



**Politecnico
di Torino**

Politecnico di Torino

Ingegneria Informatica

A.a. 2022/2023

Sessione di laurea Luglio 2023

**Analisi, progettazione e sviluppo del
processo di onboarding di Reply su
TamTamy**

Relatore:

Luigi DE RUSSIS

Relatore aziendale:

Simone MAZZOTTA

Candidato:

Gabriele Vincenzo Pio

MINISCHETTI

Indice

Elenco delle figure	v
1 Introduzione	1
1.1 Onboarding: cos'è?	1
1.2 TamTamy & Onboarding in Reply	3
1.3 Obiettivo della tesi	4
1.4 Struttura della tesi	5
2 Background	6
2.1 Stato dell'arte	6
2.1.1 Onboarding con AI a supporto	6
2.1.2 Onboarding con VR a supporto	7
2.1.3 Gamification	8
2.2 Descrizione tecnica di TamTamy	14
2.2.1 TamTamy backend	14
2.2.2 TamTamy frontend	14
2.2.3 Ambiente di sviluppo	16
3 Analisi	17
3.1 Interviste	17
3.2 Personas	19
3.2.1 Reply intern	19
3.2.2 Neo-laureato (Consultant)	20
3.2.3 Consultant & Senior Consultant	21
3.2.4 Lead	21
3.2.5 Manager	22
3.2.6 Partner	23
3.2.7 Acquired company	24
3.2.8 Reply returner	24
3.3 Requisiti funzionali	25
3.4 Journey	26

4	Progettazione	28
4.1	Flusso onboarding su TamTamy	28
4.2	Primo accesso	28
4.2.1	FirstAccess	29
4.3	Profilo utente	31
4.3.1	Onboarding Task	33
4.3.2	Widget delle user project cover	35
4.3.3	Widget dei suggerimenti - contenuti	37
4.3.4	GeCo donut chart	37
4.3.5	Widget delle statistiche	38
4.3.6	Medagliere	39
4.4	Homepage	40
4.4.1	Events around you	41
4.4.2	Widget dei suggerimenti	43
4.4.3	Highlight	47
4.5	Miscellaneo & Possibili evoluzioni	49
4.5.1	Funzionalità invita	49
4.5.2	Avatar	50
4.5.3	Task randomici	51
4.5.4	Chatbot	52
5	Sviluppo	53
5.1	FirstAccess	54
5.1.1	Frontend	55
5.1.2	Backend	55
5.2	Onboarding task	57
5.2.1	Frontend	57
5.2.2	Backend	58
5.3	Widget delle user project cover	61
5.3.1	Frontend	61
5.3.2	Backend	61
5.4	GeCo donut chart	62
5.4.1	Frontend	62
5.4.2	Backend	63
5.5	Widget delle statistiche	64
5.5.1	Frontend	64
5.5.2	Backend	64
5.6	Medagliere	65
5.6.1	Backend	65
5.7	Events around you	66
5.7.1	Frontend	66

5.7.2	Backend	66
5.8	Widget dei suggerimenti	67
5.8.1	Frontend	67
5.8.2	Backend	68
5.9	Highlight	70
5.9.1	Frontend	71
5.9.2	Backend	71
5.10	Funzionalità invita	71
5.10.1	Backend	71
6	Risultati ottenuti e Conclusioni	73
6.1	Risultati ottenuti	73
6.2	Conclusioni	76
	Bibliografia	78

Elenco delle figure

2.1	L’Octalysis, il framework della gamification	9
3.1	Journey su TamTamy e Meply	27
4.1	Mapping journey-flusso onboarding	29
4.2	Le tre slide di presentazione	29
4.3	Nuova pagina di First Access, parte 1	30
4.4	Nuova pagina di First Access, parte 2	30
4.5	Primo accesso al proprio profilo su TamTamy	31
4.6	Widget di gamification per il profilo dell’utente su TamTamy	32
4.7	Widget dei task di onboarding, vista primi N mesi	33
4.8	Widget dei task di onboarding, dettaglio del task	33
4.9	Modale “All tasks”	34
4.10	Widget dei task di onboarding, vista dopo N mesi	35
4.11	Carosello delle User Project Cover	36
4.12	Nuova project cover da aggiungere al carosello	36
4.13	Carosello contenuti, variante onboarding	37
4.14	GeCo donut chart	38
4.15	Statistiche sulle connessioni dell’utente	38
4.16	Statistiche sugli eventi e sui berries	39
4.17	Medagliere con due certificati e un badge	39
4.18	Nuova Homepage di TamTamy	40
4.19	Widget “Events Around You”	42
4.20	Area di configurazione del widget “Events Around You”	42
4.21	Carosello Contenuti, variante 3	44
4.22	Carosello Gruppi & Pagine, variante 1	45
4.23	Carosello Persone, variante 2	46
4.24	Carosello Eventi, variante 1	47
4.25	Highlight in modalità “wide”	47
4.26	Highlight video con descrizione	48
4.27	Highlight con property “wide” a false	48

4.28	Modale di invito	49
4.29	Notifica vista da Gabriele Minischetti	50
4.30	Notifica vista da Michael Scott	50
4.31	Mockup della pagina di creazione dell' avatar	51
4.32	Esempio di task randomico	52
4.33	Pagina Chatbot	52
5.1	Architettura dei widget	54
5.2	Pagina d'errore	56
5.3	Composizione widget FE	57
5.4	Composizione widget FE	68
6.1	Vecchia pagina di First Access	74
6.2	Nuova pagina di First Access	75

Capitolo 1

Introduzione

1.1 Onboarding: cos'è?

Con il termine “onboarding” si indica il processo di inserimento di un nuovo dipendente all'interno di una realtà lavorativa. Lo scopo è quello di supportare il nuovo assunto, permettendogli di adattarsi velocemente e senza intoppi alla nuova dimensione aziendale. Autori come Talya N. Bauer e Berrin Erdogan definiscono l'onboarding come il processo attraverso il quale i nuovi dipendenti passano dall'essere outsider all'essere insider [1]. In effetti si tratta di una fase che non può e non deve essere assolutamente trascurata poiché fondamentale per far nascere, e nel caso sviluppare e consolidare, il rapporto tra dipendente e azienda. Diversi studi infatti hanno dimostrato che:

- il costo stimato causato dalla “partenza” di un dipendente può valere dal 100% al 300% del suo stipendio annuale [2] oltre alla spesa non calcolabile causata dalla perdita di produttività e coinvolgimento dei dipendenti [3];
- circa il 33% dei nuovi assunti cerca un nuovo lavoro entro i primi sei mesi di lavoro [2];
- circa il 23% dei nuovi dipendenti cambia lavoro entro il primo anno [2];
- le aziende con un buon processo d'inserimento riescono a migliorare la produttività del 70% e incrementare la fidelizzazione dei nuovi arrivati dell'82% [4];

Il periodo dell'onboarding può essere il momento più critico nell'esperienza di un dipendente all'interno di un'azienda, e può avere delle ripercussioni, anche durature, sul coinvolgimento dello stesso, sulle sue prestazioni e sulla sua fidelizzazione [2]. Da ciò si evince che l'onboarding ha molteplici obiettivi. Il primo fra questi è la formazione del dipendente, per fargli acquisire tutte le competenze e le conoscenze

per poter diventare pienamente operativo. Si parla infatti di “Time To Proficiency” (TTP), ovvero la quantità di tempo che va dal primo giorno lavorativo al giorno in cui il nuovo assunto diventa completamente indipendente [5]. Il TTP può essere misurato in ore, giorni, mesi o anche anni, e varia, ovviamente, in base al settore e al ruolo: il TTP per un nuovo banchiere può valere dagli 11 ai 14 mesi, per un nuovo pilota 1500 ore, per un nuovo rappresentante di vendita approssimativamente un anno [6]. È evidente che per l’azienda il TTP rappresenta una spesa data dai costi di formazione del dipendente e dalla scarsa produttività iniziale dello stesso. Occorre quindi definire una chiara ed efficiente strategia di onboarding per portare, il prima possibile, il neo-assunto ad essere indipendente, preparato e pronto ad affrontare le quotidiane sfide lavorative.

La formazione non è l’unico obiettivo dell’onboarding: assicurare una completa preparazione al dipendente, renderlo sempre più competente e lasciare poi che questi vada via in un’altra azienda non è certamente una strategia vincente! È importante, quindi, fidelizzare il dipendente, fargli sviluppare un senso di appartenenza che lo motivi a rimanere e incrementare la propria produttività. Durante l’onboarding bisogna curare l’inserimento nel nuovo arrivato nel team per farlo sentire parte integrante, rendere chiari gli obiettivi e condividere i valori aziendali, ricordandosi sempre che se i dipendenti si trovano bene difficilmente lasceranno l’azienda. Secondo la Society for Human Resource Management (SHRM), le quattro “C” per un onboarding di successo sono [7]:

- **Compliance:** definizione delle linee guida, ovvero le politiche aziendali, i regolamenti, le norme giuridiche;
- **Clarification:** il neo-assunto deve comprendere qual è il suo ruolo nell’azienda e quale può essere il suo contributo;
- **Culture:** condivisione della cultura aziendale composta da tutte le prassi interne, sia formali che informali;
- **Connection:** per riuscire a formare una squadra vincente l’azienda deve agevolare la creazione di rapporti interpersonali tra i dipendenti.

Dare importanza solo al primo punto, concedendo magari qualche chiarimento sul ruolo e trascurando gli aspetti legati alla cultura aziendale e ai rapporti lavorativi significa fare “Onboarding passivo”. Certamente, delle nozioni di Culture e Connection potrebbero arrivare ai dipendenti nel loro percorso, ma in maniera informale o comunque non strutturata, non organizzata e soprattutto senza un coordinamento volto a massimizzare il successo dell’onboarding. In questi casi l’onboarding viene visto come una checklist, un insieme di task da completare. Un onboarding pro-attivo, invece, richiede un’attenzione dedicata ad ognuna delle quattro “C” oltre che un coinvolgimento dello staff HR nelle varie fasi e, se eseguito correttamente, comporta [7]:

- Maggiore soddisfazione sul lavoro per i dipendenti;
- Turnover più bassi;
- Livelli di prestazione più elevati;
- Meno stress.

1.2 TamTamy & Onboarding in Reply

TamTamy è il social network di Reply. Si tratta di una webapp utilizzata principalmente per condividere informazioni di vario tipo:

- corsi di formazione;
- meeting e live con esperti;
- argomenti di attualità e trend tecnologici;
- iniziative ed eventi aziendali;
- FAQ e comunicazioni di servizio.

Oltre a questo, è molto utile anche per accedere facilmente ad una serie di strumenti con cui i replyer possono potenzialmente avere a che fare ogni giorno:

- tool per prenotare la scrivania, per aprire ticket, per consuntivare;
- la pagina di welfare, dei corporate beneficts, di metasalute;
- la libreria dei progetti a cui han lavorato le varie company negli anni.

Attualmente l'onboarding in Reply tende ad essere di tipo passivo, sebbene ci siano tutte le premesse per poter ambire ad un processo di inserimento pro-attivo. Il primo giorno vengono presentati i colleghi, gli uffici e l'ambiente di lavoro in generale, e consegnati e configurati gli opportuni strumenti. La formazione viene garantita affiancando al nuovo arrivato colleghi più esperti per un periodo di tempo variabile. Il giorno stesso dell'assunzione viene inviata un'email al dipendente - "WELCOME ON BOARD!" - in cui vengono presentate velocemente diverse iniziative, opportunità e strumenti aziendali. Circa 20 giorni dopo, al nuovo assunto arriva una seconda e ultima email - "HOW'S EVERYTHING?" - in cui vengono date altre informazioni, sempre però in maniera approssimativa. Infine, entro il primo mese viene organizzato un evento online, l'INDUCTION, in cui un team HR si occupa di:

- spiegare ai nuovi arrivati le politiche aziendali e le norme comportamentali (per esempio, come segnalare le proprie assenze, come consuntivare);
- raccontare la storia di Reply;
- illustrare le possibilità di carriera;
- rispondere alle domande più frequenti (per esempio, dove è possibile scaricare le buste paga) e chiarire eventuali dubbi.

1.3 Obiettivo della tesi

Il problema dell'attuale onboarding di Reply è che si pone davvero poca importanza sulla "cultura" e le "connessioni" aziendali seppur in azienda ci siano davvero tante iniziative il cui scopo è la condivisione di valori, la creazione e il rafforzamento dei rapporti lavorativi. Lo stesso discorso vale anche per le diverse opportunità di crescita, che vengono sì raccontate ma in maniera superficiale e poco efficiente. Le due email, infatti, strabordano di informazioni e questo rappresenta un grosso rischio, poiché ignorarle potrebbe significare precludersi davvero tante possibilità in Reply. A ciò si aggiunge TamTamy, che viene a stento presentato e che invece dovrebbe avere un ruolo chiave per:

- condividere il materiale formativo;
- presentare i vari step del processo di onboarding al nuovo arrivato, in modo da rendere chiaro qual è il percorso d'inserimento e specializzarlo, direzionarlo in base alle passioni e gli interessi del neo-assunto;
- tracciare i progressi e le attività svolte dal replyer, così da riuscire a tirare su dati e statistiche utili per svolgere eventuali analisi per migliorare l'esperienza lavorativa dei dipendenti (dando loro un motivo in più per rimanere e non andar via in altre aziende).

L'obiettivo di questo progetto di tesi, quindi, è definire un nuovo processo di onboarding da integrare a quello esistente in modo da renderlo più completo, capace di soddisfare le diverse esigenze dei nuovi assunti sfruttando anche l'AI. Le informazioni vanno dilazionate nel tempo, vagliate tenendo conto del percorso lavorativo dei nuovi arrivati, offrendo loro, nella maniera meno noiosa possibile, solo ciò che possono effettivamente trovare utile. Al tempo stesso bisogna tracciare il percorso di ogni neo-assunto per scoprire possibili intoppi, capire cosa va bene e cosa invece va assolutamente rivisto o corretto. Il motto che deve ispirare il lavoro è "Connect the dots" proprio perché bisogna creare una nuova narrazione in grado di far conoscere e valorizzare meglio tutte le iniziative e i potenti strumenti che

già esistono in Reply. Verranno utilizzate quindi le meccaniche di gamification per ottenere un maggior livello di coinvolgimento e le AI per personalizzare i servizi offerti all'utente, verrà data ai nuovi assunti la possibilità di scoprire Reply un po' alla volta e specializzarsi seguendo le proprie passioni, ma soprattutto TamTamy avrà un ruolo da protagonista nel processo.

1.4 Struttura della tesi

L'organizzazione del documento di tesi è la seguente:

- Capitolo 2: background. Si studierà lo stato dell'arte per poi procedere con l'analisi di TamTamy, delle sue tecnologie e dell'ambiente di sviluppo;
- Capitolo 3: analisi. Si ricercheranno tutti i possibili profili del nuovo assunto, le conseguenti diverse esigenze da soddisfare nel percorso di onboarding e quindi i requisiti funzionali;
- Capitolo 4: progettazione. Verrà definito tutto il nuovo flusso di onboarding da portare su TamTamy;
- Capitolo 5: sviluppo. Si descriverà e racconterà tutto il processo di sviluppo;
- Capitolo 6: risultati ottenuti e conclusioni. Si valuteranno i risultati ottenuti e si commenteranno i possibili miglioramenti da apportare.

Capitolo 2

Background

In questo capitolo si esamina l'effettivo punto di partenza del progetto di tesi. Nello stato dell'arte si andranno a ricercare alcuni esempi di onboarding da prendere come reference e, al tempo stesso, si esplorerà il concetto di gamification. Infine si analizzeranno TamTamy e la sua architettura logica, in modo da definire quale approccio avere nella fase di sviluppo.

2.1 Stato dell'arte

I due esempi di onboarding presentati qui di seguito risultano di significativo rilievo poiché si focalizzano principalmente su trend tecnologici recenti di maggior rilevanza.

2.1.1 Onboarding con AI a supporto

L'intelligenza artificiale è forse uno dei trend tecnologici dalle prospettive più interessanti. Dall-E 2, in grado di generare immagini secondo le richieste dell'utente, e chatGPT 3, capace di intrattenere conversazioni complesse, sono solo due delle più recenti dimostrazioni delle potenzialità dell'IA: le possibilità di impiego sono davvero molteplici e sono certamente destinate ad aumentare col passare del tempo. L'onboarding può essere affiancato da questo potente strumento per fornire al nuovo assunto un'esperienza ottimizzata e al tempo stesso adattabile. Non a caso, secondo lo State of Employee Onboarding report del 2022 [8] il 68% delle aziende prese in considerazione sta usando l'intelligenza artificiale e l'automazione nei processi di assunzione e di onboarding:

- per il 53% nell'induction e nei processi di orientamento;
- per il 53% nell'automazione dei report nelle risorse umane;

- per il 44% nel training per ruolo specifico;
- per il 52% nel training generale;
- per il 40% per la documentazione iniziale.

Lato azienda, l'AI può essere d'aiuto nel collezionare dati e riorganizzarli secondo certe esigenze. Per esempio, è possibile, date le informazioni dei nuovi assunti catalogarli per sesso, stato civile, dati demografici, in maniera da assegnare loro i vantaggi contestuali opportuni [9]. Al tempo stesso l'AI può essere anche da supporto per fornire statistiche, individuando pattern e trend del processo di inserimento. In questo senso, l'intelligenza artificiale è una sorta di assistente virtuale capace di svolgere tutti i compiti ripetitivi e ridondanti, minimizzando di conseguenza gli errori e lasciando il team di HR libero di occuparsi dei task più strategici [9]. Lato utente, il principale vantaggio sta nella possibilità di ottenere un percorso di onboarding altamente personalizzato e quindi molto più coinvolgente e motivante. È possibile infatti cambiare, in base alle esigenze dell'utente, la lingua utilizzata nelle interazioni. Allo stesso modo, date le scelte e le abilità del nuovo assunto, si potrà avere un percorso di inserimento piuttosto che un altro, raccomandando questo o quell'evento, questa o quel corso di formazione [9]. Non è da trascurare anche l'utilità delle chatbot capaci di fornire assistenza immediata qualora un utente dovesse avere uno o più dubbi, riducendo inoltre il numero di ticket destinati al team HR.

2.1.2 Onboarding con VR a supporto

L'emergenza sanitaria del 2020 e i conseguenti provvedimenti presi, hanno innegabilmente portato le diverse aziende a guardare da un'altra prospettiva la maniera di lavorare, sperimentando nuove soluzioni per risolvere problemi senza precedenti. La VR, in questo senso, ha permesso di vivere l'esperienza dell'onboarding da remoto in modo meno distaccato e noioso. Come già visto nel capitolo 1, infatti, in un percorso di inserimento di successo non si deve assolutamente trascurare l'aspetto sociale legato alla creazione e alla cura del rapporto tra i vari componenti del team e il nuovo arrivato. Proprio per questo motivo Accenture ha sfruttato la realtà virtuale per realizzare un campus innovativo per i nuovi assunti: il "One Accenture Park" [10]. Si tratta di un ambiente virtuale composto da vari uffici identici, in tutto e per tutto, a quelli fisici, con l'aggiunta di una nuova area chiamata "Nth Floor" in cui i dipendenti possono ritrovarsi e, perché no, persino giocare. In questo modo è possibile vivere esperienze più coinvolgenti e appassionanti rispetto alle classiche riunioni su Zoom (o Teams ecc.), fare networking e scoprire la cultura aziendale indipendentemente dal luogo in cui gli utenti si trovano fisicamente [10]. I benefici della VR, però, non sono solo questi. Avere a disposizione una simulazione

del tutto simile all' ambiente di lavoro reale rende, infatti, possibile formare il nuovo arrivato in maniera più completa e al sicuro da qualsiasi tipo di incidente. L'utente è libero quindi di sperimentare, provare e riprovare a svolgere task particolari, gestire casi critici, scoprire le possibili conseguenze di un errore senza causare danni di alcun tipo. Addirittura, in molti casi sono le company stesse a spingere, in questi ambienti protetti, i nuovi assunti a sbagliare, in modo da allenare le "soft skill" come ad esempio la capacità di ragionare nella maniera più lucida possibile in situazioni fortemente stressanti [11] o la tolleranza al rischio [12]. Il risultato è, di conseguenza, un onboarding più efficace proprio perché le competenze sviluppate dall'utente sono frutto delle proprie esperienze, grazie a una partecipazione attiva attraverso la simulazione e lo svolgimento di mansioni quotidiane e/o straordinarie.

2.1.3 Gamification

Per gamification si intende l'applicazione di elementi motivanti e di engaging, tipici del gaming, in contesti non ludici con lo scopo di favorire l'apprendimento e/o la produzione stimolando, in generale, l'utente ad essere più partecipe. In altre parole, si cerca di trasformare un'attività noiosa in divertente e coinvolgente. Un esempio pratico di gamification può essere la progress-bar di LinkedIn. Il valore di un social si misura dalla quantità di informazioni che gli utenti inseriscono nel sistema, e questo il team di sviluppatori di LinkedIn l'aveva capito. Il problema era che non tutti compilavano in maniera completa il proprio profilo, poiché l'atto di inserire tutti i dati richiesti veniva visto come un'operazione tediosa. Pertanto, per motivare gli user, gli sviluppatori di LinkedIn hanno introdotto una progress-bar per indicare il livello di completezza del profilo. In questa maniera gli utenti, ogni qual volta inserivano i dati richiesti in una certa sezione vedevano la progress-bar avanzare, provando una sensazione di crescita e soprattutto di avvicinamento al raggiungimento del proprio obiettivo (trovare lavoro). Il risultato finale è stato sensazionale: nonostante la progress-bar non avesse richiesto grossi tempi di sviluppo, la completezza media dei profili LinkedIn è aumentata del 20% [13]. La gamification, quindi, ha come obiettivo la ricerca dei motivi per cui le persone si sentono coinvolte, divertite e appagate mentre giocano, per applicarli in contesti di non-gaming in maniera da far sentire gli utenti allo stesso modo. Uno dei primi a capire questo concetto è stato Yu-Kai Chou il quale ha realizzato l'Octalysis, il framework della gamification, composto da 8 Core Drives.

Gli 8 core drive sono quindi le fondamenta della gamification e per ognuno di essi sono possibili varie strategie, di cui verranno riportati alcuni esempi.



Figura 2.1: L'Octalysis, il framework della gamification

Core Drive #1: Chiamata all'azione

La chiamata all'azione è l'elemento motivante che spinge l'utente ad iniziare a giocare. L'obiettivo è quello di dargli la sensazione di star facendo qualcosa di estremamente importante per la quale lui è "il prescelto". Nei videogame molto spesso si usa il contesto apocalittico: una certa minaccia incombe, il mondo sta per essere distrutto ma per fortuna che c'è l'utente, l'unico in grado di salvare la situazione! Esempi di applicazione di questo concetto in un contesto di non-gaming sono:

- Wikipedia quando, dopo una donazione, ringrazia per il contributo dato al mondo della "conoscenza libera";
- Apple quando riesce ad instillare nei propri clienti un forte senso di appartenenza al brand, una lealtà tale da portare gli utenti ad acquistare ad occhi chiusi i nuovi prodotti;
- i college americano quando riescono a fidelizzare i propri studenti a tal punto da renderli i futuri finanziatori.

Il core drive #1 è applicabile in qualsiasi fase del “viaggio”, ma generalmente è preferibile usarlo durante la fase della scoperta o la fase di onboarding [14].

Core Drive #2: Senso di crescita

La crescita è una parte fondamentale del viaggio e, pertanto, non va assolutamente trascurata. È essenziale “celebrare” i progressi fatti dall’utente solo, però, se e quando ha senso farlo: un trofeo ha valore e, soprattutto, è apprezzato solo se frutto di impegno e difficoltà superate. Una medaglia deve far sentire l’utente soddisfatto, fiero ed orgoglioso dei propri traguardi, dei propri successi. Il focus della progettazione, quindi, non deve essere su quali “game element” utilizzare, ma su come potrebbe sentirsi l’utente [13].

Una possibile applicazione del Core Drive #2 sono i **punti e punteggi**. Gli Status Points sono un altro “game element”, oltre alla progress-bar, molto utile per mostrare all’utente i suoi progressi, motivandolo ad andare avanti. Internamente, inoltre, permettono al sistema di capire quanto l’user sia vicino alla win-state. I punti sono di due tipi:

- “One-Way Status Points”, si possono solo accumulare;
- “Two-Way Status Points”, si possono guadagnare ma anche perdere.

