

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

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.
- 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 che possa essere usato da turisti che visitano quei luoghi di interesse.

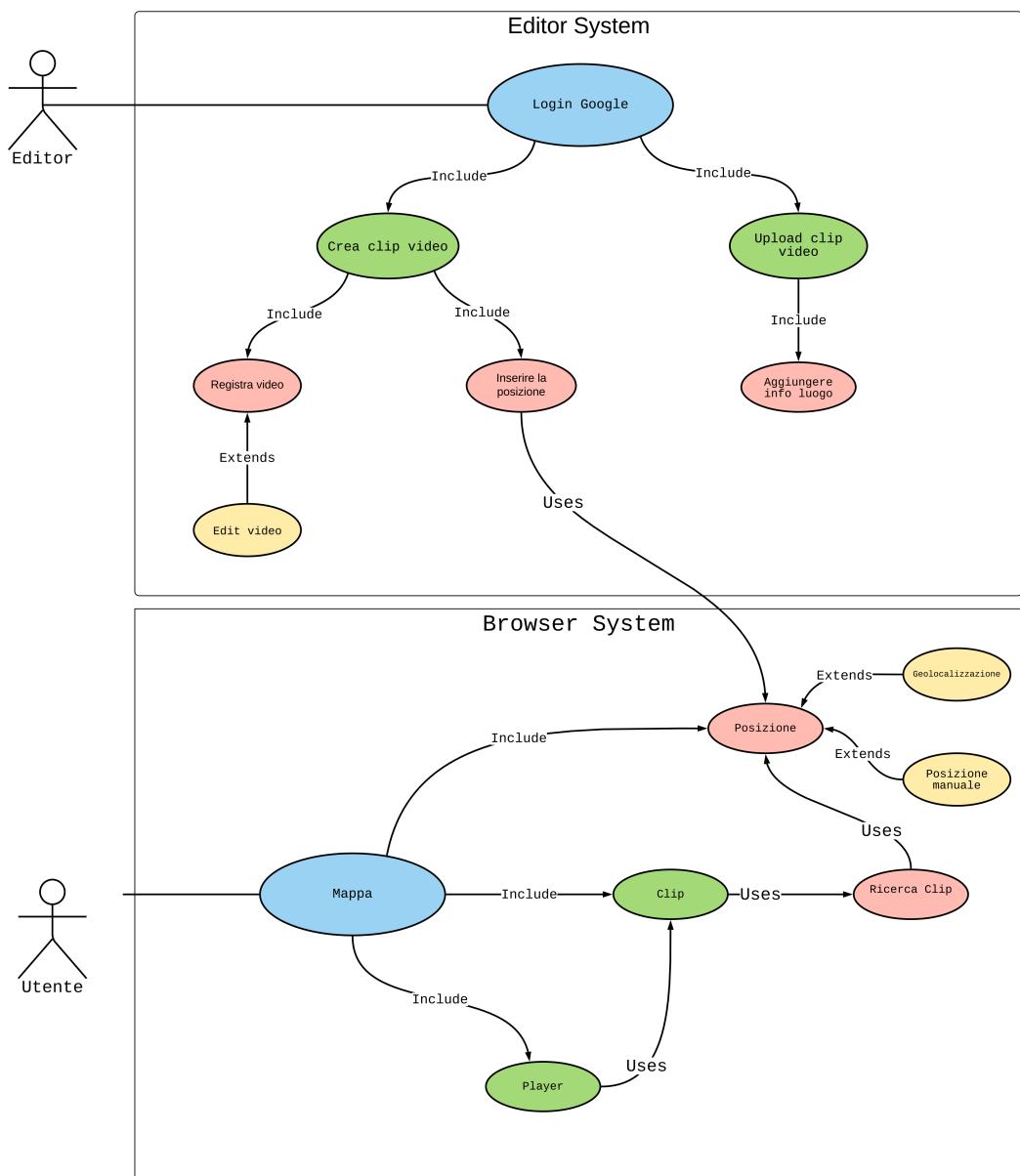


Grafico UML dei casi d'uso della web application.

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.

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'antempria 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.

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

Software CAS

Taiga

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**

The screenshot shows a user story card titled "#53 Descrizione Wikipedia". The card includes a summary: "Come utente voglio poter vedere la descrizione di un luogo tratta da Wikipedia". It lists "Related tasks" and "Attachments". The right sidebar displays project metrics: 0 UX, 0 Design, 0 Front, 13 Back, and 13 total points. It also shows the "Assigned to" field with Filippo Bartolucci, 1 Watcher (Filippo Bartolucci), and buttons for adding watchers and a trash bin.

