



Progettazione e realizzazione di Litto: un'applicazione per l'aiuto all'organizzazione e allo studio di interessi

04/02/2022

—

Introduzione

In questo progetto ho voluto creare **Litto**: un'applicazione che aiuta l'utente a portare a termine i propri obiettivi in modo organizzato e temporizzato. L'idea iniziale del progetto era fornire una possibile soluzione ad un problema frequente: iniziare un "obiettivo" e non rispettare i tempi prestabiliti, o addirittura non portarlo a termine. In questo caso il termine "obiettivo" acquisisce un significato più ampio: tutti quei propositi, azioni, impegni che hanno come scopo quello di essere completati in tempi più o meno definiti.

Durante la prima parte del progetto è stata validata ed integrata l'idea iniziale. Oltre all'idea, infatti, era necessario capire i comportamenti e le necessità di utenti reali di fronte alla realizzazione dei propri obiettivi. Dopo aver svolto alcune interviste, è stato possibile definire le personas a cui è rivolta l'applicazione, gli scenari in cui queste avrebbero utilizzato Litto e le caratteristiche funzionali che quest'ultima avrebbe dovuto avere per rispondere alle necessità degli utenti.

Nella seconda parte sono stati creati i mockups e i prototipi dell'applicazione andando a confrontare le necessità trovate precedentemente e la realizzabilità dell'applicazione. In questa fase è stato possibile applicare gli accorgimenti necessari per rendere usabile l'applicazione. Con l'implementazione finale, apportando alcune modifiche ai prototipi, è stata realizzata l'applicazione.

L'ultima fase è stata il test di usabilità che ha permesso sia di verificare se l'applicazione sviluppata fosse riuscita a raggiungere, sia se ci fossero criticità nell'usabilità dell'applicazione stessa. Dal test è stato possibile capire quali lavori futuri fossero necessari per rendere più user friendly l'applicazione.

Needfinding

Una volta individuato il problema iniziale, è stato necessario capire se tale problema è condiviso anche da altre persone iniziando una fase di needfinding. In particolare, in questa fase il focus si è incentrato sul capire il rapporto che le persone hanno con i propri obiettivi considerando: quali obiettivi ha ogni persona, quale è la routine che generalmente viene utilizzata per portarli a termine, quali sono le difficoltà che si incontrano durante la realizzazione, quali sono i principali motivi per cui non si raggiunge un obiettivo ed infine se un'applicazione potrebbe aiutare a completare un obiettivo.

Sulla base di questi interrogativi sono state fatte cinque interviste a persone appartenenti a categorie diverse: Leonardo, Laura, Noemi, Lisa e Grazia. Da queste interviste è emerso come si possano categorizzare queste persone secondo varie caratteristiche: come di solito completano gli obiettivi, se sarebbero disposti al possibile utilizzo di un'applicazione per portare avanti i propri obiettivi, se usano già un'applicazione simile a quella proposta, se è abituato a riservare del tempo ai propri obiettivi durante la settimana.

Personas

Sono state individuate 3 categorie principali che rappresentano le 3 personas su cui si focalizzerà lo sviluppo del progetto:

I. Leonardo (Primary Persona)



Leonardo è uno studente universitario che vive in famiglia. Ha molti interessi che cerca di portare avanti insieme allo studio. Questi generalmente riguardano attività di gruppo in cui ama confrontarsi, scambiare opinioni ed avere obiettivi comuni con gli altri membri. Ha anche passioni che porta avanti in modo autonomo, studiando e documentandosi principalmente online. Ha notato però che spesso non trova informazioni precise su ciò su cui è interessato e che portare avanti certe passioni autonomamente è meno stimolante e cadenzato. Al momento utilizza Discord per dubbi specifici e Internet in generale per interessi generici.

II. Lisa (Secondary Persona)



Lisa lavora da anni e ha una famiglia. I suoi interessi sono sia legati alla famiglia, sia personali. Porta avanti i propri obiettivi in modo molto preciso, ma il problema principale che riscontra è che generalmente non riesce a dedicare il tempo che vorrebbe ai suoi interessi perché deve mantenere la casa e deve lavorare; questo non accade in ferie e in momenti di tranquillità. Le piace condividere alcuni dei propri obiettivi con altre persone, sia in famiglia che al lavoro, mentre altri preferisce di no. Usa giornalmente mezzi tecnologici ma non per condividere e pubblicare niente; infatti, può essere attirata da una condivisione di obiettivi con altre persone ma probabilmente non ne farebbe un uso costante.

III. Grazia (Negative Persona)



Grazia è in pensione e si dedica principalmente alla famiglia. Ha sempre avuto molti interessi che ora riesce a portare avanti dedicandogli tutto il tempo necessario. Le piace condividere il suo tempo con le sue amiche e portare avanti con loro alcune delle sue passioni. Altre invece le porta avanti da sola informandosi, cercando materiale in libreria e leggendo libri. Le capita spesso di informarsi con dei post social sulla storia della sua città, ma non è molto interessata all'utilizzo della tecnologia.

Vision Statements e Scenari

Una volta individuate le primary e secondary personas, a cui quindi viene indirizzato lo sviluppo dell'applicazione, è stato necessario delineare alcuni scenari di utilizzo dell'applicazione. In particolare, l'enfasi è caduta su quegli scenari in cui l'applicazione avrebbe permesso alle personas di risolvere un problema legato al conseguimento di obiettivi. L'identificazione di questi scenari quindi si è suddivisa nelle seguenti fasi: individuazione del problema legato al tema degli obiettivi, definizione del vision statement con cui l'applicazione avrebbe potuto aiutare l'utente, studio di uno scenario in cui l'utente viene attivamente aiutato dall'applicazione. Di seguito viene riportato un esempio.

Problema: Leonardo ha un interesse da tanto tempo ma non sa da dove iniziare a studiarlo e come organizzarne lo studio

Vision Statement: L'applicazione dovrebbe aiutare Leonardo a creare un nuovo obiettivo e a deciderne l'organizzazione (cosa studiare, come suddividere i materiali di studio e in quali tempi). L'applicazione potrebbe fornire anche una lista di obiettivi già creati che quindi Leonardo non deve organizzare.

Context Scenario: Leonardo mentre cammina per andare all'università pensa spesso alla sua passione che non ha mai portato avanti e a come poter iniziare a studiarla. Apre Litto e cerca la parola chiave del suo interesse e trova velocemente dei percorsi brevi da intraprendere per avere un'infarinatura dei suoi concetti chiave. Il percorso che ha trovato è calibrato alle sue capacità iniziali e, una volta capita l'organizzazione, decide di iniziarlo. Leonardo appena inizia il percorso è molto contento perché finalmente ha trovato il materiale che non avrebbe mai scoperto da solo e l'applicazione gli permette facilmente di consultare tutto il percorso avendo un'interfaccia minimale e moderna. Leonardo, quindi, realizza il proprio piccolo obiettivo in qualche settimana ed è pronto per iniziarne un altro, avendo acquisito una conoscenza di base e cercando di andare in profondità. L'app gli consiglia altri percorsi correlati e Leonardo ne inizia subito un altro.

