Innometri...	file:///private/var/folder...	2020-02-13 23:27-31...	2020-02-13 23:27-38...	00:00:16			
	App Focus	Atom	com.github.atom	file:///Applications/Ato...	2020-02-13 23:27-03...	2020-02-13 23:27-06...	00:00:03			
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 22:29-20...	2020-02-13 22:29-24...	00:00:03			
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 22:29-05...	2020-02-13 22:29-39...	00:15:00	anyang9 Easily add sp...	https://www.talater.co...	
	idle	com.apple.Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 22:28-58...	2020-02-13 22:29-03...	00:00:05	Preferr...	/leafless/	
From	App Focus	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:28-53...	2020-02-13 22:28-53...	00:00:03	5 Voice Control JavaSc...	https://queyhouse.co...	
	idle	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:28-18...	2020-02-13 22:28-24...	00:00:06	anyang9 Easily add sp...	https://www.talater.co...	
	App Focus	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:27-59...	2020-02-13 22:28-53...	00:00:04	anyang9 Easily add sp...	https://www.talater.co...	
	App Focus	Oversight Helper	com.objective-see.Ove...	file:///Applications/Ove...	2020-02-13 22:27-01...	2020-02-13 22:27-08...	00:00:06	anyang9 Easily add sp...	https://www.talater.co...	
	idle	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:27-01...	2020-02-13 22:27-08...	00:00:06	anyang9 Easily add sp...	https://www.talater.co...	
	App Focus	UserNotificationCenter	com.apple.UserNotification	file:///System/Library/C...	2020-02-13 22:27-45...	2020-02-13 22:27-47...	00:00:02	anyang9 Easily add sp...	https://www.talater.co...	
	App Focus	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:27-39...	2020-02-13 22:27-45...	00:00:06	anyang9 Easily add sp...	https://www.talater.co...	
	App Focus	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:27-34...	2020-02-13 22:27-34...	00:00:04	Nuova scheda	chrome://newtab/	
	App Focus	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:27-34...	2020-02-13 22:27-34...	00:00:04	Nuova scheda	chrome://newtab/	
	idle	Google Chrome	com.google.Chrome	file:///Applications/Goo...	2020-02-13 22:27-15...	2020-02-13 22:27-29...	00:00:14	anyang9 Easily add sp...	https://www.talater.co...	
To	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 22:27-05...	2020-02-13 22:27-15...	00:00:10	TaiAntering9 See...	https://github.com/fal...	
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 22:27-00...	2020-02-13 22:27-05...	00:00:05	5 Voice Control JavaSc...	https://queyhouse.co...	
	idle	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 22:26-55...	2020-02-13 22:27-00...	00:00:05	voice-commands.js	https://jimmybyrun.git...	
	App Focus	Finder	com.apple.Finder	file:///System/Library/C...	2020-02-13 22:26-50...	2020-02-13 22:26-85...	00:00:35	voice-commands.js	https://jimmybyrun.git...	
	idle	Finder	com.apple.Finder	file:///System/Library/C...	2020-02-13 22:26-41...	2020-02-13 22:26-45...	00:00:03	voice-commands.js	https://jimmybyrun.git...	
	App Focus	Finder	com.apple.Finder	file:///System/Library/C...	2020-02-13 22:26-21...	2020-02-13 22:26-26...	00:00:20	jimmybyrun/voice-co...	https://github.com/jim...	
	idle	Xcode	com.apple.Xcode	file:///Applications/Xco...	2020-02-13 22:54-37...	2020-02-13 22:54-44...	00:00:07			
	App Focus	Xcode	com.apple.Xcode	file:///Applications/Xco...	2020-02-13 22:54-37...	2020-02-13 22:54-44...	00:00:07			
	App Focus	Xcode2	com.apple.Xcode2	file:///Applications/Xco...	2020-02-13 22:54-27...	2020-02-13 22:54-31...	00:00:03			
	App Focus	Xcode2	com.apple.Xcode2	file:///Applications/Xco...	2020-02-13 22:54-27...	2020-02-13 22:54-31...	00:00:02			
Send Metrics	App Focus	Finder	com.apple.Finder	file:///System/Library/C...	2020-02-13 23:54-24...	2020-02-13 23:54-25...	00:00:00			
	idle	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:54-23...	2020-02-13 23:54-23...	00:00:01	Where am I ?	https://beta191038.w...	
	App Focus	com.apple.Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-33...	2020-02-13 23:54-09...	00:00:08	Where am I ?	https://beta191038.w...	
	idle	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-35...	2020-02-13 23:54-30...	00:00:05	Accedi - Account Go...	https://accounts.googl...	
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-37...	2020-02-13 23:53-00...	00:23	accounts.google.com	https://accounts.googl...	
	idle	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-39...	2020-02-13 23:53-04...	00:30	Forwarding ...	https://accounts.googl...	
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-39...	2020-02-13 23:53-04...	00:30	Where am I ?	https://beta191038.w...	
	idle	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-43...	2020-02-13 23:53-34...	00:00:10	Where am I ?	https://beta191038.w...	
	App Focus	Safari	com.apple.Safari	file:///Applications/Saf...	2020-02-13 23:55-42...	2020-02-13 23:53-24...	00:18	Preferiti ...	favorites://	
	App Focus	Telegram	ru.kesoper.Telegram	file:///Applications/Tel...	2020-02-13 23:52-16...	2020-02-13 23:52-16...	00:00:02			