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.

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.

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.

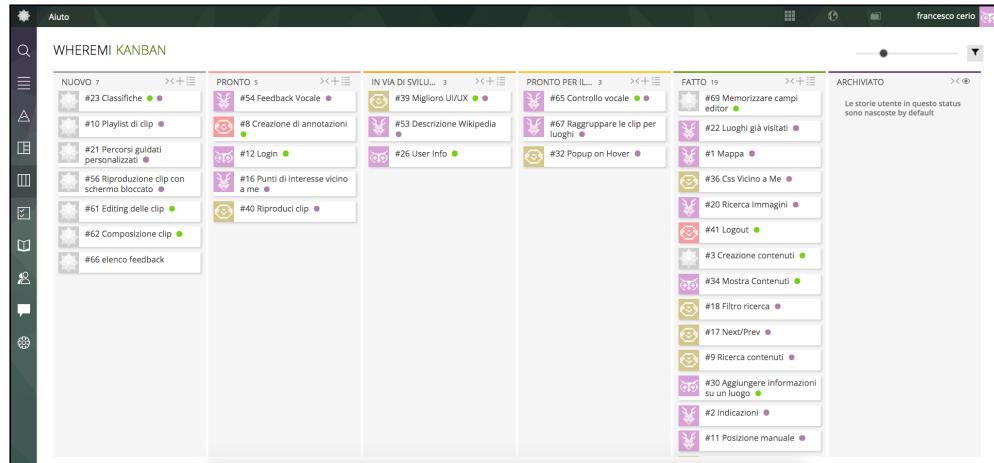
Epic user story

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.

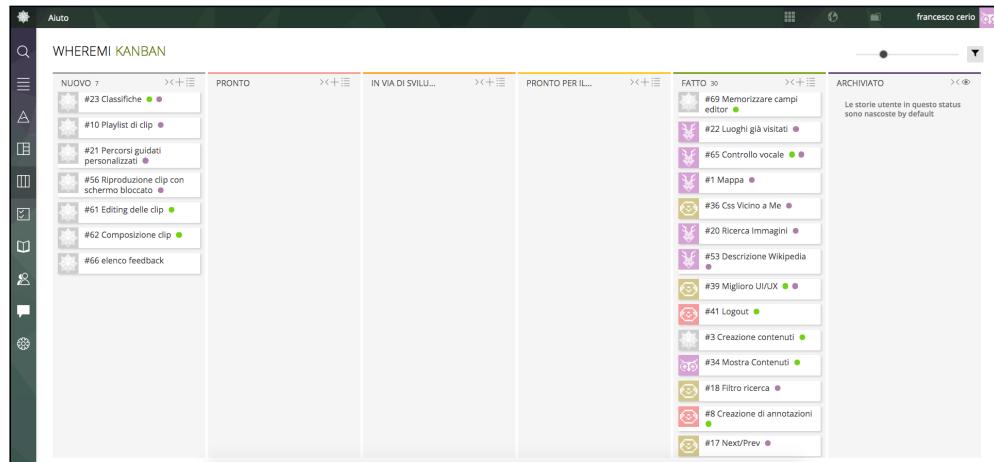
Per la gestione degli issue ci siamo affidati sempre a Taiga perché lo abbiamo trovato più che sufficiente per i nostri scopi, dal momento che per ogni issue per di assegnare tipo, gravità, priorità e status.

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.

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.

Issues

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	critical	low	▲ 0	#64 Alert wikipedia	Closed ▾	05 Mar 2020	Filippo Ba...
bug	critical	low	▲ 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	critical	low	▲ 0	#52 Visualizzazione rotta con display 2560*1440 e maggiori	Closed ▾	05 Mar 2020	Matteo C...
bug	critical	low	▲ 0	#51 Login non sempre funzionante	Closed ▾	05 Mar 2020	francesco...
bug	critical	critical	▲ 0	#38 Dettagli in nearMe	Closed ▾	12 Feb 2020	umberto ...
bug	critical	low	▲ 0	#37 NearMe pulsanti	Closed ▾	06 Feb 2020	umberto ...
bug	critical	low	▲ 0	#27 Zoom mobile	Closed ▾	09 Feb 2020	Filippo Ba...
bug	critical	low	▲ 0	#25 Indicazioni non vanno con posiziona manuale	Closed ▾	11 Dec 2019	Filippo Ba...
bug	critical	low	▲ 0	#24 Posizione attuale doppia	Closed ▾	11 Dec 2019	Filippo Ba...
bug	critical	low	▲ 0	#15 Redirect dopo login	Closed ▾	09 Feb 2020	francesco...
bug	critical	low	▲ 0	#14 Sovraposizione elementi da mobile	Closed ▾	09 Feb 2020	Filippo Ba...
bug	critical	low	▲ 0	#7 Icone navbar sovrapposte in landscape	New ▾	26 Nov 2019	 Not assig...
bug	critical	low	▲ 0	#6 Route di navigazione non si cancella	Closed ▾	11 Dec 2019	Filippo Ba...
bug	critical	low	▲ 0	#5 Indicazioni in lingua italiana	Closed ▾	11 Dec 2019	Filippo Ba...
bug	critical	low	▲ 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.

