

Università Politecnica delle Marche

Facoltà di Ingegneria

Dipartimento di Ingegneria dell'Informazione

Corso di Laurea in Ingegneria Informatica e dell'Automazione



Tesi di Laurea

Applicazione di tecniche per migliorare la User Experience
in un'app relativa all'Università Politecnica delle Marche

Application of techniques to improve User Experience in an
app relative to the Polytechnic University of Marche

Relatore

Prof. Domenico Ursino

Candidata

Chiara Mazzucchelli

Anno Accademico 2018-2019

“L'unico modo di fare un ottimo lavoro è amare quello che fai.”

Indice

Introduzione	3
1 Lo User Experience Design	7
1.1 Un po' di storia	7
1.1.1 Nascita dell'UX	7
1.1.2 Primi casi di user experience	9
1.1.3 Applicazioni tecnologiche	10
1.2 Progettazione	11
1.2.1 L'importanza delle persone	11
1.2.2 Fasi di progetto	14
2 Tecniche di User Experience Applicate	21
2.1 Descrizione del progetto	21
2.2 Tecniche di UX utilizzate	22
2.2.1 Discovering	23
2.2.2 Co-designing	26
3 Analisi dei Requisiti e Progettazione	29
3.1 Analisi dei Requisiti	29
3.1.1 Risultati del questionario	29
3.1.2 Requisiti	32
3.2 Progettazione	32
3.2.1 Card Sorting	32
3.2.2 Wireframe	35
3.2.3 Diagrammi UML	38
4 Implementazione della Componente Front-end e Manuale Utente	41
4.1 Implementazione della componente Java	41
4.1.1 Implementazione della classe <code>MainActivity.java</code>	41
4.1.2 Implementazione dei fragment	42
4.2 Implementazione della componente XML	43
4.2.1 Implementazione del file <code>activity_main.xml</code>	44
4.2.2 Implementazione dei fragment	44
4.3 Manuale Utente	60

VIII Indice

4.3.1	Home Page	61
4.3.2	Schermata “Materia”	61
4.3.3	Schermata “Mazzo”	62
4.3.4	Schermata “Eventi”	63
4.3.5	Schermata “Note”	64
4.3.6	Schermata “Profilo”	64
5	Discussione, conclusioni e sfide future	67
5.1	Discussione	67
5.1.1	Ruolo della UXD nella realizzazione delle app	67
5.1.2	Lezioni apprese	68
5.2	Conclusioni e uno sguardo al futuro	69
	Ringraziamenti	71
	Riferimenti bibliografici	73

Elenco delle figure

1.1	Discipline comprese nella User Experience	8
1.2	Il grafico mostra l'aumento degli esperti di UX e una proiezione del loro numero negli anni a venire. Si sottolinea che anche i dati relativi agli anni passati sono delle stime, in quanto non esiste una definizione specifica e chiara di "UX specialist"; le ricerche condotte dal <i>Nielsen Norman Group</i> dichiarano, infatti, che su 1045 esperti del settore trovati, questi hanno 210 diversi titoli lavorativi.	12
1.3	L'immagine mostra le sfaccettature della User Experience identificate da Peter Morville.	13
1.4	Le sei macrofasi principali dello sviluppo di un prodotto digitale.	15
2.1	Curva dell'oblio di Hermann Ebbinghaus.	22
2.2	Matrice SWOT.	24
2.3	Screenshot del questionario	27
3.1	Aggettivi che descrivono " <i>Assistente di Studio</i> "	30
3.2	Valutazione dei servizi offerti	30
3.3	Valutazione generale dell'applicazione	30
3.4	Valutazione della facilità di utilizzo	31
3.5	Valutazione di alcuni aspetti dell'applicazione	31
3.6	Card Sorting creato tramite <i>Optimal Workshop</i>	33
3.7	Mappa dell'applicazione	35
3.8	Wireframe che mostrano la schermata delle flashcard, quella della materia selezionata e quella del singolo mazzo	36
3.9	Wireframe che mostrano la schermata degli eventi, quella degli eventi nel calendario e quella per l'aggiunta di un nuovo evento	37
3.10	Wireframe che mostrano la schermata delle note e quella del profilo.	38
3.11	Use Case Diagram	39
3.12	Activity Diagram	39
4.1	Screenshot della home page, ovvero la schermata "Flashcard"	62
4.2	Screenshot della schermata "Materia"	62
4.3	Screenshot della schermata "Mazzo"	63

X **Elenco delle figure**

4.4	Screenshot della schermata “Eventi”	64
4.5	Screenshot della schermata “Note”	65
4.6	Screenshot della schermata “Profilo”	65

Elenco dei listati

4.1	Implementazione della classe <code>MainActivity</code>	41
4.2	Implementazione del fragment <code>EventiFragment</code>	42
4.3	Implementazione dell'adapter <code>PageAdapter</code>	43
4.4	File <code>activity_main.xml</code>	44
4.5	Implementazione del file XML di <code>FragmentEventi</code>	45
4.6	Implementazione del file XML di <code>FragmentFlashcard</code>	46
4.7	Implementazione del file XML di <code>FragmentMateria</code>	51
4.8	Implementazione del file XML di <code>FragmentMazzo</code>	55
4.9	Implementazione del file XML di <code>FragmentNote</code>	57
4.10	Implementazione del file XML di <code>FragmentProfilo</code>	59

Introduzione

In un mondo sempre più tecnologico e pervaso da prodotti digitali che tentano di dare un contributo positivo ad ogni aspetto della vita quotidiana dell'uomo, la chiave per il successo non è più la funzionalità del proprio progetto, ma la sua usabilità e la sua capacità di offrire all'utente qualcosa che lo faccia sentire soddisfatto. Tutto ciò significa che l'applicazione o il sito web deve, ovviamente, garantire il funzionamento dei servizi che offre e far sì che l'utente abbia una risposta a ciò che cerca, ma lo deve fare assicurando un prodotto facile da utilizzare, intuitivo e ammaliante. Jakob Nielsen, informatico e imprenditore danese, fa notare che:

Sul Web, l'usabilità è una condizione necessaria per la sopravvivenza. Se un sito web è difficile da usare, le persone se ne vanno. Se la home page non riesce a indicare chiaramente cosa offre un'azienda e cosa possono fare gli utenti sul sito, le persone se ne vanno. Se gli utenti si perdono su un sito web, se ne vanno. Se le informazioni di un sito web sono difficili da leggere o non rispondono alle domande chiave degli utenti, se ne vanno. Noti uno schema in tutto questo?

È proprio in questo contesto che entra in gioco e si diffonde il concetto di *User Experience*, ovvero la tendenza a progettare un prodotto ponendo al centro dell'attenzione l'esperienza dell'utente. Tutto il processo di progettazione si sposta, dunque, sul pubblico, cercando di comprendere cosa l'utente finale si aspetta dal prodotto, così da poter creare qualcosa fatto su misura e che combaci con i suoi bisogni e i suoi desideri.

Lo *User Experience Designer* è la figura chiave di tutto questo processo, colui che deve essere in grado di coniugare l'aspetto ingegneristico della progettazione con quello empatico e psicologico dell'ascoltare e comprendere gli altri, con l'obiettivo di realizzare qualcosa che funzioni e che susciti emozioni positive. Per raggiungere il suo scopo, lo *User Experience Designer* ha degli strumenti a sua disposizione e delle tecniche che lo aiutano nel suo intento, ma la regola d'oro rimane una, ovvero quella di dare spazio agli utenti finali, che si "sporcheranno le mani" in prima persona per progettare il prodotto che utilizzeranno.

Il coinvolgimento dei futuri utenti è, dunque, essenziale e avviene anche in più momenti dell'iter progettuale, a partire dalla stesura dei requisiti, passando per la costruzione dell'architettura del prodotto e giungendo, infine, alla fase di testing, che può e deve avvenire più volte, proprio per assicurarsi che non si perda mai di

vista l'obiettivo finale. Le tecniche per coinvolgere gli utenti sono molteplici, ma vanno tenuti in mente alcuni accorgimenti. Il primo è che è sempre meglio stare a contatto con le persone per comprenderle in maniera più profonda e per studiare il modo con cui si rapportano alla tecnologia; il secondo è fare in modo che i momenti in cui si richiede la partecipazione delle persone siano costruttivi per il progettista, ma divertenti per i partecipanti, proponendo attività che all'apparenza possono sembrare dei giochi, ma che, in realtà, sono in grado di fornire le informazioni necessarie.

Alla luce di quanto scoperto entrando nel mondo dello *User Experience Design*, ci siamo resi conto che esistono molti prodotti digitali ben funzionanti e potenzialmente utili, ma che non vengono apprezzati a pieno dagli utenti in quando non sono intuitivi, le loro modalità di utilizzo non sono molto chiare, o, semplicemente, non hanno interfacce grafiche in grado di suscitare gradimento negli utilizzatori. Fa parte di questi prodotti che non hanno avuto il giusto apprezzamento l'applicazione creata dal collega Kevin Cela, chiamata *Assistente di Studio*, uno strumento in grado di offrire un supporto alla vita dello studente universitario, sostenendolo anche nella fase dello studio, tramite la metodologia di ripasso delle flashcard.

L'obiettivo di questa tesi è diventato, dunque, quello di riprogettare tale applicazione, basandosi sui concetti appresi dallo studio dello *User Experience Design*, per dare nuova luce e nuove opportunità a questo prodotto.

Assistente di Studio è un'applicazione che prevede tre funzionalità principali. La prima è il calendario, ovvero una sezione in cui è possibile segnare eventi importanti, come esami, lezioni e ricevimenti. La seconda è la sezione che permette il salvataggio delle note, quindi di appunti importanti che è necessario ricordare. La terza è la sezione delle flashcard, probabilmente la funzionalità più caratteristica dell'applicazione. In tale sezione è possibile creare dei mazzi di carte, ciascuna contenente un concetto con la propria definizione o la propria spiegazione, per poi studiarli: l'app mostra il concetto e lo studente deve spiegarlo, allenando la sua memoria. Per quanto utile e innovativo, questo prodotto non viene utilizzato al meglio, per cui si è deciso di riprogettarlo, per cercare di migliorare l'esperienza utente.

L'applicazione *Assistente di Studio* si è trovata, quindi, al centro di una campagna di *User Experience*, volta a comprendere cosa di questo prodotto non ha soddisfatto le aspettative, cosa poteva essere migliorato e cosa gli utenti desideravano ritrovare nell'app.

L'applicazione è stata, in primo luogo, oggetto di un'analisi *SWOT*, condotta dal progettista per comprenderne i punti di forza e di debolezza; successivamente, l'app è stata sottoposta a un gruppo di utenti del target selezionato, che l'hanno testata e hanno compilato un questionario per indicare le loro impressioni. Il risultato è stato molto soddisfacente, in quanto è vero che è emerso che l'indice di gradimento non era elevato, ma sono state avanzate molte proposte, che si sono rivelate estremamente utili. Le risposte del questionario sono state analizzate in dettaglio e, dopo aver stilato l'elenco dei requisiti da soddisfare, si è deciso di coinvolgere nuovamente gli utilizzatori finali, tramite la tecnica del *card sorting*. Quest'ultima è una delle tecniche caratteristiche dello *User Experience Design*, che permette di progettare l'architettura di un prodotto digitale insieme agli utenti stessi, basandosi sui loro modelli mentali.