La quantità di punti “mossa”, deve essere ponderata: bisogna fare in modo che gli utenti siano soddisfatti e non annoiati o delusi dal punteggio. Gli Status Point, inoltre, possono essere assoluti o marginali: nel primo caso indicano la quantità totale di punti guadagnati dall’utente, mentre nel secondo caso indicano la quantità di punti coinvolti per una data sfida [13].

Core Drive #3: Creatività

Un concetto che dà valore al gioco è quello di cercare di spronare l’utente ad usare la creatività, come fanno i mattoncini Lego. La creatività, infatti, comporta un coinvolgimento capace di fidelizzare gli user, convincendoli a rimanere anche una volta arrivati alla “fase di endgame” (fase in cui ormai sono stati conquistati tutti gli win-state). Avere la possibilità di rifare le cose in modo totalmente diverso, sperimentare nuove soluzioni anche bizzarre è uno stimolo per i “veterani” capace di tenere in vita un gioco potenzialmente concluso da tempo [15].

Core Drive #4: Proprietà & Possesso

Il concetto di proprietà è anch’esso utile per stimolare la fidelizzazione: quando un utente possiede qualcosa vuole migliorarlo, proteggerlo e averne di più, soprattutto

se ha speso del tempo a personalizzarlo secondo i propri gusti. Farmville ne è stato una chiara dimostrazione [16].

Il concetto di Proprietà può essere implementato attraverso i **punti scambiabili**. È stata già vista la differenza tra One-Way Status Points e Two-Way Status Points. I primi sono solo cumulabili, mentre i secondi possono essere sia guadagnati che persi o spesi. I “punti scambiabili” sono una sotto-categoria di Two-Way Status Points in quanto possono essere barattati, in maniera strategica, per ottenere oggetti di valore. È importante tenere bene a mente che avere un’economia di questo tipo non significa necessariamente che l’esperienza sia coinvolgente: la chiave di successo sta nell’impegno profuso nel processo, quanto lavoro è stato necessario per riuscire ad acquistare quel particolare item [16].

La “**costruzione della ricompensa**” è un’altra tecnica tipica del Core Drive #4. Secondo uno studio [17] “l’effetto IKEA” è un’curiosa reazione che porta le persone ad affezionarsi particolarmente ai propri mobili IKEA. Ciò accade principalmente perché questi richiedono del tempo per essere montati e, soprattutto, “partecipazione” da parte del consumatore. Una volta assemblato il prodotto, i clienti si sentono particolarmente soddisfatti proprio perché consapevoli di aver contribuito, con il proprio impegno e con il proprio tempo, alla realizzazione di quel mobile. Lo stesso principio viene usato nelle logiche di ricompensa di diversi videogame. Anziché dare all’utente un premio già completo, è più conveniente, dopo ogni vittoria, concedergliene un pezzettino con una certa libertà di personalizzazione. In questa maniera l’utente sarà motivato a continuare a giocare anche per poter continuare a costruire la propria ricompensa [16].

L’**effetto Alfred**, invece, si manifesta quando gli utenti sentono che un certo prodotto/servizio sia così personalizzato in base alle proprie esigenze da non riuscire neanche ad immaginare di poter utilizzarne uno diverso. Personalizzare l’esperienza in base al tipo di utente permette, quindi, di ottenere alti livelli di fidelizzazione [16].

Core Drive #5: Influenza sociale e relazionale

L’aspetto sociale/relazionale non è affatto da sottovalutare. “L’uomo è un animale sociale” e pertanto vive, cresce, impara grazie alle relazioni umane. Proprio per questo motivo il Core Drive #5 pone il proprio focus sulla comune ed ineluttabile tendenza umana a voler stabilire connessioni e a paragonarsi con gli altri [18].

Una prima forma di relazione umana in un’azienda è il **tutoraggio**. Avere un mentore aiuta i nuovi dipendenti ad imparare più velocemente e assorbire maggiormente la cultura aziendale, aumentando in modo efficace la soddisfazione sul posto di lavoro. D’altro canto è anche una maniera per dare nuovi stimoli ai “veterani”. Sarebbe l’ideale trovare la migliore associazione “mentore-nuovo arrivato” tenendo conto anche delle caratteristiche individuali.

Un'altra forma di tutoraggio, spesso utilizzata sul web, sono i **forum** che, non di rado, vengono usati anche per scambiarsi semplicemente opinioni sugli argomenti più disparati. Il tallone d'Achille di questa modalità di tutoraggio è l'inattività: in questo caso gli utenti sarebbero demotivati a prendere l'iniziativa ed aprire un nuovo argomento di dibattito. Conviene, quindi, creare dapprima una community salda, con un forte "potenziale di conversazione" da sfogare, per l'appunto, su un forum [18].

Anche le **missioni di gruppo** sono una tecnica di marketing molto efficace. Un famoso gioco di successo che utilizza questa strategia è World of Warcraft. In WoW, ci sono molte missioni così impegnative da richiedere un'intera squadra di 40 giocatori, ciascuno specializzato in un diverso settore. Ciò ha spinto molti utenti a raggrupparsi in clan/gilde e a suddividere le mansioni garantendo, di conseguenza, dei regolari accessi al gioco [18].

Un altro possibile esempio di applicazione del Core Drive #5 sono i **pulsanti per vantarsi** e i **Tout Flag**. Nei videogame, infatti, esistono due modi per vantarsi con i propri amici [18]:

- esplicitamente, con "festeggiamenti rumorosi" fatti per ostentare i propri successi. Un esempio potrebbe essere FIFA che, dopo aver segnato un gol, mette a disposizione del gamer una serie di esultanze che si attivano pigiando una certa sequenza di pulsanti;
- implicitamente, con manifestazioni silenziose ma comunque efficaci. Si parla di "Tout Flag", ovvero trofei, badge, premi esibiti sul proprio profilo, visibili da possibili visitatori.

Core Drive #6: Scarsità & Impazienza

La scarsità non è necessariamente dannosa, anzi, può essere una vantaggiosa opportunità: se sfruttata bene può generare un forte desiderio nell'utente, a tal punto da renderlo impaziente. In altre parole, se si desidera incentivare un determinato comportamento, una buona idea potrebbe essere quella di porvi un limite. Nel marketing questo concetto è noto come "Principio della scarsità" [19]. La rarità di un certo prodotto, la sua esclusività, ne accresce il valore e rende chi lo possiede parte di un gruppo elitario. Quando si acquista una vettura Ferrari, ad esempio, si accetta di rispettare un codice etico imposto dalla casa automobilistica. L'auto, quindi, non può e non deve essere trascurata, riverniciata o messa all'asta. Il cliente che non rispetta queste regole non potrà più comprare vetture del Cavallino Rampante. Potrebbe sembrare una strategia controproducente, eppure questa forma di scarsità dà un senso di privilegio, di appartenenza all'élite che genera ancora più desiderio attorno ad un prodotto già di per sé raro e quindi prezioso.

Uno strumento perfetto per dare all'utente una percezione di scarsità è il tempo. A prova di ciò, Yu-Kai Chou porta un efficace esempio: i camion della spazzatura. Il fatto che l'immondizia venga ritirata solo alcune volte a settimana, anziché tutti i giorni, spinge le persone a non procrastinare l'azione di portare fuori i rifiuti. Probabilmente, se l'appuntamento fosse quotidiano, le persone paradossalmente inizierebbero ad accumulare molta più spazzatura in casa [20]. **Prove a tempo, sfide su appuntamento** sono perfette per creare negli utenti un senso di urgenza tale da spingerli a non rimandare.

Core Drive #7: Imprevedibilità & Curiosità

Jesse Schell, famoso videogame designer, definisce la parola "divertimento" come "piacere con delle sorprese" [21]. L'imprevedibilità è quindi l'arma più potente per prevenire la noia e ad accrescere la curiosità dell'utente: in altre parole, rende possibile tenere "in vita" il gioco per molto più tempo. Se un determinato contenuto ha diversi finali alternativi, perché non vederli tutti? Se un videogame è open-world, perché non esplorarlo? Cosa succede se si ripete tutta l'esperienza videoludica usando un altro personaggio strutturato in tutt'altra maniera, con tutt'altre caratteristiche? La curiosità e l'imprevedibilità sono due elementi motivanti troppo potenti per poter essere ignorati.

Le **mystery box** sono pacchi misteriosi ottenuti dopo aver superato una certa sfida o raggiunto un dato traguardo. Il contenuto potrebbe essere qualsiasi oggetto, da quello più comune a quello più raro, e ciò contribuisce a rendere il premio eccitante [22].

Gli **easter eggs** aggiungono un livello di sorpresa in più: sono "sorprese a sorpresa", ricompense inaspettate ottenute dopo l'attivazione di un trigger inatteso. La peculiarità degli easter eggs sta nel passaparola che riescono a generare. Inoltre, se il trigger sembra essere casuale, gli easter eggs spingono la gente a fare speculazioni su quale possa essere stata la loro origine [22].

Core Drive #8: Perdita & Dipendenza

Questo ultimo core drive gioca sulla paura che l'utente ha di perdere i progressi fatti e di togliere, quindi, senso a tutto il tempo speso. Farmville, ad esempio, sfruttava questo timore per costringere, in un certo senso, l'utente a connettersi regolarmente, pena la perdita del raccolto. Il danno veniva mostrato anche graficamente, con immagini demoralizzanti: l'avatar era triste, le piantagioni devastate, gli animali affamati. In questa maniera l'utente veniva motivato a diventare più proattivo. Bisogna tenere bene a mente, però, che all'utente deve essere chiaro cosa fare per evitare eventi di questo tipo, altrimenti potrebbe essere tentato ad abbandonare tutto [23].

Spesso possono essere proposte all'utente delle **opportunità evanescenti**: per vincere un dato premio bisogna risolvere un task, superare una prova, rispondere ad una domanda entro una certa quantità di tempo. Persa l'occasione non è dato sapere se e quando se ne avrà una nuova. Per enfatizzare la fugacità dell'opportunità e creare nell'utente un maggiore senso di urgenza vengono spesso usati timer [23].

2.2 Descrizione tecnica di TamTamy

2.2.1 TamTamy backend

TamTamy è una webapp che si avvale di uno stack tecnologico composto da diversi layer, alla base della quale si posiziona **Spring**, un framework impiegato per lo sviluppo di applicazioni su piattaforma java che, in questo caso, serve per gestire il lifecycle di tutte le classi.

Appena sopra vi è **Struts**, il framework utilizzato per gestire il ciclo di vita di tutte le pagine della web app, il popolamento del data model per il frontend e la gestione delle chiamate API. Struts introduce il concetto di action, ovvero una classe java che si compone di servizi per poter operare sui dati in ingresso, fare query al DB, effettuare operazioni di lettura e scrittura. Vi è, quindi, un mapping tra URL ed action che avviene all'interno del sistema TamTamy in file .xml che iniziano per "struts": struts.xml, struts-user.xml, struts-error.xml ecc.

Altrettanto importante è il framework **Hibernate** che fornisce un servizio di Object-Relational Mapping (ORM), permettendo di astrarre il DB nel codice. In altre parole, traduce le tabelle del database in classi java. All'avvio, Hibernate controlla che tutte le tabelle in suo possesso siano coerenti con quelle nel DB. Se necessario, lo aggiorna aggiungendone di nuove o integrando le colonne mancanti. Non rimuove mai però.

2.2.2 TamTamy frontend

A frontend, invece, si utilizza il framework **Vue.js (v2)** per renderizzare tutti i componenti HTML, ovvero template precedentemente realizzati che vengono richiamati in più punti del codice e configurati attraverso determinati parametri. La peculiarità di questo approccio sta proprio nella modularità, che permette una migliore gestione del codice in termini di ordine, di quantità e di tempo. Un elemento ricorrente in più pagine, infatti, non andrà implementato ogni volta da zero, ma semplicemente importato e inizializzato, ottenendo, così, dei file più puliti e con molte righe di codice in meno. I benefici riguardano anche le performance: l'utente infatti, dopo aver eseguito una certa operazione, non dovrà più attendere il caricamento dell'intera pagina, ma solo quello dei componenti coinvolti.

In TamTamy sono stati definiti 3 tipi di componenti:

1. Componente basic, di bassissimo livello. Può essere, per esempio, un pulsante o uno spinner di caricamento;
2. Componente, di più alto livello. Può contenere uno o più componenti basic. Un esempio può essere una card di un gruppo o di un utente;
3. Widget, di alto livello. Ha funzionalità più complesse e interagisce col backend. Può essere composto da uno o più componenti o componenti basic.

Ciclo di vita di un componente

Di base, il ciclo di vita di un componente Vue si suddivide in quattro fasi:

1. Creation;
2. Mounting;
3. Updating;
4. Destroying.

I lifecycle hooks sono funzioni che permettono di associare del codice ad una di queste fasi [24].

La lista dei lifecycle hooks usati su TamTamy è la seguente:

- `beforeCreate()`;
- `created()`;
- `beforeMount()`;
- `mounted()`;
- `beforeUpdate()`;
- `beforeDestroy()`;
- `destroyed()`.

Options e Data

In ogni file `.vue` le varie options possibili sono [25]:

- `data()`: dichiarata come una funzione, ritorna il data object iniziale;
- `props`: un elenco/hash di attributi che accettano valori passati dal componente padre;

- `computed`: funzioni che eseguono delle computazioni che restituiscono un valore che poi viene memorizzato. Dopo la prima chiamata, ogni volta che si proverà ad accedere alla `computed` in questione, verrà ritornato sempre lo stesso valore senza rieseguire le computazioni. Se una variabile all'interno della `computed` cambia valore, alla successiva invocazione il valore ritornato verrà ricalcolato;
- `methods`: l'insieme dei metodi dell'istanza `Vue`;
- `watch`: funzione il cui nome indica una `property`. Nel caso in cui l'elemento monitorato cambi valore, la funzione viene eseguita.

2.2.3 Ambiente di sviluppo

In totale vi sono 4 ambienti di sviluppo:

1. `local`: vera e propria area di sviluppo. È accessibile e visibile solo dal singolo sviluppatore. Sfruttando `Eclipse` si riesce ad emulare sul proprio computer il comportamento del server. Il computer si collegherà a sé stesso, all'indirizzo `127.0.0.1`.
2. `svilTamTamy`: ambiente di test utilizzato principalmente per fare test di carico. Ha una struttura abbastanza simile a “`prod`” (si appoggia a 2 server);
3. `testTamTamy`: ambiente di test, simile a “`svilTamtamy`” ma ha una struttura più semplice, (si appoggia ad un solo server). Viene usato per test meno pesanti e per mostrare i nuovi sviluppi completi ai designer;
4. `prodTamTamy`: il vero social. Si appoggia a 4 server.

Capitolo 3

Analisi

Quali sono i requisiti del progetto? Quali sono le esigenze che un nuovo arrivato può avere? Un neo-laureato alla prima esperienza lavorativa necessiterà delle stesse informazioni di cui può aver bisogno un nuovo manager? Chiaramente no. È utile, quindi, delineare le personas e definire i requisiti funzionali in modo da ricavare il journey.

Per “personas” si intendono profili lavorativi, identikit ideali che indicano gli effettivi bisogni, interessi, comportamenti e ambizioni.

Per “journey”, invece, si intende l’effettivo percorso di onboarding, “il viaggio dell’utente” verso il raggiungimento dell’obiettivo del progetto: l’acquisizione di informazioni e competenze, la scoperta delle varie opportunità di crescita e della cultura aziendale.

In alcuni casi personas diverse avranno journey allineati e unificabili.

3.1 Interviste

Per definire e costruire tutti i possibili profili aziendali è stato necessario effettuare delle interviste, sia in larga scala che in modo mirato. Su TamTamy è stato pubblicato un content in cui, tramite un form di Google, venivano poste ai replyer le seguenti 14 domande (laddove necessario, sono state indicate tra parentesi le possibili opzioni):

1. Let’s start with some profiling questions.
How long have you been working at Reply?
2. What is your job role (Consultant, Senior Consultant, Lead, Manager, ...)?
3. Before Reply, did you have other work experiences in the same sector?

4. Ok, now we can start with the survey.
Select, among the following initiatives, the ones you have heard about at least more often.
(Options: XChange, Reply Challenge, Training courses, FunAtReply initiative)
5. Select, among the following platforms, the ones you frequent with a good frequency.
(Options: Learn.Share.Remix, Chick Norris TV, Find the Expert, CoPs, Reply Library)
6. You generally use TamTamy for:
(Options: Community, Access tools, Manage work organization, Training)
7. Which TamTamy tools do you consider most important? And why?
8. Have you ever left Reply? If yes, from 1 to 5 how easy was it to readapt to the changes when you returned?
9. Are you part of a company acquired by Reply? If yes, from 1 to 5 how easy was it to adapt to the new working reality?
10. Was the induction useful for you?
11. Did you have a mentor or guide figure when you started out?
12. From 1 to 5 how do you rate the current onboarding process?
13. What would you like more attention to be paid to?
14. What did you like the most instead?

Al fine di raggiungere quanti più utenti possibili, questo contenuto è stato tenuto come “pinned” nell’homepage di TamTamy per tre settimane. In questo modo è stato visto da 6749 replyer diversi di cui 965 hanno compilato il form.

Oltre a questo tipo di interviste sono stati effettuati dei colloqui “live” in modo da riuscire ad affrontare meglio e in maniera più discorsiva il tema dell’onboarding. Gli intervistati sono stati selezionati randomicamente (e anche in base alla loro disponibilità) tra quelli che hanno compilato il Google form. Con ognuno di loro si è ripercorsa la carriera in Reply, soprattutto la fase iniziale, cercando di capire quali informazioni sono mancate e quali invece sono state più utili, quanto la comunicazione sia stata efficace, quanto sia conosciuta la community di Reply e quali sono le iniziative più apprezzate. Per avere opinioni più eterogenee si è cercato di intervistare replyer di country diverse e dal title diverso.

3.2 Personas

3.2.1 Reply intern

La prima personas ricavata dalle informazioni ottenute dalle varie interviste è il “Reply intern”. In molti casi si tratta di studenti con contratto da stagista, motivo per cui le principali difficoltà sono essenzialmente due:

1. Poco tempo per adattarsi alle dinamiche di lavoro, poiché queste hanno sicuramente priorità più bassa rispetto agli obiettivi accademici da raggiungere. Sebbene la curiosità non manchi, il tempo a disposizione è certamente inferiore rispetto alle 40 ore settimanali;
2. Accesso limitato a molti strumenti, iniziative ed eventi. Secondo le politiche aziendali, infatti, con un contratto da stagista (o simili) non è consentito esplorare il “mondo Reply” nella sua interezza. Su TamTamy, per esempio, non è possibile accedere a diverse piattaforme, così come i contenuti presenti in bacheca sono un piccolissimo sottoinsieme di tutti i content del social. La principale conseguenza negativa di queste limitazioni si ripercuote sulle informazioni da passare al replayer, poiché frammentate e decentralizzate.

Ciononostante, nella maggior parte delle interviste emergeva una forte ed entusiastica curiosità di:

- scoprire Reply e le sue diverse company, di cosa si occupano, cosa effettivamente fanno, quali sono i settori in cui operano;
- conoscere le dinamiche aziendali di base;
- comprendere i ruoli nel team, incluso il proprio, e cosa gli altri si aspettano.

Il profilo delineato, quindi, è quello di un giovane alla prima esperienza lavorativa (part-time), che vuole capire quale percorso intraprendere e in quale settore. Per questo motivo, le informazioni di cui ha bisogno non devono andare troppo nello specifico, ma devono essere un po’ più superficiale ed eterogenee, in modo da aiutare a sviluppare una visione d’insieme. Si è deciso, pertanto, di porre i seguenti punti come obiettivi da raggiungere:

- Costruire relazioni con i colleghi, anche di altre company;
- Iniziare a sviluppare soft e hard skills;
- Ricevere un adeguato tutoraggio, che tenga conto del difficoltoso equilibrio tra studio e lavoro.

3.2.2 Neo-laureato (Consultant)

Il successivo profilo individuato è quello del neo-laureato assunto, che a differenza del Reply intern offre più garanzie (in termini di tempo messo a disposizione ma anche di energie, focus, ecc.), anche a causa della soluzione contrattuale certamente meno precaria. La frase più significativa estratta dalle interviste, più rappresentativa delle difficoltà, è:

“Il mondo del lavoro per me è una nuova realtà di cui so spaventosamente poco.”

La totale assenza di esperienza lavorativa rappresenta il principale tallone d’Achille per i neo-laureati, nonché causa di:

- mancanza di consapevolezza sulle iniziative utili per crescere e formarsi;
- scarsa conoscenza degli strumenti di lavoro;
- gestione non ottimale del tempo.

Tra le esigenze manifestate, invece, le due principali riguardavano:

- comprensione delle dinamiche aziendali;
- crescita professionale (intesa in termini di formazione).

Il neo-laureato quindi è un profilo che soffre la propria inesperienza ma che, nonostante ciò, è “affamato di conoscenza” ed è estremamente volenteroso di imparare e crescere. Per questo motivo, gli obiettivi definiti sono:

- Conoscere le opportunità di carriera, in modo da capire sia quale direzione dare alla propria e sia per capire qual è la struttura interna di Reply;
- Costruire relazioni con i colleghi, sia con quelli più esperti che possono fare da mentori, sia con quelli meno esperti che comunque possono essere “compagni di viaggio” nel percorso di formazione;
- Sviluppare soft e hard skills;
- Scoprire Reply e sperimentare le varie possibilità di crescita essa che offre (xChange, Learn, Reply Challenges);
- Trovare almeno un “mentore” a cui rivolgersi in caso di necessità;
- Raggiungere l’autonomia lavorativa;
- Imparare ad utilizzare i vari strumenti utili per il lavoro.

3.2.3 Consultant & Senior Consultant

La categoria relativa ai Consultant & Senior Consultant conta circa il 67% dei dipendenti totali di Reply. Tralasciando i neo-laureati, il cui profilo è stato già esaminato nel paragrafo precedente, molti dei consultant rimanenti hanno ammesso di aver avuto uno scarso interesse nell'esplorare il mondo e la cultura aziendale. Emblematica è stata la frase "Sono così tanto concentrato sul mio lavoro quotidiano da ignorare cosa succede in Reply e cosa le altre company stiano facendo". A ciò si aggiunge anche il numero di riunioni che normalmente tende ad aumentare con la seniority (a causa della maggiore esperienza maturata) e che può rendere il lavoro piatto e sempre meno entusiasmante.

In generale, i desideri espressi per il nuovo processo di onboarding sono stati:

- Opportunità per crescere professionalmente e perfezionare le proprie skills (soprattutto per i consultant, desiderosi di affermarsi);
- Più possibilità per creare rapporti lavorativi ed opportunità vantaggiosi (soprattutto per i senior consultant, proiettati ormai al successivo scatto di carriera).

Date queste premesse, per questo tipo di personas, sono stati posti i seguenti obiettivi:

- Diventare un esperto ed essere riconosciuto come tale (soprattutto per i consultant);
- Portare la propria carriera al livello successivo (soprattutto per i senior consultant, che possono diventare o team lead o manager o addirittura partner);
- Trovare nuovi stimoli per evitare la noiosa ripetitività lavorativa.

3.2.4 Lead

In Reply con il termine "Lead" ci si riferisce a un collega che ha maturato una grandissima esperienza su certi argomenti ma che, a differenza dei manager, ha deciso di non partecipare in maniera attiva e diretta alla ricerca di nuove opportunità profittevoli per la company. Proprio per questo motivo al team leader viene assegnata come principale responsabilità la guida "tecnica" del team, che prevede:

- Definizione della tabella di marcia dei progetti da completare;
- Disponibilità a dare consigli, suggerimenti, spiegazioni ai propri colleghi della stessa company/business unit ma anche a colleghi di altre company/business unit;

- Essere un mentore per i colleghi junior che ambiscono ad avere una conoscenza più approfondita in un certo settore;
- Interesse nel presentare al team le iniziative di Reply (utili per crescere professionalmente) e motivarlo ad aderirvi;
- Mediazione sulla fattibilità dei progetti e sulle tempistiche che questi richiederebbero.

Tutte queste responsabilità comportano come aspetti negativi:

- Poco tempo per continuare a specializzarsi a causa della perdita di contatto con la parte operativa dei progetti;
- Difficoltà nel reperire contenuti più approfonditi e avanzati.