Sprint

Per lo sviluppo 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.

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 user story che riguardano la creazione e il caricamento dei video.

Risultato sprint

Alla fine del primo sprint avevamo una versione molto base dell'applicazione.

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 (blue, green, red) are placed on the map. The interface includes navigation controls (+, -, ▲, ▼) and a search bar at the top. Bottom navigation icons include WhereAMI, VicinoMe, Cerca, Categorie, and Profilo.
- 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 heading "I Miei Contributi". The bottom navigation bar includes 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 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 WhereAMI, VicinoMe, Cerca, Categorie, and Profilo.

Risultato primo sprint.

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

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. Below it, "Language: English". Under "Informazioni Clip", it lists: Purpose: Come, Content: Storia, Audience: Medie. Navigation icons are at the bottom.

Clip Information: Shows two clips: "SFPHF800+" and "Piazza Maggiore". Each clip has columns for Language (Italiano), Distance (771m, 262m), Audience (Agen, Medie), Purpose (Perché, Perché), and two buttons "Dettagli" and "Indicazioni". Below these, another section shows "PalazzoPepoli" with similar details.

Risultato secondo sprint.

Terzo sprint

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

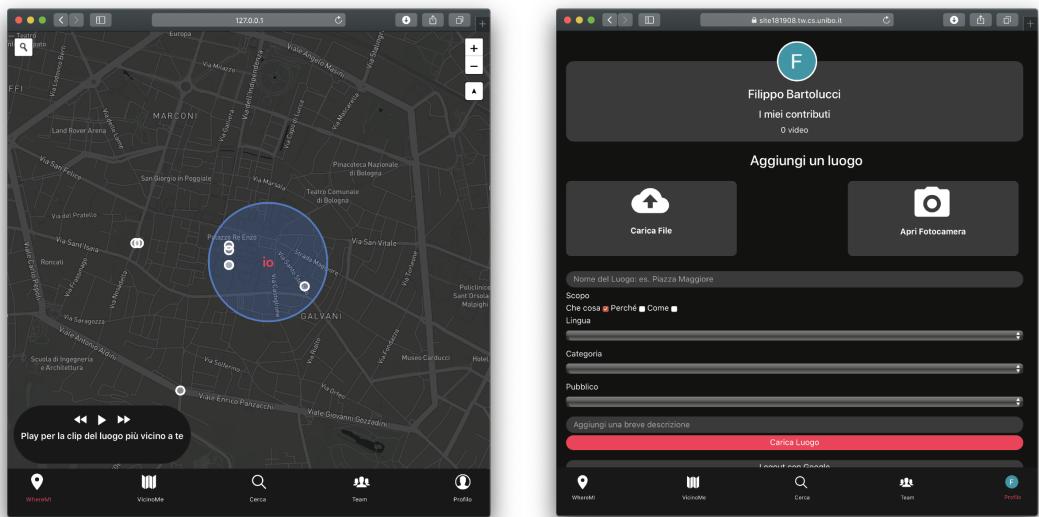
Inizialmente questo era stato programmato per essere lo sprint finale, dato che eravamo giunti a ridosso della data di scadenza per la presentazione del progetto. Ci siamo, quindi concentrati nell'aggiungere alcune delle feature facoltative, nel sistemare i bug e ripulire il codice secondo i report di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

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.

Quarto sprint

Durata: 20/02/2020 - 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.