Gli altri scenari principali identificati sono:

- Lisa è appena tornata a casa dal lavoro e vorrebbe dedicarsi ai suoi interessi. È molto stanca e fra poco dovrà dedicarsi alla famiglia. Lisa non fa niente perché sa che questo interesse le porterebbe via molto tempo. Lisa però si ricorda dell'applicazione in cui ricerca il proprio interesse. Scopre che ci sono alcuni piani che richiedono durata limitata e che sono già organizzati, almeno Lisa deve occuparsi solamente di portare avanti il proprio interesse invece che organizzarlo. Lisa dedica circa 15 min al giorno e riesce a completare l'obiettivo in circa 2 mesi.
- Alla prima lezione del primo anno, Leonardo è online (o a causa pandemia o per altri suoi motivi). Il professore invita tutti gli studenti a seguire il corso sulla pagina Moodle e gli consiglia di utilizzare Litto in cui il materiale del corso è suddiviso temporalmente. Leonardo scarica l'app, si registra velocemente e trova in modo semplice il corso del professore (del suo corso di laurea). Nota che la pagina del corso è ben organizzata con varie sezioni ed in particolare, essendo curioso dei temi del corso, vede che nelle

varie settimane sono riportate anche risorse fornite dagli studenti passati (libri, appunti, tracce di esame ecc). Decide poi di aggiungere una foto al suo profilo in modo che gli altri ragazzi, appena torna all'università, lo riconoscano. Appena il professore comunica le date d'esame, Leonardo si organizza con i suoi amici e si danno una scadenza per preparare l'esame dandosi sotto obiettivi comuni. In questo modo tutti possono tener traccia del loro livello di preparazione.

Requisiti Funzionali

Una volta delineati gli scenari di utilizzo dell'applicazione sono stati identificati i requisiti funzionali dell'applicazione. In particolare, i seguenti requisiti sono quelli principali, che poi verranno espansi nella sezione successiva riguardante il prototyping:

1. Registrare un nuovo account utente dalla pagina iniziale
2. Mostrare una pagina utente
3. Creare un nuovo obiettivo dalla pagina home
4. Cercare un obiettivo tramite keywords dalla pagina home
5. Gestire un obiettivo tramite la pagina dedicata

Prototyping

In questa fase sono stati analizzati tutti i requisiti funzionali e tradotti in un'interfaccia utente. In prima istanza è stato creato un mockup e successivamente è stato creato un prototipo dell'applicazione, in entrambi i casi utilizzando il software Figma e nel secondo caso utilizzando la libreria di componenti Taiga UI integrata anche in Angular, framework con cui è stata sviluppata l'applicazione vera e propria. Nello sviluppo, per rendere l'applicazione più moderna e più piacevole da usare, è stata scelta la libreria Taiga UI in quanto più vicina agli standard Material Design 3 rispetto alla libreria Angular Material che rispetta gli standard Material Design 2. La creazione dei due stadi di design, mockup e prototipo, ha permesso di studiare più approfonditamente la disposizione delle componenti dell'applicazione e di applicare le buone pratiche della Human Computer Interaction necessarie a creare un'interfaccia utente usabile e coerente. Infatti, questi due step di prototipazione hanno fatto emergere alcune criticità sia di disposizione delle componenti, sia di usabilità (per esempio l'elevato excise in alcuni scenari, la mancanza di affordances e signifier sufficienti) che sono state prontamente corrette prima della fase di sviluppo.

Nella fase di prototipazione sono stati scelti i colori principali dell'applicazione. Sono stati scelti come colori primario e secondario quelli dei bottoni di *Esempio di utilizzo dei colori primario e secondario* rispettivamente a destra e a sinistra in quanto sono colori che forniscono vivacità ma permettono comunque un utilizzo prolungato dell'applicazione essendo colori tenui.



Figura 1 Esempio di utilizzo dei colori primario e secondario rispettivamente a destra e a sinistra

La creazione dei prototipi si è incentrata in particolare sul rendere fruibile un obiettivo. In questo caso il concetto di obiettivo è integrato all'interno del piano: una risorsa in cui sono riportati i materiali necessari allo studio del piano suddivisi in "step" di una settimana. Ogni step ha un sottoobiettivo da raggiungere e una lista ordinata di materiali da consultare. Ogni utente può creare un suo piano personalizzato o seguire un piano già creato da altri utenti.

Nella maggioranza delle pagine riportate è presente la footer bar di Figura 2, il principale elemento per navigare all'interno dell'applicazione.



Figura 2 Esempio di footerbar

In particolare, nel prototipo, e poi nell'implementazione reale, questa è composta solamente da icone che risultano sufficientemente esplicative ed evitano l'appesantimento di ulteriori scritte. Infatti, tutte le icone rappresentano un mapping diretto con i concetti di: homepage, obiettivi attivi, creazione di un piano e user page.

Ognuno dei requisiti identificati in precedenza ha richiesto lo sviluppo di una o più pagine dedicate all'interno dell'applicazione, in conformità al Single Responsibility Principle.

1. Registrare un nuovo account utente dalla pagina iniziale

In questa pagina l'utente deve essere in grado sia di fare log in, sia di registrarsi all'interno dell'applicazione, e in quest'ultimo caso di fornire sia le informazioni personali necessarie, sia di selezionare i propri interessi così da permettere all'app di consigliare da subito alcuni piani pertinenti.