Tenendo conto sia delle responsabilità e delle difficoltà, si è quindi deciso di porre come elementi fondamentali del percorso di onboarding:

- Corsi di aggiornamento/formazione sul team leading;
- Formazione avanzata sulla cultura aziendale, soprattutto sulle iniziative che più possono tornare utili ad un lead;
- Partecipazione ad eventi, talk e round tables della propria area di competenza, in modo da diffondere il più possibile il proprio sapere;
- Sostegno (anche economico) nell'espansione del proprio sapere.

3.2.5 Manager

Dalle interviste effettuate ai diversi manager è emerso che le principali difficoltà percepite sono:

1. Focus di alto livello che rende difficile essere concentrato sulle singole iniziative;
2. Gestione del tempo tra le calls/riunioni e il lavoro autonomo;
3. Mancanza di tempo per approfondire la cultura di Reply, gli strumenti e le struttura di lavoro.

Di questi disagi, il terzo è quello più grave e va assolutamente risolto, anche perchè il manager ha come responsabilità:

- Accogliere e guidare i nuovi replyer, soprattutto nel primo periodo;
- Organizzare e promuovere eventi per il team;

- Presentare al team le iniziative di Reply (utili per crescere professionalmente) e motivarlo ad aderirvi.

In altre parole, la conoscenza delle possibilità di Reply per un manager è essenziale. Oltre a ciò, il manager ha anche come compiti:

- Mantenere le relazioni con i clienti;
- Fiutare nuove opportunità di business;
- Fare da intermediario tra senior e partner.

L'onboarding di un manager deve quindi aiutarlo a:

- Acquisire una conoscenza avanzata del contesto Reply;
- Imparare a gestire relazioni e risorse.

3.2.6 Partner

L'ultimo profilo individuato che coincide con un job-role in Reply è quello del partner, il cui ruolo principale consiste nel guidare la company nella giusta direzione. Per fare ciò è necessario:

- Conoscere tutti gli aspetti legali, le politiche e le spese relative alla gestione dei team;
- Padroneggiare ogni strumento di Reply e saper condividere il proprio sapere al riguardo con i dipendenti;
- Fiutare nuove opportunità di business;
- Comprendere l'ecosistema Reply;
- Organizzare riunioni, eventi ed attività per condividere la propria visione;
- Acquisire “tonnellate” di informazioni con il minimo sforzo.

Interessante è stata la proposta avanzata durante un'intervista: “Sarebbe bello avere una guida alla cultura Reply perché è mio dovere guidare i manager verso la giusta direzione e non ho tempo per cercare le informazioni da solo”

Il tempo è il principale nemico percepito, infatti sono state segnalate:

- difficoltà nel gestire il tempo tra le calls/riunioni e il lavoro autonomo;
- mancanza di tempo per approfondire ulteriormente la cultura di Reply, gli strumenti e le strutture di lavoro.

Gli obiettivi del processo di inserimento di un partner sono quindi:

- Capire come promuovere il networking tra le aziende;
- Conoscere a menadito, supportati da una guida, il mondo Reply;
- Comprendere come muoversi tra gli aspetti legali, le politiche e le spese relative alla gestione dei team.

3.2.7 Acquired company

“Ci siamo sentiti tutti come persi nel bosco. Abbiamo avuto accesso ad alcuni strumenti, processi e sistemi, ma non sapevamo cosa farci o come interagire con le altre company.”

Diversi colleghi delle company acquisite hanno segnalato la difficoltà nell’adattarsi al cambio “repentino” dal precedente ambiente di lavoro al nuovo. Effettivamente entrare in una nuova realtà senza avere punti di riferimento, colleghi più esperti pronti a fare da mentori, non è affatto facile. Occorre che i nuovi colleghi:

- si sentano non solo benvenuti, ma anche rassicurati su quello che sta succedendo;
- abbiano un’ idea chiara di quali sono le aspettative attese dall’azienda;
- capiscano il nuovo ambiente di lavoro e vengano incoraggiati alla collaborazione con altre company;
- imparino ad utilizzare in modo efficiente gli strumenti aziendali.

Gli obiettivi, anche in questo caso, variano in base al ruolo, ma una base comune può essere:

- Cercare il networking tra le aziende;
- Imparare ad utilizzare gli strumenti aziendali;
- Adattarsi all’ambiente di lavoro di Reply.

3.2.8 Reply returner

I returner sono ex-colleghi tornati in Reply dopo un periodo di assenza passato a lavorare altrove. Sebbene si tratti di persone che hanno già avuto modo di vivere l’esperienza aziendale in passato non è detto che questi l’abbiano fatto, così come è anche vero che col tempo possono esserci stati dei cambiamenti anche importanti: già solo la Reply Organization, per esempio, subisce variazioni ogni anno! Non a caso, le difficoltà più segnalate dai colleghi sono state:

- Dover riadattarsi alla routine aziendale;
- Mancanza di conoscenza di ciò che è cambiato mentre si era via.

Gli obiettivi da raggiungere, quindi, variano da ruolo a ruolo, ma hanno come punti in comune:

- Tornare a vivere l'azienda;
- Riscoprire i valori aziendali e gli strumenti utilizzati;
- Stabilire nuove relazioni nella company e in Reply;
- Scoprire i cambiamenti più importanti.

3.3 Requisiti funzionali

Requisiti funzionali	
R1.	Unire i puntini
R2.	Utilizzare la gamification per aumentare l'engagement
R3.	Presentare TamTamy al nuovo arrivato
R4.	TamTamy deve avere un ruolo da protagonista
R5.	Dare consigli in base all'utente e al suo profilo aziendale
R6.	Assegnare task in base al profilo aziendale del replyer
R7.	Estendere la durata dell'onboarding
R8.	Aiutare Manager e Partner nel loro ruolo nell'onboarding

Il primo requisito (R1) serve a ricordare l'obiettivo del progetto: fare onboarding sfruttando le iniziative e gli strumenti che già esistono in Reply.

Per aumentare l'engagement è necessario utilizzare le meccaniche di gamification (R2). Non devono essere implementati necessariamente tutti i core drive descritti in precedenza, ma soltanto quelli che risultano essere più utili date le caratteristiche del widget. La "filosofia" della gamification, infatti, vuole che ci si concentri di più sulle finalità che sui singoli game elements utilizzati. Tutto il flusso deve avvenire su TamTamy (R4) e pertanto occorre anche fornire al nuovo arrivato una panoramica generale del social oltre che curarne la presentazione(R3).

TamTamy non deve essere solo presentato ma anche esplorato a fondo, in modo da far comprendere agli user i molteplici utilizzi del social e rendere quest'ultimo uno strumento da usare con una certa regolarità. Inoltre, per via delle esigenze diverse delle personas bisogna realizzare percorsi specifici che si adattino agli interessi degli utenti, in modo da destinare all'opportuno target di riferimento strumenti, contenuti, task, eventi e attività (R5 e R6).

Come già accennato nel capitolo 1 le informazioni vanno dilazionate nel tempo per evitare che queste vengano perse o ignorate, e pertanto va estesa la durata dell'onboarding (R7).

Infine, per aiutare i Manager e i Partner a seguire i processi di inserimento di cui sono responsabili, bisogna esplicitare in una dashboard tutte le informazioni che possono essere utili (R8).

3.4 Journey

Tutto il processo di onboarding è stato diviso in 4 fasi:

Pre-onboarding L'insieme di tutte gli step che vanno dal periodo preliminare all'assunzione fino a poco dopo. Questa fase è stata inserita a seguito di diverse segnalazioni, emerse dalle interviste, su alcuni disservizi avuti non appena entrati in Reply (welcome kit o PC aziendale arrivati in ritardo, per esempio) e per rendere più facile ai manager/partner il tracciamento delle informazioni riguardanti la nuova assunzione. La piattaforma di riferimento, in questo caso, è Meply, uno strumento accessibile da TamTamy utilizzato al momento solamente per mostrare al partner o al manager o all'utente in sessione il percorso di crescita di quest'ultimo da un punto di vista prettamente contrattuale. Si è pensato, quindi, di aggiungere alla piattaforma una nuova sezione "New Entries" dove il manager o il partner o chiunque altro si stia occupando di seguire la nuova assunzione potrà vederne lo stato di avanzamento e monitorare lo stato di consegna del PC e del cellulare aziendale, del welcome kit ecc. da una dashboard.

Welcome on board Insieme di attività/iniziativa proposte entro i primi N mesi di contratto. Questa fase è stata separata da quella successiva per evitare di dare troppe informazioni in poco tempo (problema segnalato in molte interviste) e per presentare, quindi, solo l'essenziale: gli strumenti che il replyer dovrà usare già da subito e i task che vanno svolti entro una certa data (per esempio, i corsi sulla sicurezza). La durata è stata, in via preventiva, parametrizzata: per ora il numero N di mesi è uguale a 6 per via della durata media del periodo di prova e delle scadenze dei task accennati precedentemente. Non è escluso, tuttavia, che in futuro il valore di N possa cambiare. I task proposti saranno guidati e soprattutto mandatori, in modo da far scoprire già da subito all'utente TamTamy e la sua utilità principale. Le attività svolte saranno tracciate e rese visibili al manager e al partner su Meply;

Onboarding Long Term Insieme di attività/iniziativa proposte dal mese numero N al mese numero M di contratto. La differenza con la fase precedente sta nel fatto che l'obiettivo dell' Onboarding Long Term è quello di far scoprire

tramite TamTamy tutte le opportunità e le potenzialità di Reply. Come per N, anche M è un parametro il cui valore in futuro potrebbe cambiare ma che adesso vale 24. Le attività svolte saranno tracciate e rese visibili al manager e al partner su Meply;

Reply Employees Experience Insieme di attività/iniziative proposte da dopo il mese numero M di contratto. Sebbene da questo momento in poi l'effettivo processo di onboarding si può considerare concluso, si è deciso comunque di continuare a utilizzare i nuovi strumenti (di TamTamy e Meply) per tracciare il percorso del replayer, condividere la cultura aziendale e incentivare la creazione di connessioni.

Di seguito uno schema riassuntivo di quanto affermato.

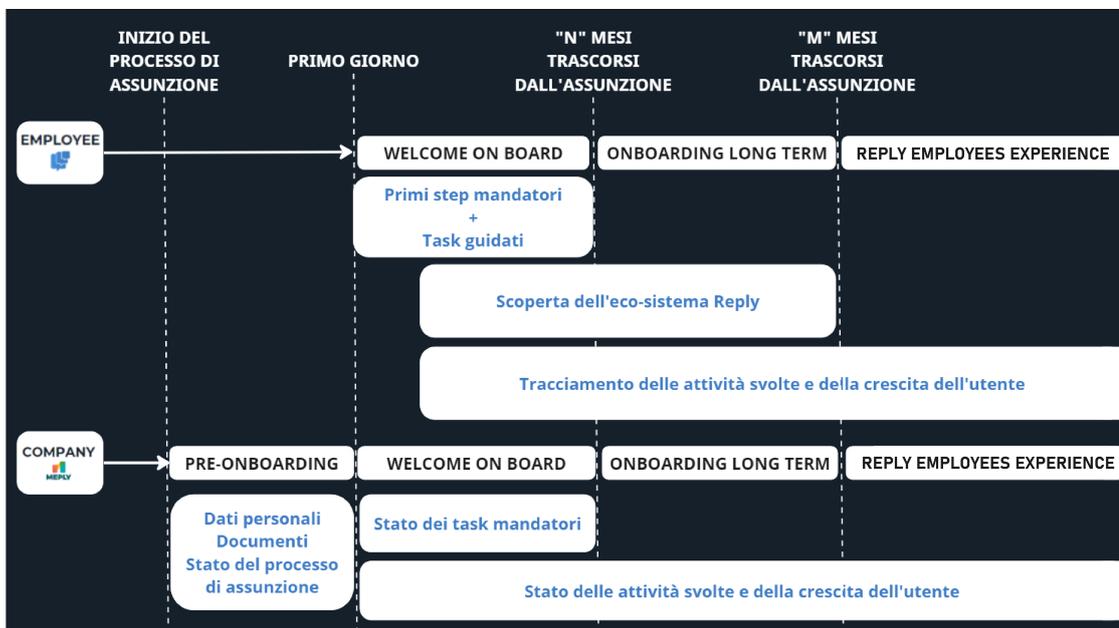


Figura 3.1: Journey su TamTamy e Meply

Capitolo 4

Progettazione

Poiché R4, definito nel capitolo precedente, richiede che tutto il flusso debba avvenire su TamTamy, delle 4 fasi del journey verranno prese in considerazione solo le ultime 3: Welcome on board, Onboarding Long Term e Reply Employees Experience. La fase di pre-onboarding, essendo completamente su Meply, è stata completamente esclusa dal flusso di progettazione.

4.1 Flusso onboarding su TamTamy

Il flusso progettato per l'onboarding su TamTamy parte dalla primissima pagina con cui si interfaccia l'utente: la "FirstAccess". Qui inizialmente si spiega al nuovo arrivato cos'è TamTamy, come richiesto dal R3, e successivamente si procede con la creazione della nuova utenza. Completata la registrazione si atterra sulla pagina del profilo dell'utente, dove il replayer troverà task da svolgere (R6), consigli e spiegazioni sui vari strumenti e portali accessibili da TamTamy (R5) e widget che raccontano la crescita professionale dell'user all'interno di Reply (R2). Infine l'ultimo step del flusso riguarda l'homepage di Tamtamy, dove sono stati aggiunti widget di suggerimenti, anche specifici, al fine di soddisfare il R1.

Sia la pagina del profilo dell'utente che l'homepage di TamTamy hanno a che fare con tutte le 3 fasi successive al pre-onboarding, con l'unica differenza che il profilo dell'utente subisce delle variazioni quando si passa da una fase all'altra. La FirstAccess invece deve essere visualizzata solo nella fase di Welcome on board, in particolare nei primissimi giorni successivi all'assunzione.

4.2 Primo accesso

Il redesign di TamTamy è iniziato nel 2018 ed è attualmente ancora in corso. Si tratta, infatti, di un'operazione lenta che si protrae ancora nel tempo poiché bisogna

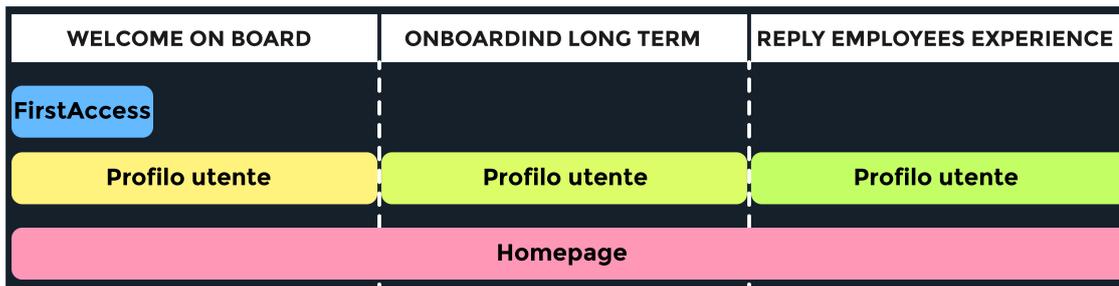


Figura 4.1: Mapping journey-flusso onboarding

andare in contro alle effettive esigenze del cliente. Capita quindi spesso che, per soddisfare nuove richieste, è necessario aggiornare vecchie pagine. La “FirstAccess” ne è un esempio.

4.2.1 FirstAccess

Il primo widget con cui l’utente viene a contatto è il widget di FirstAccess, il cui compito è quello di accogliere il nuovo arrivato su TamTamy e creare la nuova utenza. Vengono inizialmente mostrate 3 slide che si susseguono per dare al replyer una prima presentazione del social (R3).

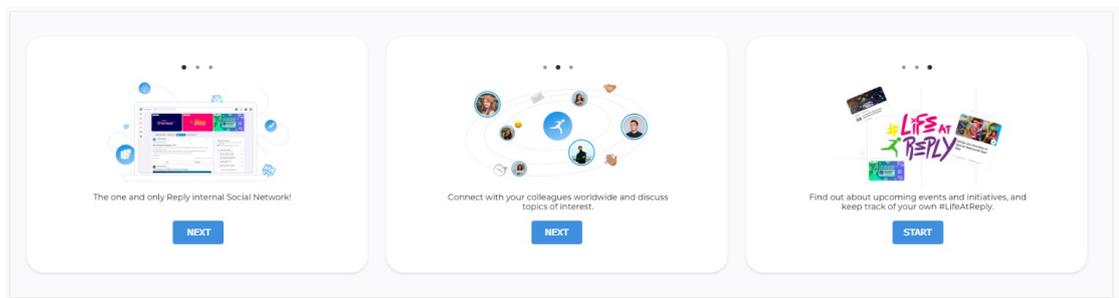


Figura 4.2: Le tre slide di presentazione

Successivamente, si procede con l’effettiva creazione dell’utenza e ciò avviene in due fasi. Nella prima parte si ripropongono gli stessi elementi della vecchia interfaccia ma con la nuova grafica di TamTamy. La seconda parte, invece, è stata pensata per dare al nuovo arrivato la possibilità di personalizzare la propria esperienza sul social. L’idea è di generare l’effetto Alfred del Core Drive #4 (R2 e R5): personalizzare un servizio a tal punto da renderlo insostituibile.

L’utente, quindi, indica tra le varie chips proposte quelle contenenti gli argomenti che reputa più interessanti. I topic scelti aiuteranno Machine Learning a selezionare i contenuti da consigliare all’user in sessione.

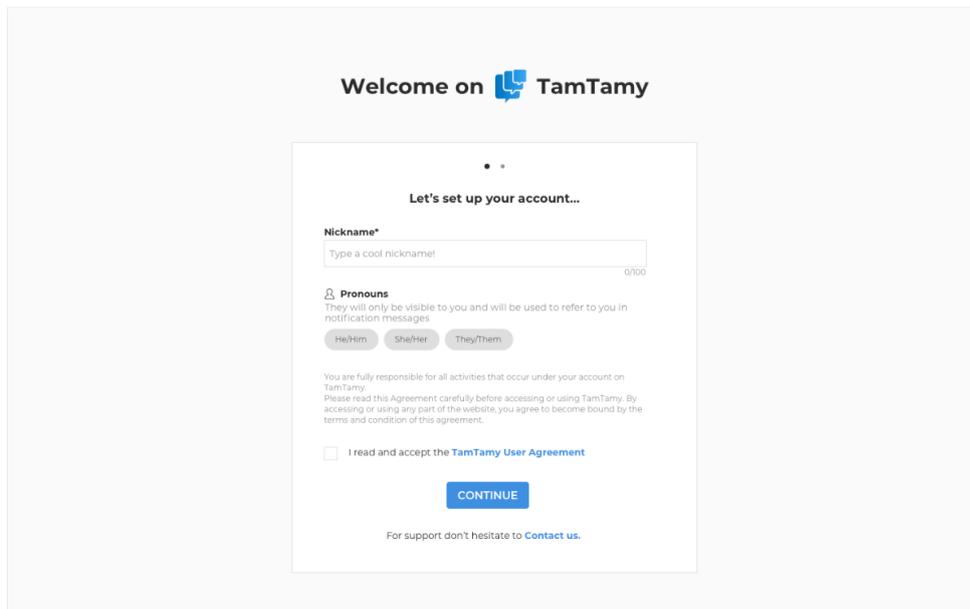


Figura 4.3: Nuova pagina di First Access, parte 1

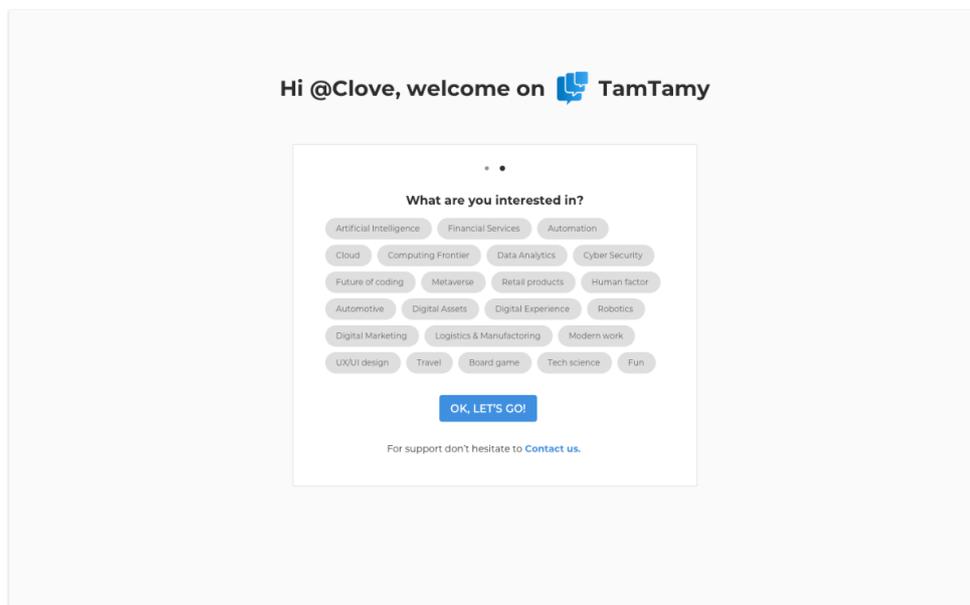


Figura 4.4: Nuova pagina di First Access, parte 2

4.3 Profilo utente

Completata la configurazione iniziale nella pagina di FirstAccess si viene reindirizzati sul proprio profilo di TamTamy su cui apparirà la prima modale che darà il via alla gamification pensata per i replyer.

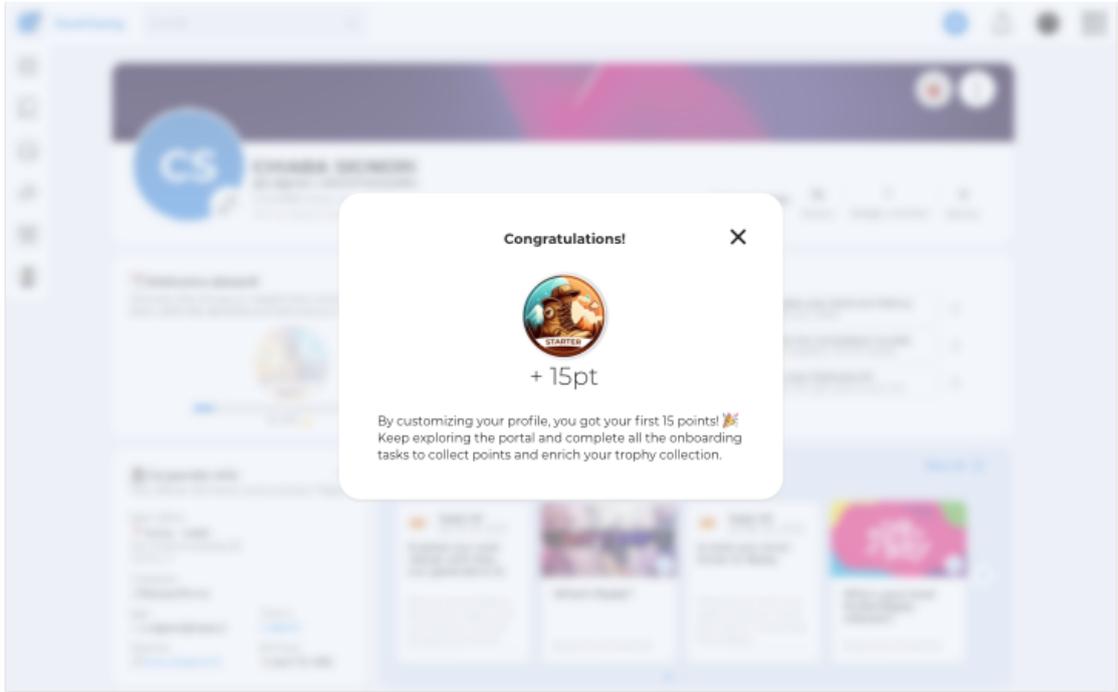


Figura 4.5: Primo accesso al proprio profilo su TamTamy

Il profilo dell'utente, infatti, ha un ruolo fondamentale nel nuovo processo di onboarding in quanto è il perno centrale per proporre le diverse attività ai nuovi arrivati. Sono stati inseriti, quindi, diversi widget ispirati principalmente dai Core Drive #2 (Senso di crescita) e #4 (Proprietà & Possesso) per soddisfare i R2: la pagina deve raccontare i progressi fatti dall'user, mostrare le certificazioni prese, i progetti a cui si è preso parte, il contributo che si è dato. In altre parole, si è cercato di "materializzare" il percorso di crescita degli utenti: prendersi cura del proprio profilo TamTamy significa prendersi cura della propria carriera in Reply.