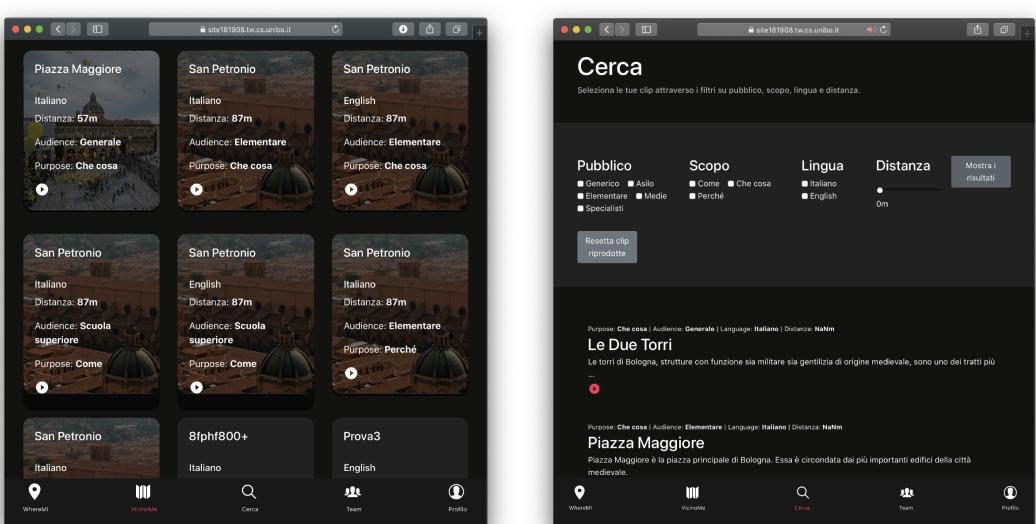
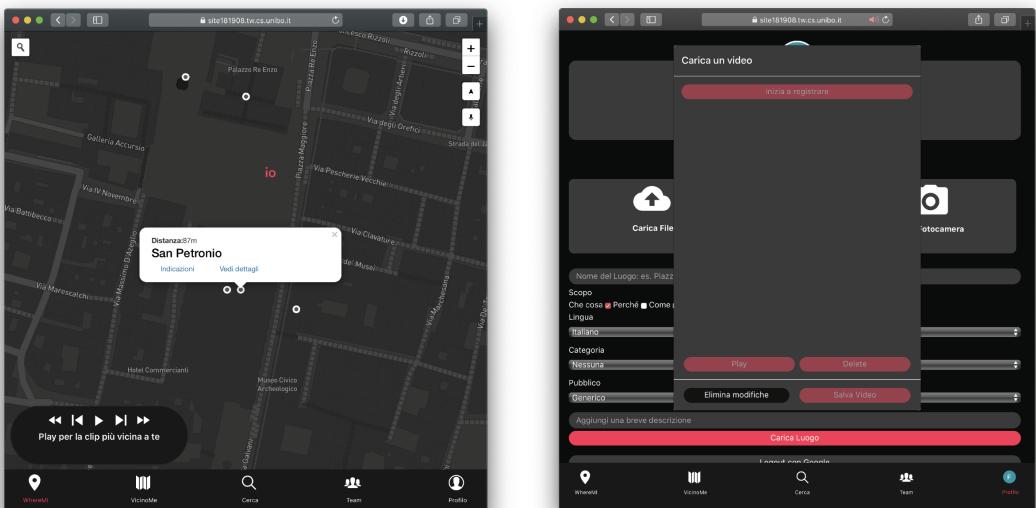


Quarto sprint

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

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 sistematati gli ultimi bug.



Risultato quarto sprint.

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.

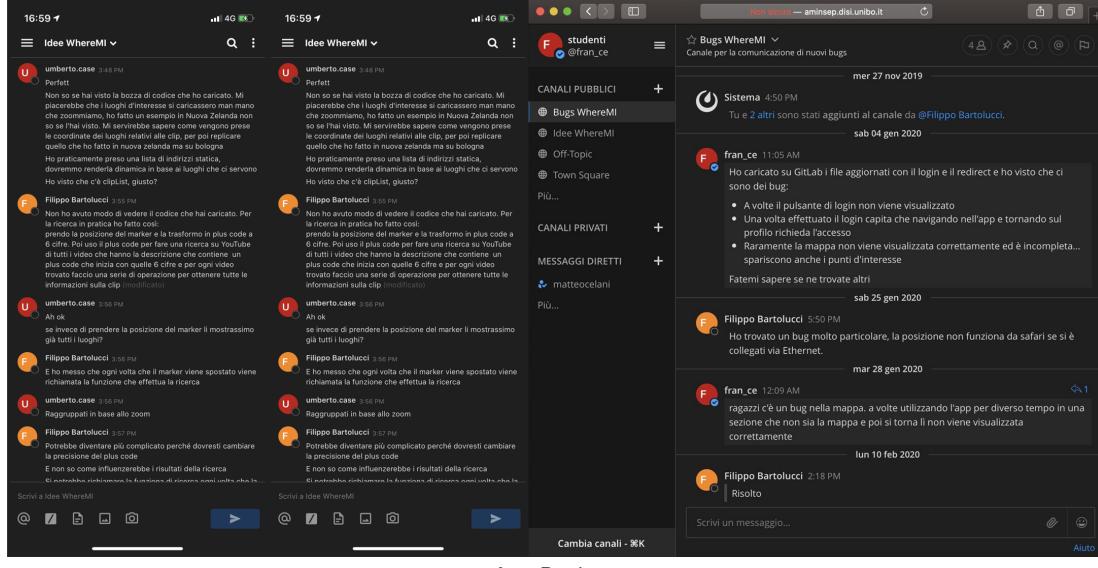
The screenshot shows the 'Active branches' section of a GitLab project. It lists three branches:

- master** (default, protected): Last commit was 6e39c69c - Relazione 1 hour ago. It has 55 commits and 0 merges. Actions: Merge request, Compare, Filter dropdown, Delete.
- correzioni-vitali** (merged): Last commit was 8b9e6f94 - Merge branch 'master' into 'correzioni-vitali' 1 month ago. It has 55 commits and 0 merges. Actions: Merge request, Compare, Filter dropdown, Delete.
- EditorSviluppo** (merged): Last commit was c546f23c - Merge branch 'master' into 'EditorSviluppo' 1 month ago. It has 170 commits and 0 merges. Actions: Merge request, Compare, Filter dropdown, Delete.

Divisione dei branches.

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

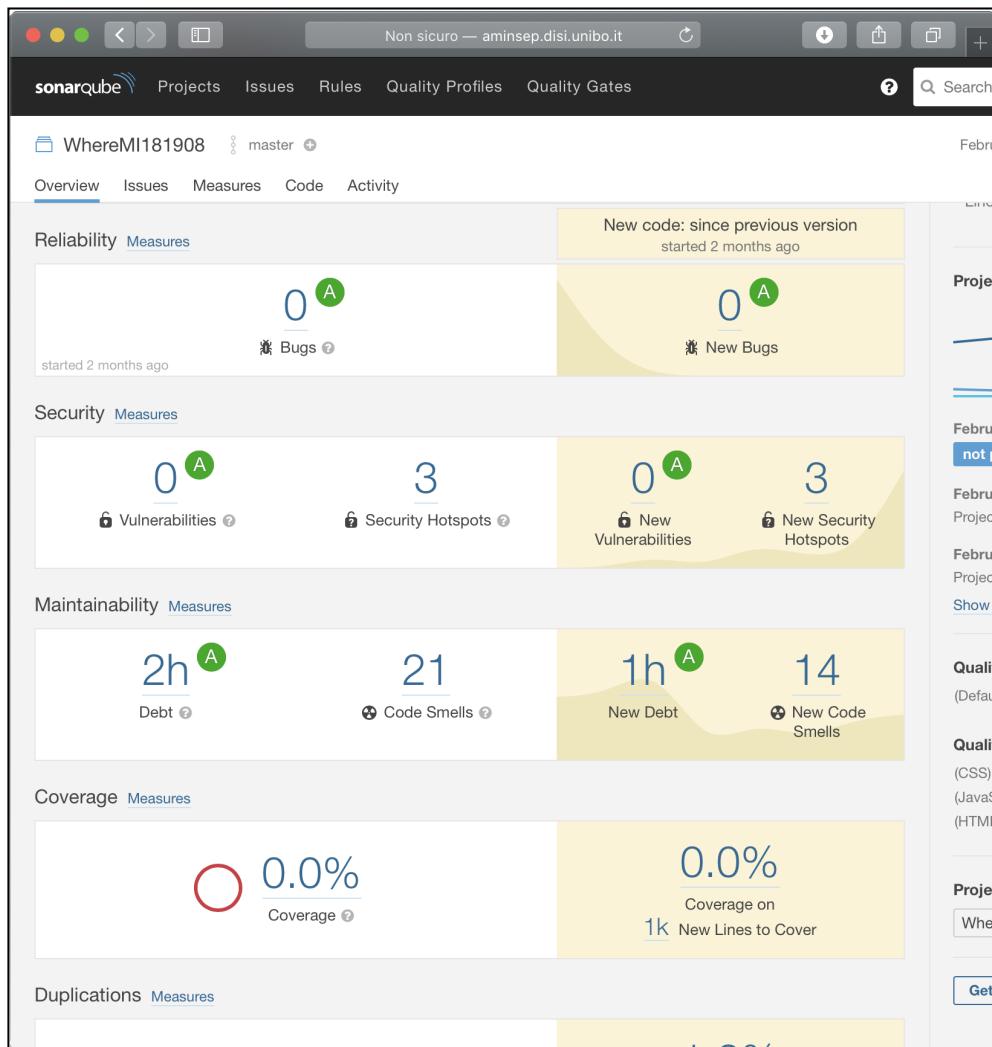
L'App mobile si è presentata molto intuitiva e semplice nell'utilizzo, come ogni applicazione di messaggistica. Sul sistema operativo Android non abbiamo riscontrato problemi di utilizzo.