È emerso che la disposizione di alcuni elementi dell'applicazione non era fun-

zionale, e ciò influenzava negativamente la facilità di utilizzo del prodotto stesso. Inoltre, gli utenti hanno riscontrato delle pecche dal punto di vista grafico e intuitivo ma hanno, anche, proposto nuove funzionalità da integrare, per poter rendere *Assistente di Studio* uno strumento realmente utile. Grazie all'utilizzo di queste tecniche è stato, perciò, possibile riprogettare l'app e realizzare le nuove interfacce grafiche, in grado di attirare nuovi utenti e renderli soddisfatti.

In questa tesi racconteremo tutta questa esperienza, iniziando con il comprendere che cos'è lo *User Experience Design*, per poi applicare ciò che si è appreso, fino ad ottenere un prodotto riprogettato e in grado di offrire una piacevole esperienza utente.

La tesi è strutturata come di seguito specificato:

- Nel primo capitolo verrà proposta una panoramica sullo *User Experience Design*. Dopo averne raccontato la nascita e l'evoluzione, si descriveranno le macrofasi principali che vanno seguite per poter attuare una buona progettazione UX, ponendo particolare attenzione nel descrivere l'importanza delle persone nella progressione del progetto.
- Nel secondo capitolo, verrà illustrato il progetto che si intende sviluppare e si descriveranno in maniera dettagliata le tecniche che verranno applicate per il conseguimento dell'obiettivo imposto.
- Nel terzo capitolo verranno stabiliti i requisiti che l'applicazione deve rispettare, per poi definirne l'architettura; infine si procederà con la sua progettazione; in particolare, verranno descritti nel dettaglio la struttura finale del prodotto e le sue funzionalità, tramite mappe concettuali, wireframe e diagrammi UML.
- Nel quarto capitolo verrà illustrata l'implementazione della componente front-end dell'app, ponendo attenzione ai file XML che descrivono la struttura delle interfacce grafiche; si concluderà, poi, con un manuale utente che, con il supporto di screenshot, illustrerà il funzionamento dell'applicazione.
- Nel quinto capitolo verranno tratte le conclusioni, verranno descritte le lezioni apprese tramite questo progetto e verranno delineati alcuni possibili sviluppi futuri.

Lo User Experience Design

Lo User Experience Design (UXD), in italiano “progettazione dell’esperienza utente”, è un approccio che ha l’obiettivo di valutare come le persone si rapportano a un prodotto, un oggetto o un processo. Lo UXD può essere definito come un processo iterativo che tiene conto del bagaglio esperienziale dell’utente: i criteri per giudicare la qualità di un prodotto non sono più soltanto l’efficienza e l’efficacia, ma viene dato sempre più peso e valore all’usabilità, alla facilità d’uso, ai sentimenti e alle emozioni che il prodotto suscita nell’utilizzatore. In altre parole, lo UXD è il processo atto a ideare prodotti, sia digitali che fisici, che siano utili, facili da usare e piacevoli da utilizzare. Si tratta di migliorare l’esperienza che le persone hanno durante l’interazione con il prodotto, per assicurarsi che trovino valore nel servizio che viene loro offerto.

1.1 Un po’ di storia

La User Experience è un campo relativamente nuovo, nato circa 100 anni fa. Tomer Sharon (Managing Director, Head of User Research & Metrics at Goldman Sachs) definisce la UXD come *“l’arte e la scienza di generare emozioni positive tra le persone che interagiscono con prodotti o servizi”*. Per giungere a questo obiettivo non è sufficiente essere esperti di design. Questo approccio si basa, infatti, su discipline diverse, dall’ergonomia, all’usabilità, al marketing, all’architettura e alla psicologia (Figura 1.1). A un primo impatto, qualsiasi definizione di User Experience può far pensare a un processo applicato in ambito tecnologico, riguardante, in particolare, applicazioni, siti web e software. È vero, ma non è sempre stato così.

1.1.1 Nascita dell’UX

Ripercorrendo all’indietro la storia, si può notare che dei primi accenni di User Experience lasciavano già delle tracce nel 4000 a.C., con il filosofo cinese Feng Shui. Ovviamente non si trattava di UX nel senso odierno del termine, ma deve essere intesa più come ricerca del benessere dell’uomo. Il filosofo studiava la disposizione spaziale degli oggetti in relazione al flusso di energia (chi). In poche parole, Feng

Questo termine indica una disciplina scientifica che analizza il rapporto tra l'uomo e l'ambiente di lavoro, per migliorare la produttività e insieme alleggerire lo sforzo psicofisico del lavoratore.

Con un salto temporale si arriva al 1430 circa con Leonardo da Vinci. Michael J. Gelb nel suo libro *"How to think like Leonardo da Vinci"* racconta di come il Duca di Milano abbia commissionato al grande artista la progettazione di una cucina per un grande banchetto. Egli accettò il lavoro e con grande estro costruì dei nastri trasportatori per portare i prodotti alimentari ai preparatori. Fu un disastro in quanto i nastri funzionavano in maniera troppo irregolare, rovinando il cibo. Nonostante ciò si può notare come, centinaia di anni prima della Rivoluzione Industriale, già Leonardo da Vinci aveva utilizzato la tecnologia per cercare di aiutare e alleggerire il lavoro manuale.

1.1.2 Primi casi di user experience

Le fondamenta dell'UX, com'è conosciuta oggi, risalgono agli inizi del 1900, quando l'uomo iniziò ad avere un ruolo centrale nei processi produttivi industriali. Parliamo del periodo post Rivoluzione Industriale: le aziende si stavano espandendo, la manodopera specializzata diminuiva e la tecnologia cercava di rimpiazzare queste mancanze utilizzando i primi macchinari. In questo scenario si distinguono due personaggi: Frederick Winslow Taylor e Henry Ford, entrambi all'avanguardia nel tentativo di rendere il lavoro umano più organizzato, efficiente e produttivo.

Taylor fu un ingegnere meccanico e uno dei primi consulenti aziendali, impegnato in ricerche per migliorare l'interazione tra operai e i loro strumenti e apparecchiature. Nel 1911 scrisse un libro, *"The Principles of Scientific Management"*, in cui afferma che una gestione più sistematica del sistema produttivo è la soluzione alle inefficienze. Nello stesso periodo anche Henry Ford cercava di massimizzare l'efficienza della produttività umana, studiando un metodo nuovo per far interagire i lavoratori con i macchinari complessi. Capì che aggiungere ai macchinari funzionalità sofisticate non era più sufficiente e comprese che era necessaria una figura nuova, il progettista, con l'obiettivo di creare interfacce intuitive per rendere comprensibili e di facile utilizzo anche i sistemi più complessi.

Questi studi e queste nuove tecniche per la produzione di massa ispirarono altre grandi menti del periodo. Ad esempio, lo psicologo e pioniere dell'ergonomia Paul M. Fitts riprogettò le cabine di pilotaggio degli aerei, migliorandole e rendendo i voli più sicuri. Man mano che la tecnologia si espandeva dalle fabbriche al settore militare, fino a coinvolgere i consumatori stessi, è diventato estremamente importante progettare interfacce utente facili e comprensibili a tutti, in modo da poter sfruttare al meglio le capacità umane.

Verso la metà del XX secolo l'efficienza industriale e l'ingegno umano contribuivano a migliorare le idee sviluppate da Taylor e Ford, ma miravano anche a creare ambienti produttivi più armoniosi. Il caso più eclatante fu l'industria Toyota, che sviluppò il famoso sistema di produzione incentrato sull'uomo. Non solo l'azienda cercava di rendere più efficiente la produzione, ma, cosa mai vista prima, il sistema di produzione si basava sul rispetto per le persone, considerando i lavoratori come fattore chiave nel processo di evoluzione e miglioramento. Uno dei principi fondamentali della Toyota era proprio il coinvolgimento dei lavoratori stessi nella

risoluzione dei problemi e nell'ottimizzazione dei processi di cui erano parte. Questo rappresenta un passo fondamentale nella storia della User Experience: per la prima volta si poneva attenzione all'uomo e alla sua interazione con la tecnologia.

Qualche anno più tardi, nel 1955, Henry Dreyfuss, designer industriale americano, scrisse il saggio *"Designing for People"*. Dreyfuss è considerato una figura chiave nella storia dell'UX in quanto ha ridisegnato molti dei prodotti di più largo consumo del tempo, migliorandone l'usabilità, e ha descritto questi metodi, utilizzati ancora oggi. Famosa è la frase estratta al suo libro, capace di riassumere il concetto base dell'UX:

Quando il punto di contatto tra il prodotto e le persone diventa un punto di attrito, allora il [designer] ha fallito. D'altra parte, se le persone sono rese più sicure, più a proprio agio, più desiderose di acquistare, più efficienti o semplicemente più felici, dal contatto con il prodotto, il designer è riuscito.

Gli ingegneri non sono state le uniche figure a contribuire allo sviluppo della User Experience. Un altro personaggio molto significativo, ma non così scontato, è stato Walt Disney, considerato da molti il primo UX designer. Durante la progettazione di quello che sarebbe poi diventato Disney World, Walt Disney descrisse il progetto come *"sempre nello stato di divenire, un luogo in cui le ultime tecnologie possono essere utilizzate per migliorare la vita delle persone"*. In ogni caso Walt Disney era ossessionato dal creare esperienze magiche e coinvolgenti per gli spettatori, e per farlo incitava i suoi *"Imagineers"* a conoscere il pubblico e a mettersi nei loro panni. Se questo non è User Experience, senza dubbio gli si avvicina parecchio.

1.1.3 Applicazioni tecnologiche

Nonostante tecniche di UX venissero applicate già da tempo in campo manifatturiero, la User Experience non è stata riconosciuta come un campo di studi a sé stante fino alla fine del '900, quando queste metodologie iniziarono ad essere applicate in campo informatico.

Il processo di diffusione dell'UX in campo tecnologico si è avviato negli anni '70, in parallelo con la distribuzione dei primi computer. In questo periodo psicologi e ingegneri lavoravano fianco a fianco per cercare di migliorare l'esperienza utente. I risultati più rilevanti vennero raggiunti da PARC, un centro di ricerca *Xerox* fondato per esplorare la tecnologia in relazione con i posti di lavoro. Proprio in questo centro, sotto la guida dell'ingegnere Bob Taylor, vennero sviluppati gli strumenti più avanzati e duraturi di interazione tra uomo e macchina, tra cui il mouse e la *Graphical User Interface* (GUI), utilizzati anche oggi. Il lavoro di PARC ha notevolmente e profondamente influenzato la prima interfaccia utente commercializzata: ci troviamo nel 1984 quando viene rilasciata la prima versione dell'*Apple Macintosh*. Da questo momento in poi, Apple si troverà sempre in prima linea per quanto riguarda la User Experience.

Nei primi anni '90 l'ingegnere elettronico e psicologo cognitivo Donald Norman si unisce allo staff Apple come User Experience Architect, diventando la prima persona a ricoprire questo ruolo in un'azienda. Il termine *User Experience* è stato coniato da Norman in persona per definire ciò che fino a quel momento era stato descritto come *Human Interface Research*. Dichiarò infatti che:

ho inventato il termine perché pensavo che Human Interface e Usabilità fossero troppo limitate: volevo ricoprire tutti gli aspetti dell'esperienza della persona con un sistema, incluso il design industriale, la grafica, l'interfaccia, l'interazione fisica, e il manuale.

Il suo libro *The Design of Everyday Things*, pubblicato nel 1988, in cui sostiene che il design si debba basare più sull'usabilità e sulle funzionalità che non sull'estetica, resta tutt'ora una pietra miliare per tutti gli esperti di User Experience.