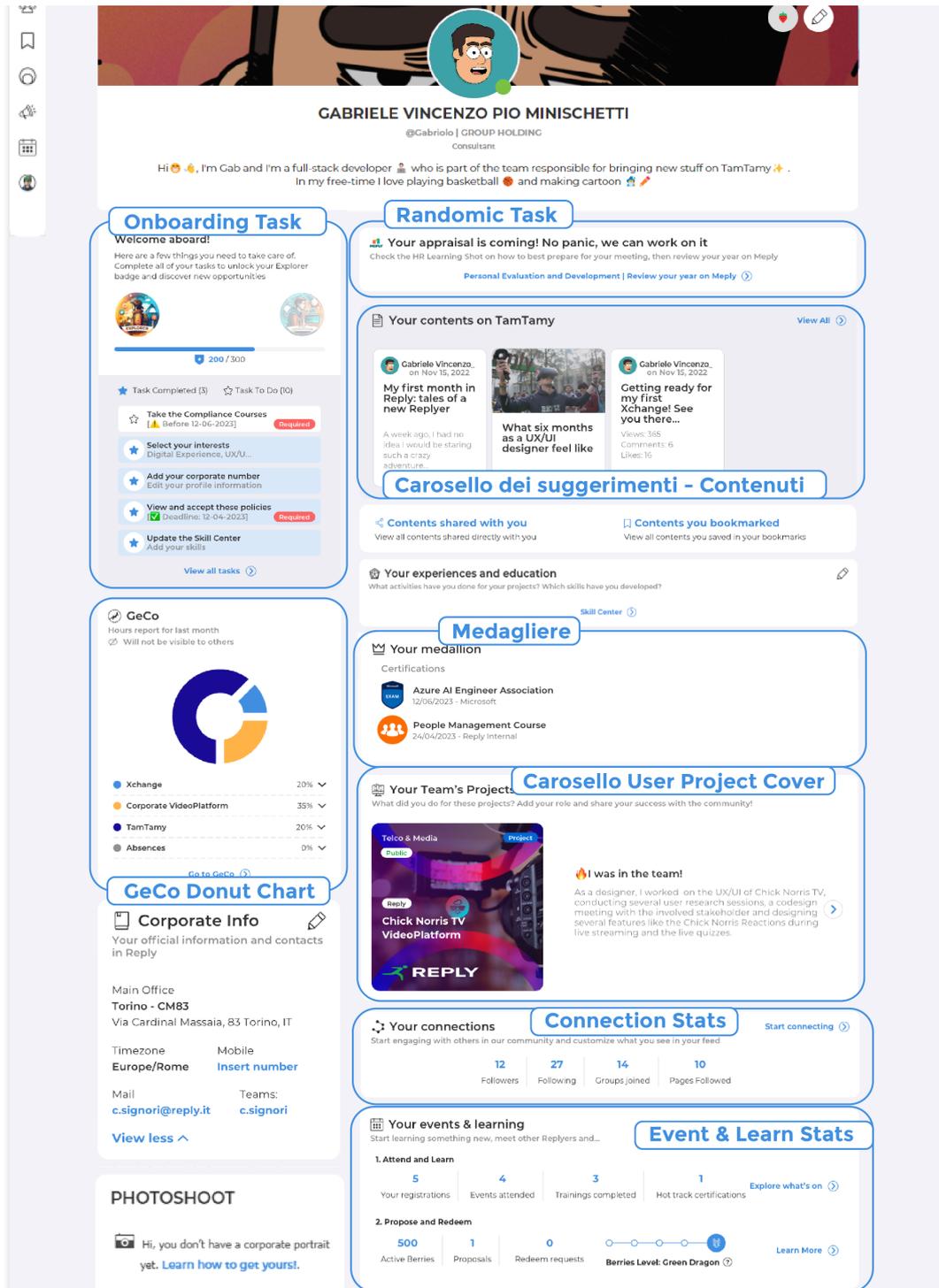


Figura 4.6: Widget di gamification per il profilo dell'utente su TamTamy

4.3.1 Onboarding Task

Il widget “Onboarding Task” è stato pensato per proporre al nuovo arrivato brevi task da risolvere e tracciarne allo stesso tempo lo stato di avanzamento. Per evitare di assegnare troppe attività già da subito, il widget si compone di 3 livelli, ognuno dei quali ha un badge rappresentativo e un insieme di task da proporre:

1. livello starter: gli obiettivi proposti riguardano la configurazione del proprio profilo, l'accettazione delle policy, il ritiro del welcome kit e il completamento dei Compliance Courses (corsi informativi sulla sicurezza sul lavoro ecc);
2. livello explorer: fase in cui si invita l'utente a esplorare TamTamy e gli strumenti che mette a disposizione. L'insieme di task, in questo caso, può variare da personas a personas (un partner ha a disposizione strumenti non accessibili ai consultant, per esempio) e anche da company a company. Ognuna di queste, infatti, è libera di contattare il gruppo di sviluppatori per richiedere l'inserimento di uno o più task specifici;
3. livello insider: i task proposti riguardano la partecipazione a eventi/iniziative Reply. Come per il livello 2, le proposte possono variare da personas a personas e da company a company.

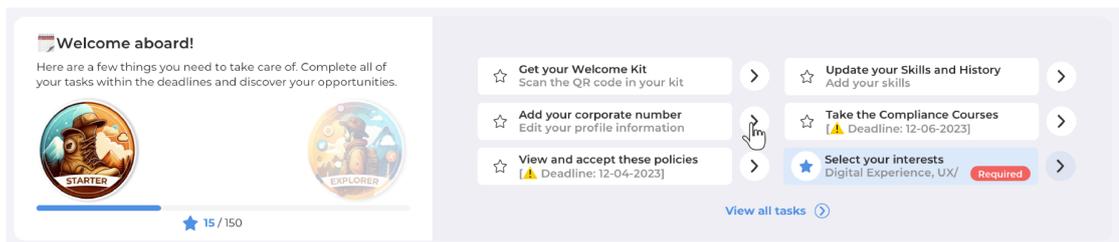


Figura 4.7: Widget dei task di onboarding, vista primi N mesi

Per alcune proposte è possibile accedere a una sezione di dettaglio, contenuta nel widget stesso, utile a spiegare cosa fare per riuscire a svolgere il task.

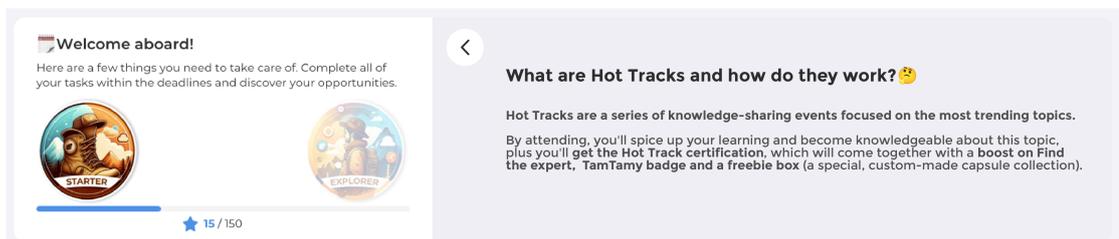


Figura 4.8: Widget dei task di onboarding, dettaglio del task

In altri casi, cliccando sulla freccetta, si viene reindirizzati ad una pagina in particolare. Cliccando, invece, su “View all task” si apre una modale contenente tutti i task che verranno proposti all’utente per il livello in questione.

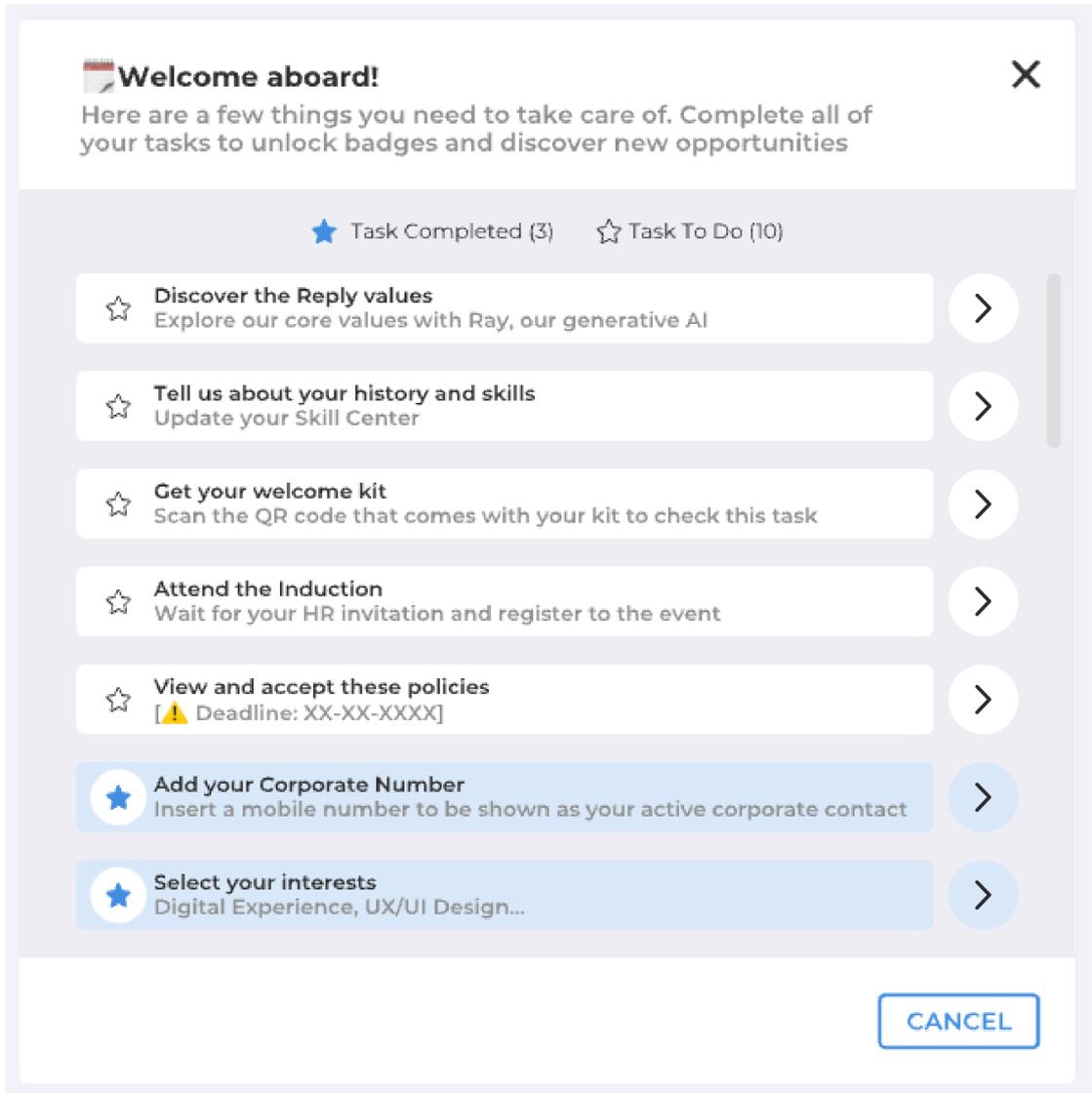


Figura 4.9: Modale “All tasks”

Con l’inizio della fase “Onboarding Long Term” il widget “perde di priorità” e pertanto occupa meno spazio. Con l’inizio della fase “Reply Employees Experience”, il widget viene chiuso definitivamente. Durante la fase “Welcome on board” i task impaginati sono 6 (due colonne da tre task l’una). Successivamente, si impaginano solo 5 task. In entrambi i casi i task ancora da svolgere hanno maggiore priorità.

Il widget è visibile solamente al proprietario del profilo e al manager e al partner di riferimento (R8).

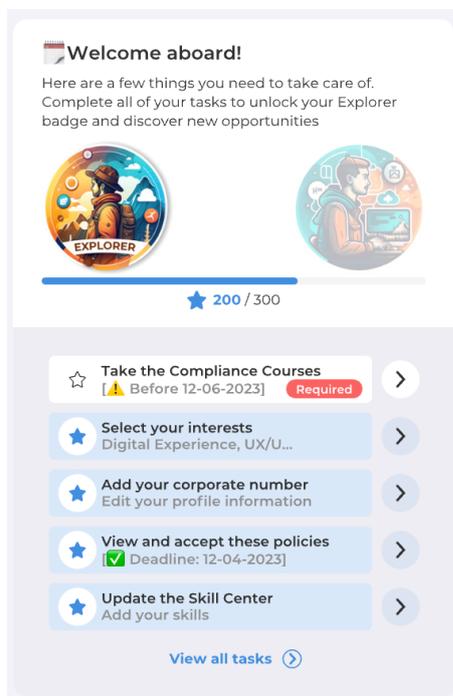


Figura 4.10: Widget dei task di onboarding, vista dopo N mesi

4.3.2 Widget delle user project cover

Le project cover sono “brevi racconti” di progetti conclusi a cui si è lavorato per una certa quantità di tempo. Vengono raccolte nel portale “Project Library” (accessibile da TamTamy) e il loro scopo è principalmente quello di fornire delle reference per nuovi progetti dagli aspetti simili. Si è pensato, quindi, di sfruttare le project cover come badge da esibire con orgoglio sul proprio profilo. Il widget, quindi, si basa non solo sul Core Drive legato alla crescita, ma anche sul Core Drive #5 (Influenza sociale e relazionale), in particolare sul concetto di “Tout Flags”. Il carosello, pertanto, mostra 7 project cover (come indica il design system) a cui l’utente ha preso parte e per ognuna di esse vi è un breve testo di massimo 255 caratteri (poiché già la project cover scende nei dettagli) che riassume quale sia stato il suo contributo.

La configurazione delle card avviene nella pagina della project cover in modo da forzare l’utente a scoprire ed esplorare il portale di Project Library. Le User Project Cover indicate come “preferite” avranno maggiore priorità nel carosello. Se

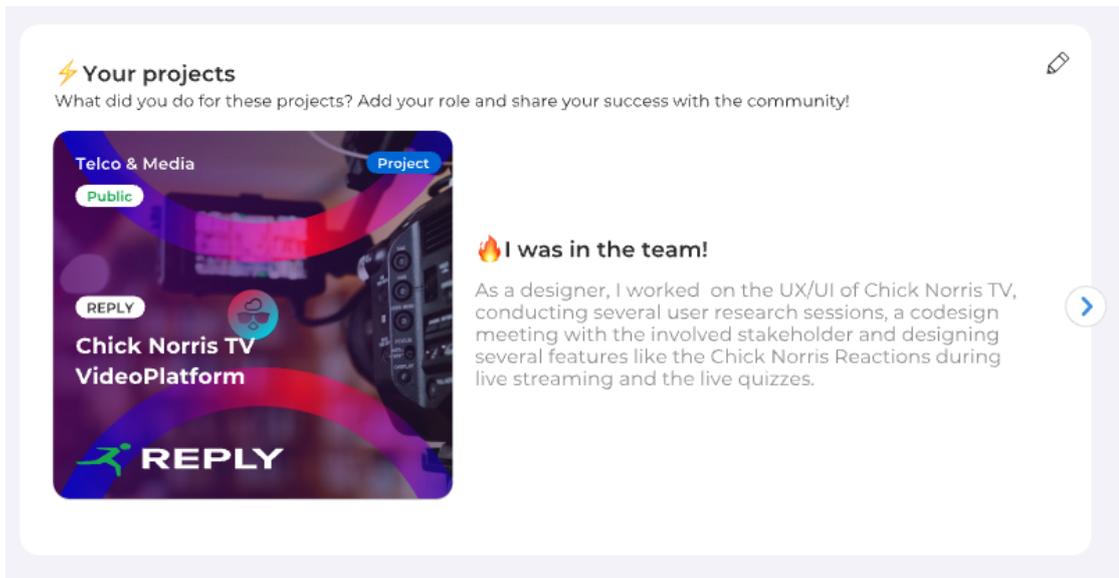


Figura 4.11: Carosello delle User Project Cover

queste sono inferiori di 7, le restanti verranno scelte randomicamente (tra le User Project Cover “non preferite”). Il widget appare sul profilo dell’utente con l’inizio della fase “Onboarding Long Term”, per dare allo user il tempo di “accumulare” qualche project cover. Lo stesso discorso non vale per i “Returner” per cui il widget sarà disponibile già da subito (in questo caso potrebbero già esserci dei lavori fatti in passato). Se non ci sono card da mostrare, il widget si chiude.

The image shows a form titled "WHAT WAS YOUR CONTRIBUTION TO THIS PROJECT?" with a star icon in the top right corner. Below the title is the instruction "Describe your contribution, the challenges you faced and the solution provided". There are two input fields: "Title" and "Description". A blue "SAVE" button is located at the bottom right of the form.

Figura 4.12: Nuova project cover da aggiungere al carosello

Come richiesto dal R2 anche per questo widget si utilizza la gamification, in particolare la “Costruzione della ricompensa” (Core Drive #3: Creatività) e “Tout

Flags” (Core Drive #5: Influenza sociale e relazionale).

4.3.3 Widget dei suggerimenti - contenuti

Il carosello dei contenuti è lo stesso widget che viene utilizzato nell’homepage (di cui si parla nel paragrafo “Widget dei suggerimenti”).

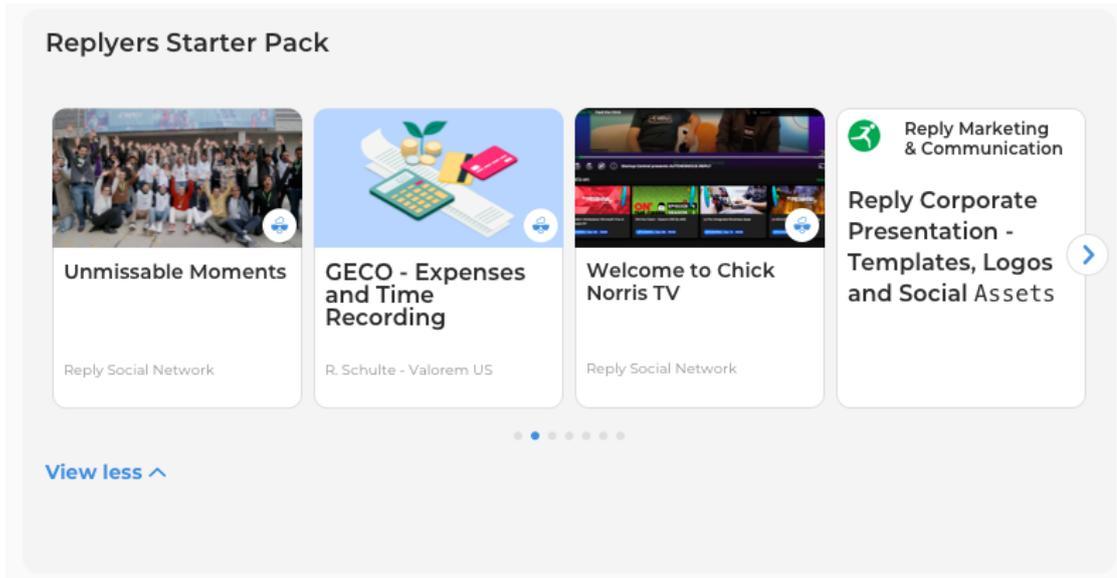


Figura 4.13: Carosello contenuti, variante onboarding

La differenza sta nelle varianti mostrate, che in questo caso sono due:

- Variante 3: contenuti editoriali per l’onboarding (stessa variante proposta nell’homepage e trattata più avanti);
- Variante 4: contenuti creati dall’utente proprietario del profilo.

4.3.4 GeCo donut chart

GeCO è la piattaforma, accessibile da TamTamy, utilizzata per consuntivare, indicare quotidianamente su quali progetti si è lavorato. Il GeCO donut chart è un grafico a “ciambella” che mostra le ore spese a lavorare sui vari progetti nell’ultimo mese (Core Drive #2: Senso di crescita). Il widget è visibile solo dal proprietario del profilo e si attiva con l’inizio della fase “Onboarding Long Term” (dopo N mesi passati dall’assunzione). Cliccando su “Go to GeCo” si viene reindirizzati al portale di GeCo.

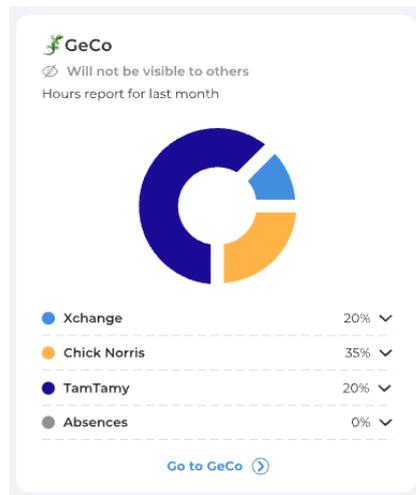


Figura 4.14: GeCo donut chart

4.3.5 Widget delle statistiche

Entrambi i widget delle statistiche forniscono dei numeri col fine di quantificare un particolare aspetto della propria crescita all'interno di Reply (Core Drive #2: Senso di crescita).

Connection stats

Il widget "Connnection stat" fornisce i numeri sulle connessioni create dall'utente in questione.

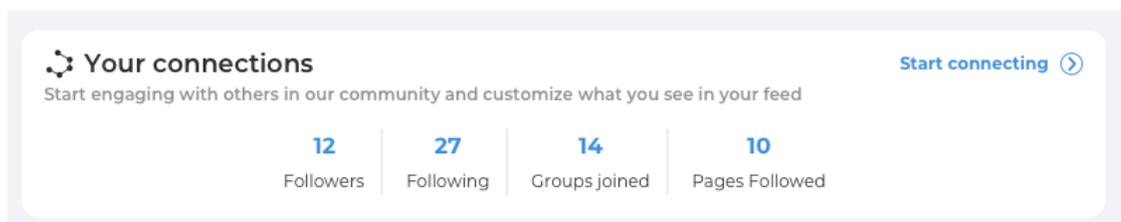


Figura 4.15: Statistiche sulle connessioni dell'utente

Event & Learn stats

Il widget delle statistiche degli Event & Learn è stato pensato per incitare gli utenti ad essere attivi: vengono esplicitate delle informazioni che, inoltre, in futuro potrebbero essere utili per capire perché un determinato utente ha un determinato punteggio.

Il widget delle statistiche si divide in due parti:

1. la sezione legata agli eventi, che indica quanto un utente sia interessato e partecipa a questo tipo di iniziative;
2. la sezione legata alla piattaforma “Learn.Share.Remix”, lo strumento più potente di Reply di diffusione di conoscenza.

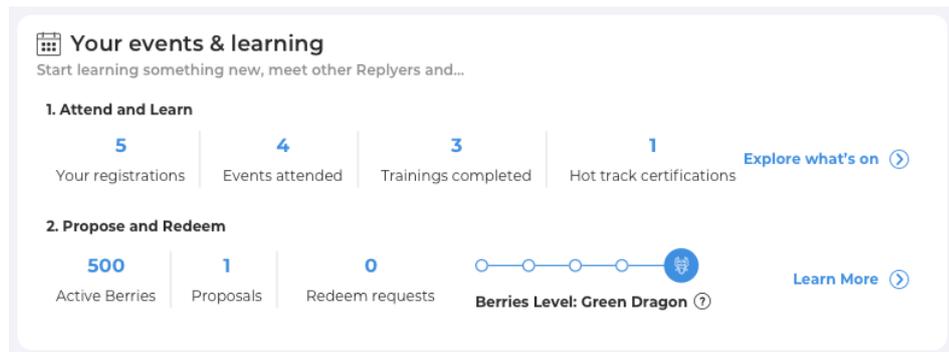


Figura 4.16: Statistiche sugli eventi e sui berries

4.3.6 Medagliere

Su TamTamy esiste già il widget dei certificati, il cui scopo è quello di mostrare le certificazioni ottenute dall'utente. Il medagliere ne è un'evoluzione. Al suo interno ci sono due sezioni: una dedicata alle certificazioni, una dedicata ai badge di TamTamy. Questo widget è un esempio di “Tout Flags” (Core Drive #5: Influenza sociale e relazionale).