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.

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 bug che non avevamo subito notato e risolverli: infatti, lo strumento rende possibile trovare quale porzione del codice contiene un bug e consultare una documentazione che spiega come risolverlo.

Writing cookies is security-sensitive

Security Hotspot Minor Main sources cert, cwe, owasp-a3, sans-top25-porous Available Since Nov 26, 2019 SonarAnalyzer (JavaScript) Constant/Issue: 5min

Writing cookies is security-sensitive

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.

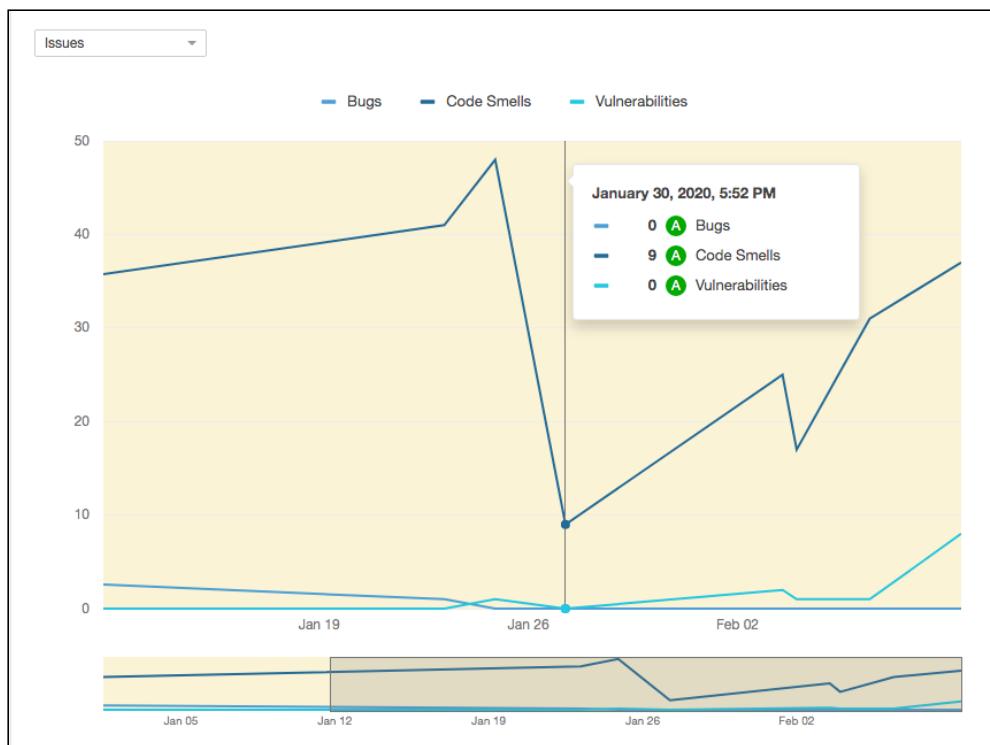
Ask Yourself Whether

- sensitive information is stored inside the cookie.