Si può identificare un altro step importante per la crescita dell'UX nel 2007, con il rilascio sul mercato del primo *iPhone*. Prima del lancio, Steve Jobs promise che l'iPhone sarebbe stato lo smartphone più facile da utilizzare in commercio. Non solo mantenne la promessa fatta, ma cambiò completamente il panorama dei dispositivi mobili per sempre, catapultando l'Apple tra le più grandi aziende tecnologiche. Il suo grande successo è dovuto, in gran parte, alla User Experience che questi nuovi smartphone garantivano, di gran lunga superiore a quella offerta da tutti gli altri prodotti in circolazione.

Dal 2007 il campo della User Experience si è evoluto rapidamente, andando di pari passo con gli sviluppi e i traguardi raggiunti dalla tecnologia, e ci si aspetta che sarà così anche negli anni a venire. Ciò avrà come conseguenza l'aumento del numero degli esperti del settore (Figura 1.2), con sempre più skill e conoscenze richieste, per non perdere il passo del progresso tecnologico.

Le persone continueranno sempre a sperimentare prodotti e servizi nuovi e con una complessità che rispecchierà la tecnologia e l'organizzazione interna. Il compito dello UX designer sarà sempre quello di comprendere tecnologia, organizzazione e utenti finali, con l'obiettivo di progettare prodotti più intuitivi e coinvolgenti, utilizzabili da tutti e che aggiungano un valore alla vita delle persone.

1.2 Progettazione

Entriamo nel vivo della questione: qual è il processo che deve essere attuato quando si esegue un lavoro di User Experience? Purtroppo non esiste una risposta precisa a questa domanda: non c'è, infatti, un percorso meccanico da seguire. La progettazione all'insegna dell'UX è un processo creativo e per questo non è costituito da passaggi rigidi e da regole formali. Si possono solo individuare dei concetti fondamentali da rispettare e delle macrofasi indicative molto flessibili.

1.2.1 L'importanza delle persone

La normativa *ISO 9241-210* definisce la User Experience come

le percezioni e le reazioni di un utente che derivano dall'uso o dall'aspettativa d'uso di un prodotto, sistema o servizio.

È chiaro, dunque, che uno dei concetti fondamentali della progettazione UX è la centralità e l'importanza delle persone. Un buon prodotto digitale deve, innanzitutto, dichiarare in maniera schematica, ma estremamente chiara, il contenuto del sito o applicazione. Federico Badaloni, UX designer per *Il Gruppo L'Espresso*, sostiene che:

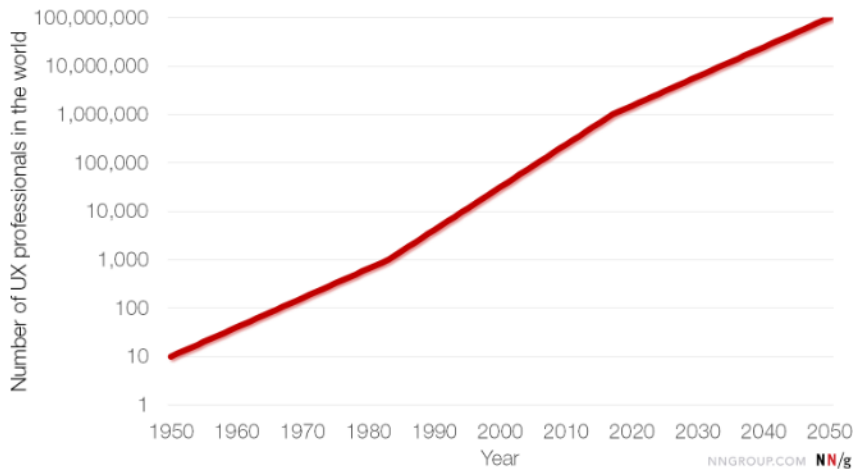


Figura 1.2: Il grafico mostra l'aumento degli esperti di UX e una proiezione del loro numero negli anni a venire. Si sottolinea che anche i dati relativi agli anni passati sono delle stime, in quanto non esiste una definizione specifica e chiara di "UX specialist"; le ricerche condotte dal *Nielsen Norman Group* dichiarano, infatti, che su 1045 esperti del settore trovati, questi hanno 210 diversi titoli lavorativi.

Quando non riesci ad esprimere in una frase qual è la promessa che fa un sito, allora quel sito non serve a niente.

Ci sono anche altri parametri che rendono un sito o un'applicazione un buon prodotto digitale, e sono tutti strettamente collegati al modo con cui gli utenti si rapportano al prodotto stesso. Peter Morville, architetto dell'informazione, ha identificato sette requisiti da rispettare per creare un sito ben progettato (Figura 1.3). Questi sono:

- *Usabile*: l'interfaccia deve essere facilmente utilizzabile.
- *Utile*: i contenuti dell'interfaccia devono essere originali e devono fornire risposte soddisfacenti alle richieste degli utenti.
- *Attrattivo*: la comunicazione visiva deve essere chiara e piacevole, con ogni elemento di design utilizzato appositamente per evocare emozioni ed apprezzamenti.
- *Accessibile*: il sito deve garantire a tutti la possibilità di essere utilizzato, ovvero deve poter essere consultabile anche da persone con difficoltà fisiche o tecnologiche.
- *Trovabile*: i contenuti devono essere trovabili e localizzabili con facilità all'interno del sito o applicazione.
- *Credibile*: il sito deve fornire contenuti e informazioni chiare e vere; gli utenti devono fidarsi e credere in ciò che viene detto loro.
- *Valido*: il sito deve fare in modo che l'esperienza dell'utente sia di valore e significativa.

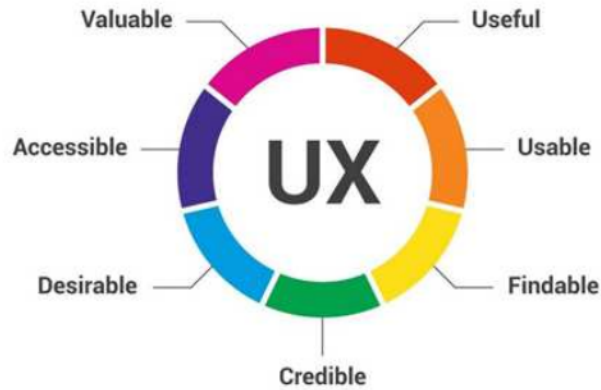


Figura 1.3: L'immagine mostra le sfaccettature della User Experience identificate da Peter Morville.

Al giorno d'oggi, con la tecnologia in continua evoluzione, questi requisiti hanno avuto bisogno di essere integrati. Un buon prodotto digitale deve, anche, essere:

- *Autoesplicativo*, ovvero chiaro nell'informazione e negli obiettivi.
- *Onesto*, ovvero propositivo nel cercare di migliorare sempre e correggere gli errori.
- *Modulabile*, cioè deve poter essere modificabile e sviluppabile nel tempo.
- *Sociale*, ovvero deve garantire dialogo con gli utenti.
- *Pervasivo*, cioè utilizzabile da differenti strumenti.

Tutte queste caratteristiche mettono in luce una considerazione fondamentale: il progettista non deve ideare un prodotto digitale basandosi sui suoi gusti personali. Non è lo User Experience Designer il futuro utilizzatore dell'applicazione o del sito e per questo il prodotto non deve rispecchiare le sue preferenze. Se un prodotto è pensato per un determinato gruppo di utenti, allora, nella progettazione, si deve tener conto dell'opinione dei diretti interessati. Anzi, il prodotto deve essere creato insieme alle persone che lo utilizzeranno, affidando agli utenti finali un ruolo chiave e centrale in tutto il processo. La prima regola di uno User Experience Designer è, dunque, imparare ad ascoltare: dalla conversazione con gli utenti possono venire alla luce soluzioni innovative che permetteranno di accelerare il processo creativo.

Come coinvolgere gli utenti

Lo User Experience Designer deve coinvolgere le persone. Innanzitutto deve considerare che queste possono essere divisi in due categorie, ovvero gli utenti finali, cioè coloro che utilizzeranno il prodotto, e gli stakeholder. In quest'ultimo gruppo rientrano tutti coloro che appartengono alle aree operative, i tecnici, i responsabili della comunicazione, del marketing e, più in generale, tutte le persone che vogliono collaborare alla riuscita del progetto. Ovviamente questi due gruppi avranno un grado di conoscenza ed esperienza diverso, ma non per questo i contributi forniti

da un gruppo sono più importanti degli altri. A prescindere da questa divisione, esistono delle tecniche per coinvolgere le persone che si possono applicare ad entrambi i gruppi: ciò che cambierà sarà soltanto il contenuto. Indipendentemente dalla categoria considerata l'obiettivo è identico: raccogliere informazioni e dati per poter sviluppare un prodotto che soddisfi i desideri di chi lo utilizzerà.

Il coinvolgimento delle persone è indispensabile lungo tutto il processo di progettazione. Ci sono, infatti, tre momenti in cui è importante richiedere la partecipazione attiva delle persone:

- Discovering;
- Co-designing;
- Testing.

La prima fase consiste nella raccolta delle informazioni e costituisce la prima tappa della progettazione. Attraverso svariate tecniche, come, ad esempio, l'intervista one-to-one, l'intervista di gruppo o i focus group, i questionari e, infine, le cultural probes, lo User Experience Designer raccoglie materiale, ascolta gli utenti e li osserva interagire con il prodotto. Tutte queste informazioni serviranno per comprendere i desideri e i bisogni degli utenti e, quindi, per stabilire i requisiti del sito o dell'applicazione.

La fase di codesing è successiva a quella di discovering e consiste nell'ordinare le informazioni raccolte nella fase precedente e analizzarle. L'obiettivo è produrre idee e soluzioni ai problemi emersi. Le tecniche utilizzate per coinvolgere le persone sono di due tipi, esistono, infatti, le tecniche di brainstorming, utilizzate per produrre o ordinare idee, e quelle di elicitazione, che aiutano, invece, a comprendere i modelli mentali degli utenti finali e creare un'architettura del sito che rispetti tali modelli.

L'ultima fase, quella di testing, è una fase particolare. Essa consiste, essenzialmente, nel porre l'utente di fronte al prodotto e nell'analizzare le sue reazioni. È molto utile per capire cosa si sta sbagliando e cosa si può migliorare. La particolarità di questa fase è che non è l'ultimo passo da compiere prima di rilasciare il prodotto, ma è una pratica che deve essere eseguita dall'inizio del progetto utilizzando, ad esempio, dei prototipi, fino alla fine dello sviluppo del prodotto, e anche dopo il suo rilascio, per poter studiare eventuali aggiornamenti.

Le tecniche da utilizzare durante la progettazione vanno stabilite a inizio progetto ma sono estremamente flessibili e adattabili. Questo significa che non esiste uno schema preciso che stabilisca quali tecniche è più opportuno utilizzare, ma queste devono essere valutate e scelte in base al contesto, al tempo e al budget a disposizione. L'importante è interagire e coinvolgere le persone il più possibile, anche in momenti diversi e con obiettivi differenti, per poter realizzare un prodotto digitale fatto su misura dell'utilizzatore.

1.2.2 Fasi di progetto

Come già accennato in precedenza, la progettazione UX non è costituita da degli step precisi da rispettare. Stiamo, infatti, parlando di un processo creativo ed elastico e ciò che si può fare è tentare di riassumerlo in delle macrofasi, per fornire un'idea della direzione da prendere. Anche parlando di macrofasi le opinioni sono differenti, alcuni esperti ne individuano tre, altri quattro, altri cinque, e così via.