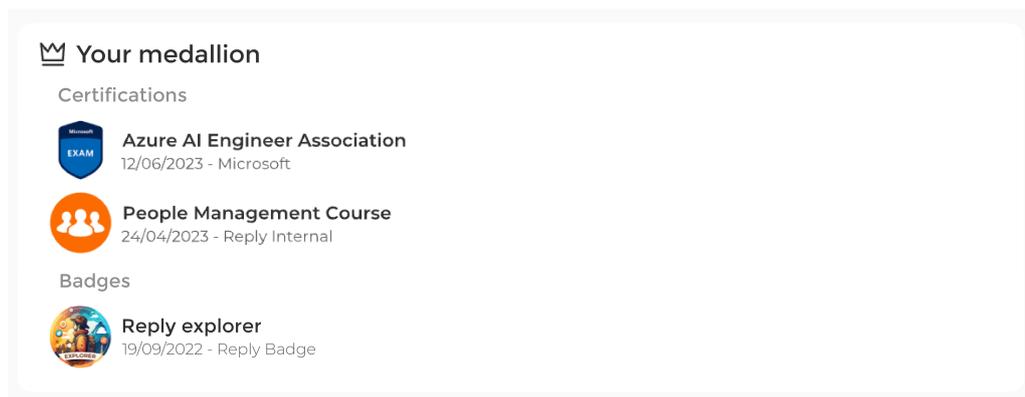


Figura 4.17: Medagliere con due certificati e un badge

4.4 Homepage



Figura 4.18: Nuova Homepage di TamTamy

L'homepage di TamTamy è il punto di contatto tra la community di Reply e i singoli utenti. Qui vengono mostrati i contenuti in base a tre possibili criteri

alternativi:

- “All Replyers”, l’insieme dei contenuti creati nel gruppo in cui vengono inseriti tutti i replyer;
- “Your feed”, l’insieme dei contenuti creati dalle pagine e dagli user seguiti dall’utente in sessione;
- “Your feed”, l’insieme dei contenuti creati dalle pagine e dagli user seguiti dall’utente in sessione;
- “Recommended”, l’insieme dei contenuti consigliati da Machine Learning. Questa funzionalità è stata recentemente introdotta.

Al fine di creare, come suggerito dal core Drive #5, una community salda e attiva si è deciso quindi di aggiungere alcuni widget utili a incentivare la nascita di connessioni e a promuovere meglio eventi e attività.

4.4.1 Events around you

Il widget “Events around you” si colloca in spalla, nell’homepage di TamTamy, e serve per consigliare, in modo mirato, eventi agli utenti sfruttando le potenzialità del Machine Learning (effetto Alfred, Core Drive #4). Il widget presenta, nella parte superiore, un menù a tendina utile a selezionare tra tutte le città in cui è presente almeno una sede Reply quella per cui si vogliono conoscere gli eventi. La città che viene mostrata di default è la stessa dell’utente in sessione, a meno che questi non ne selezioni una dal menù a tendina: in questo caso la città di default diventa quella scelta dall’user. Le iniziative, inoltre, vengono mostrate secondo il seguente ordine di priorità:

1. priorità evento;
2. fuzzy Date (raggruppamento per distanza della data) secondo il seguente ordine di priorità
 - near, entro 7 giorni;
 - medium, tra 7 e 15 giorni;
 - far, oltre i 15 giorni.
3. modalità, secondo il seguente ordine di priorità:
 - fisico;
 - ibrido;
 - online.

4. distanza geografica dell'evento;
5. quanto l'utente potrebbe essere interessato al topic dell'evento;
6. tag;
7. data;
8. id.

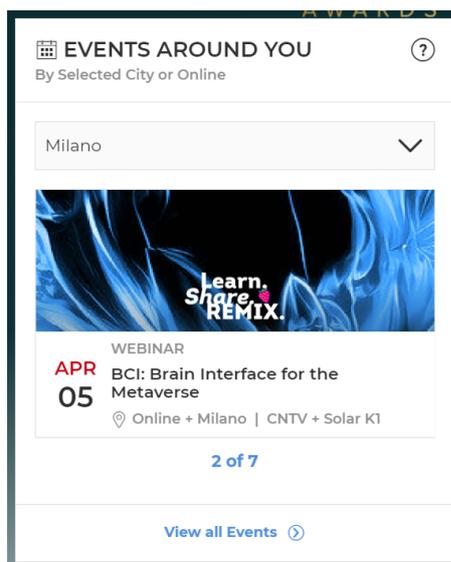


Figura 4.19: Widget “Events Around You”



Figura 4.20: Area di configurazione del widget “Events Around You”

Gli eventi consigliati vengono scelti da una lista che viene passata settimanalmente nella pagina di Carousel Management dal team RSN (Reply Social Network), responsabile di organizzare e promuovere attività con l'obiettivo di migliorare la

condivisione delle conoscenze e sviluppare una cultura comune in Reply. Questa lista è un JSON che contiene, per ogni evento, le informazioni di configurazione necessarie per poter eseguire l'ordinamento.

4.4.2 Widget dei suggerimenti

Il carosello dei suggerimenti è un widget il cui scopo è quello di incentivare la scoperta della cultura aziendale attraverso la creazione di connessioni lavorative (R1). In totale ci sono quattro tipi di caroselli, ognuno dei quali può avere diverse varianti. Tutti i caroselli dei suggerimenti sono collocati nel body della homepage di TamTamy, ognuno però con diversa posizione. Per ognuno di essi viene mostrata, ad ogni refresh della homepage, una variante scelta in maniera casuale. Se la variante selezionata propone come risultato una lista vuota, viene scartata e al suo posto, sempre casualmente, ne viene scelta un'altra. Se tutte le varianti hanno come risultato una lista vuota, il widget si chiude.

Contenuti

Il carosello dei contenuti serve per dare visibilità a contenuti che normalmente non comparirebbero nella homepage, poiché caricati su un altro gruppo o in un altro portale di TamTamy o magari non di recente.

Questo widget viene mostrato in posizione n.3 e le varianti progettate sono tre:

- Variante 1: contenuti presi dal portale CoP, un'insieme di gruppi in cui vengono approfondite tematiche di attualità relative al mondo tech. Il footer contiene un pulsante che, se pigiato, rimanda al portale CoP;
- Variante 2: contenuti presi dal gruppo della company dell'utente in sessione. Il footer contiene un pulsante che, se pigiato, rimanda al gruppo della company;
- Variante 3: contenuti editoriali per l'onboarding. Sono content informativi che vengono proposti in base ai mesi trascorsi dall'assunzione e al title dell'utente in sessione. Per essere discriminati si usano dei "reserved tag", tag il cui utilizzo è consentito solo ad alcuni replyer (configurati a DB):
 - onboarding+level+one: per i contenuti che devono essere consigliati a tutti gli utenti che sono in Reply da 0 a N mesi (il valore N di default vale 6, Welcome on board);
 - onboarding+level+two: per i contenuti che devono essere consigliati a tutti gli utenti che sono in Reply da N a M mesi, con $N > M$ (il valore M di default vale 24 mesi, Long Term Onboarding);

- onboarding+level+three: per i contenuti che devono essere consigliati a tutti gli utenti che sono in Reply da più di M mesi (Reply Employees Experience);
- onboarding+collaborator: per i contenuti che devono essere consigliati solo ai collaborator;
- onboarding+consultant: per i contenuti che devono essere consigliati solo a consultant e senior consultant;
- onboarding+lead: per i contenuti che devono essere consigliati solo ai lead;
- onboarding+manager: per i contenuti che devono essere consigliati solo ai manager e senior manager;
- onboarding+partner: per i contenuti che devono essere consigliati solo ai partner e associate partner.

Un contenuto redatto per i consultant e senior consultant che sono in Reply da meno di N mesi avrà quindi i tag “onboarding+level+one” e “onboarding+consultant”.

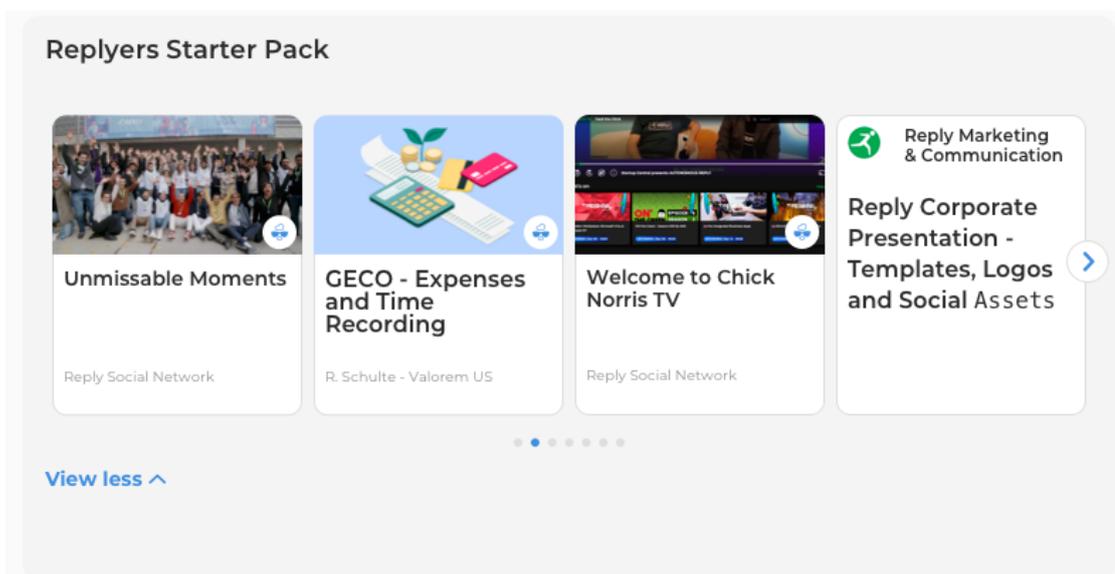


Figura 4.21: Carosello Contenuti, variante 3

Gruppi & Pagine

Il carosello dei gruppi e delle pagine ha, rispetto a quello delle persone e quello degli eventi, maggiore priorità poiché soddisfa di più i requisiti richiesti: networking e

condivisione della cultura aziendale. Proprio per questo motivo occupa, nel body della homepage di Tamtamy, la posizione n. 6: viene visualizzato dopo i primi due contenuti.

Le varianti disponibili sono due:

- Variante 1: gruppi di tendenza, ovvero l'insieme dei gruppi di Tamtamy particolarmente attivi di cui l'utente non è membro. Il footer contiene un pulsante che, se pigiato, rimanda alla sezione "All Groups";
- Variante 2: pagine ufficiali importanti, ovvero l'insieme delle pagine dei central services non seguite dall'utente e non filtrate per country. Il footer contiene un pulsante che, se pigiato, rimanda alla sezione "All Pages".

Per ogni variante, i gruppi o le pagine (che di fatto sono gruppi in cui solo pochi utenti hanno il permesso di pubblicare contenuti) vengono mostrati secondo un ordinamento che da maggiore priorità a quelli più attivi (Core Drive #5: Influenza sociale e relazionale - forum).

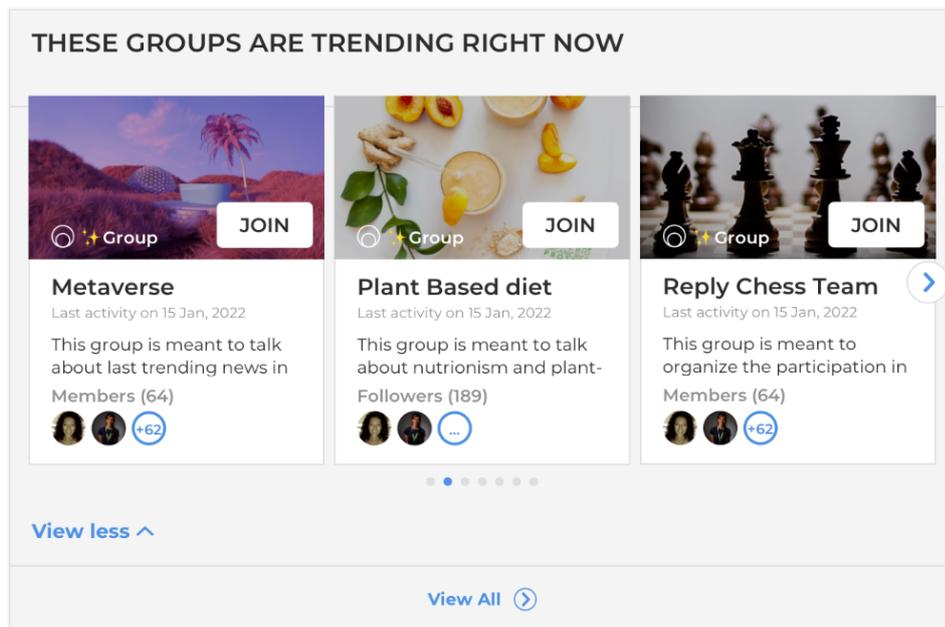


Figura 4.22: Carosello Gruppi & Pagine, variante 1

Persone

Il carosello delle persone è terzo per importanza e, pertanto, viene visualizzato in posizione n. 9. Il suo scopo è quello di incentivare la creazione di collegamenti, di connessioni tra i vari replyer secondo 3 criteri alternativi:

- Variante 1: persone più attive su TamTamy;
- Variante 2: persone della stessa company dell'utente in sessione;
- Variante 3: persone della stessa city dell'utente in sessione.

Per ogni variante, gli utenti vengono mostrati secondo un ordinamento che dà maggiore priorità a chi è più attivo (Core Drive #5: Influenza sociale e relazionale - forum).

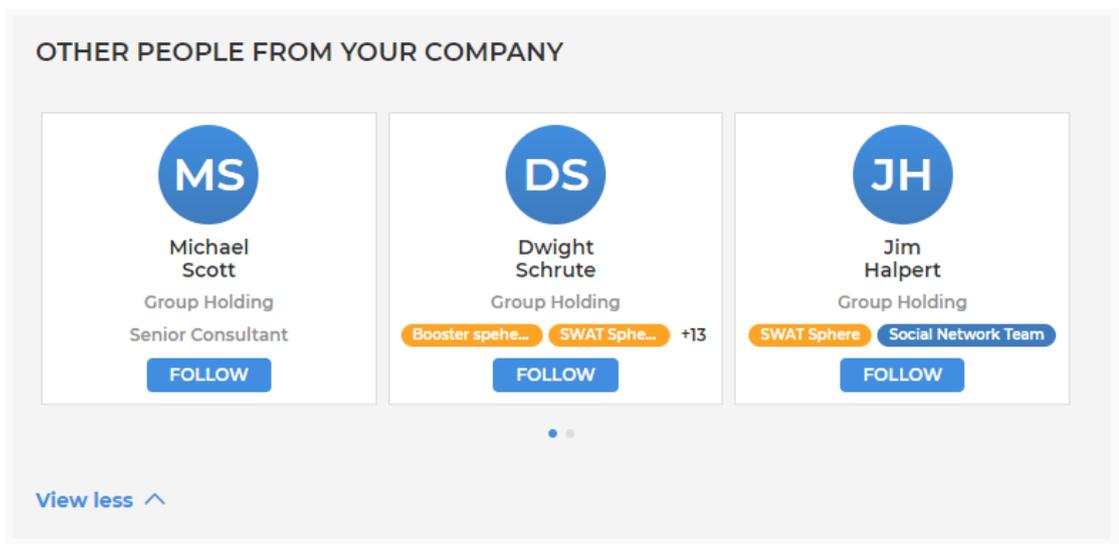


Figura 4.23: Carosello Persone, variante 2

Eventi

Il carosello degli eventi viene posto in posizione n.12 a causa dell'altro widget degli eventi consigliati già presente in spalla nella homepage di TamTamy. La differenza tra i due widget sta nel fatto che, in questo caso, gli eventi proposti vengono presi direttamente dal DB, senza alcuna elaborazione da Machine Learning.

Le varianti disponibili sono:

- Variante 1: eventi del programma Learn.Share.Remix e, pertanto, formativi. Il footer contiene un pulsante che, se pigiato, rimanda al portale Learn.Share.Remix;
- Variante 2: eventi localizzati nella stessa city dell'utente. Il footer contiene un pulsante che, se pigiato, rimanda alla sezione "All Events".

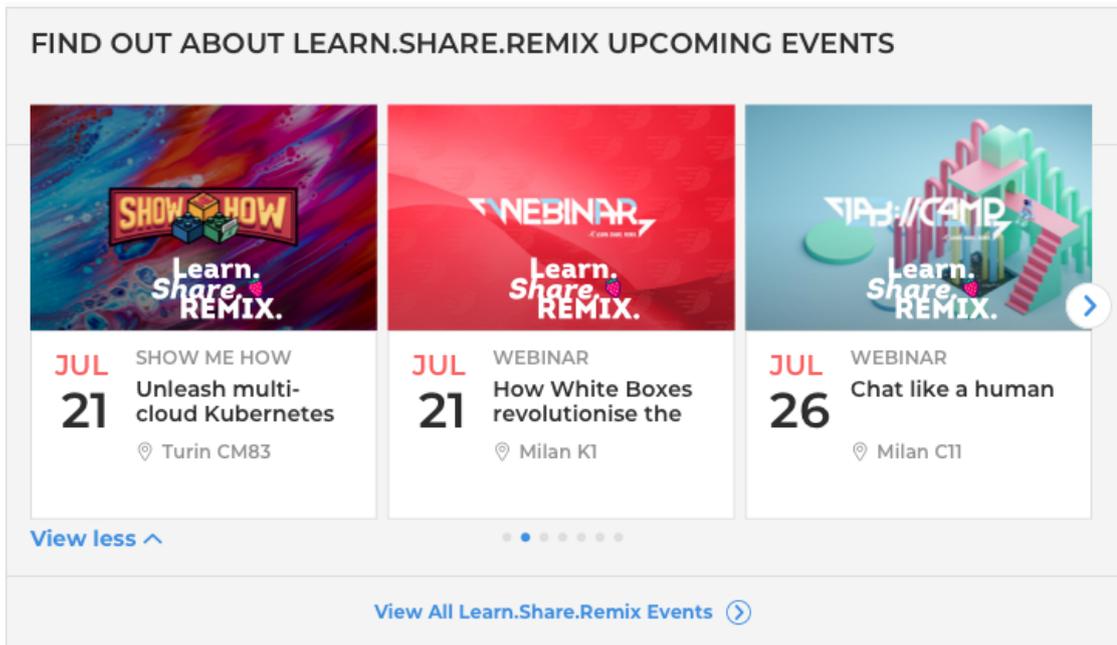


Figura 4.24: Carosello Eventi, variante 1

4.4.3 Highlight

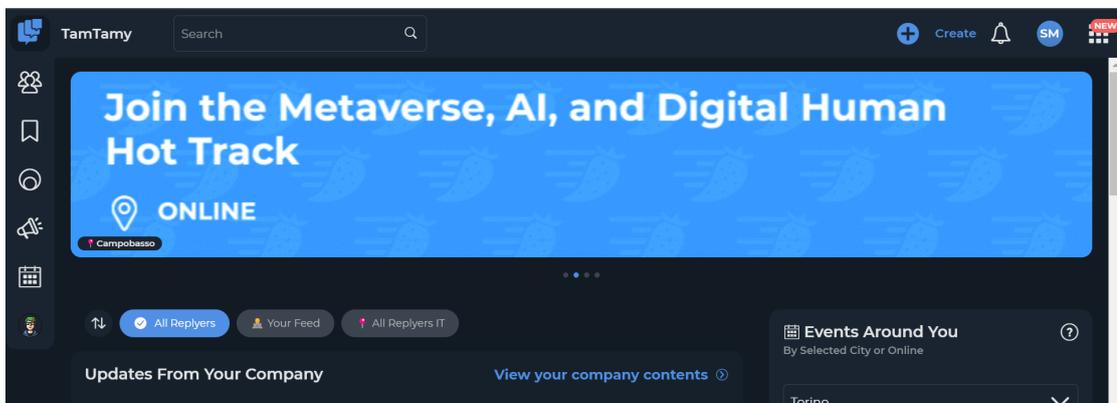


Figura 4.25: Highlight in modalità “wide”

Il widget degli highlight è uno strumento pensato per mettere in primo piano comunicazioni editoriali di diverso tipo, e viene posto nell’ head del body dell’home-page di TamTamy. Si tratta di un carosello contenente, per ogni pagina, una due o tre card. Una card highlight può contenere un’immagine o un’animazione lottie (animazioni vettoriali basate su JSON) o un video. Se, in fase di configurazione

si setta l’highlight come “wide” (mettendo la proprietà “wide” a true), la card occuperà tutto lo spazio disponibile nella pagina del carosello. La stessa cosa accade se l’highlight è un video e ha una descrizione (che verrà mostrata in un’area adiacente al player).

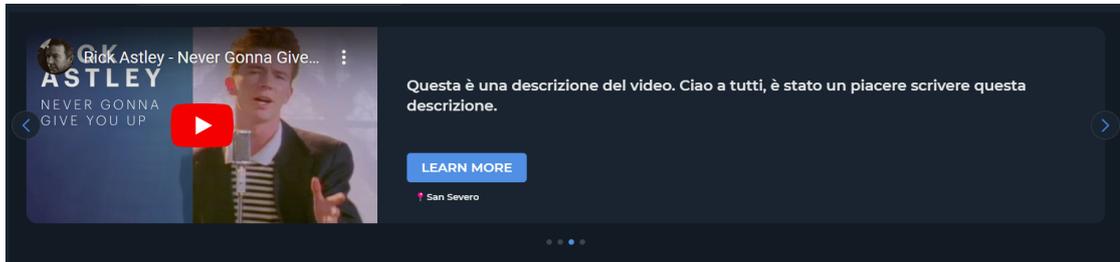


Figura 4.26: Highlight video con descrizione

In alternativa, le card divideranno tra loro lo spazio disponibile, per un massimo di tre highlight per pagina.

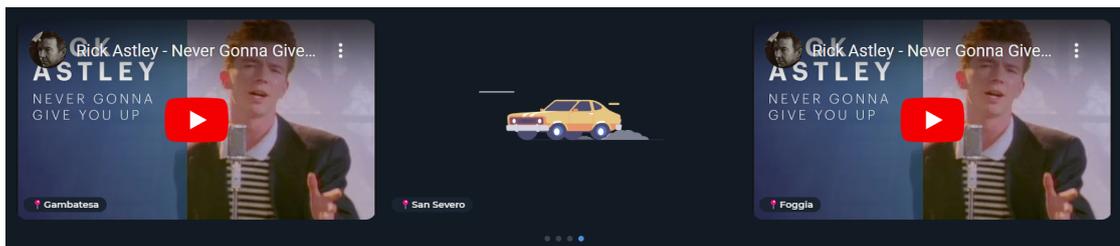


Figura 4.27: Highlight con property “wide” a false

Se si clicca sulla card, in caso di highlight-video il video viene fatto partire, mentre in caso di highlight-image o highlight-animation si viene reindirizzati al link passato in fase di configurazione. Per avere lo stesso comportamento per i video si dovrà cliccare sul pulsante “LEARN MORE”. È possibile, per un highlight, indicare la località (che verrà mostrata in un ribbon) e filtrare per city o per country (mostrare l’highlight solamente agli utenti di una certa città o di una certa country). Se non ci sono highlight da mostrare, il widget si chiude.

Poiché la configurazione del widget è permessa solo ai colleghi di RSN (Reply Social Network), si è deciso di quindi di non procedere con la creazione di una pagina di gestione degli highlight troppo elaborata: per configurare il widget degli highlight si passa un JSON nella prima text-area della pagina di “Carousel Management”.

4.5 Miscellaneo & Possibili evoluzioni

4.5.1 Funzionalità invita

Una buona strategia per sfruttare le potenzialità del Core Drive #5 (Influenza sociale e relazionale) è quella di aggiungere la possibilità di invitare colleghi agli eventi. L'idea è quella di dare due diversi significati all'azione:

1. “ci andiamo insieme?”: un consultant junior invita un collega a partecipare insieme ad un evento;
2. “ti consiglio questo evento”: un team lead invita un collega junior a iscriversi ad un evento.