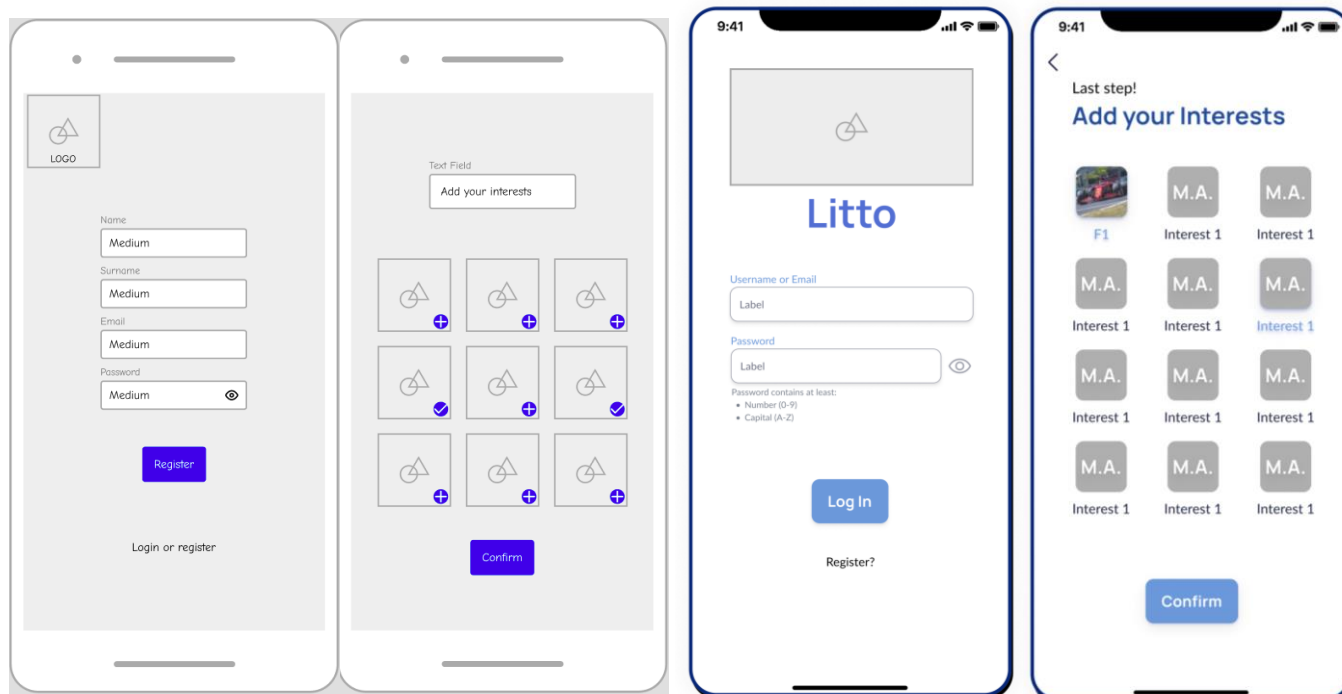


Figura 3 Mockups e prototipi della fase di registrazione/login e scelta degli interessi

2. Mostrare una pagina utente

Nella pagina utente è possibile avere una overview delle proprie informazioni generali e soprattutto dei propri interessi, dei piani completati e del livello generale che si ha.

Ogni interesse aggiunto ha un livello, calcolato sulla base di quanti piani riguardanti quell'interesse sono stati completati dall'utente. In questo modo l'utente può avere una panoramica generale intuitiva sui suoi successi e sugli aspetti da migliorare.

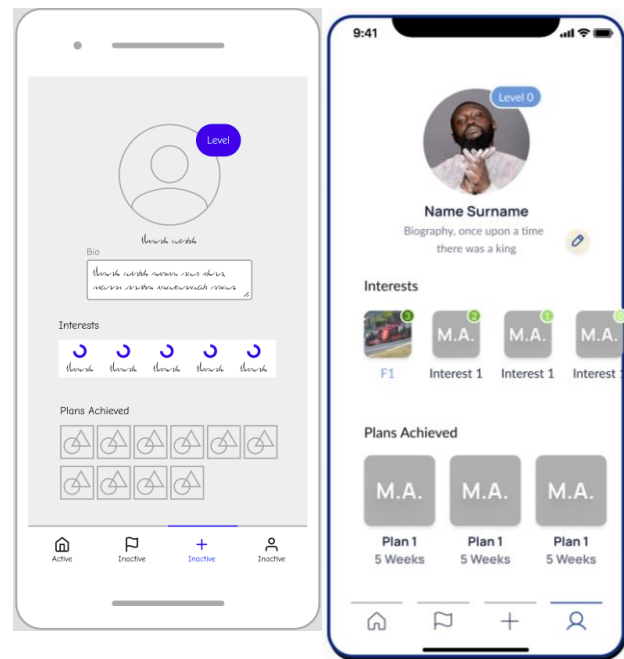


Figura 4 Mockup e prototipo della pagina utente

3. Home page

In Figura 5 a destra è possibile vedere la prima versione della home page in cui sono presenti: una barra di ricerca che permette di cercare velocemente un piano, la lista degli steps da completare nella settimana corrente, una overview più dettagliata degli steps da completare ed infine la lista dei piani consigliati sulla base dei propri interessi.

Nel prototipo è stata rimossa l'overview dettagliata dello step, poiché risulta una ripetizione di quanto riportato sopra. Cliccando sullo step corrente si apre la pagina di overview dello step in cui si può consultare il relativo materiale, mentre cliccando sul recommended plan si apre la pagina di overview di tutto il piano.

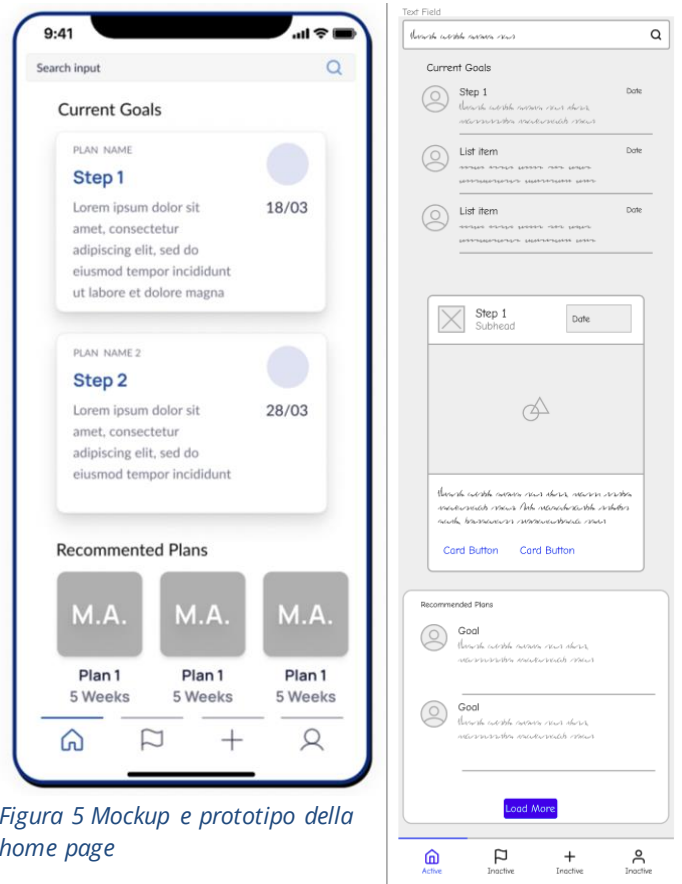


Figura 5 Mockup e prototipo della home page

4. Creare un nuovo piano dalla pagina home

La fase di creazione del piano è risultata una delle più complesse da realizzare in quanto l'utente per creare un piano deve cercare, organizzare ed aggiungere il materiale necessario. Nel mockup sono state disposte solamente le componenti principali necessarie a creare un nuovo piano, mentre nel prototipo è aumentato notevolmente il livello di dettaglio.