Per cercare di delineare al meglio i passaggi fondamentali per la creazione di un buon prodotto digitale, parleremo di sei macrofasi: una fase preparativa, quattro fasi di elaborazione e una di evoluzione (Figura 1.4). Al centro di ognuna di queste fasi troveremo, ovviamente, le persone e le loro esperienze, che influenzeranno i progettisti nella scelta della direzione da percorrere e degli obiettivi da raggiungere, e li aiuteranno a tornare sulla “retta via” qualora questi obiettivi vengano persi di vista durante lo sviluppo del prodotto.



Figura 1.4: Le sei macrofasi principali dello sviluppo di un prodotto digitale.

Fase 1: Pianificazione

Questa prima fase è il momento in cui si raccolgono le prime informazioni e si valutano i tempi e le metodologie che verranno utilizzate. Si parla di analisi conoscitiva, durante la quale si iniziano a fissare i primi requisiti e le risorse che servono per soddisfarli. Il primo step in assoluto consiste nel parlare con il committente per stabilire aspettative e obiettivi.

Il secondo step consiste, invece, nello scegliere la metodologia che verrà usata durante lo sviluppo del prodotto. Gli approcci possono essere di tre tipi:

- *A cascata*: consiste in una sequenza lineare di passaggi, in cui si passa allo step successivo solo quando si è sicuri che lo step corrente sia concluso.
- *Agile*: è uno sviluppo rapido basato sulla collaborazione veloce dei membri del team. Il processo viene diviso in microprocessi a se stanti, ciascuno portato avanti da un collaboratore; alla fine vengono fusi i vari risultati.
- *Big Design Up Front*: è una variante meno rigida del metodo a cascata, che prevede fasi della progettazione piuttosto ampie e flessibili.

Alla fine di questa fase è dunque essenziale aver stabilito obiettivi e aver compreso qual è il problema da affrontare perché

design solves a problem; to provide a solution, you first need to understand the problem.

Fase 2: Ricerca e Analisi

Stabiliti gli obiettivi generali del progetto si prosegue con una ricerca di informazioni più consistente e profonda. È questo il momento di ascoltare tutte le persone coinvolte e osservarle mentre interagiscono con il prodotto, sia esso un prototipo o una versione preesistente da aggiornare. Segue l'analisi, che consiste nel comprendere, a partire dai dati raccolti, cosa pensano gli utenti e cosa desiderano. La tipologia

di indagine da condurre varia a seconda che il prodotto da realizzare sia completamente nuovo oppure che esso si basi su una versione già esistente ma ormai obsoleta. Nel primo caso l'analisi sarà più generica, mentre nel secondo caso si baserà sul prodotto già esistente (AS IS). Si eseguirà, dunque, una SWOT *analysis* con l'obiettivo di mettere in luce i punti di forza e di debolezza del prodotto da riprogettare.

Si passerà dunque a intervistare gli utenti per apprendere come si rapportano al prodotto, non solo in termini quantitativi (i click e i percorsi che fanno gli utenti all'interno di un sito sono dati che si hanno già a disposizione). Ciò che ci interessa sono le esperienze degli utenti, ma anche le loro emozioni e i loro sentimenti, scaturiti dall'interazione con il prodotto. I modi di rapportarsi con le persone sono svariati:

- Interviste strutturate: vengono poste domande fisse a risposta chiusa.
- Interviste "naturali": le domande sono aperte in modo tale che l'intervistato sia libero di rispondere.
- Storytelling: gli intervistati vengono sottoposti a poche domande aperte che li guidano e li stimolano nel raccontare eventi ed esperienze vissute.
- Analisi dei task: all'intervistato viene chiesto di disegnare un processo del sito, ovvero di spiegare come si comporterebbe in una determinata situazione. Lo scopo è ottenere una schematizzazione dei malfunzionamenti.
- Diari: è uno strumento di auto-documentazione. All'intervistato viene chiesto di annotare giorno per giorno elementi della propria vita personale che coinvolgono il prodotto che si sta progettando.
- Cultural probes: sono una variante creativa dei diari. Invece che annotare riflessioni ed esperienze, agli intervistati viene fornito un kit di materiali diversi, utili a raccogliere informazioni in maniera informale e originale.

Combinando i risultati delle interviste agli utenti con quelli delle interviste agli stakeholder, si è in grado di riorganizzare i dati. I dati si dividono innanzi tutto per similarità attraverso tecniche come il diagramma di affinità; meccanicamente si elabora una classificazione delle richieste da soddisfare in base alla priorità.

Infine, come ultimo step, si esegue un'analisi competitiva per valutare come si posiziona il sito o l'applicazione che stiamo progettando all'interno del mercato e per valutare la concorrenza.

Fase 3: Creazione

Sulla base di quanto scoperto durante l'interazione con gli utenti, si passa, ora, a ideare soluzioni ai problemi riscontrati. Il primo step consiste nel creare una mappa concettuale che riassume chi è coinvolto nel progetto e i processi che coinvolgono quest'ultimo. In poche parole la mappa deve rappresentare i ragionamenti del team di progettazione, utilizzando strumenti semplici, come fogli e penne o lavagne, per evidenziare il funzionamento e la logica del prodotto. Il secondo passo è produrre un documento chiamato *personae* che ha l'obiettivo di fornire un quadro degli utenti, del loro comportamento e delle loro aspettative. Non si utilizzano persone reali, ma vengono creati dei profili verosimili di possibili utenti che hanno determinati bisogni, determinate esperienze e che si rapporteranno con il prodotto in un certo modo. Si riesce a creare, così, un prospetto degli utenti finali per valutare l'efficacia della progettazione. In pratica, si reperiscono dati da altre ricerche, per esempio dati

ISTAT o CENSIS, per creare questi profili e poi si cerca di renderli il più verosimili possibile aggiungendo un nome, una foto, dati anagrafici, un background reale, frasi pronunciate, e qualsiasi informazione aggiuntiva che riesca a rendere credibile il personaggio. Tutto ciò serve per mettere a fuoco un determinato target e valutare il suo rapporto con il prodotto.

L'ultimo passo consiste nel costruire con gli utenti finali l'architettura del sito o dell'applicazione. Esistono svariate tecniche, chiamate tecniche di elicitazione, che possono essere applicate. Una delle più utilizzate è quella del *card sorting*, che consiste nel fornire ai partecipanti una serie di cartoncini con sopra scritti i contenuti che dovranno essere presenti nella piattaforma e una loro spiegazione. Chi partecipa dovrà raggruppare i cartoncini in maniera logica, creando, così, delle categorie più ampie di concetti tra loro collegati. Questo metodo risulta molto utile per comprendere come ragionano le persone e i loro modelli mentali e per capire cosa ritengono importante e cosa, invece, considerano superfluo. Alla fine della sessione di card sorting emergerà la struttura che dovrà avere il sito per essere intuitiva e facile da usare.

Fase 4: Sviluppo

Durante questa fase si arriva a realizzare un prodotto verosimile, ovvero il prototipo. Si entra quindi nel vivo della progettazione, costruendo le basi dell'architettura informativa, ovvero l'insieme organizzato di informazioni restituite all'utente. Per arrivare ad avere un prodotto finito ci sono vari step da completare.

1. È fondamentale definire subito l'*impianto iniziale dei contenuti*, soprattutto alla luce del fatto che la sequenzialità degli stessi è un modello ormai sorpassato a causa dell'ipertesto. Il contenuto digitale è diventato liquido e l'assetto delle pagine dipende dalla relazione tra i contenuti. Questi possono essere organizzati essenzialmente in tre modi diversi. Il primo è la struttura gerarchica, o ad albero, in cui il sistema parte da una home e si snoda in diverse aree, a loro volta divise in sottoaree. Un'altra struttura è quella multidimensionale in cui il contenuto non si estende in profondità ma in direzione orizzontale; in questo modo un'informazione non ha una posizione univoca, ma può essere richiamata da più parti del sito senza il bisogno di doverla replicare ogni volta. L'ultima struttura è quella relazionale, basata su un sistema di organizzazione altamente dinamico: non esiste più una collocazione fissa dei contenuti, ma le aree prendono forma in base al clustering prodotto da uno o più criteri esterni.
2. Ora si può procedere alla realizzazione della *mappa del sito*, che può essere più o meno profonda, grande o piccola; l'importante è che sia autoesplicativa. Esistono degli elementi e dei costrutti standard che permettono di rendere le mappe esaustive, intuitive e comprensibili a tutti, anche ai non esperti.
3. Il prossimo passo è progettare i *supporti logico-semantic*, ovvero gli strumenti che aiutano nel recupero dell'informazione. Il supporto principale è quello che permette la ricerca delle informazioni all'interno della piattaforma, attuabile tramite tag e keyword, ovvero collegamenti tra contenuti basati su parole in comune.
4. Durante questo passo ci si occupa della creazione dei *wireframe*, letteralmente "fil di ferro". Essi sono delle illustrazioni schematiche delle varie pagine del sito

o dell'applicazione che mostrano la disposizione del contenuto; si basano sull'utilizzo di linee semplici e senza grafica. Questi disegni servono essenzialmente a mostrare l'architettura delle schermate, evitando la distrazione della grafica.

5. Il prossimo documento da preparare è il *flowchart* che rappresenta in maniera dinamica la relazione tra utente e sito. La differenza con le mappe risiede nel fatto che il diagramma dei processi può essere solo sequenziale: scelta una funzionalità, esso spiega tutti i passaggi che devono essere compiuti affinché l'utente raggiunga un obiettivo o esegua una determinata azione.
6. Un altro documento da preparare è l'*analisi funzionale*, che contiene una descrizione specifica e dettagliata di ogni servizio offerto dalla piattaforma. È un documento ripetitivo e poco originale che descrive minuziosamente tutti i passaggi che vengono compiuti dall'interazione dell'utente con la piattaforma fino al conseguimento dell'obiettivo. Risulta essere molto utile per gli sviluppatori in quanto, leggendo tale documento, quest'ultimi hanno un'idea precisa di cosa devono realizzare, senza dover interpretare le richieste del committente.
7. Come ultima cosa, il progettista deve iniziare a lavorare all'*infrastruttura tecnologica*: viene creata una piattaforma ad hoc personalizzata che realizzi le funzioni che deve avere il prodotto finale, o viene scelta un'infrastruttura già esistente che sappia fare ciò che si era proprosti di realizzare. In ogni caso l'infrastruttura tecnologica è composta da più elementi, come server e database, ma quello che interessa allo User Experience Designer è il fatto che è divisa nella componente di back end, che organizza i contenuti, e nella componente di front end, che è l'interfaccia con cui l'utente interagisce con la piattaforma. Il compito dello User Experience Designer è proprio quello di mettere in contatto questi due elementi e di allinearli.

Ora si è in grado di realizzare un prototipo che, come gli esperti sottolineano, non è un artefatto bensì un processo iterativo in grado di trasformarsi ed evolversi nel tempo. La prototipazione può essere eseguita in più momenti lungo tutto l'iter progettuale, ogni volta con scopi differenti. Il prototipo può essere cartaceo o digitale, in base alle esigenze, ovvero in base all'aspetto che si vuole mettere in risalto per essere testato dalle persone.

Fase 5: Testing

Il prodotto viene testato in tutte le sue componenti. Questa fase non viene eseguita solo prima del rilascio del prodotto finito. In realtà è un'attività iterativa da svolgere sin dall'inizio del progetto. Questa è una fase cardine in quanto i risultati che vengono conseguiti da ogni test servono per mantenere il progetto aderente alla realtà e in linea con gli obiettivi da raggiungere, per rilevare difficoltà, gradimenti e aspettative che verranno tramutati in consigli da seguire per poter migliorare la User Experience del prodotto. Testare un prodotto digitale ormai è molto semplice e veloce. Bisogna, però, stare attenti in quanto i dati che vengono raccolti sono molteplici e provengono da utenti che non sono esperti del settore. È necessario, quindi, saper interpretare correttamente i dati.