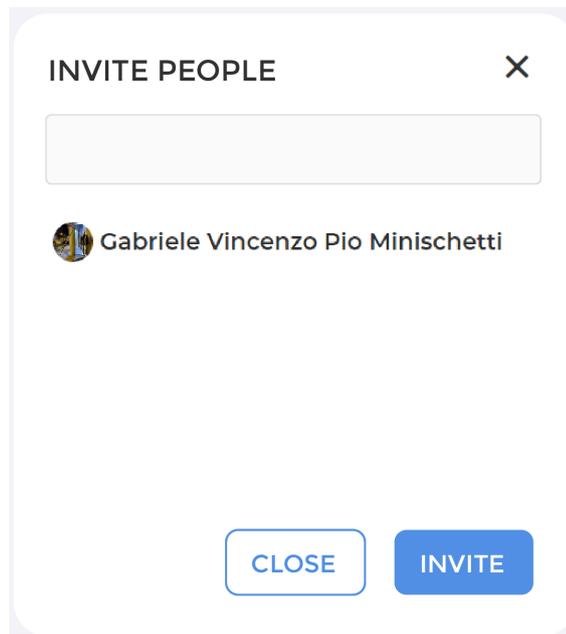


Figura 4.28: Modale di invito

La differenza tra i due casi sta nel fatto che nel primo l'evento viene visto, in ottica di Gamification, come una “missione di gruppo”: un utente A può invitare un utente B soltanto se l'utente A è già iscritto. Il secondo caso, invece, può essere considerato o come una forma di tutoraggio (un utente più esperto consiglia un evento ad un collega con meno esperienza) o come un semplice consiglio tra colleghi. Per poter accedere alla seconda funzionalità bisogna avere un title maggiore o uguale a quello di consultant senior. L'invito si traduce in una notifica (web e email) che porta alla scheda dell'evento consigliato. In questa maniera, l'utente

invitato ha la possibilità di informarsi in maniera adeguata (data e ora dell'evento, tematiche trattate ecc.) e decidere se iscriversi o meno.

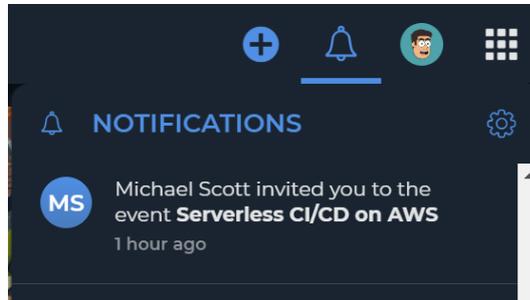


Figura 4.29: Notifica vista da Gabriele Minischetti

In caso di iscrizione, si invia una notifica di aggiornamento all'utente invitante.

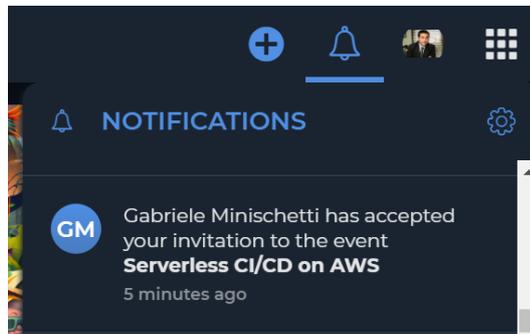


Figura 4.30: Notifica vista da Michael Scott

In futuro è da valutare se iniziare a tracciare gli inviti (Quante persone ha invitato Utente x? Quante di queste si sono iscritte all'evento? Quante hanno effettivamente partecipato?) in modo da sfruttare questa funzionalità anche per fare punteggio nell'eventuale competizione interna.

4.5.2 Avatar

Una delle prime idee accantonate è stata quella di far creare all'utente, subito dopo la FirstAccess, un avatar personalizzato. L'obiettivo era quello di sfruttare al meglio le potenzialità del Core Drive #4 (Proprietà & Possesso). Si era pensato, quindi, di associare il repler ad una sua rappresentazione digitale ulteriormente personalizzabile con item "acquistabili" solo tramite punti spendibili guadagnati svolgendo le diverse attività di onboarding. Tutto ciò è sembrato, però, essere fine a sé stesso: in assenza di un contesto di utilizzo (realtà virtuale, reaction in stile

Instagram ecc.), un avatar digitale sarebbe ancora uno strumento di engagement valido? Queste perplessità, unite alle difficoltà implementative (comunque superabili) e di mantenimento sono il motivo per cui l'idea è stata, almeno per il momento, messa da parte.

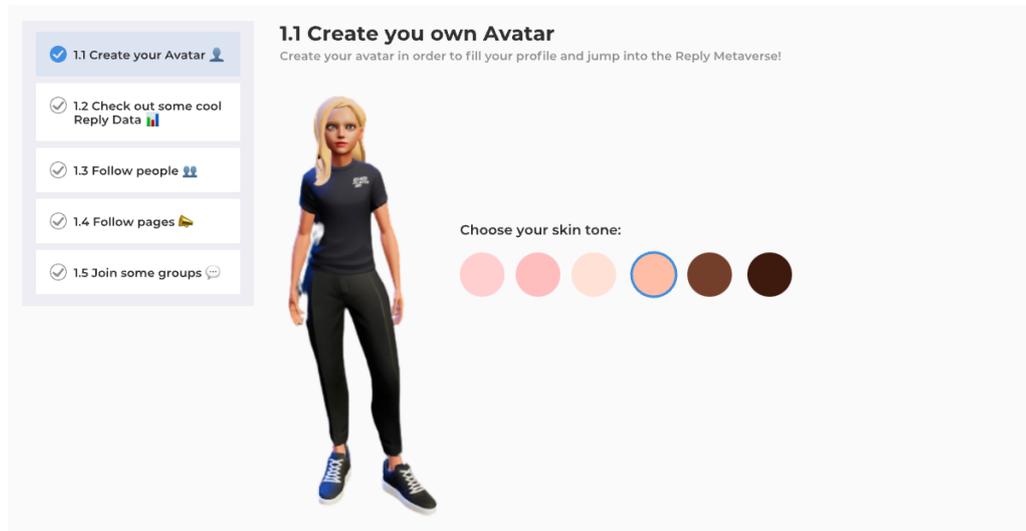


Figura 4.31: Mockup della pagina di creazione dell' avatar

4.5.3 Task randomici

Il widget dei task randomici va inserito nella posizione 1 della pagina del profilo dell'utente e, a differenza del widget "Onboarding task", non ha come obiettivo la scoperta del mondo Reply, ma l'incentivazione ad essere attivi. L'idea alla base del widget è quella di sfruttare le potenzialità del Core Drive #7 (Imprevedibilità e curiosità) e pertanto:

1. non sarà sempre visibile, ma apparirà sui profili di alcuni utenti (scelti randomicamente) e vi rimarrà solo per una certa quantità di tempo (ancora da definire);
2. il task proposto al replyer sarà scelto randomicamente da una lista che varia da personas a personas;
3. se si esegue il task prima che questo "scada" (ovvero, prima che il widget sparisca dal profilo dell'utente) sarà possibile, con una certa probabilità (ancora da definire), riscattare un premio: o punti per l'eventuale competizione interna o un gadget scelto randomicamente tra quelli messi a disposizione.

In questa maniera si sfruttano gli “appuntamenti” del Core Drive #6 (Scarsità e impazienza) e le “opportunità evanescenti” del Core Drive #8 (Perdita & Dipendenza).

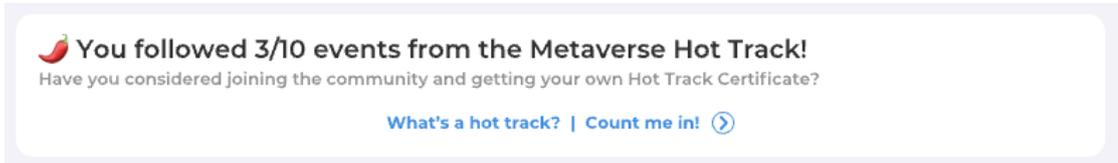


Figura 4.32: Esempio di task randomico

Il widget è stato inserito tra gli sviluppi futuri sia per il suo legame con l’eventuale competizione interna sia per non togliere, nel frattempo, priorità al widget “Onboarding task”.

4.5.4 Chatbot

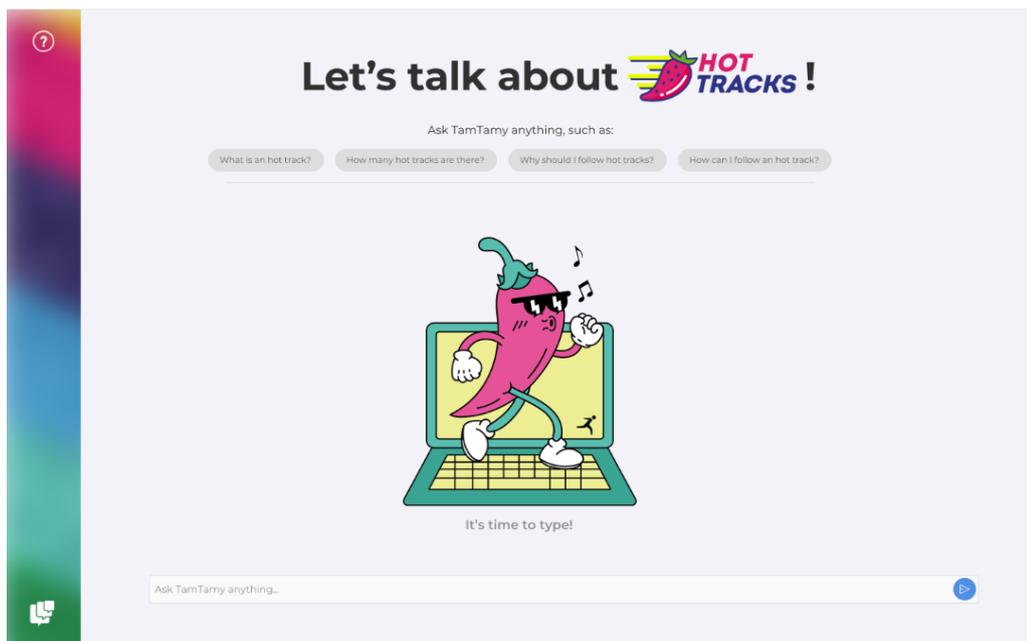


Figura 4.33: Pagina Chatbot

Una cosa molto utile soprattutto a chi ha poca conoscenza del mondo Reply potrebbe essere un Chatbot basato su AI, specializzato quindi nella conversazione con veri utenti e capace di dare prontamente risposte utili e soddisfacenti.

Capitolo 5

Sviluppo

Ogni widget si compone di una parte frontend e di una parte backend. La prima consiste nell'interfaccia (con cui interagisce l'utente) e un insieme di operazioni di più alto livello (primi controlli, chiamate API ecc.). La seconda invece si occupa di inviare, già da prima del caricamento del widget, dati necessari al FE. I metodi responsabili di ciò sono due:

- `buildOptions()` che ritorna informazioni che sono uguali in ogni contesto (per esempio, il titolo di una stessa sezione mostrata a due utenti diversi) o che vengono recuperate facilmente dalla tabella delle option del widget. Il bean prodotto è un “OptionsBean” (esempio: `ExampleOptionsBean`);
- `buildData()` che si occupa di passare tutto ciò che deve essere calcolato, anche in base al contesto (ovvero in caso di dati che variano da utente ad utente, da account ad account). Il bean prodotto è un “DataBean” (esempio: `ExampleDataBean`).

Il nome della componente frontend viene scritto sempre in kebab-case ed ha come prefisso “tt” (esempio: `tt-example`). La componente backend, invece, ha il nome in CamelCase e presenta sempre “Widget” come suffisso (esempio: `ExampleWidget`). Non tutti i dati vengono forniti in maniera sincrona al FE, molti arrivano in modo asincrona dalle API. Ciò accade per evitare di rallentare troppo il caricamento della pagina oppure come conseguenza di scelte prese dall'utente o a seguito di operazioni effettuate dallo stesso. Le API vengono implementate a backend nelle classi il cui nome, scritto in CamelCase, termina col suffisso “Controller” (esempio: `ExampleController`). Le richieste delle API vengono soddisfatte dai servizi, implementazioni di logiche di business che al loro interno possono chiamare anche altri servizi e/o DAO (Data Access Object, responsabili di interagire direttamente con il DB). Tra BE e FE, le informazioni viaggiano all'interno dei bean, che variano in base al contesto.

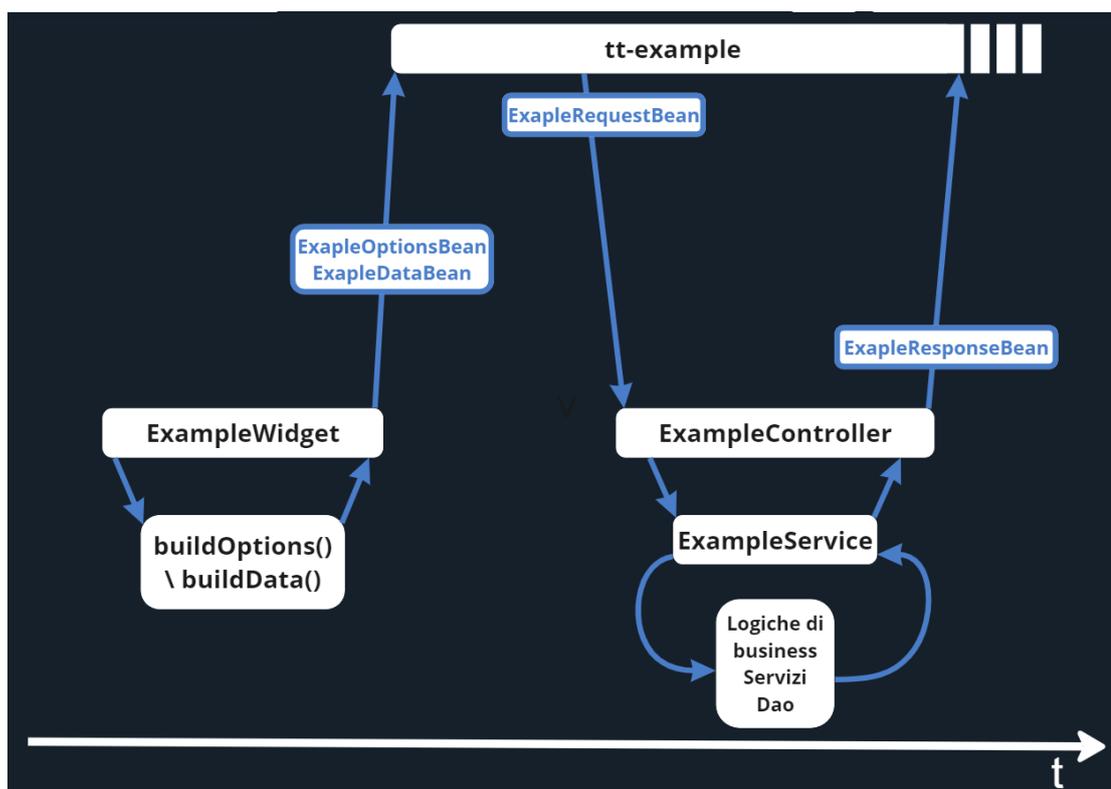


Figura 5.1: Architettura dei widget

5.1 FirstAccess

Al primo accesso, l'utente inserisce la propria mail aziendale e la password. Avviene, quindi, l'autenticazione sul server di Login di Reply (Active Directory) tramite il protocollo LDAP (Lightweight Directory Access Protocol), un protocollo di Microsoft per gestire in maniera centralizzata le autenticazioni delle utenze. L'autenticazione è del tipo SSO (Single Sign-On) per evitare di effettuare il login per ogni servizio di TamTamy. Se le credenziali di accesso sono corrette e se l'utente è già presente nella tabella Kc_User si viene reindirizzati all'homepage altrimenti, se non esiste alcuna tupla sull'user in questione, si viene reindirizzati alla pagina di firstAccess per completare la registrazione del nuovo replyer. In questa pagina sono stati inseriti sia il widget di presentazione che il widget di FirstAccess.

5.1.1 Frontend

tt-first-access

All'avvio il widget mostra la prima delle tre slide di presentazione di TamTamy. Cliccando sul pulsante "START" dell'ultima slide si disattiva la sezione relativa all'accoglienza su TamTamy e si attiva quella relativa alla creazione della nuova utenza. Il passaggio dalla prima sezione alla seconda avviene gradualmente, grazie ad un'animazione sull'opacità resa possibile dal tag `<transition>`. La sezione di creazione dell'utente si compone di due slide. Dalla prima si può accedere alla seconda pigiando il pulsante "Continue" che chiama la funzione `continue()` responsabile di consentire il passaggio solo se si superano dei controlli preliminari:

- l'utente ha inserito un nickname;
- il nickname scelto non contiene caratteri speciali;
- l'utente ha selezionato un genere;
- l'utente ha spuntato la casella.

Superati i controlli si attiva lo slider e si accede alla seconda slide in cui l'utente può scegliere fino a un massimo di 5 chips. Al click del pulsante "OK, LET'S GO" si chiama la funzione `doResolve()`, la quale avvia il processo di creazione dell'utenza. In questo caso il controllo preliminare è solo sul numero di chips selezionate che deve essere maggiore di 0 e minore di 5. Come per la slide numero uno, qualora dovesse essere individuato un problema verrebbe stampato il messaggio d'errore sotto al rispettivo componente. Se non ci sono errori, si chiama quindi la funzione `stateOn()` che disattiva la possibilità di interagire con tutti i componenti eccetto che con lo slider: all'utente viene lasciata la possibilità di muoversi tra le due slide. Nel frattempo, si procede inserendo le informazioni nel bean di richiesta che viene passato come parametro all'API. In caso di successo, la registrazione è avvenuta correttamente e l'utente viene reindirizzato al proprio profilo di TamTamy. In caso di fallimento, invece, si stampa il messaggio d'errore sotto al component coinvolto.

5.1.2 Backend

FirstAccessAction

La `FirstAccessAction` è l'action che si occupa di gestire la pagina del primo accesso. Nella vecchia implementazione il metodo `signup()` era responsabile sia di caricare la pagina che di creare l'utenza al termine della registrazione. Nella nuova implementazione il metodo è stato rinominato in `input()` e si occupa semplicemente di caricare la pagina, mentre la gestione della registrazione del nuovo utente viene lasciata all'API `"FirstAccessController"`. Proprio per questo motivo di tutti i controlli

che venivano fatti nella vecchia action, nella nuova è rimasto solo quello sull'utente in sessione: ci si assicura che l'utente sia effettivamente al suo primo accesso su TamTamy tramite il metodo `getSession()`. In caso di errore, tramite il metodo `buildErrorPage()` si reindirizza l'utente alla seguente pagina d'errore. In caso di

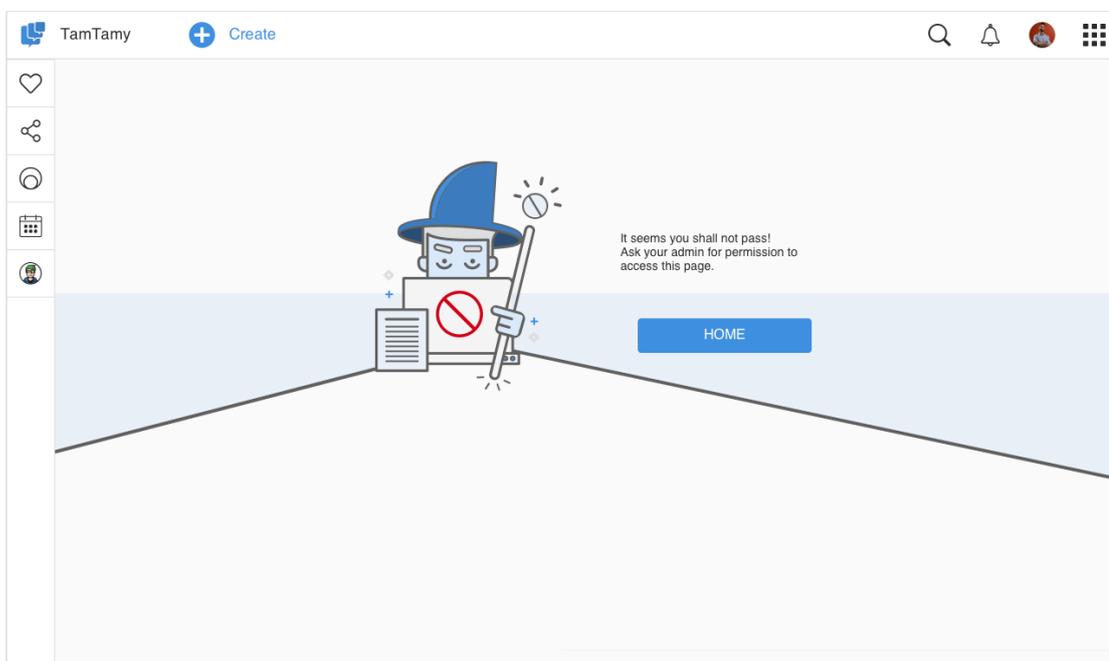


Figura 5.2: Pagina d'errore

successo, invece, si procede con l'inizializzazione della pagina tramite il metodo `buildPageLayout()` che, eseguendo delle find in DB, permette di risalire ai widget che devono essere caricati.

FirstAccessWidget

Il widget BE è stato implementato per fornire al FE, tramite il metodo `buildOption()` la lista di topic da proporre all'utente.

FirstAccessController - PUT & POST

I topic proposti non possono essere sempre gli stessi: col passare del tempo e l'arrivo di nuovi trend tecnologici potrebbero nascere nuovi argomenti di interesse e morirne altri. La PUT serve proprio per risolvere questo problema: quando Machine Learning effettua una modifica, passa alla PUT del `FirstAccessController` un `RequestBean` contenente la lista con tutti i topic. Lato TamTamy si chiama il servizio `checkAndUpdate()` che dapprima effettua gli opportuni controlli (verifica

che la lista non sia vuota e che il chiamante abbia il giusto token identificativo) e poi, attraverso il servizio `updateMachineLearningTopic()`, provvede ad aggiornare il valore memorizzato in DB.

La POST della `FirstAccessController` si occupa invece di gestire la registrazione del nuovo user. Il servizio responsabile di ciò è il `checkAndCreateUser()` che:

1. si assicura che il nuovo utente non sia già registrato a TamTamy e che le informazioni nel bean di richiesta siano valide (il nickname scelto non deve essere vuoto, non deve avere una lunghezza superiore ai 30 caratteri, non è stato già scelto da un altro utente ecc.);
2. inserisce l'utente in tutti i gruppi di default tramite il servizio `addUserToDefaultGroups()`;
3. memorizza l'accettazione del TamTamy User Agreement e della policy tramite i servizi `saveAcceptAgreement()` e `handleAutomaticPolicy()`;
4. invia, tramite il servizio `sendFavouriteTopicsToMachineLearning()`, i topic scelti dall'utente a Machine Learning.

Se gli step numero 3 e 4 non vanno a buon fine verrà mostrata all'utente una pagina di errore di TamTamy, nella quale gli si comunicherà che la creazione non è andata a buon fine e lo si inviterà ad aprire un ticket.

5.2 Onboarding task

5.2.1 Frontend

Lo sviluppo del widget frontend ha richiesto la realizzazione di due nuove componenti: la `tt-task-list` e la `tt-task-card` (contenuta all'interno della `tt-task-list`).

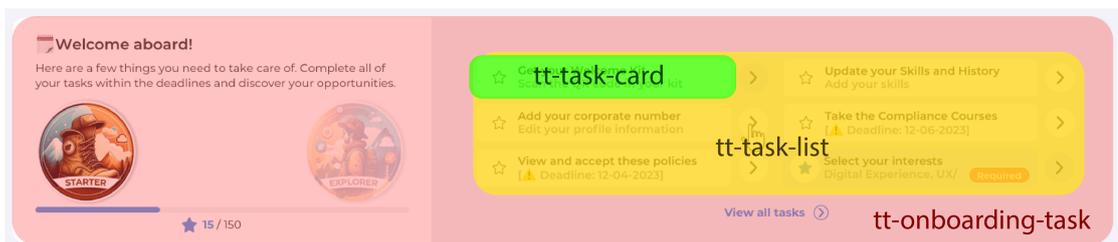


Figura 5.3: Composizione widget FE

tt-task-card, tt-task-list & tt-onboarding-task

La `tt-task-card` è la rappresentazione del singolo task. Per ogni attività riporta il titolo, il sottotitolo e un'icona per indicare se il task è stato svolto o meno. Per questo motivo le tre options accettate dalla card sono: `title` (String), `subtitle` (String) e `Done` (Boolean).