Nel prototipo è possibile vedere come il piano sia suddiviso in "step" ed all'interno di ogni step siano presenti vari materiali da consultare (video, pagine web, ecc). Questi possono essere aggiunti al piano con il menù a tendina "add material" che permette di scegliere quale materiale aggiungere. A seconda del materiale selezionato appaiono caselle di testo differenti (per esempio link a YouTube e caselle di testo in cui scrivere liberamente come in Figura 7). I materiali aggiunti possono essere quindi ordinati con il componente apposito.

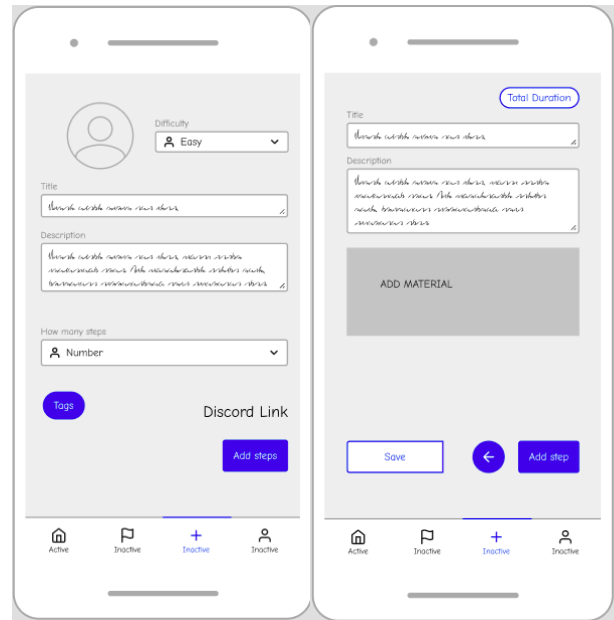


Figura 6 Mockup della creazione del piano

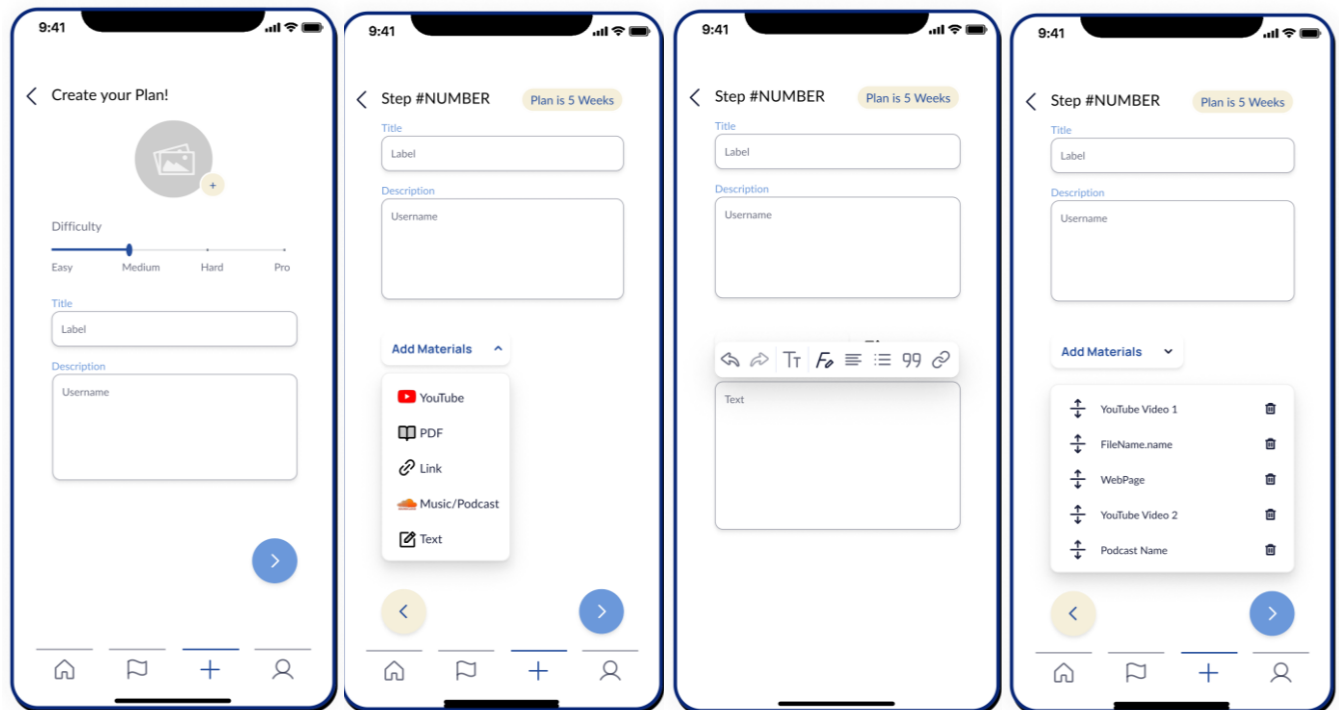


Figura 7 Prototipo della creazione del piano, con aggiunta del materiale "testo" ed esempio di riordinamento

5. Ricercare un piano tramite keywords dalla pagina home

Come riportato in precedenza, dalla pagina home è possibile ricercare un piano o un tag (parola chiave del piano) o il nome di una pagina (feature non implementata nell'applicazione reale). Per ogni piano è riportata anche la durata, in modo che l'utente abbia subito un'idea se quel piano può interessargli anche considerandone la durata. La ricerca tramite tag può permettere all'utente di trovare piani a cui non avrebbe pensato autonomamente.