Il processo di testing si può riassumere in tre fasi:

1. *Pianificazione*: si decidono gli obiettivi e i passaggi fondamentali. In poche parole, si decide cosa va testato, come, da chi, quando e dove svolgere il test. Si

possono, infatti, svolgere diversi tipi di test: si può testare il prodotto intero o solo una sua parte, il test si può eseguire dal vivo o in remoto, può essere qualitativo o quantitativo, etc. Per quanto riguarda chi esegue il test, è necessario selezionare a grandi linee il target. Una volta fatto ciò, non è necessario far testare il prodotto a decine di persone diverse: per evidenziare i problemi bastano 5 persone.

2. *Attività di test*: prima dell'avvio della sessione di testing è necessario redigere due documenti. Il primo è la guida al test che stabilisce cosa le persone devono fare esattamente; in poche parole, esso descrive il compito di ogni partecipante al test. Il secondo sono le regole di facilitazione, ovvero ciò che deve fare il mediatore che conduce e guida gli utenti nel test.
3. *Analisi dei risultati*: nella sessione di test viene raccolto molto materiale, anche in formati diversi, per procedere, immediatamente dopo, al suo ordinamento. È importante, come prima cosa, evidenziare i problemi chiave che sono emersi ed elaborare una scala di priorità, per capire quali di questi problemi va affrontato con più urgenza e come. I risultati del test vanno, poi, presentati al cliente, meglio se in modo originale, utilizzando, ad esempio, la tecnica dello storytelling.

Fase 6: Evoluzione

Dopo il rilascio del prodotto vengono innanzitutto elaborate le soluzioni che sono state adottate durante il progetto e viene redatto un documento che servirà a ricostruire la storia del prodotto per facilitare un futuro aggiornamento. Successivamente si proseguirà raccogliendo feedback dagli utenti per capire se è necessario un ulteriore sviluppo e per comprendere cosa c'è da migliorare.

I due documenti principali da redigere sono la guida di stile, che descrive cosa si è progettato, come mantenere il prodotto e farlo evolvere, e il pattern design, o pattern library, che è essenzialmente un archivio di soluzioni e problemi ricorrenti nella progettazione del prodotto. Il ciclo di vita di un prodotto digitale non termina, infatti, con il rilascio del sito o applicazione, ma è necessario un monitoraggio costante per valutare se c'è qualcosa da migliorare e per sviluppare eventuali aggiornamenti. In questo modo il prodotto rimane sempre all'avanguardia e non diventa mai obsoleto.

Tecniche di User Experience Applicate

Dopo la panoramica sulle varie fasi per una buona progettazione della UX, in questo capitolo entriamo nel dettaglio del nostro progetto. Verrà, dunque, descritto ciò che si intende realizzare e si affronteranno le prime fasi della progettazione. Illustreremo, quindi, le tecniche di User Experience che si intendono applicare per il conseguimento dell'obiettivo.

2.1 Descrizione del progetto

L'obiettivo posto al centro di questa tesi è quello di analizzare un'applicazione già esistente, in particolare quella sviluppata dal collega di corso Kevin Cela, valutare pregi e difetti, e riprogettarla affinché l'usabilità e l'esperienza utente siano le migliori possibili. L'applicazione in questione si chiama *Assistente di Studio* ed è pensata per supportare gli studenti nello studio.

L'applicazione è composta da tre sezioni. La prima è costituita da un calendario, su cui poter fissare eventi, scadenze, ma anche esami, consegne o appuntamenti. Una sezione contiene le note, per poter memorizzare appunti sulle lezioni o, in generale, qualsiasi cosa possa essere utile ricordare. Infine, la sezione più particolare e importante dell'applicazione contiene mazzi di flashcard. Le flashcard sono delle carte che contengono delle informazioni su di un loro lato e la relativa descrizione sull'altro: possono racchiudere, ad esempio, un termine tecnico con la sua definizione, o una domanda con la sua risposta. Tali flashcard costituiscono un metodo innovativo per aiutare gli studenti nella memorizzazione.

Secondo gli studi condotti alla fine del 1800 dal filosofo e psicologo tedesco Hermann Ebbinghaus, la velocità con cui l'uomo dimentica informazioni è maggiore nel breve periodo, e diminuisce con il passare del tempo, definendo la cosiddetta "*curva dell'oblio*" (Figura 2.1). Ma la scoperta più importante fu il fatto che Ebbinghaus comprese che la memorizzazione di un'informazione è più efficace se distribuita in più sessioni d'apprendimento. Si assimila meglio una nozione se ripetuta una volta al giorno per cinque giorni, piuttosto che cinque volte in un giorno unico. Ripetere in momenti dilazionati nel tempo, quindi, fa sì che la curva dell'oblio si appiattisca.

Alla luce di ciò, il metodo di apprendimento tramite flashcard permette di stimolare la memoria a ricordare, e aiuta a ripetere gli stessi concetti in momenti diversi.

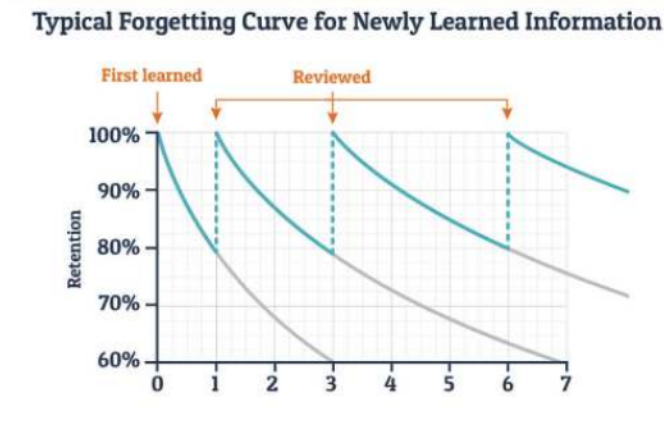


Figura 2.1: Curva dell'oblio di Hermann Ebbinghaus.

L'ideale sarebbe creare, ad esempio, le flashcard subito dopo lezione, ma iniziare a ripeterle dal giorno successivo. In ogni caso, nell'applicazione è possibile creare le proprie flashcard e dividere queste ultime in gruppi, ad esempio per materia, o addirittura per argomento.

Nelle prossime sezioni inizierà la vera progettazione, o meglio riprogettazione, dell'applicazione appena descritta, per poterne migliorare la User Experience. Il processo di progettazione è stato descritto in maniera teorica nel capitolo precedente, ma essendo, comunque, un iter elastico e adattabile al compito da svolgere, sono state fatte delle precise scelte progettuali, adattandole al caso in esame. Alcuni step delle varie fasi di progettazione non verranno eseguiti, perchè non ritenuti utili per i nostri obiettivi.

2.2 Tecniche di UX utilizzate

In questa sezione, iniziamo la progettazione dell'applicazione ponendo l'attenzione sulle tecniche di User Experience che verranno applicate nel corso della realizzazione dell'applicazione.

Iniziamo con la prima fase della progettazione, quella di pianificazione. In questo step dovremmo stabilire, attraverso un'analisi conoscitiva, le linee guida e gli obiettivi principali del progetto. Dovremmo parlare con il committente del lavoro per comprendere propositi e aspettative. Dal momento che il progetto non è effettivamente commissionato da un'azienda, ma rimane all'interno di un contesto universitario, le direttive da rispettare non prevedono di dover realizzare un'applicazione migliore delle altre presenti sul mercato, e nemmeno di dover raggiungere un determinato numero di iscrizioni o di click. L'obiettivo primario che stabiliamo è quello di cercare di migliorare l'esperienza di uno studente supportandolo nello studio, realizzando uno strumento che sia tanto utile quanto facile da usare, intuitivo e usabile.

Si sottolinea il fatto che non stiamo creando un'applicazione ex novo, ma ne stiamo modificando una già esistente. Questo fattore è molto importante, e i prossimi passaggi saranno influenzati da questo dettaglio; per riprogettare l'applicazione ci possiamo basare, infatti, su ciò che non ha soddisfatto le esigenze degli utenti, per poter migliorare il prodotto, ma possiamo, anche, mantenere funzionalità che, al contrario, sono state gradite e apprezzate.

Infine, dobbiamo stabilire che tipo di approccio metodologico vogliamo adottare. Come spiegato nel capitolo precedente, gli approcci possibili sono tre: quello a cascata, quello agile e il Big Design Up Front. In realtà, dal momento che il progetto non è sviluppato da un team, i cui componenti si possono dividere i compiti per completarli in parallelo o in sequenza, non ha molto senso individuare un approccio particolare, perchè, in ogni caso, tutti i compiti verranno svolti dalla stessa persona.

2.2.1 Discovering

Una volta stabiliti gli obiettivi primari e la metodologia da seguire, è necessario raccogliere informazioni: inizia, dunque, la fase di ricerca. Si è già stabilito che lo scopo è la riprogettazione di un'applicazione; la raccolta di informazioni sarà, quindi, divisa in due parti. La prima consiste in una navigazione approfondita del prodotto già esistente; la seconda, invece, prevede il coinvolgimento degli utenti finali.

Analisi dell'App esistente

Iniziamo la raccolta di informazioni analizzando il prodotto già disponibile, per poterne capire la struttura, la logica, i servizi che offre e come questi funzionano. Una volta compreso a fondo il funzionamento dell'applicazione, si può procedere con la SWOT *analysis*, con l'obiettivo di mettere in luce i punti di forza e di debolezza del prodotto stesso.

La SWOT *analysis*, o matrice SWOT (Figura 2.2), è uno strumento di pianificazione strategica, usato per focalizzare l'attenzione sui punti cruciali della propria attività, o del proprio prodotto, in modo tale da poter, poi, sviluppare un piano di marketing efficace. La tecnica è attribuita all'economista statunitense Albert Humphrey, che la sviluppò all'Università di Stanford fra gli anni '60 e '70, e prevede la valutazione di un prodotto, o di un progetto, sulla base dei punti di forza (*Strengths*), dei punti di debolezza (*Weaknesses*), delle opportunità (*Opportunities*) e delle minacce (*Threats*). Il risultato di quest'analisi è, normalmente, una tabella, formata da quattro caselle, ciascuna contenente in maniera schematica i fattori che rientrano nella categoria corrispondente.

Per poter compilare tale tabella, è necessario eseguire un'analisi del prodotto, e compilare un documento descrittivo che contenga tutte le informazioni trovate, così da avere una “fotografia” del prodotto stesso. Analizziamo, dunque, l'applicazione in questione, oggetto di questa riprogettazione.

Assistente di Studio è un'applicazione semplice, chiara e minimalista, e, per questo, estremamente intuitiva. L'app è costituita da tre pagine principali (*Calendario*, *Flashcards* e *Note*), ciascuna inizialmente priva di qualsiasi elemento, ma munita di un pulsante per poter aggiungere contenuti.