La `tt-task-list`, invece, è la rappresentazione della lista di attività proposte all'utente. Come option accetta la lista di task (Array di Object), il `NumberOfPagination` (Number) e il `NumberOfColumn` (Number). L'array ricevuto, prima di essere passato all'html, viene rimaneggiato dalla funzione `computeTaskList()`. Questa ordina la lista in primis per `Done` (dando maggiore priorità alle attività non svolte) e poi per chiave. Successivamente si chiama la funzione `computeTaskSublists()` che raggruppa gli elementi in base al `NumberOfColumn` e ritorna una lista di liste. Infine, se `NumberOfPagination > 0`, il risultato ottenuto dalla `computeTaskSublists()` viene filtrato in modo da passare all'html solo i primi N elementi (con $N = \text{NumberOfPagination}$). Nell'html si usa un doppio `v-for` per costruire la lista di task. Il `v-for` interno contiene una `tt-task-card` e un'arrow. Alla `tt-task-card`, come `subtitle` si passa la deadline, se presente, o in alternativa il sottotitolo ricevuto dal BE. Al click dell'arrow se la variabile `Link` (ricevuta anch'essa dal BE) è valorizzata si aprirà una nuova pagina al link indicato, altrimenti tramite un booleano (combinato alla direttiva `v-if`) si aprirà la pagina di dettaglio del task che mostrerà la `Description` ricevuta dal BE.

Il `tt-onboarding-task` si adatta in base all'option ricevuta:

- `big` si estende come fa nel body. Contiene una `tt-task-list` con `NumberOfColumn = 2` e `NumberOfPagination = 6`;
- `small` occupa uno spazio orizzontale più ristretto, sviluppandosi maggiormente in verticale. Contiene una `tt-task-list` con `NumberOfColumn = 1` e `NumberOfPagination = 5`.

La lista di task, assieme al livello in corso e l'eventuale livello successivo, viene ritornata dall'API chiamata nel lifecycle hooks `created`. Per ogni livello di gamification sono indicati, all'interno del `tt-onboarding-task.json`, titolo, sottotitolo e icona del Badge. Al click del footer si apre una modale contenente, anch'essa, una `tt-task-list` però con `NumberOfColumn = 1` e `NumberOfPagination = 0`.

5.2.2 Backend

Come deciso in fase di progettazione, al termine della fase `Welcome on board` il widget perde di priorità e, dal body, viene spostato in spalla. Per rendere ciò possibile sono stati inseriti nella pagina del profilo dell'utente due widget, uno in

spalla e uno nel body. L'attivazione di uno o dell'altro avviene grazie a un comando inserito nella colonna "ENABLED" della tabella dei widget:

- "\$networkUser.onboardingPhase == "Welcome_on_board"" per il widget nel body;
- "\$networkUser.onboardingPhase == "Onboarding_Long_Term"" per il widget in spalla;

Infine è stata aggiunta, nella KcNetworkUser una nuova colonna relativa alla fase di onboarding in cui si trova l'utente: l' onboardingPhase.

OnboardingTaskWidget

Il widget BE è la componente meno complessa di tutto il flusso. La buildOptions ritorna, nell'OptionsBean, il numberOfPagination (ovvero il numero di task che il widget deve mostrare per pagina) e il type ("small" per quello in spalla, "big" per quello nel body).

I controlli per la visibilità del widget sono gli stessi del UserDonutChartWidget.

OnboardingTaskController - GET

Inizialmente, per la GET, si era pensato a due soluzioni alternative per la ricerca delle attività da proporre all'utente:

- Eseguire una query per ogni task
 - VANTAGGIO: nessuna ridondanza di informazioni introdotta;
 - SVANTAGGIO: tempi di computazione molto elevati;
- Creare una nuova tabella dentro cui memorizzare se un task è stato svolto o meno
 - VANTAGGIO: tempi di computazione estremamente ridotti;
 - SVANTAGGI: ridondanza di informazione + controlli disseminati nel codice che a lungo andare diventano inutili (Esempio: ogni volta che un utente si iscrive a un evento bisogna leggere in DB se il task relativo alla registrazione degli eventi è stato già svolto. Per ora questi controlli non si farebbero più nel caso in cui l'utente fosse nella fase "Reply Employees Experience", ma non è detto che questa condizione non cambi in futuro);

Poiché gli svantaggi introdotti da entrambe le alternative non erano trascurabili, si è deciso di implementare una soluzione ibrida. È stata quindi creata una nuova tabella, la "KcUserTasks", in cui ogni tupla è composta da UserID (chiave),

Badge (chiave) e TasksValue (contenente un JSON). Lato codice, dato l'utente si prende il TasksValue per cui il Badge ha una data nulla (poichè il Badge è un oggetto KcBadge, si sfrutta Hibernate: "b.date = NULL"). Il valore ottenuto viene trasformato in una map<String, Boolean> e in base al badge si chiama l'opportuno servizio :

- Badge = Onboarding_Level_1 => getOnboardingLevelOneTasks();
- Badge = Onboarding_Level_2 => getOnboardingLevelTwoTasks();
- Badge = Onboarding_Level_3 => getOnboardingLevelThreeTasks().

All'interno del servizio, per ogni task si controlla se il valore nella mappa è a TRUE. Se a FALSE, o se la chiave è inesistente, si fanno le opportune ricerche in DB e nel caso si aggiorna la mappa. Per i "task custom" si controlla anche la Company dell'utente. La mappa ritornata viene riconvertita in JSON e memorizzata in DB al posto della precedente (ovviamente solo se ci sono differenze con quella di partenza). Oltre a ciò, si controllano i valori della mappa: se sono tutti a TRUE si invoca il servizio "onboardingNextLevel()", che aggiunge una nuova tupla sia nella KcBadge (relativa al badge appena ottenuto) che nella KcUserTasks (relativa al nuovo livello, se c'è). In quest'ultimo caso si sfrutta l'opportuno servizio "getOnboardingLevel..." (visto poc'anzi) a cui si passa come parametro una mappa vuota. Per risalire ai dettagli di ogni task, per ogni chiave della mappa finale si fa una ricerca nella KcTask, altra tabella creata ad hoc, che per ogni task indica:

- Chiave, lo stesso valore usato nei JSON e nelle mappe come chiave;
- Titolo da mostrare al FE;
- Deadline, un timestamp da sommare alla data di assunzione del replyer;
- Sottotitolo da mostrare al FE (alternativa alla deadline);
- Link, a cui reindirizzare l'utente al click della freccia;
- Description, una descrizione da mostrare nella sezione di dettaglio del widget (alternativa al link).

Tutte queste informazioni, assieme al booleano di risoluzione del task, vengono passate al costruttore del TaskBean. Il ResponseBean inviato al FE (e memorizzato in cache) contiene una lista di TaskBean e i Badge del livello in corso e dell'eventuale livello successivo.

Ogni volta che si esegue una particolare azione su TamTamy (esempio, ci si iscrive a un evento), si guarderà il valore della variabile onboardingPhase contenuta nel networkUser e se diverso da "Reply_Employees_Experience" si svuoterà la cache dei task onboarding.

5.3 Widget delle user project cover

5.3.1 Frontend

`tt-user-project-cover-carousel`

In base alla configurazione ricevuta dal backend, il frontend decide se renderizzare o meno il widget (tramite la direttiva `v-if`). In caso positivo, si invoca l'API nel lifecycle hooks "created". Se la lista ritornata è vuota, il widget si chiude con un animazione resa possibile dal tag `<transition>` e sparisce, altrimenti, tramite direttiva `v-for`, si passa ogni elemento alla `tt-user-project-cover-card` (card creata ad hoc) responsabile di rappresentare l'oggetto.

5.3.2 Backend

`UserProjectCoverCarouselWidget`

La `buildOptions()` invia al FE il titolo, il sottotitolo, un identificativo dell'API da invocare e il numero di card da mostrare (`numberOfResult`).

La `buildData()`, invece, è responsabile di calcolare se il widget deve essere renderizzato o meno. Attraverso il servizio "isReturner()" si controlla, guardando nella `kc_user_meta`, se il proprietario del profilo è un returner:

- caso `TRUE`, si mette a `TRUE` il booleano nel `UserProjectCoverCarouselDataBean`;
- caso `FALSE`, si calcola tramite il servizio `getJoinDate()` il numero di mesi trascorsi dall'assunzione e confrontandolo col valore della costante "ONBOARDING_PHASE_ONE_VALUE" (che indica la durata della fase "Welcome on board") si imposta di conseguenza il booleano nel `UserProjectCoverCarouselDataBean`.

`UserProjectCoverCarouselController` - GET, POST & PUT

Le project cover sono considerate come contenuti di tipo "project_cover" e pertanto vengono memorizzate a DB nella tabella "kc_content". Gli utenti associati alla project cover sono registrati nella `kc_content_user`, ognuno in una tupla che ha per chiave primaria l'id del contenuto e l'id dell'utente. I titoli e i testi delle `UserProjectCover` (ovvero le project cover che compaiono nel profilo dell'utente) vengono inseriti nella `kc_user_meta` che ha come chiavi primarie l'userID e uno string identificativo del meta (che in questo caso ha come valore l'id del contenuto).

Il DAO `getProjectCoverByUser()` ricava dalla `kc_content_user` la lista di "potenziali" `UserProjectCover`, predicando sull'id del proprietario del profilo. Sfruttando il

risultato ottenuto, il DAO `getUserProjectCoverMeta()` ricerca nella `kc_user_meta` l'insieme delle Project Cover di cui l'utente ha già "narrato" il proprio operato. Sempre dalla `kc_user_meta` si estrae anche la tupla contenente la lista di UserProjectCover preferite dal proprietario del profilo. Se quest'ultima possiede n valori, con $n \leq \text{numberOfResult}$ (prop inviata dal widget BE), dall'insieme ritornato dalla `getUserProjectCoverMeta()` ne verranno scelti randomicamente m , con $m = \text{numberOfResult} - n$. Se il numero di UserProjectCover trovate in DB è inferiore a `numberOfResult` ci si comporterà di conseguenza, selezionando ovviamente un numero inferiore di elementi. Il risultato ottenuto viene quindi memorizzato in cache, inserito nella `UserProjectCoverCarouselResponseBean` e ritornato dalla GET.

La POST viene chiamata dalla pagina della Project Cover. In primis vengono eseguiti i controlli sull'utenza (per verificare che l'user sia associato alla Project Cover di cui sta cercando di scrivere) e dei valori immessi. Superati i check si procede quindi con la creazione dell' `userMeta` del titolo e del testo e, nel caso, la creazione o l'aggiornamento della lista delle `userProjectCover` preferite. Infine si svuota la cache associata all'utente in sessione.

La PUT funziona nella stessa maniera della POST, con l'unica differenza che lo `userMeta` del titolo e del testo non viene creato, ma preso dal DB e aggiornato.

5.4 GeCo donut chart

5.4.1 Frontend

tt-user-donut-chart & tt-donut-chart

La struttura HTML del widget FE del `tt-user-donut-chart` è molto semplice, in quanto la principale funzionalità (quella del grafico a "ciambella") è stata implementata nel "`tt-donut-chart`", un nuovo componente creato ad hoc. Quest'ultimo accetta come props (con `required` a true) un `Object` al cui interno sono indicati gli elementi da rappresentare nel grafico e rispettiva percentuale. L'object in questione viene quindi elaborato dalla computed "`stats()`", responsabile di associare ad ogni elemento un colore, per un massimo di 20 colori. I colori del grafico, infatti, non vengono scelti casualmente, ma selezionati dal Design System, dall'apposita (nuova) sezione "`$stats-colors`" nel file "`_tamtam.scss`". In totale i colori sono $39 + 1$ poichè i colleghi di GeCo hanno indicato 35 come numero massimo di commesse a cui un Replyer ha lavorato in un mese. Il numero medio di commesse rimane, tuttavia, molto più basso: circa 5 al mese. La computed `stats()` non ha solamente il compito di associare i colori agli elementi, ma è responsabile anche di assicurarsi che questi non siano più di 39. Qualora non fosse così, gli elementi con percentuali più basse verrebbero uniti sotto la voce "Other" (e le loro percentuali sommate).

5.4.2 Backend

UserDonutChartWidget

Lo UserDonutChartWidget è stato pensato per implementare un qualsiasi grafico a “torta” nella pagina dell’utente. La sua configurazione avviene, come per tutti i widget, tramite options: in DB si indica il titolo, uno String in base al quale il FE decide quale API chiamare, l’URL del footer e la visibilità. Quest’ultimo valore, a differenza dei primi tre, non viene inviato al frontend ma serve alla buildData() per indicare, tramite un booleano inserito nell’ UserDonutChartDataBean, se il widget deve essere visibile o meno dall’user in sessione.

La visibilità può avere due valori:

- “ALL”, il widget verrà sempre renderizzato;
- “LIMITED”, il widget verrà renderizzato solo se l’utente in sessione è il proprietario del profilo, o il suo manager o il partner della sua company. Gli step per effettuare questi controlli sono, quindi, 2:
 1. si controlla il valore Booleano “isMyProfil” presente nell’ EntityContext-Bean, un bean, calcolato a monte dalla UserAction, che viene passato ad ogni widget della pagina dell’utente;
 2. si invoca il DAO isUserACompanyTitle(), che prende come parametro un userID, il nome di una company e un title. Ovviamente il DAO viene chiamato due volte. Nei due casi si passa sempre lo stesso userID (quello dell’utente in sessione) e la stessa company (quella del proprietario del profilo) ma title diversi: prima “MANAGER” poi “PARTNER”.

UserDonutChartController - GET & PUT

Come per le GET dello StatsController e del RecommendedListController, anche nella GET dell’UserDonutChartController si decide quale servizio invocare in base a un certo valore contenuto dal bean di richiesta. In questo caso l’unico servizio ad essere stato implementato è il getGeCoStats() che controlla se nella UserDonutChartCache sono già presenti le statistiche dell’utente indicato nel bean di richiesta. Se sono già salvate delle informazioni, queste vengono inserite nel bean di risposta e inviate al FE. Altrimenti, tramite il servizio getStatsFromGeCo() si contatta GeCo per ottenere le statistiche desiderate. In questo caso, se il risultato ritornato è valido viene memorizzato in cache, altrimenti si ritorna un messaggio di errore.

Per avere dei dati sempre aggiornati è stata realizzata anche la PUT, il cui compito è quello di svuotare la UserDonutChartCache dell’user indicato nel bean di richiesta. Questo metodo viene chiamato da GeCo ogni volta che un utente effettua una nuova consuntivazione.

5.5 Widget delle statistiche

L'organizzazione della struttura del widget delle statistiche è simile a quella del carosello dei suggerimenti: widget BE specifici associati allo stesso widget FE che si adatta in base alle esigenze. Le associazioni inserite nella classe enum “WidgetType” sono:

- CONNECTION_STATS che associa il tt-stats al ConnectionStatsWidget;
- EVENT_AND_LEARN_STATS che associa il tt-stats al EventAndLearnStatsWidget.

5.5.1 Frontend

tt-stats

Il tt-stats è il guscio più esterno del widget FE e definisce la forma di tutte le varianti e l'organizzazione degli elementi ricorrenti: icona, titolo, sottotitolo e statistiche. Queste ultime vengono calcolate in maniera asincrona da un'API, la GET della StatController. In caso di fallimento la lista sarà vuota. In caso di successo si effettua un mapping tra le label (prese dal JSON “tt-stats-options.json”) e i risultati ottenuti. Questo passaggio è stato inserito per evitare, qualora un domani si dovesse presentare la necessità di cambiare una delle label, di andare a modificare anche il backend. La lista finale viene passata al partial, responsabile di rappresentarla.

Partial

Le statistiche sono gestite dai due rispettivi partial: tt-connection-stats e tt-event-and-learn-stats. Ognuno degli elementi per cui mostrare le statistiche viene ricavato dalla lista passata dal widget padre e rappresentato dalla tt-stat-card, un nuovo componente creato ad hoc. Se la lista è vuota si stamperà un messaggio in cui si inviterà l'utente a contattare l'assistenza.

5.5.2 Backend

StatsWidget

Lo StatsWidget si occupa di indicare qual è la “specializzazione” del widget delle statistiche. Oltre al titolo, il sottotitolo, l'icona e eventuali URL, il widget BE invia alla parte FE anche il nome della variante scelta. Per adesso le possibilità sono due:

- “CONNECTION”, per indicare la variante “Connection stats”;

- “EVENT&LEARN” per indicare la variante “Event&Learn stats”.

StatsController - GET

In base alla variante riportata nel StatsRequestBean si decide il comportamento dell’API:

- caso “CONNECTION” si chiameranno i servizi:
 - countFollow(), che esegue tramite DAO due count in DB per ritornare il numero di follower e utenti seguiti;
 - countGroups&Pages(), che tramite DAO conta in DB il numero di pagine e gruppi seguiti.
- caso “EVENT&LEARN” si chiameranno i servizi:
 - computeEventsStats(), responsabile di eseguire, tramite DAO, query per ottenere il numero totale di registrazioni, il numero di eventi frequentati, il numero di trainigs e il numero di certificazioni “Hot Track” prese ;
 - computeLearnStats(), il cui compito è quello di calcolare le statistiche dell’utente relative al portale “Learn.Share.Remix”.

In ogni caso le informazioni vengono inserite in una mappa di tipo <String, Long> che verrà inviata al frontend all’interno del bean di risposta.

5.6 Medagliere

5.6.1 Backend

CertificationController - GET

Come già anticipato nel capitolo di progettazione, l’ API per farsi ritornare le certificazioni già esiste e pertanto è stato necessario aggiornarla. Le certificazioni non sono memorizzate da TamTamy, ma da Reply Discovery: la vecchia API, tramite il servizio getCertificationFromDiscovery() si faceva ritornare la lista completa delle certificazioni. Con questo nuovo sviluppo è stato chiesto a Discovery di fornire un nuovo meta in base al quale, lato TamTamy, catalogare i dati ricevuti. Questo meta viene poi mappato a FE con l’opportuna icona da mostrare.

Per i badge di TamTamy, invece, è stata creata una nuova tabella: la kc_badge. Ogni colonna indica un badge diverso e contiene la data di acquisizione del riconoscimento: questa scelta è stata fatta sia per ottimizzare la quantità di informazioni (se non c’è alcuna data significa che il badge non appartiene al medagliere dell’utente), sia perchè di uno stesso badge ci possono essere diverse versioni (Reply Award 2023,

Reply Award 2024 ecc.). Il metodo `getTamTamyBadge()` dato l'utente, restituisce un oggetto `KcBadge` al cui interno sono indicate tutte le "targhette di TamTamy" conquistate dall'utente.

Infine, sono stati aggiornati sia il `ResponseBean` che la `Cache` dedicata, in modo tale da rendere possibile la memorizzazione di tutte queste nuove informazioni.

5.7 Events around you

5.7.1 Frontend

`tt-event-carousel`

Il widget è parzialmente asincrono: le informazioni statiche (titolo, sottotitolo, icona, ecc.) vengono passate dal widget BE, mentre gli eventi vengono ritornati dall'API chiamata nel lifecycle hooks "created". Poiché la stessa API deve essere chiamata ogni volta che l'utente sceglie una nuova città dal menù a tendina, è stata creata una watch della variabile "selectedCity" che, all'occorrenza, si occupa di invocare la funzione `fetchData()` responsabile di far partire la chiamata API.

5.7.2 Backend

`EventCarouselWidget`

Per il widget BE sono stati implementati sia `buildOptions()` che `buildData()`. Il primo ritorna a FE informazioni come il titolo, il sottotitolo, l'icona, la label del footer ecc, mentre il secondo è responsabile di passare al FE la città di default del widget per l'user in sessione e l'insieme contenente tutte le città in cui c'è almeno una sede Reply.

`EventRecommendedController - POST`

L'`EventRecommendedController` viene utilizzata sia dal `Carousel Management` che dal widget "Events around you".

Nel primo caso tramite la `createBody()`, in particolare grazie all'invocazione del servizio `checkAndSaveBeanRecommended`, si controlla dapprima la correttezza del `requestBean` e si procede poi con:

1. il completamento delle informazioni da passare a ML: la data, per ogni evento indicato nella lista, viene estratta dal DB;
2. la conversione della lista di eventi in JSON tramite il metodo `buildJsonOut`;
3. l'invio del JSON all'endpoint fornito da ML.

La `indexBody()` invece è l'API responsabile di ritornare gli eventi selezionati e ordinati da ML. Il servizio `getEventListBeanRecommended()` controlla dapprima nella cache se ci sono dei risultati già salvati per la città per cui si richiede la lista di eventi. Se non ce ne sono si invoca il metodo `getRecommendedEventFromMachineLearningServer()` che contatta l'endpoint fornito da ML. L'`userID` dell'utente in sessione viene criptato e passato come argomento, assieme al numero di risultati richiesti e la città per cui si vogliono conoscere gli eventi. La risposta di ML è un JSON che viene, quindi, convertito in una lista che viene salvata in un bean di risposta che viene memorizzato in cache e ritornato a FE.

5.8 Widget dei suggerimenti

I diversi caroselli hanno una struttura frontend molto simile tra loro, ma criteri di selezione del tutto diversi. Pertanto si è deciso di realizzare un unico widget FE che si adatta in base alle configurazioni passate dal BE.

5.8.1 Frontend

tt-recommended-list

Il `tt-recommended-list` contiene al suo interno:

- componenti e logiche comuni ad ogni carosello `recommended`;
- `partial`, pezzi di codice responsabili della personalizzazione del widget FE. In totale sono quattro, uno per ogni carosello `recommended`, e ne viene attivato, tramite una direttiva `v-if`, solo quello corrispondente alla configurazione scelta dal BE.

Nel `tt-recommended-list-options.json` sono stati definiti per ogni tipo di carosello e per ogni variante:

- il titolo del widget;
- se mostrare o meno il footer del widget;
- la label da inserire nell'eventuale footer;
- il link a cui fare il redirect se si clicca sulla label dell'eventuale footer.

Per evitare di affaticare il caricamento dell'homepage si è deciso di rendere il widget asincrono: i suggerimenti arrivano dal Widget BE, ma si ottengono a seguito di una chiamata API effettuata nel lifecycle hooks `created`. Se la lista ritornata è vuota, il widget si chiude con un animazione resa possibile dal tag `<transition>` e sparisce, altrimenti viene passata al `partial` che si occuperà di rappresentarla.

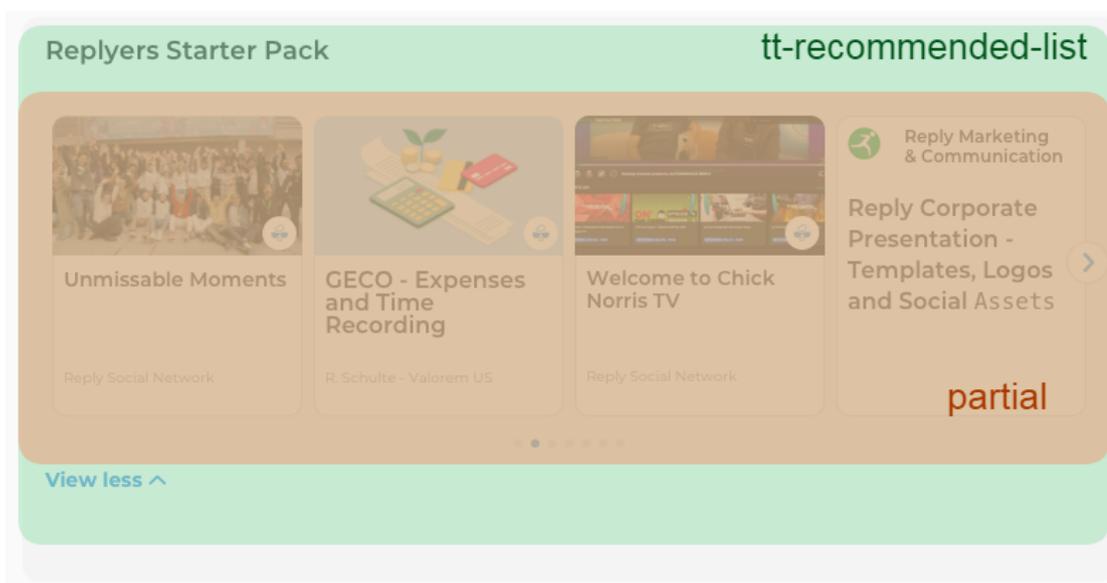


Figura 5.4: Composizione widget FE

5.8.2 Backend

UpdateGroupActivityCoefficientJob & UpdateUserActivityCoefficientJob

Come definito nel capitolo di progettazione, sia il carosello dei gruppi che quello delle persone mostrano dei risultati ordinati in base a quanto questi siano attivi. Trattandosi di due widget da inserire nella homepage di TamTamy, bisognava fare in modo che le query non fossero particolarmente complesse, in modo da restituire le liste con l'ordinamento richiesto, senza però sovraccaricare troppo la pagina. Proprio per questo motivo è stata aggiunta una nuova colonna "activity_coefficient" sia nella kc_group (tabella contenente tutti i gruppi di TamTamy) che nella kc_user (tabella contenente tutti gli utenti di TamTamy) in maniera tale da poter salvare, per ogni gruppo e utente, il rispettivo "coefficiente di attività". Per fare ciò è bastato aggiungere una nuova variabile di tipo long sia nella classe KcUser che nella classe KcGroup. Hibernate, poi, si è occupato di aggiornare le effettive tabelle del DB. Il valore di questi due nuovi coefficienti, invece, viene aggiornato ogni domenica da due job (processi schedulati grazie alla libreria Quartz): UpdateGroupActivityCoefficientJob e UpdateUserActivityCoefficientJob. In queste due classi è stato quindi implementato il metodo executeInternal responsabile di:

1. Calcolare il nuovo valore del coefficiente di attività. Per fare ciò in un caso viene chiamato il servizio "computeGroupActivityCoefficient()" mentre nell'altro il "computeUserActivityCoefficient()". Entrambi i metodi eseguono prima

una query nel DB per contare, rispettivamente per ogni gruppo e per ogni utente, il numero di contenuti e commenti creati dalla domenica precedente. Successivamente calcolano il nuovo coefficiente di attività, assegnando un punto per ogni contenuto e mezzo punto per ogni commento;

2. Aggiornare tutti gli “activity_coefficient” nel DB.

RecommendedListWidget

Il widget recommended BE serve solo a portare informazione al widget FE circa il tipo di carosello (People, Group, Content o Event), il numero di risultati da mostrare e le varianti tra cui scegliere. Per il carosello dei contenuti suggeriti, per esempio, le varianti proposte variano in base a dove si trova il widget:

- in homepage le varianti proposte sono: contenuti onboarding, contenuti del gruppo CoPs, contenuti creati dal gruppo della company dell’utente;
- nel profilo utente le varianti proposte sono: contenuti onboarding e contenuti creati dal proprietario del profilo.