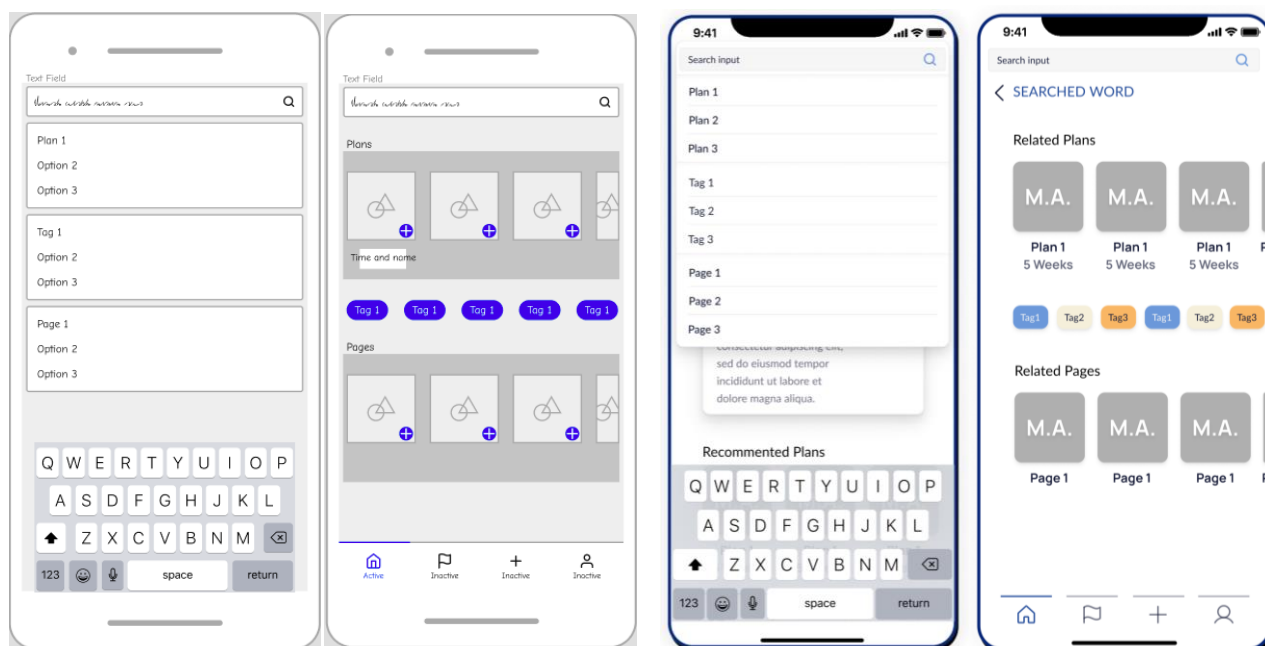


Figura 8 Mockup e prototipo dell'interfaccia di ricerca e della pagina che ne mostra i risultati

6. Gestire un piano tramite la pagina dedicata

Il piano, come scritto precedentemente, è la risorsa principale dell'applicazione. Questo contiene l'obiettivo che l'utente vuole completare ed è suddiviso in sezioni (steps) da completare settimanalmente. Ogni step contiene a sua volta vari materiali da consultare durante la settimana.

La pagina iniziale del piano offre una breve overview della durata e della sua suddivisione in steps. Nel prototipo di Figura 9, quest'ultima componente viene resa tramite un carosello che permette una navigabilità molto intuitiva e veloce fra gli steps, anziché tramite una lista.

Una volta che l'utente inizia il piano può scegliere quando far iniziare e finire il piano e soprattutto ha accesso alla consultazione dello step della prima settimana.

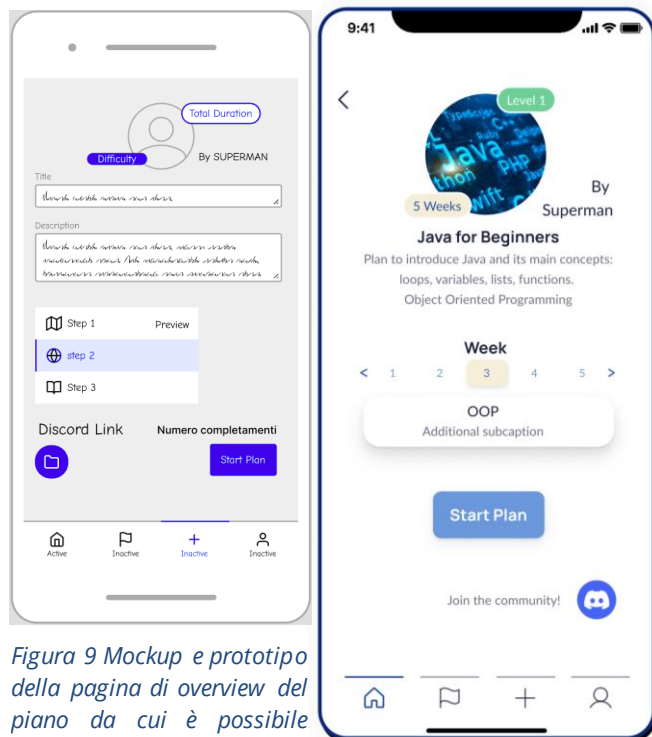


Figura 9 Mockup e prototipo della pagina di overview del piano da cui è possibile iniziare il piano stesso

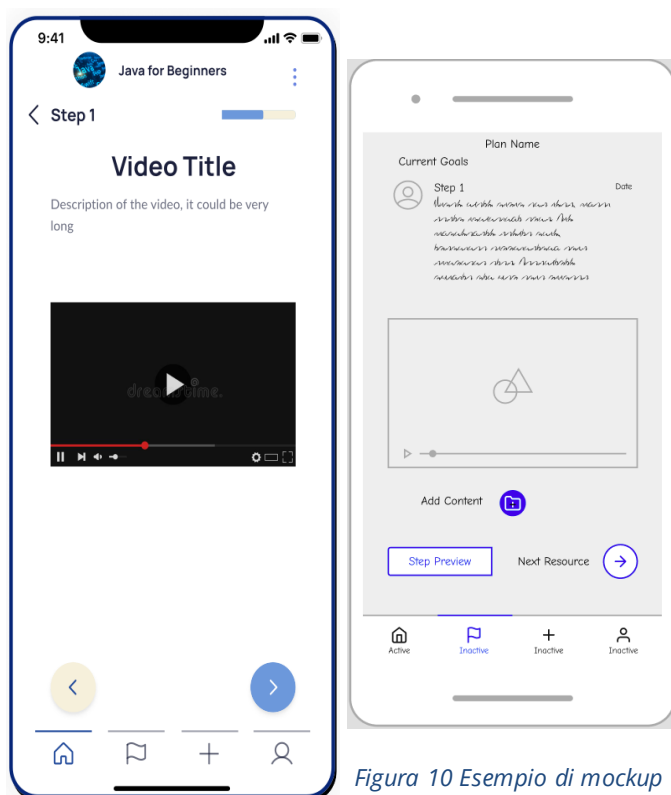


Figura 10 Esempio di mockup e prototipo della pagina di consultazione del materiale di un piano. In questo caso di un video YouTube

È stata predisposta una schermata aggiuntiva per la consultazione degli steps. Qui, infatti, vengono mostrati i vari materiali disponibili, permettendo di navigare fra loro con le frecce predisposte e, grazie alla progressbar, l'utente può facilmente capire quanto manca dal termine del piano. In queste schermate la parte rilevante è il materiale in sé, che occupa la parte principale dello schermo e che deve ricalcare fedelmente l'utilizzo giornaliero che viene fatto del certo materiale. L'utente, arrivato all'ultimo materiale dello step, può completare lo step e proseguire con il successivo. Quando lo step completato è l'ultimo del piano, questo verrà aggiunto ai piani completati dall'utente nella sua user page.