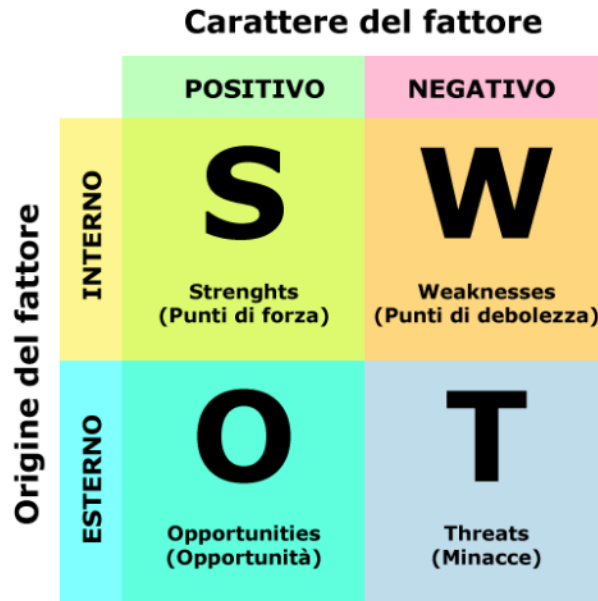


Figura 2.2: Matrice SWOT.

Calendario consente di creare un evento: cliccando il pulsante, contraddistinto da un “+”, è possibile selezionare la materia, la data, l’ora e aggiungere una descrizione. Una volta registrati gli eventi, questi verranno visualizzati uno sotto l’altro, dal più prossimo al più lontano. Inoltre, è possibile modificare o eliminare un evento tenendo premuto sullo stesso e selezionando l’opzione corrispondente.

Flashcards permette di creare un nuovo mazzo di carte, contraddistinto da materia, nome e descrizione. I mazzi creati vengono visualizzati uno sotto l’altro. Una volta creato il mazzo, si può cliccare su *Gestisci Mazzo* per creare le carte vere e proprie. Si aprirà una nuova pagina con presente il tasto “+”, tramite il quale si possono aggiungere le carte, ciascuna con il proprio titolo e contenuto. Tutte le carte e tutti i mazzi possono essere modificati o eliminati, tenendo premuto su di essi e selezionando l’azione corrispondente. Infine, ogni mazzo presenta il tasto *Studia*. Una volta premuto, le carte create verranno mostrate in maniera casuale; in particolare verrà ne visualizzato il titolo; solo cliccando sul tasto *Mostra il contenuto della carta* verrà mostrato l’altro lato della stessa. A questo punto comparirà la domanda “*Come ti è sembrato questo concetto?*”, con due opzioni di risposta: *Facile* o *Difficile*. Se viene selezionato *Difficile*, la prossima volta che viene studiato il mazzo, quella carta verrà mostrata tra le prime.

Note permette, sempre tramite il pulsante “+”, di aggiungere una nota, specificandone la materia, il titolo e il corpo. Le note saranno visualizzate per materia, una sotto l’altra. Tenendo premuto sulla nota, si può decidere se modificarla o eliminarla.

In alto a destra di ciascuna di queste tre pagine principali, a fianco del nome dell’applicazione, è presente il tasto, contraddistinto da tre punti verticali, che con-

sente di aprire una tendina con le sezioni *Gestisci Materie*, *Impostazioni* e *Aiuto*. *Gestisci Materie* permette di aggiungere una materia, caratterizzata dal nome e da un colore, selezionabile da una tavolozza o inserendo il nome RGB. *Impostazioni* contiene due opzioni. La prima permette di abilitare le notifiche del calendario; la seconda permette di scegliere se visualizzare il calendario intero, o se mostrare solo gli eventi salvati. Se si decide di mostrare il calendario, gli eventi non saranno più disposti uno sotto l'altro, ma verranno mostrati solo quando si consulta il giorno corrispondente. Se è stato selezionato un altro giorno, però, non verrà indicato quand'è il prossimo evento, nè ogni altro evento che è stato registrato. Infine, *Aiuto* contiene la spiegazione di ogni pulsante o funzione presenti nell'app.

Bisogna specificare, però, che non è possibile creare alcun evento, mazzo, o nota, finchè non si è aggiunta la materia nella sezione *Gestisci Materie*. Quest'ultima risulta, quindi, essere la prima azione da compiere quando si utilizza l'applicazione, e il fatto che non si trovi nelle pagine principali può creare confusione. In ogni caso, l'applicazione scende fino al terzo livello, per cui non ha struttura gerarchica, ma, piuttosto, si estende in senso orizzontale. Essa non contiene servizi online e non ha bisogno di un meccanismo di autenticazione. Il logo risulta essere ben studiato e intuitivo, ma all'interno dell'applicazione non sono presenti altre immagini, se non dei simboli stilizzati delle tre pagine principali, posizionate nei pulsanti corrispondenti.

Dopo aver analizzato l'applicazione, è possibile portare a termine la SWOT analysis. Ai fini della nostra progettazione, però, risultano utili solo i dati relativi ai fattori interni; ci interessano, quindi, solo i punti di forza e quelli di debolezza, per cui la tabella che costruiremo presenterà soltanto due caselle. Essa viene monitorata nella Tabella 2.1.

<i>Punti di forza</i>	<i>Punti di debolezza</i>
Spiegazione di tutte le sezioni e termini	Funzione principale nascosta
Semplice da usare	Poche immagini
Buona gestione delle flashcard	Calendario non gestito al meglio
Minimale	Obbligo di compilare il campo "Descrizione"

Tabella 2.1: SWOT analysis dell'applicazione *Assistente di Studio*

Interazione con gli utenti

Tramite l'analisi appena fatta, abbiamo descritto l'applicazione e abbiamo capito come funziona. Ma, come è stato fatto notare già nel capitolo precedente, la progettazione UX si basa sui desideri, sulle critiche e sulle aspettative degli utenti finali. È, perciò, estremamente importante riuscire a coinvolgerli nelle varie fasi di progetto, iniziando dalla raccolta delle informazioni. Per poter stabilire gli obiettivi dell'applicazione e i requisiti da soddisfare è necessario sapere cosa gli utenti si aspettano dal prodotto che vogliamo offrire, cosa vorrebbero migliorare rispetto, ad esempio, a una versione antecedente, e che funzionalità desiderano trovare.

Esistono molteplici tecniche differenti che possono essere utilizzate per raccogliere informazioni dagli utenti finali, e non c'è un preciso criterio per scegliere tra un metodo e l'altro. Sono tutti validi, ma una tecnica può essere più efficace rispetto a un'altra in base al particolare progetto che si sta seguendo e in base ad altri fattori, quali budget e tempo a disposizione.

Ai fini di questa riprogettazione, la metodologia scelta per raccogliere informazioni è il questionario online, in quanto risulta essere conveniente sia in termini di tempo che di elaborazione dei dati. Inoltre, essendo l'applicazione utile in ambito scolastico o universitario, il target prescelto è quello degli studenti, sicuramente più propositivi nel compilare un questionario che nell'essere sottoposti a un'intervista, di durata, sicuramente, superiore.

I questionari, infatti, sono un mezzo rapido, economico ed estremamente pratico per recuperare informazioni, molto utili per fotografare una situazione e recuperare molti dati contemporaneamente. Il difetto di questa tecnica, però, è la rigidità dello strumento: è molto facile capire cosa non piace, ma non lo è altrettanto conoscere il perchè e come migliorare. Ogni questionario non deve durare più di 10 minuti, per non scoraggiare l'intervistato ad abbandonare prima del completamento, e può essere strutturato in più moduli, se è articolato e tratta di più argomenti. Esistono tre tipologie di domande che possono essere utilizzate in un questionario:

- *Checkbox*: sono domande che prevedono come risposta “sì” o “no”. È buona norma prevedere sempre una risposta alternativa, come, ad esempio “altro”. Sono la tipologia di domande più semplice e veloce, sia per l'utente che per chi elabora le risposte.
- *Risposta multipla*: sono domande che prevedono più di una possibile risposta. Queste vengono elencate, ad esempio, tramite un menù a tendina o a discesa e l'utente può scegliere solo un'opzione. Un'alternativa può essere quella di chiedere all'utente di esprimere una propria valutazione: si chiama *scala di Likert* ed è una tecnica che permette di raccogliere pareri e sentimenti nei confronti di un prodotto.
- *Risposta libera*: sono le domande che permettono agli utenti di esprimere meglio se stessi. Non sono presenti risposte tra cui scegliere, ma viene lasciato uno spazio bianco che gli intervistati devono utilizzare per scrivere, liberamente, il loro parere. Sono le più difficili da elaborare, ma sono utili quando non è necessario quantificare i risultati e quando si effettua uno studio più esplorativo.

Per raccogliere le informazioni per questo progetto, il questionario è stato realizzato tramite *Moduli Google*. È composto da 14 domande, alcune a risposta chiusa, alcune che richiedono una votazione tramite la scala di Likert e alcune a risposta aperta, con la possibilità di suggerire idee. Il questionario verrà inviato, come detto precedentemente, solo a studenti e, insieme al modulo, verrà allegata l'applicazione in formato *apk*. Verrà, quindi, chiesto agli utenti di testare l'app per qualche giorno prima di compilare il modulo. Si allegano gli screenshot del questionario (Figura 2.3). Le risposte verranno, poi, analizzate nel prossimo capitolo.

2.2.2 Co-designing

Concludiamo illustrando un'altra tecnica di UX, sebbene quest'ultima verrà applicata in un secondo momento. La fase di co-designing prevede, infatti, il coinvolgi-

Applicazione "Assistente Di Studio"

Il questionario è stato creato come supporto a un progetto universitario. Lo scopo è raccogliere informazioni e pareri circa l'applicazione "Assistente Di Studio", per poi procedere a una riprogettazione dello stesso basata sui dati raccolti. Si prega di rispondere seriamente, in quanto le risposte che date saranno effettivamente utilizzate. Grazie per il vostro tempo.

*Campo obbligatorio

A primo impatto, come hai ritenuto 'Assistente Di Studio'? *

Piacevole
 Accattivante
 Poco curata
 Caotica
 Intuitiva
 Intrigante
 Noiosa
 Banale

C'è qualcosa che ritieni frustrante in 'Assistente Di Studio'?

La tua risposta

Se potessi cambiare qualcosa in 'Assistente Di Studio', cosa cambieresti e perché?

La tua risposta

Quale servizio ritieni fondamentale? *

Scegli

Dai una valutazione ai seguenti aspetti di 'Assistente Di Studio': *

	1	2	3	4	5
Facilità di utilizzo	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Chiarezza dei servizi	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Gestione delle materie	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Funzionabilità	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Aspetto grafico	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Notifiche	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Come valuti i servizi offerti da 'Assistente Di Studio'? *

	Inaufficiente	Sufficiente	Buono	Ottimo
Calendario	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Flashcards	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
Note	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Se dovessi dare una valutazione generale ad 'Assistente Di Studio', che voto gli daresti? *

1 2 3 4 5

Perché?

La tua risposta

Quanto ritieni facile da usare 'Assistente Di Studio'? *

1 2 3 4 5

Poco Molto

Come miglioreresti 'Assistente Di Studio'? Dicci le tue idee

La tua risposta

C'è qualche altra funzionalità che vorresti aggiungere all'applicazione?

La tua risposta

Consideri 'Assistente Di Studio' uno strumento utile per lo studio? *

Sì
 No

Raccomanderesti 'Assistente Di Studio' a un amico? *

Sì
 No

Quali sono i tuoi studi? *

Scuola secondaria di secondo grado
 Università
 Altro

Figura 2.3: Screenshot del questionario

mento degli utenti finali, ma ciò avviene solo dopo aver rielaborato i dati ricavati nella fase di discovering, ovvero dopo aver stabilito i requisiti che l'applicazione deve soddisfare.