RecommendedListController - GET

La RecommendedListController, in base al chiamante, invoca il servizio opportuno, specifico per ogni tipo di suggerimento (KcContent, KcGroup, KcUser o KcEvent). Di base tutti e quattro i servizi hanno lo stesso comportamento:

1. si dichiarano le variabili e si entra in un ciclo while da cui si esce solo quando la lista dei risultati non è vuota o l’array “variants” non contiene più varianti;
2. utilizzando un randomizer si estrae dall’array “variants” una variante. Quest’array è una copia dei valori contenuti dalla classe enum “VariantStatus” che, per ogni tipo di carosello, specifica le varianti implementate: per il carosello dei gruppi, per esempio, il metodo groupValues() ritorna solo i valori “VARIANT_ONE” e “VARIANT_TWO”. L’array risultante viene ulteriormente filtrato dai valori passati dal widget BE. Per il carosello dei contenuti suggeriti, per esempio, le varianti implementate sono 4, che diventano 3 per l’homepage (VARIANT_ONE, VARIANT_TWO, VARIANT_THREE) e 2 per il profilo dell’utente (VARIANT_THREE e VARIANT_FOUR);
3. data la variante, si controlla prima se è già presente una lista in cache:
 - (a) se in cache non v’è alcuna lista, si entra nello switch case specifico della variante estratta e si esegue la query opportuna. Se la query dà come risultato una lista vuota, si rimuove la variante dall’array “variants” e si

esegue un nuovo ciclo while e così via. Se la query ritorna, invece, una lista di risultati, si salva tutto in cache e si esce dal ciclo while;

(b) se in cache è già presente una lista, la si scarica e si esce dal ciclo while;

4. una volta fuori dal ciclo while, utilizzando il costruttore opportuno (uno per il caso della lista vuota, uno per il caso della lista valorizzata) si dichiara e si ritorna il bean di output inserendovi la lista di risultati, la variante scelta e, nel caso della variante “company” del widget dei contenuti, il link del gruppo della company.

5.9 Highlight

Per configurare i vari highlight (da inserire nel JSON caricato nella pagina di Carousel Management) sono state create le seguenti variabili:

- videoUrl: accetta l'url del video da mostrare;
- animation: accetta l'url dell'animazione lottie da usare come sfondo della card;
- backgroundImage: accetta l'url dell'immagine (o della gif) da usare come sfondo della card. Se assente o se il valore è nullo (e l'highlight non è né un video, né un'animazione), come immagine di sfondo ne verrà scelta randomicamente una tra quelle passate dal BE;
- wide: se a true, la card si estenderà su tutto lo spazio disponibile;
- location: se presente e se valorizzata, farà apparire un ribbon contenente la location;
- filteredCountry: se valorizzato permette di mostrare un certo highlight solo agli utenti della country indicata;
- filteredCity: se valorizzato permette di mostrare un certo highlight solo agli utenti della city indicata. Se sia filteredCity che filteredCountry sono valorizzati, si filtrerà per città;
- link: accetta il link della pagina a cui reindirizzare l'utente;
- videoDescription: Accetta la descrizione del video.

5.9.1 Frontend

tt-header-highlight

Il widget degli highlight, di fatto, è un carosello che per ogni pagina può contenere una, due o tre “highlight-card”. A decidere quante card mostrare per ogni pagine è la computed “computedHighlight()”, che prende in ingresso il JSON e ritorna un array di array. In base al numero di oggetti contenuti dagli array interni (minimo uno, massimo tre) verrà quindi scelto quante card renderizzare per pagina. Il criterio con cui si decide quanti elementi inserire negli array interni è basato sulle prop “wide” e “videoDescription” (se la prima è settata a “true” o se la seconda è valorizzata, assieme alla variabile videoUrl, nell’array ci deve essere solo l’oggetto in questione) e sulla modalità di visualizzazione (in “modalità cellulare” si visualizzerà una sola card per pagina). Prima di inserire l’highlight nell’array si controlla anche se è stato imposto in filtraggio per country o per city.

5.9.2 Backend

HeaderHighlightWidget

Tramite il metodo “buildOptions()” il widget BE preleva dal DB e ritorna al widget FE le tre seguenti informazioni:

1. il JSON caricato nella prima text-area della pagina di Carousel Management;
2. la city dell’utente in sessione, utile per filtrare gli highlight se la variabile “filteredCity” è valorizzata;
3. le immagini di default di cui a FE se ne sceglierà una randomicamente qualora “backgroundImage”, “animation” e “videoUrl” non dovessero essere valorizzate.

5.10 Funzionalità invita

5.10.1 Backend

InviteUsersController - POST

Il servizio checkAndCreate() invocato dalla POST della classe “InviteUsersController” ha il compito di produrre la notifica che verrà poi inviata all’utente. Prima però è necessario effettuare gli opportuni check, tramite il metodo validate() della classe InviteUsersCreateValidationStrategy. Si controlla quindi:

1. che l’evento esista;

2. che il title dell'utente invitante sia superiore a quello di Consultant. In caso negativo, sarà necessario un ulteriore controllo per verificare che l'utente in questione sia effettivamente iscritto all'evento;
3. che l'utente invitato esista;
4. che l'utente invitato possa iscriversi all'evento. Infatti, per poter anche solo visualizzare un evento è necessario far parte del gruppo di visibilità con cui è mappato l'evento in questione nella Kc_Group_Meta;
5. che l'utente invitato non sia già iscritto;
6. che l'utente invitato non sia stato già contattato, per quell'evento, dall'utente invitante.

Qualora una di queste condizioni non dovesse essere rispettata, verrebbe prodotta una `FieldsValidationException` mappata a FE con l'opportuno messaggio di errore da mostrare all'utente invitante. Se invece si superano i controlli, tramite il servizio `save()` della classe "NoticeService" si memorizza in DB un nuovo oggetto di tipo `KcNotice` contenente tutte le informazioni necessarie per produrre la notifica.

Capitolo 6

Risultati ottenuti e Conclusioni

In questo capitolo conclusivo verranno analizzati i risultati ottenuti, effettuando una valutazione requisito per requisito. Si procederà, quindi, con una panoramica di quanto fatto e infine si definiranno i prossimi step.

6.1 Risultati ottenuti

Prima di procedere con l'implementazione, il flusso pianificato è stato mostrato al "Board", l'organo di governo di Reply nonché committente del progetto. I feedback ricevuti sono stati positivi e pertanto si è potuto procedere con la fase di sviluppo. Stessa sorte non è toccata al progetto parallelo, il pre-onboarding su Meply, per cui sono state richieste diverse modifiche a diversi processi.

È possibile, quindi, effettuare la seguente analisi requisito per requisito:

R1, "Unire i puntini" Gli strumenti di TamTamy vengono adeguatamente presentati dal Widget "Onboarding Task". Allo stesso modo, anche i nuovi widget dell' homepage rendono possibile la valorizzazione di eventi, gruppi, pagine e utenze TamTamy. A ciò si aggiungono quei widget presenti nella pagina del profilo dell'utente, che si legano fortemente a uno strumento in particolare: il GECO Donut Stats con GECO, il widget delle User Project Cover con le Project Cover, l'Event & Learn Stats con il portale Learn.Share.Remix. Il requisito, pertanto è da considerarsi **soddisfatto**;

R2, "Utilizzare la gamification per aumentare l'engagement" Di tutti i Core Drive, solo 2 non sono stati presi in considerazione nella fase di progettazione:

1. Core Drive #1 - Chiamata all'azione, più adatto per la parte narrativa gestita dal team RSN;
2. Core Drive #3 - Creatività, che non ha trovato nessuna possibile applicazione in alcun widget.

Tutti gli altri Core Drive sono stati utilizzati, soprattutto nella pagina del profilo del replayer, con lo scopo di aumentare il livello di coinvolgimento. Poiché nel capitolo in cui sono stati definiti i requisiti era già stato specificato che non dovevano essere necessariamente utilizzati tutti i Core Drive ma soltanto quelli che risultavano essere più utili date le caratteristiche dei widget, anche questo requisito è stato **soddisfatto**. Resta comunque **da verificare l'efficacia** della soluzione implementata, sia con i tester che con l'effettivo utente finale;

R3, “Presentare TamTamy al nuovo arrivato” A differenza del vecchio processo di onboarding, ora TamTamy viene introdotto adeguatamente durante la fase di registrazione. Le 3 slide, infatti, forniscono le informazioni di contestualizzazione necessarie all'utente per poter iniziare a capire a cosa il social possa servire. Oltre ciò, un contributo da non sottovalutare è dato dal redesign grafico della firstAccess, la prima interfaccia di TamTamy con cui il nuovo arrivato viene in contatto che causa, inevitabilmente, il primissimo giudizio dato al social. Sicuramente, un'interfaccia più fresca, aggiornata, al passo con i tempi evita che l'utente possa pensare già da subito di avere a che fare con uno strumento obsoleto. Il requisito è stato quindi **pienamente soddisfatto**;

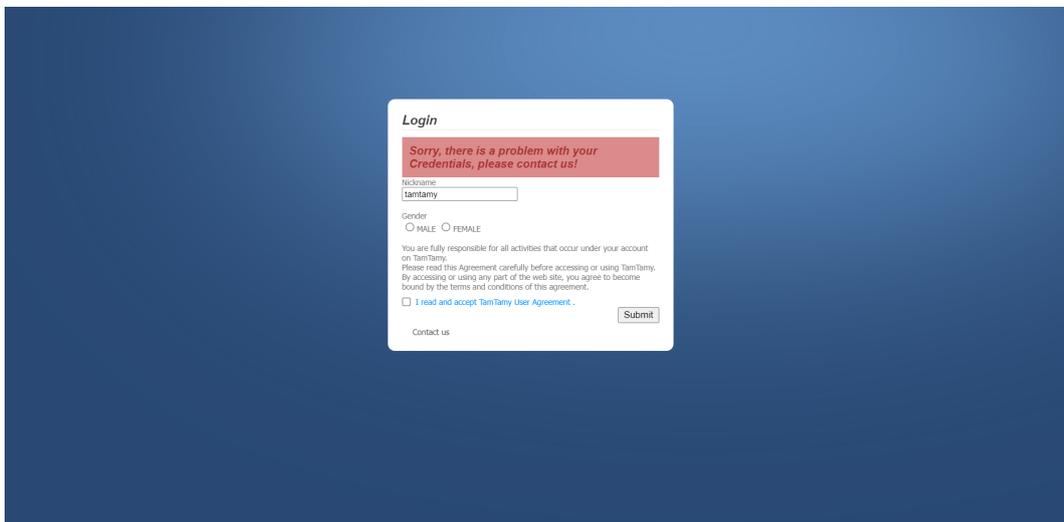


Figura 6.1: Vecchia pagina di First Access

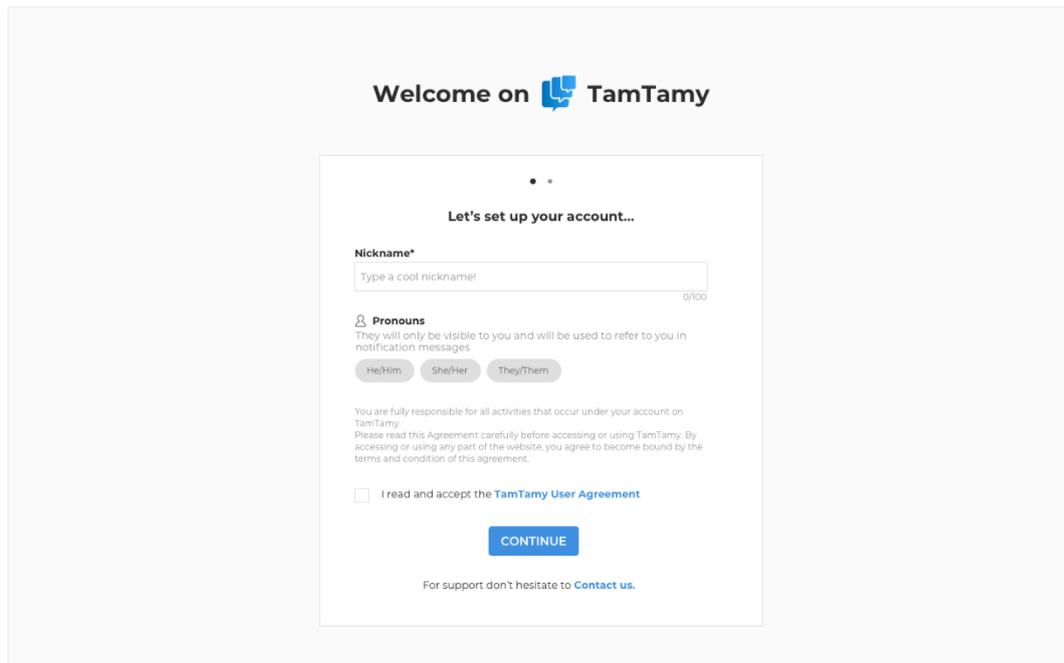


Figura 6.2: Nuova pagina di First Access

- R4, “TamTamy deve avere un ruolo da protagonista”** Il journey è ambientato per 3/4 su TamTamy: solo la fase di pre-onboarding (assieme alle dashboard) è su Meply. TamTamy, quindi, ha certamente un ruolo da protagonista nel nuovo processo di onboarding. Il requisito è stato **soddisfatto**;
- R5, “Dare consigli in base all’utente e al suo profilo aziendale”** I widget responsabili dei consigli dati agli user sono tutti quelli nuovi implementati nell’homepage di TamTamy. Di questi non tutti si specializzano in base alle caratteristiche dell’utente, ma comunque è presente una buona componente di personalizzazione. Il requisito è stato quindi **soddisfatto in buona parte**;
- R6, “Assegnare task in base al profilo aziendale del replyer”** Sia il widget “Onboarding Task” che il “Randomic Task” (che verrà sviluppato prossimamente) propongono al nuovo arrivato attività che possono variare in base al profilo aziendale individuato nel replyer. Da questo processo però manca il profilo dei Partner poichè il Board ha deciso che non fosse prioritario lavorarci. Requisito **soddisfatto in buona parte**;
- R7, “Estendere la durata dell’onboarding”** La durata dell’onboarding, come descritto nel paragrafo del journey del capitolo 3, è stato esteso da 1-2 mesi a 2 anni. Le informazioni ora vengono dilazionate nel tempo, riproposte ad ogni

accesso e divise in base alla fase del Journey in corso. I contenuti di onboarding consigliati dal widget dei suggerimenti nel profilo dell'utente, per esempio, sono diversi: durante la fase di "Welcome on board" si danno informazioni di maggiore priorità, mentre nella fase di "Onboarding Long Term" si danno informazioni più approfondite. Il requisito è quindi **pienamente soddisfatto**;

R8, "Aiutare Manager e Partner nel loro ruolo nell'onboarding" La richiesta dietro questo obiettivo era di fornire delle dashboard che mostrassero dati e informazioni di vario tipo (stato di avanzamento dell'assunzione, partecipazione a eventi ecc) utili a partner e manager per monitorare il processo di onboarding. Come definito nel Journey le dashboard erano da implementare su Meply, mentre TamTamy avrebbe dovuto solamente fornire i dati prodotti dalle fasi "Welcome on board", "Onboarding Long Term" e "Reply Employees Experience". Tuttavia, poichè il progetto presentato per Meply non è stato approvato, su TamTamy non è stata nè implementata nè progettata alcuna soluzione per il requisito, che pertanto è da considerarsi **non soddisfatto**.

6.2 Conclusioni

L'onboarding è il biglietto da visita di un'azienda, oltre che un fattore estremamente determinante nella scelta, da parte di un dipendente, se rimanere o meno. Per creare un nuovo flusso di onboarding per Reply su TamTamy si è proceduto per step, il primo dei quali è stata la fase di studio in cui:

- è stato esplorato TamTamy, sia come utente che come sviluppatore;
- sono state effettuate ricerche in letteratura per trovare possibili reference;
- si è approfondito il concetto di gamification.

Parallelamente sono state condotte delle interviste, live e tramite Google form, in modo da ottenere le informazioni propedeutiche per poter costruire le personas. Sono stati quindi definiti i requisiti ed è stato delineato il journey. Completata la fase di analisi è iniziata, dunque, quella di progettazione in cui è stato pianificato tutto il nuovo processo di onboarding. Quest'ultimo parte dalla primissima pagina di TamTamy con cui l'utente viene a contatto ed arriva all'homepage, passando per i vari strumenti accessibili dal social e il profilo dell'utente, fondamentale per rappresentare la crescita del replyer nel tempo. I "mattoncini" del nuovo processo di inserimento sono stati i widget, ognuno con le proprie finalità e funzionalità, implementati, nella fase di sviluppo, sfruttando le già esistenti architetture del social: Vue per il frontend, Struts, Spring e Hibernate per il backend. Infine, sono state effettuate delle valutazioni, requisito per requisito, al fine di verificare se e quanto gli obiettivi siano stati raggiunti.

Il primissimo futuro step è quello di portare l'intero flusso su un ambiente di test e lasciarlo testare agli SWAT, i tester di Reply. Seguirà, poi, una fase di correzione di eventuali problemi emersi e segnalati alla fine della quale si potrà finalmente procedere con il rilascio in ambiente di produzione. Resta anche in TODO l'integrazione con Meply per soddisfare l'unico requisito non soddisfatto: aiutare Manager e Partner nel loro ruolo nell'onboarding.

Bibliografia

- [1] T. N. Bauer e B. Erdogan. «Organizational socialization: The effective onboarding of new employees.» In: *American Psychological Association* (2011) (cit. a p. 1).
- [2] K. Ferrazzi. *Technology Can Save Onboarding from Itself*. Mar. 2015. URL: <https://hbr.org/2015/03/technology-can-save-onboarding-from-itself> (cit. a p. 1).
- [3] T. Petrucelli. «Winning the “cat-and-mouse game” of retaining millennial talent.» In: *Strategic HR Review* (2017) (cit. a p. 1).
- [4] M. Laurano. «The true cost of a bad hire». In: *Brandon Hall Group* (2015) (cit. a p. 1).
- [5] J. Williams e S. Rosenbaum. *Learning Paths: Increase Profits by Reducing Time it takes Employee to get up-to-speed*. Pfeiffer, ago. 2004 (cit. a p. 2).
- [6] R. K. Attri. *Accelerated Proficiency for Accelerated Times: A Review of Key Concepts and Methods to Speed Up Performance*. Speed To Proficiency Research: S2Pro©, mag. 2020 (cit. a p. 2).
- [7] T. N. Bauer. «Onboarding New Employees: Maximizing Success». In: *Onboarding White Paper Series* (2013) (cit. a p. 2).
- [8] Leena AI team. *2022 - The State of Employee Onboarding in the US*. URL: <https://leena.ai/state-of-employee-onboarding-united-states> (cit. a p. 6).
- [9] Vanhishikha Bhargava. *The role of AI in employee onboarding and how the future looks*. URL: <https://leena.ai/blog/ai-employee-onboarding/> (cit. a p. 7).
- [10] Thomas Stone. *HOW ACCENTURE USES VIRTUAL REALITY TO ONBOARD EMPLOYEES AND MORE*. URL: <https://www.i4cp.com/productivity-blog/how-accenture-uses-virtual-reality-to-onboard-employees-and-more> (cit. a p. 7).
- [11] virsabi team. *Onboarding & training with Virtual Reality*. URL: <https://virsabi.com/onboarding-and-training-with-vr/> (cit. a p. 8).

-
- [12] Mary K. Pratt. *6 potential uses for VR during onboarding*. URL: <https://www.techtarget.com/searchhrsoftware/feature/6-potential-uses-for-VR-during-onboarding> (cit. a p. 8).
- [13] Yu-Kai Chou. *The 8 Core Drives of Gamification (#2): Development and Accomplishment*. URL: <https://yukaichou.com/gamification-study/8-core-drives-gamification-2-development-accomplishment/> (cit. alle pp. 8, 10).
- [14] Yu-Kai Chou. *The 8 Core Drives of Gamification (#1): Epic Meaning & Calling*. URL: <https://yukaichou.com/gamification-study/8-core-drives-gamification-1-epic-meaning-calling/> (cit. a p. 10).
- [15] Yu-Kai Chou. *The 8 Core Drives of Gamification (#3): Empowerment of Creativity & Feedback*. URL: <https://yukaichou.com/gamification-study/8-core-drives-gamification-3-empowerment-creativity-feedback/> (cit. a p. 10).
- [16] Yu-Kai Chou. *The 8 Core Drives of Gamification (#4): Ownership and Possession*. URL: <https://yukaichou.com/gamification-study/8-core-drives-gamification-4-ownership-possession/> (cit. a p. 11).
- [17] Michael I. Norton. *The IKEA effect: When labor leads to love*. URL: <https://www.sciencedirect.com/science/article/abs/pii/S1057740811000829> (cit. a p. 11).
- [18] Yu-Kai Chou. *The 8 Core Drives of Gamification (#5): Social Influence & Relatedness*. URL: <https://yukaichou.com/gamification-study/8-core-drives-of-gamification-5-social-influence-relatedness/> (cit. alle pp. 11, 12).
- [19] Robert B. Cialdini. *Le armi della persuasione*. Giunti, 2013 (cit. a p. 12).
- [20] Yu-Kai Chou. *The 8 Core Drives of Gamification (#6): Scarcity & Impatience*. URL: <https://yukaichou.com/gamification-study/8-core-drives-gamification-6-scarcity-impatience/> (cit. a p. 13).
- [21] Jesse Schell. *The Art of Game Design: A Book of Lenses*. A K Peters/CRC Press, 2008 (cit. a p. 13).
- [22] Yu-Kai Chou. *The 8 Core Drives of Gamification (#7): Unpredictability & Curiosity*. URL: <https://yukaichou.com/gamification-study/the-8-core-drives-of-gamification-7-unpredictability/> (cit. a p. 13).
- [23] Yu-Kai Chou. *The 8 Core Drives of Gamification (#8): Loss & Avoidance*. URL: <https://yukaichou.com/gamification-study/8-loss-and-avoidance/> (cit. alle pp. 13, 14).
- [24] Vue.js team. *Instance Lifecycle Hooks*. URL: <https://v2.vuejs.org/v2/guide/instance.html#Instance-Lifecycle-Hooks> (cit. a p. 15).

BIBLIOGRAFIA

- [25] Vue.js team. *Options / Data*. URL: <https://v2.vuejs.org/v2/api/#Options-Data> (cit. a p. 15).