Sviluppo

Implementazione

In questa fase è stata sviluppata l'applicazione partendo dai prototipi realizzati durante la fase precedente. Il framework utilizzato è Angular con la libreria di componenti Taiga UI, come riportato in precedenza. Poiché non è possibile compilare un'applicazione mobile nativamente in Angular, è stato necessario integrare il progetto in Ionic che, utilizzando Capacitor, ha permesso di compilare l'applicazione sia per Android che per iOS. L'applicazione si interfaccia con un server Flask creato ad hoc, in grado di rispondere correttamente a tutte le richieste HTTP eseguite dal frontend. In questo caso il backend è molto semplice, infatti non è stato integrato un vero e proprio DBMS ma i dati scambiati sono stati memorizzati su dei dizionari Python. Poiché l'applicazione ed il server generalmente non sono connessi sotto la stessa rete, il server è stato esposto tramite il servizio Ngrok che permette di fare tunneling ed esporre un URL temporaneo da cui raggiungere il server. Il codice del progetto può essere consultato nella pagina GitHub <https://github.com/loremacchia/litto-ui>, in particolare nella cartella src.

L'applicazione creata è una Single Page Application, ovvero la pagina mostrata all'utente è sempre la stessa ma dinamicamente vengono modificate le componenti al suo interno, questo è reso possibile da Angular utilizzando tags HTML specifici. Lo schema dell'applicazione quindi si basa su "components" contenenti sia codice HTML da mostrare all'interno della pagina renderizzata, sia codice TypeScript per interfacciarsi con la vista e gestire input dall'utente ed interazioni con il backend. Queste ultime erano svolte utilizzando i "services", classi TypeScript predisposte solamente a fare richieste HTTP al server e ritornare le risposte secondo gli standard attesi dalla componente. Per fare ciò sono stati introdotti anche dei "model", oggetti che rappresentano dei Data Transfer Object e che permettono di gestire un dato di ritorno come un oggetto appartenente ad una classe definita e non come semplice file Json.

Avendo una panoramica del funzionamento, è possibile vedere come l'applicazione sia stata sviluppata seguendo il paradigma Model View Controller: in ogni componente è presente sia la componente View (HTML+CSS) sia la componente Controller (TypeScript), mentre il Model è rappresentato dal backend che gestisce i dati ricevuti e risponde alle richieste del controller.

Per minimizzare le richieste HTTP e per soddisfare i requisiti REST, in cui il backend è stateless, è stato usato il localStorage fornito da Angular. In questo modo vengono salvate in memoria locale le informazioni intermedie quando l'utente sta creando un piano o il suo profilo e nello stesso modo viene salvato l'ID dell'utente connesso.

Interfaccia

L'implementazione dell'interfaccia è molto simile al prototipo mostrato in precedenza. Sono comunque state fatte alcune modifiche sia dettate dalla realizzabilità utilizzando Taiga UI sia per migliorare l'usabilità dell'applicazione.

Per esempio, nella fase di aggiunta del materiale sono stati aggiunti il pulsante "Add" e vari bottoni con suggerimenti per rendere più chiara l'aggiunta del materiale e le relative informazioni necessarie. Inoltre, è stato introdotto il bottone Finish Editing per evitare di decidere la durata del piano come prima cosa, ma dando la possibilità di completare il piano una volta che i materiali necessari fossero completamente aggiunti.

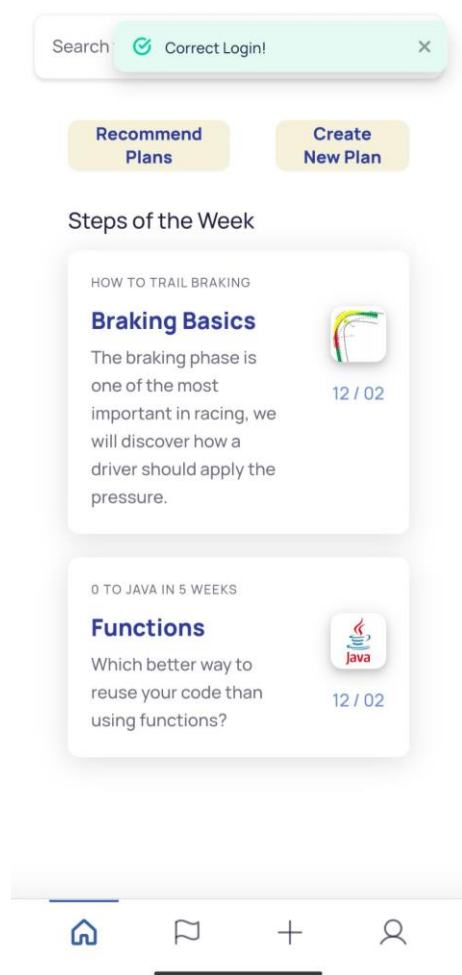


Figura 12 Implementazione finale dell'interfaccia della schermata home e della notifica push di login corretto

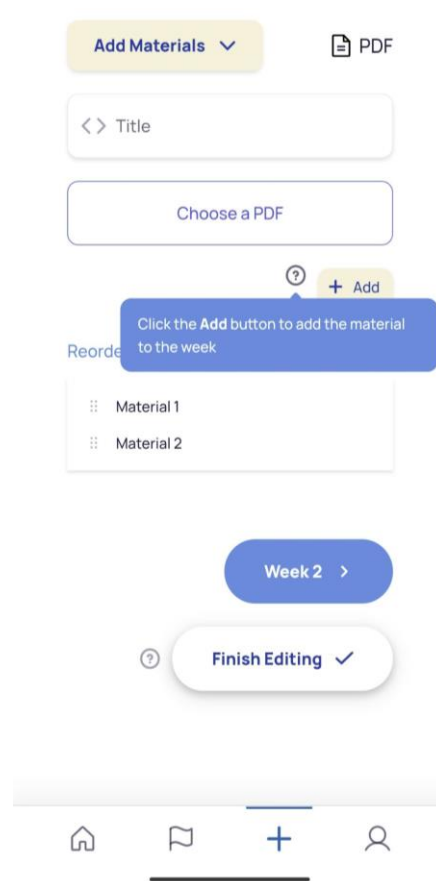


Figura 11 Implementazione finale della schermata di aggiunta dei materiali al piano

Sono state aggiunte delle notifiche push per fornire maggiore feedback all'utente. Per esempio, nella fase di login o nella fase di creazione di un piano se alcune informazioni inserite sono errate o non sono sufficienti, viene mostrata una notifica di errore. Mentre quando viene completato un piano o il login viene fatto correttamente viene mostrata una notifica di success.