La tecnica che verrà applicata è quella del *card sorting*. Questa rappresenta un metodo di classificazione, o categorizzazione, che consente di organizzare contenuti, problemi e soluzioni. Nell'ambito della progettazione di un prodotto digitale, il card sorting permette di ordinare le idee e di costruire l'architettura del sito o dell'applicazione. La struttura di uno strumento digitale, e delle informazioni che questo veicola, deve essere progettata con estrema attenzione, tenendo in considerazione esigenze informative e modalità di navigazione. Applicando questa tecnica, si posso-

no coinvolgere gli utenti finali, guidandoli nella progettazione dell'architettura del sito che andranno ad utilizzare loro stessi. Le informazioni che saranno acquisite tramite il card sorting sono utili per comprendere i modelli mentali di classificazione, per valutare la comprensibilità delle etichette e di qualsiasi cosa sia scritta nel sito o nell'applicazione; in poche parole, il card sorting permette di costruire un prodotto fatto su misura del futuro utilizzatore, senza che vi sia l'influenza del modello logico del progettista.

L'obiettivo del card sorting è, quindi, quello di trovare il modo migliore di rappresentare i contenuti in base a come gli utenti si aspettano che vengano classificati. La realizzazione di questa tecnica è molto semplice; i materiali necessari sono soltanto un mazzetto di cartoncini, o svariati post-it, un posto in cui poterli appendere, e un gruppo di utenti selezionati. Per una buona riuscita, ci sono quattro step da seguire:

1. *Creare una lista di contenuti del progetto*: selezionare tutti i contenuti del sito o dell'applicazione e scriverli ognuno su di un cartoncino, con la relativa spiegazione. Per essere più comprensibile, le voci su cui lavorare devono essere standardizzate, ovvero i contenuti devono essere omogenei e semplici da comprendere.
2. *Rintracciare un gruppo di utenti e consegnare loro il set di cartoncini*.
3. *Chiedere agli utenti di dare un ordine ai cartoncini consegnati*: il loro obiettivo è quello di suddividerli in categorie e sottocategorie che possono definire loro stessi (card sorting aperto), o in gruppi predefiniti (card sorting chiuso).
4. *Analizzare le card*: dopo aver chiesto a più persone di effettuare questa divisione dei concetti, è necessario segnarsi quante volte una card è stata posizionata e categorizzata nello stesso modo. Più uno stesso concetto è stato classificato in una determinata categoria, più rilevante è l'informazione ottenuta ai fini della costruzione della struttura del prodotto.

Sono stati nominati due tipi di card sorting, quello aperto e quello chiuso. La differenza sostanziale tra i due tipi è che nel primo le persone sono libere di organizzare i contenuti in cluster a piacere nonché di assegnare a ciascun cluster il nome che ritengono più appropriato. Nella modalità chiusa, invece, le categorie sono già stabilite e viene chiesto agli utenti solo dove posizionerebbero le card; in pratica, l'alberatura delle informazioni è già definita, ma l'utente ha il compito di testarla. Anche qui, non ci sono regole precise che stabiliscano quando usare una variante e quando l'altra, ma, in linea di massima, si consiglia di utilizzare il card sorting aperto se si sta progettando un prodotto nuovo; in questo modo si può individuare la struttura che gli utenti considerano più intuitiva e la nomenclatura che ritengono più esplicativa. Se si sta affrontando una riprogettazione, si consiglia la modalità chiusa. In questo caso lo scopo è testare se i nomi del menù sono chiari, se le categorie sono troppe o troppo poche e se la suddivisione dei contenuti è, a loro parere, intuitiva. Ovviamente l'analisi dei dati ottenuti è diversa, in quanto, in quest'ultimo caso, è necessario registrare quanti e quali sono stati gli scostamenti di classificazione dall'architettura di partenza. In ogni caso, nessuno vieta di usare una modalità ibrida, in cui i cluster sono già stati definiti, ma agli utenti viene data la possibilità di cambiare il loro nome, di cancellarli o di aggiungerne di nuovi.

Analisi dei Requisiti e Progettazione

Entriamo, ora, nel vivo del nostro progetto. In questo capitolo analizzeremo i risultati del questionario, per comprendere i pareri degli utenti finali e per poter stabilire i requisiti da soddisfare. Una volta chiarito ciò che andremo a realizzare, proseguiremo con la progettazione, stabilendo l'architettura dell'applicazione, e creando altri documenti di supporto, tra cui i wireframe e i process flow.

3.1 Analisi dei Requisiti

Il primo passo della progettazione di un'applicazione consiste nello stabilire i requisiti che il progetto dovrà soddisfare. Per far ciò, come detto nel capitolo precedente, è stato realizzato un questionario, somministrato a un campione di studenti. Analizziamo i risultati di tali questionari, per poter individuare punti di forza e di debolezza dell'applicazione secondo il parere degli utenti finali, con l'obiettivo di stilare una prima lista di requisiti da soddisfare.

3.1.1 Risultati del questionario

Analizzando le risposte raccolte tramite il questionario, si può notare che, a primo impatto, la maggior parte degli utenti descriverebbero *Assistente di Studio* in maniera negativa. Infatti, fra tutti gli aggettivi disponibili per definire l'applicazione, il più usato è stato “poco curata”, seguito da “noioso” e da “banale” (Figura 3.1).

Sono, poi, stati fatti votare i servizi offerti dall'applicazione (Figura 3.2). Riguardo al Calendario, gli utenti hanno avuto opinioni diverse, distribuendosi su tutte le valutazioni disponibili, ma presentando un picco in corrispondenza del giudizio “sufficiente”. Per quando riguarda la sezione Flashcard, i voti sono tutti concentrati intorno a “sufficiente” e “buono”, con un lieve vantaggio di quest'ultimo. Infine, per la sezione Note, i giudizi sono stati piuttosto omogenei: la maggior parte degli utenti ha ritenuto il servizio “sufficiente”.

È stato, poi, chiesto di dare una valutazione generale all'applicazione, su una scala da 1 a 5 (Figura 3.3). I risultati non sono stati troppo soddisfacenti; infatti la maggior parte degli utenti ha votato 3, solo una persona ha votato 4, gli utenti restanti hanno votato 2.

A primo impatto, come hai ritenuto "Assistente Di Studio"?

11 risposte

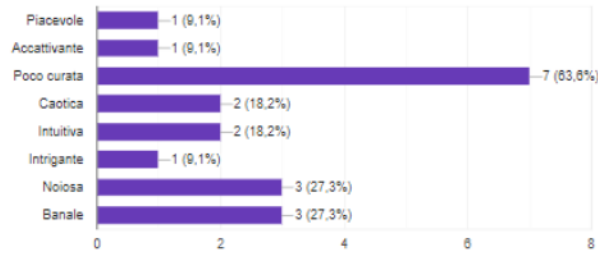


Figura 3.1: Aggettivi che descrivono "Assistente di Studio"

Come valuti i servizi offerti da "Assistente Di Studio"?

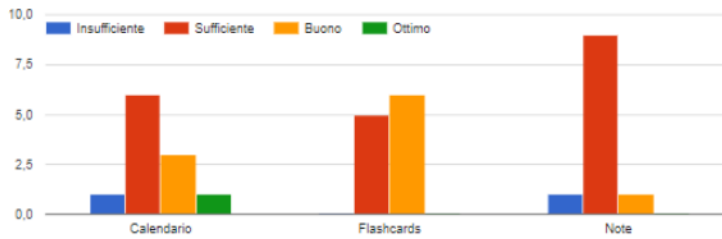


Figura 3.2: Valutazione dei servizi offerti

Se dovessi dare una valutazione generale ad "Assistente Di Studio", che voto gli daresti?

11 risposte

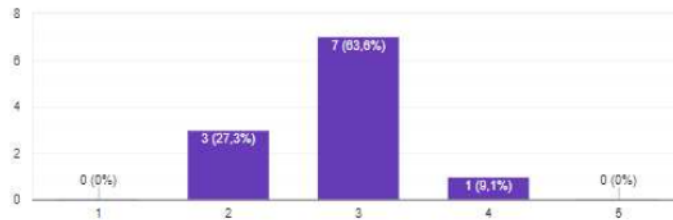


Figura 3.3: Valutazione generale dell'applicazione

Riguardo alla facilità di utilizzo, le votazioni sono state abbastanza variegata (Figura 3.4). Sempre su una scala da 1 a 5, si nota un picco in corrispondenza del 3, ma non è da dimenticare che al secondo posto troviamo il 5, con veramente poche

valutazioni negative.

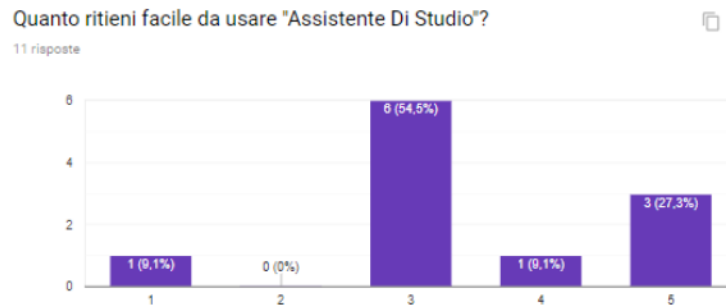


Figura 3.4: Valutazione della facilità di utilizzo

Una delle domande che ha fornito alcune delle risposte più importanti e significative è quella che chiedeva di dare una valutazione da 1 a 5 a vari aspetti dell'applicazione (Figura 3.5). Si può notare come i pareri siano molto differenti, ma come ci siano molte valutazioni negative, in particolare per l'aspetto grafico e per le notifiche. Anche la chiarezza dei servizi non ha un voto complessivamente positivo, e lo stesso accade per la gestione delle materie.

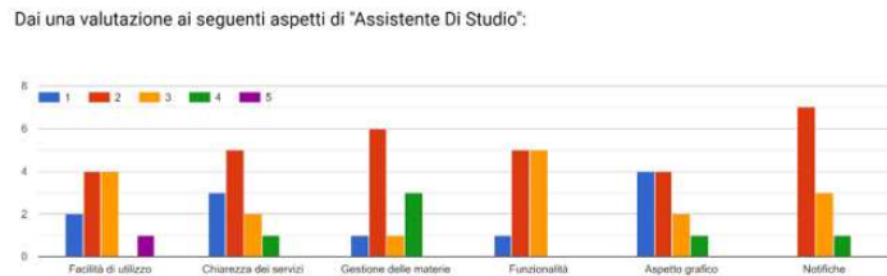


Figura 3.5: Valutazione di alcuni aspetti dell'applicazione

Altri dati indicativi sono che il 73% degli intervistati reputa *Assistente di Studio* uno strumento utile per lo studio, ma non tutti lo raccomanderebbero a un amico.

Le domande più importanti per la riprogettazione sono state, però, quelle che prevedevano una risposta aperta. È stato, così, possibile raccogliere dei pareri e dei pensieri che difficilmente sarebbero emersi tramite delle domande a risposta multipla.

3.1.2 Requisiti

Le varie risposte sono state analizzate e raggruppate per ambito, e ad ogni richiesta è stata associata una priorità. Questa è stata calcolata considerando il numero di utenti che l'hanno richiesta, la fattibilità di realizzazione e la congruenza con i risultati delle domande a risposta multipla, analizzati precedentemente. Si riportano in Tabella 3.1 tali richieste, con le relative priorità.