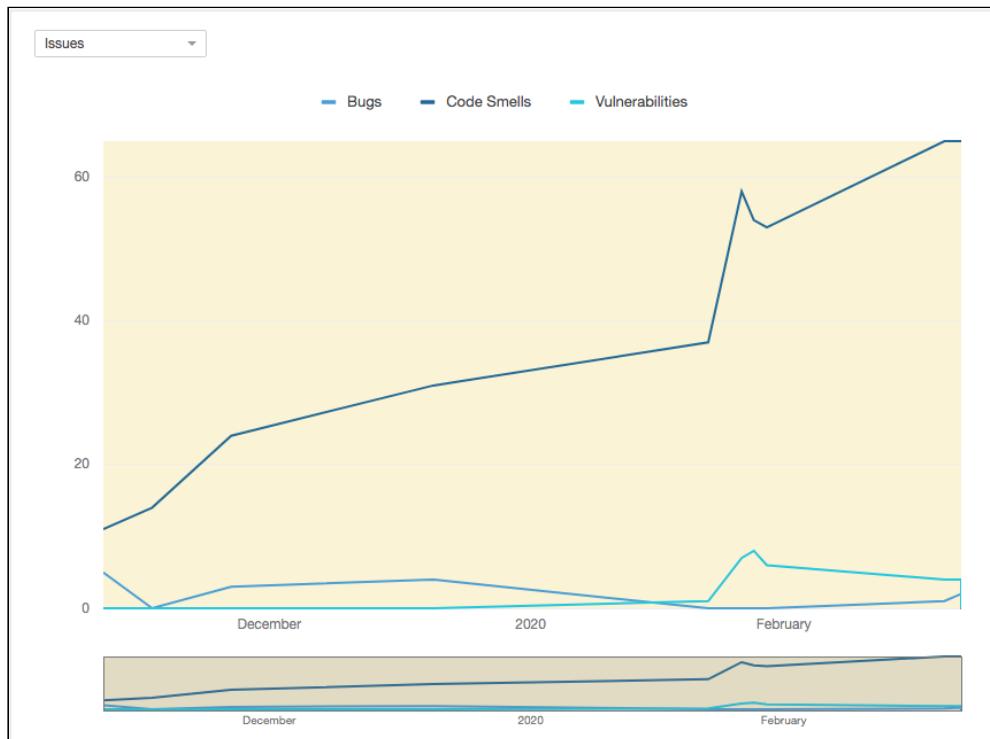
Esempio di consultazione della documentazione per la risoluzione di un bug

Schermata in cui viene presentato un bug

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

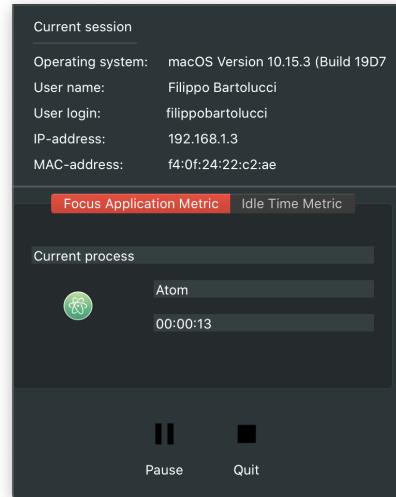
Abbiamo notato come è cambiato il grafico nel corso del tempo e di come, al crescere del numero di righe di codice, la presenza di code smell sia aumentata notevolmente. Molti di quelli riportati dallo strumento di analisi erano banali e abbiamo notato che se non rimossi o corretti, avrebbero compromesso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

Innometrics

Innometrics si compone di due applicazioni: Collector e Transfer.

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

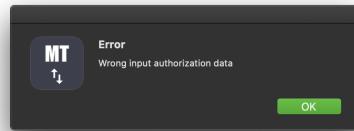
Transfer

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.