Anche nella schermata Home sono state apportate alcune modifiche all'interfaccia di ricerca e aggiungendo i due bottoni "Recommend Plans" e "Create New Plan". Soprattutto il primo bottone permette agli utenti di trovare piani a cui sono interessati senza cercarli per parole chiave, l'utilizzo dell'applicazione diventa quindi molto più efficiente.

Sono state modificate anche le varie interfacce di fruizione del piano, di aggiunta degli interessi e di ricerca. Sono riportate in Figura 13 e Figura 14.

Figura 13 Implementazione finale dell'interfaccia della schermata iniziale, della schermata di scelta degli interessi e di ricerca

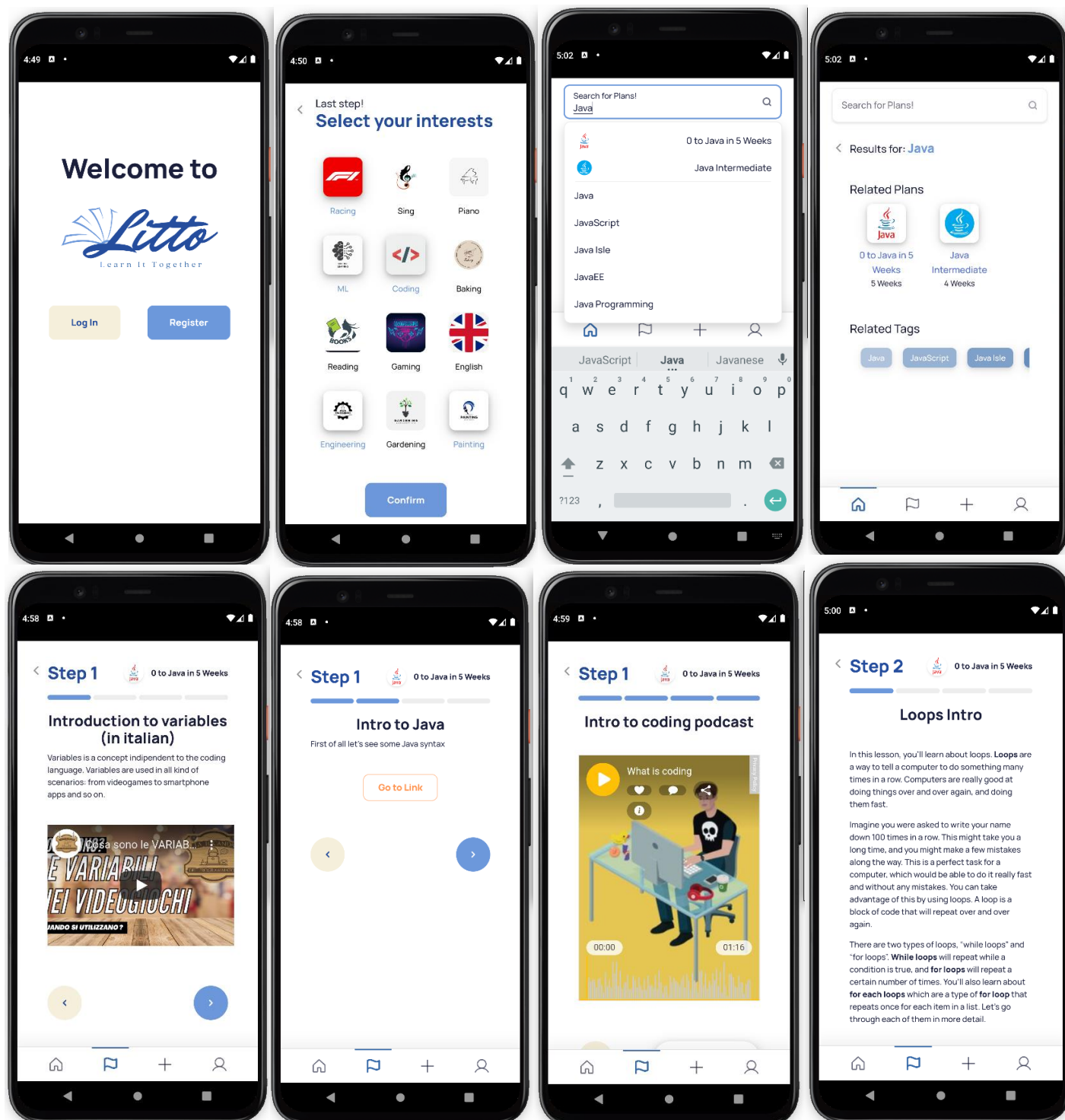


Figura 14 Interfaccia finale di consultazione dei materiali di un piano. Da sinistra troviamo: video Youtube, link ad una pagina web, podcast Spreaker e testo scritto

Usability Testing

Per verificare che l'applicazione fosse correttamente utilizzabile e che i goals definiti nella fase di need finding fossero raggiunti, è stata svolta una fase di usability testing. In particolare, vista la natura dell'app, in questo test il focus è ricaduto sul testare gli aspetti di learnability, memorability e satisfaction dell'applicazione. Ricordando anche i requisiti funzionali trovati nella fase di need finding, sono stati individuati i goals su cui basare il test, ovvero cosa si vuole comprendere dallo svolgimento dei test:

- L'utente riesce ad apprendere nuove nozioni usando l'applicazione?
- L'utente riesce a organizzare le proprie conoscenze in un piano?
- L'utente riesce a trovare le informazioni che cerca?
- L'utente riesce ad usare l'applicazione rilassandosi?
- L'utente riesce ad usare l'applicazione facilmente?
- L'interfaccia è sufficiente a soddisfare le necessità dell'utilizzo?