Innometrics Transfer

Purtroppo non siamo stati in grado di inviare i dati raccolti al server perchè, seppur avendo creato un account sul sito relativo, ogni volta che abbiamo provato ad effettuare il login compariva un messaggio come quello in figura che non ci permetteva di continuare con l'invio.



Innometrics Collector

7. Commenti finali

Arrivati alla fine dello progetto possiamo dire che l'utilizzo di questi software ci ha agevolato molto nello sviluppo, in particolare siamo riusciti a mantere un'organizzazione costante e ordinata dello stesso. Tra questi, Taiga e GitLab sono diventati per noi strumenti essenziali.

Grazie a Taiga abbiamo avuto una mappa organizzata e sempre aggiornata delle azioni da svolgere, la quale ha permesso di coordinare il lavoro tra di noi: abbiamo aggiunto tutte le idee che ritenevamo potessero essere utili al nostro progetto, la maggior parte di queste sono state poi sviluppate, testate e rilasciate nel prodotto finale. Altre invece si sono rivelate non indispensabili o difficili da realizzare per mancanza di tempo e, quindi, scartate. Taiga, dunque, ci ha permesso di avere una visione generale di come il progetto stava prendendo vita, nonostante su GitLab avessimo usato branch diversi.

Git è stato fondamentale: con il suo utilizzo abbiamo portato avanti il lavoro in modo indipendente, ma pur sempre confrontandoci. La separazione dei branch di sviluppo in GitLab ci ha permesso di tenere parti di codice diverse separate. Quando commettevamo degli errori o apportavamo modifiche che erano in contrasto con il resto del progetto potevamo facilmente risalire a vecchie versioni e ripristinare il lavoro in modo che funzionasse.

L'utilità di SonarQube si è mostrata solo dopo che il progetto aveva iniziato a prendere vita. Lo abbiamo utilizzato per scovare in modo semplice bug ed imperfezioni nel codice. Quando SonarQube segnalava un bug, questo veniva riportato affinché qualcuno del gruppo lo risolvesse. Sui file JavaScript il suo utilizzo è stato un po' limitato, in quanto non riusciva a riconoscere tutti i framework JS da noi utilizzati. Tuttavia con HTML e CSS non ha avuto nessun problema e ci ha aiutato nel ridurre il riuso del codice: siamo riusciti a scrivere un foglio di stile CSS con pochissime righe duplicate ed integrarle nel codice HTML, anch'esso con poche righe duplicate. SonarQube ci ha anche aiutato segnalando tecnologie poco sicure o deprecate, come l'utilizzo di alert. Riteniamo sia uno strumento indispensabile per lo sviluppo in team di un progetto, in quanto permette a tutti gli sviluppatori di revisionare il proprio codice, capire efficacemente i propri errori, consultare una guida in caso di dubbi sulla risoluzione e, infine, apportare le dovute modifiche.

L'utilizzo di Mattermost non si è rivelato indispensabile, gli strumenti per la comunicazione sono potenzialmente tantissimi, ma nonostante ciò abbiamo cercato di utilizzare tutte le potenzialità che offriva, come ad esempio l'app mobile e la creazione di più canali di comunicazione. I diversi canali sono stati utilizzati in modo intelligente grazie alla separazione in funzione del tipo di comunicazione. Quando abbiamo trovato problemi o avuto nuove idee o sviluppato nuove funzionalità, abbiamo scritto sul canale dedicato. Questo ci ha permesso di dare più chiarezza della tematica della discussione e di tenere pulite e organizzate le chat.

L'utilizzo di tutte queste tecnologie insieme ha portato allo sviluppo di un applicazione ben costruita che non presenta in bug, un'app fluida e che soddisfa le specifiche richieste.

Possiamo dire, per concludere, che i software CAS sono stati utili ed efficaci nello sviluppo dell'applicativo. Ci hanno permesso di semplificare molte operazioni di organizzazione e di gestione del team con professionalità ed efficienza.