InnoMetrics		Metrics Preview (Last update: 18:45-29 05-03-2020)							Clear Database	
Metric Type	Name	Bundle	Bundle Path	Timestamp Start	Timestamp End	Duration	Browser Tab Title	Browser Tab Url		
Exclusion filters:										
Keywords										
Apps	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:32-02...	2020-02-23 23:32-27...	00:00:15	tutu bologn - Ric...	https://www.google.co...	
	idle	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:31-47...	2020-02-23 23:31-53...	00:00:06	Le due Tom: Garied...	https://www.bologn...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:31-47...	2020-02-23 23:31-53...	00:00:06	Le due Tom: Garied...	https://www.bologn...	
	idle	Atom	com.github.atom	/file/Applications/Ato...	2020-02-23 23:27-59...	2020-02-23 23:31-40...	00:00:41			
	App Focus	Atom	com.github.atom	/file/Applications/Ato...	2020-02-23 23:27-48...	2020-02-23 23:31-48...	00:00:58			
	App Focus	InnometricsTransfer	com.denzan.innometri...	/file/private/var/folder...	2020-02-23 23:27-31...	2020-02-23 23:27-48...	00:00:16			
	App Focus	Atom	com.github.atom	/file/Applications/Ato...	2020-02-23 23:27-03...	2020-02-23 23:27-26...	00:00:02	preprocess - Leaflet.j...	https://dataviewflow...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:59-29...	2020-02-23 22:59-29...	00:00:03	amryangf Easly add sp...	https://www.talter...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:59-03...	2020-02-23 22:59-05...	00:00:02	Preferiti	favorites://	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:58-28...	2020-02-23 22:59-03...	00:00:05	5 Voice Control JavaSc...	https://quehyouse...	
From	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:58-33...	2020-02-23 22:58-33...	00:00:00			
	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:18-28...	2020-02-23 22:28-28...	00:00:06	amryangf Easly add sp...	https://www.talter...	
	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:18-28...	2020-02-23 22:28-33...	00:00:05	amryangf Easly add sp...	https://www.talter...	
	App Focus	Overseer Helper	com.obersee.oh-ovr...	/file/Applications/Ove...	2020-02-23 22:17-23...	2020-02-23 22:17-23...	00:00:04	amryangf Easly add sp...	https://www.talter...	
	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:17-27...	2020-02-23 22:17-27...	00:00:03	amryangf Easly add sp...	https://www.talter...	
	App Focus	UserNotificationCenter	com.apple.UserNotice...	/file/System/Library/C...	2020-02-23 22:17-27...	2020-02-23 22:17-27...	00:00:02	System	https://quehyouse...	
	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:17-27...	2020-02-23 22:17-27...	00:00:06	Nova sched	chrome://newtab/	
	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:17-27...	2020-02-23 22:17-27...	00:00:04	Nova sched	chrome://newtab/	
	App Focus	Google Chrome	com.google.Chrome	/file/Applications/Geo...	2020-02-23 22:17-27...	2020-02-23 22:17-27...	00:00:04	Nova sched	chrome://newtab/	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:17-25...	2020-02-23 22:17-29...	00:00:04	amryangf Easly add sp...	https://www.talter...	
To	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:17-25...	2020-02-23 22:17-25...	00:00:10	Talter/anyangf Spee...	https://github.com/fal...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:17-25...	2020-02-23 22:17-25...	00:00:05	5 Voice Control JavaSc...	https://quehyouse...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:17-25...	2020-02-23 22:17-25...	00:00:05	voice-commands.js	https://jimmybyum.git...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:17-25...	2020-02-23 22:17-25...	00:00:05	voice-commands.js	https://jimmybyum.git...	
	App Focus	Finder	com.apple.finder	/file/System/Library/C...	2020-02-23 22:16-26...	2020-02-23 22:16-26...	00:00:03	amryangf Easly add sp...	https://www.talter...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 22:16-25...	2020-02-23 22:16-25...	00:00:03	amryangf Easly add sp...	https://www.talter...	
	App Focus	Finder	com.apple.finder	/file/System/Library/C...	2020-02-23 22:16-26...	2020-02-23 22:16-26...	00:00:01			
	App Focus	Xcode	com.apple.Xcode	/file/Applications/Xco...	2020-02-23 22:16-43...	2020-02-23 22:16-43...	00:00:07			
	App Focus	Term2	com.googlecode.Term2	/file/Applications/Term...	2020-02-23 22:16-43...	2020-02-23 22:16-43...	00:00:00			
	App Focus	Xcode	com.apple.Xcode	/file/Applications/Xco...	2020-02-23 22:16-43...	2020-02-23 22:16-43...	00:00:00			
Send Metrics	App Focus	Finder	com.apple.finder	/file/System/Library/C...	2020-02-23 23:54-24...	2020-02-23 23:54-25...	00:00:00	Where am I ?	https://192.168.0.8...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:01	Where am I ?	https://192.168.0.8...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:01	Where am I ?	https://192.168.0.8...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:01	Accedit - Account Go...	https://accounts.googl...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:05	Forwarding ...	https://accounts.googl...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:05	Where am I ?	https://192.168.0.8...	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:05	Preferiti	favorites://	
	App Focus	Safari	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:54-23...	2020-02-23 23:54-24...	00:00:10			
	App Focus	Telegram	ru.kaspersky.Telegram	/file/Applications/Teleg...	2020-02-23 23:52-16...	2020-02-23 23:52-16...	00:00:02			
	App Focus	com.apple.Safari	/file/Applications/Saf...	2020-02-23 23:52-16...	2020-02-23 23:52-16...	00:00:05	Preferiti	favorites://		

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

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 scoprire in modo semplice bug ed imperfezioni nel codice. Quando SonarQube segnalava un bug, questo veniva riportato affinché qualcuno del gruppo lo risolvesse. Un esempio pratico del suo utilizzo è l'aiuto che ci ha dato 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.