Per comprendere al meglio questi aspetti è stato scelto l'uso delle Single Ease Questions all'interno del test. Ogni domanda riportata presentava un punteggio da 1 a 7 in cui 1 significava fortemente in disaccordo mentre 7 completamente d'accordo. Le domande proposte ad ogni candidato sono le seguenti:

1. Sono riuscito a consultare i materiali del piano senza problemi
2. Il modo di proporre i materiali all'interno del piano non mi piace
3. La ricerca di un piano è confusionaria
4. Sono riuscito a trovare il piano che mi interessava riguardo alla programmazione
5. Aggiungere il materiale al piano è stato difficile
6. Vorrei poter aggiungere tipi diversi di materiale al piano rispetto a Testo, PDF, YouTube, Spreker e Link
7. Ho trovato utile poter riordinare i materiali del piano durante la creazione
8. Ho trovato chiara la pagina di riepilogo dell'utente
9. È stato veloce registrarsi nell'applicazione
10. La seconda volta è stato più facile trovare ed iniziare un piano
11. La navigabilità dell'app non è chiara
12. I suggerimenti all'interno dell'applicazione non mi sono stati d'aiuto
13. In generale sono soddisfatto dell'applicazione

Per testare l'applicazione in modo completo sono stati predisposti vari task da svolgere. L'utente avrebbe dovuto:

- fare login con credenziali conosciute
- cercare, iniziare e completare uno step di un piano riguardante un certo ambito
- creare un piano riguardante una passione del candidato
- fare logout e registrarsi con un proprio account
- testare che il piano fosse creato correttamente

I task sono principalmente open ended e resi come scenario tasks, permettendo all'utente di utilizzare l'applicazione in modo naturale pur svolgendo task delineati dal moderatore per un test completo. I primi tre task si incentrano sul valutare la learnability dell'applicazione mentre gli ultimi due ne valutano anche la memorability.

Pilot Test

Per verificare che il test potesse essere svolto correttamente, è stato predisposto un Pilot Test precedente. Al candidato è stato proposto un test identico a quello proposto nello usability test. Durante lo svolgimento del test sono emerse svariate criticità nell'uso dell'applicazione che hanno più volte richiesto un supporto da parte del moderatore. Le criticità principali individuate sono state: non sempre l'applicazione è risultata reattiva a volte impedendone completamente l'utilizzo, non è facile capire da cosa dipenda la lunghezza del piano, una volta che l'utente ha fatto il login veniva condotto alla user page e non alla home page. Tutte queste criticità sono state patchate per quanto possibile nella versione finale, fornita per i test di usabilità. Inoltre, si è dimostrata utile una spiegazione generale dell'applicazione prima di somministrare il test.

Usability Test

Questo test è stato svolto fornendo ai candidati una breve spiegazione dell'applicazione, la lista di task da svolgere ed un link al form contenente le SEQ. Il test è stato principalmente unmoderated ma il moderatore rimaneva nelle vicinanze per errori dell'applicazione fortemente impattanti sull'esecuzione del test.

Il test è stato somministrato a quattro persone equamente distribuite fra la primary e la secondary personas e le risposte alle SEQ sono state fornite su Google Forms. In Tabella 1 sono riportate le domande proposte con le medie e le varianze degli esiti. Per valutare le medie bisogna ricordare che lo 0 rappresenta il disaccordo mentre il 7 l'essere d'accordo, mentre per le varianze minori sono e più i partecipanti sono concordi nelle risposte fornite.

Considerando gli obiettivi iniziali del test l'esito generale è risultato soddisfacente, ma sono necessarie alcune considerazioni. Durante il test sono stati riscontrati problemi nel premere alcuni bottoni dovuti alla conversione da Web App ad App Android tramite Ionic, e altri

problemi nella visualizzazione di certe componenti in dispositivi con schermi più piccoli. Questi due aspetti hanno condizionato le votazioni delle domande 1 e 5. Un'altra nota riguarda la ricerca e la creazione del piano in cui, essendo le componenti in cui risiede maggiore complessità, i candidati hanno trovato difficoltà nel completare il task e poca immediatezza.

Dal test emerge quindi che: la learnability è l'aspetto in cui i candidati hanno trovato maggiore difficoltà; quindi, l'applicazione ha una curva di apprendimento troppo ripida; la memorability risulta buona in quanto la seconda volta in cui il candidato ha dovuto cercare il piano si è trovato a suo agio; e anche per quanto riguarda la satisfaction gli utenti hanno fornito un riscontro positivo.

DOMANDA	MEDIA	VARIANZA
1. Sono riuscito a consultare i materiali del piano senza problemi	4,75	6,91
2. Il modo di proporre i materiali all'interno del piano non mi piace	1,5	0,33
3. La ricerca di un piano è confusionaria	2,25	2,25
4. Sono riuscito a trovare il piano che mi interessava riguardo alla programmazione	6,75	0,25
5. Aggiungere il materiale al piano è stato difficile	4	4
6. Vorrei poter aggiungere tipi diversi di materiale al piano rispetto a Testo, PDF, YouTube, Spreker e Link	4,75	6,91
7. Ho trovato utile poter riordinare i materiali del piano durante la creazione	6	2
8. Ho trovato chiara la pagina di riepilogo dell'utente	6,5	1
9. È stato veloce registrarsi nell'applicazione	7	0
10. La seconda volta è stato più facile trovare ed iniziare un piano	6,25	2,25
11. La navigabilità dell'app non è chiara	1,5	0,33
12. I suggerimenti all'interno dell'applicazione non mi sono stati d'aiuto	2,5	1,66
13. In generale sono soddisfatto dell'applicazione	6,25	0,25

Tabella 1 Tabella che riporta le domande svolte a seguito dei task di test e le relative media e varianza dei risultati

Conclusioni e Sviluppi Futuri

Considerando gli obiettivi definiti inizialmente nella fase di need finding, l'applicazione sviluppata ha riscosso un ottimo riscontro nella fase di usability testing. Infatti, tutti gli utenti sono riusciti a completare i task ed alcuni di loro si sono mostrati interessati ad usare l'applicazione quando possibile.

Come riportato in precedenza, i test hanno mostrato anche quali sono gli aspetti principali su cui migliorare l'applicazione:

- Il processo di creazione è stato a tratti macchinoso, dovendo riuscire a compilare una Web App in una App nativa. Questo ha portato a bug che hanno reso l'applicazione difficile da usare. In futuro potrebbe essere sviluppata direttamente un'app nativa Android o cross platform con Flutter, senza dover passare da Angular. Il porting dell'interfaccia sarebbe complicato, poiché Taiga UI è compatibile solo con Angular, ma permetterebbe anche una ristrutturazione dell'interfaccia in favore di un tema Material 3 ufficiale di Google;
- Le fasi di creazione del piano e ricerca sono risultate complesse, è quindi necessario uno studio più approfondito su come poterle cambiare, magari aiutandosi con test più frequenti e con un uso dell'applicazione ancora più naturale da parte dei candidati. Un esempio è fornire la possibilità di aggiungere tipi diversi di materiale al piano, come immagini o disegni a mano libera;
- Sviluppare un backend reale rilasciato su un server vero, e non locale con l'utilizzo di Ngrok, permette sia tempi di latenza minori, sia una maggiore coerenza e completezza dell'uso dell'applicazione;
- Come emerso dalle interviste del needfinding, l'integrazione di una community all'interno dell'applicazione sarebbe molto gradita, soprattutto dalla Primary Persona. Questo permetterebbe una maggiore serietà e passione nel portare avanti un obiettivo, che in quel modo diventerebbe un obiettivo condiviso.