<i>Richiesta</i>	<i>Priorità</i>
Aggiungere un breve tutorial iniziale	Alta
Migliorare il calendario in quanto non mostra gli eventi	Media
L'app dovrebbe ricordare il nome della materia, non solo il colore	Alta
Il pulsante "gestisci materie" dovrebbe essere più in vista	Alta
Possibilità di interagire con altri utenti per scambio di flashcard	Bassa
Possibilità di specificare quali flashcard si vogliono ripassare	Media
Cambiare il nome per renderlo più breve ma più accattivante	Bassa

Tabella 3.1: Risultati delle domande aperte del questionario

Abbiamo, dunque, individuato i requisiti del nostro progetto, associati ad una priorità di realizzazione. Tali requisiti saranno, quindi, le linee guida per la riprogettazione dell'applicazione. Nella realizzazione di queste richieste, si cercherà, in ogni caso, di migliorare sempre l'aspetto grafico per renderlo più accattivante. Le valutazioni riguardanti la grafica, infatti, non sono state positive, e molti utenti si sono lamentati del fatto che l'estetica dell'applicazione era poco curata e poco gradevole. Questi appena elencati fanno parte dei requisiti funzionali, che rappresentano funzioni che il sistema deve realizzare.

Esistono, anche, i requisiti non funzionali, ovvero proprietà che il prodotto deve avere. In questo caso, rientrano in questa categoria soltanto due requisiti, ovvero quello di realizzare l'applicazione tramite *Android Studio* e quello di utilizzare come linguaggi di programmazione *Java* e *XML*.

3.2 Progettazione

Dopo aver stabilito i requisiti da rispettare, si può passare alla progettazione vera e propria. Questa fase si divide in diversi passaggi, ciascuno affrontato qui di seguito e ciascuno accompagnato da una documentazione caratteristica.

3.2.1 Card Sorting

La prima fase della progettazione prevede, ancora una volta, il coinvolgimento degli utenti finali, questa volta tramite la tecnica del "card sorting", per poter realizzare

l'architettura dell'applicazione. Tale tecnica è già stata spiegata in maniera approfondita nel capitolo precedente; quindi, in questa sezione, non verrà illustrato passo dopo passo. È stato chiesto a un gruppo di utenti appartenenti al target dell'applicazione, quindi sempre studenti, di collaborare e di ordinare le carte preparate. Tutto ciò è stato possibile grazie all'utilizzo di *Optimal Workshop* (Figura 3.6), ovvero una piattaforma online che offre svariati strumenti utili allo User Experience Designer in fase di progettazione, tra cui, appunto, il card sorting. In questa sezione del capitolo vengono analizzati i risultati di tale tecnica.

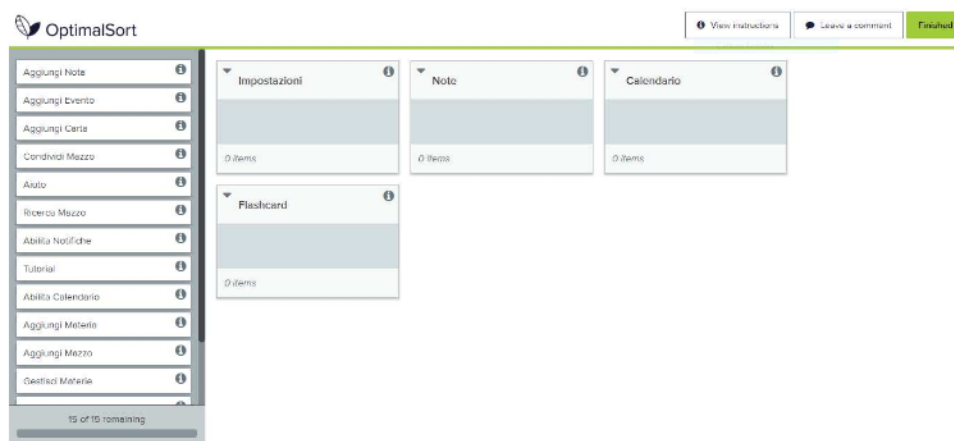


Figura 3.6: Card Sorting creato tramite *Optimal Workshop*

Analizzando i vari modi in cui le carte sono state disposte dagli utenti, sono emerse idee diverse, risultato di modelli mentali differenti. L'architettura finale dell'applicazione è stata costruita creando uno scheletro di base, formato da tutte le carte che sono state disposte dagli utenti sempre nella stessa posizione. Lo scheletro è stato, poi, integrato e completato dalle altre funzionalità, ovvero quelle corrispondenti alle carte che sono state posizionate dagli utenti in modi diversi. Si è cercato di trovare una soluzione per queste funzionalità che fosse un punto d'incontro tra le proposte dei diversi utenti, ovvero è stata creata una struttura la cui logica fosse quella maggiormente condivisa dai partecipanti alla sessione di card sorting.

Si elencano, di seguito, le modifiche che verranno effettuate all'architettura dell'applicazione:

- Si è deciso di cambiare il nome dell'applicazione. Il nuovo nome scelto è “*Study mate*”, che risulta più breve, intuitivo e accattivante.
- Si è deciso di inserire un tutorial alla prima apertura dell'applicazione, in quanto gli utenti hanno dichiarato di aver avuto problemi nel capire cosa fare al primo accesso all'applicazione stessa. Tale tutorial, successivamente, non scomparirà, ma l'utente dovrà avere la possibilità di rivederlo; per questo inseriremo un pulsante tramite il quale si potrà consultare nuovamente.
- Nel menù rimarrà presente la sezione “aiuto”, ma è stato rimosso il pulsante “gestisci materie”, in quanto l'aggiunta delle materie stesse verrà integrata nelle

altre sezioni dell'applicazione. In questo modo l'aggiunta delle materie risulterà più facile, e non si dovrà più passare per la sezione apposita, che risultava nascosta e poco raggiungibile.

- Dal menù è stata rimossa, anche, la sezione “impostazioni”, in quanto, dal momento che le impostazioni presenti si riferiscono esclusivamente alla sezione “calendario”, tali funzionalità verranno spostate direttamente nella sezione interessata.
- È stata proposta da più utenti la modifica del nome della sezione “calendario”; dal momento che il calendario è visibile solo se viene attivata la corrispondente funzionalità, si è ritenuto più corretto modificare l'etichetta in “eventi”.
- All'interno delle tre sezioni “eventi”, “flashcard” e “note”, l'utente, quando aggiunge materiale, seleziona la materia interessata. Nel caso in cui tale materia fosse nuova, la può aggiungere, associandone un colore, da qualunque punto ci si trovi, senza dover passare, quindi, per il pulsante “gestisci materie”.
- Si cercherà di migliorare il calendario per far sì che gli eventi futuri vengano segnalati.
- È emerso che gli utenti vorrebbero avere la possibilità di specificare quali flashcard ripassare. Alla luce di ciò, una volta mostrato il contenuto della carta, si potrà proseguire con la successiva, o, in alternativa, si potrà specificare che si vuole ripassare quella specifica carta. Si genererà, così, un sottomazzo, formato dalle sole carte che l'utente vorrà ripassare, al momento stesso, o in un momento successivo.
- Si agirà, anche, sulla grafica, cercando di armonizzare i colori, di inserire nuove icone più accattivanti e di rendere l'applicazione, in generale, più coinvolgente.
- Si aggiungerà, anche, la possibilità di creare un account e di condividere i propri mazzi. Se l'utente non si registra potrà usufruire solamente delle funzionalità in locale, ovvero potrà utilizzare il calendario, le note e crearsi i mazzi di flashcard. Se l'utente sceglie di registrarsi, allora potrà condividere i propri mazzi e potrà scaricare nella propria area dedicata alle flashcard mazzi creati da altri utenti, ricercandoli in un database in base alla materia.

Mappa dell'applicazione

Per comprendere meglio lo scheletro della nuova applicazione, è stata realizzata una mappa (Figura 3.7) di *Study mate*, creata tramite il servizio online *LucidChart*. L'obiettivo è quello di rendere chiara la struttura che deve avere il prodotto, prima di procedere con la progettazione di ogni schermata. Dall'immagine si nota che è presente una home page, collegata con la sezione “Aiuto” e con la sezione “Profilo”, se l'utente ha effettuato il login, o “Login”, se egli non ha ancora effettuato l'accesso. Dalla home page si diramano tre differenti pagine, ovvero “Eventi”, “Flashcard” e “Note”. In realtà la home non è altro che una di queste tre schermate, impostata, di default, come quella che viene visualizzata all'apertura. In particolare, si è scelto come pagina principale la schermata delle flashcard, in quanto i questionari hanno evidenziato il fatto che gli utenti ritengono quest'ultima funzionalità la più importante dell'applicazione.

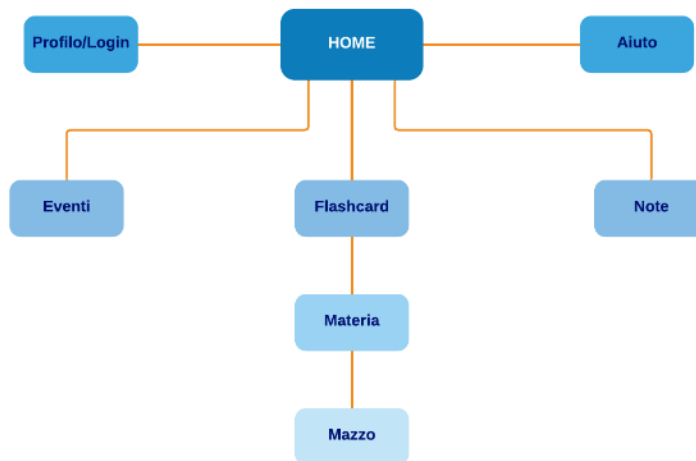


Figura 3.7: Mappa dell'applicazione

3.2.2 Wireframe

I wireframe sono un metodo per rappresentare la grafica di un prodotto digitale a livello strutturale. Essi sono essenziali per la progettazione di questi tipi di prodotti, in quanto permettono di costruire una prima rappresentazione visuale di un sito o di un'applicazione, senza entrare nel dettaglio del design vero e proprio. I wireframe permettono, infatti, di comunicare l'idea di progetto, l'architettura dell'informazione e la disposizione degli elementi nella schermata, senza fornire distrazioni dovute a colori, etichette e immagini. Essi si realizzano, infatti, in bianco e nero, utilizzando linee semplici e costrutti standard autoesplicativi, così da creare uno “schizzo” del prodotto, semplice e comprensibile.

Schermata “Flashcard”

Il wireframe in Figura 3.8(a) mostra la schermata della sezione “flashcard”, che coincide con la home page dell'applicazione. Questa “pagina” contiene le materie che sono state aggiunte e, sotto ciascuna, vengono visualizzati i titoli dei mazzi di flashcard relativi a ogni materia. L'utente può studiare le carte di tutta una determinata materia, le carte di un solo mazzo, oppure può selezionare la materia per visualizzare i dettagli dei suoi mazzi. Se la materia non è ancora stata salvata, si può procedere con la sua creazione premendo sul pulsante corrispondente.

Schermata “Materia”

Una volta selezionata la materia dalla pagina delle flashcard, si aprirà una nuova activity, illustrata nel wireframe in Figura 3.8(b), che mostra tutti i mazzi che

riguardano la materia stessa. Ogni mazzo è suddiviso in due sottomazzi, uno che contiene le carte da ripassare e uno con le carte che, invece, l'utente ritiene di sapere. Da qui l'utente può decidere se studiare l'intera materia, un mazzo specifico o, addirittura, un sottomazzo. Inoltre si possono aggiungere nuovi mazzi o selezionarne uno per visualizzarne le carte. Se si è eseguito il login, allora è presente anche il pulsante che permette di ricercare mazzi creati da altri utenti.

Schermata “Mazzo”

Il wireframe in Figura 3.8(c) mostra la pagina che viene visualizzata quando si seleziona uno specifico mazzo. Qui vengono mostrate le singole carte, con titolo e relativo contenuto. È anche presente il pulsante per aggiungere nuove carte.

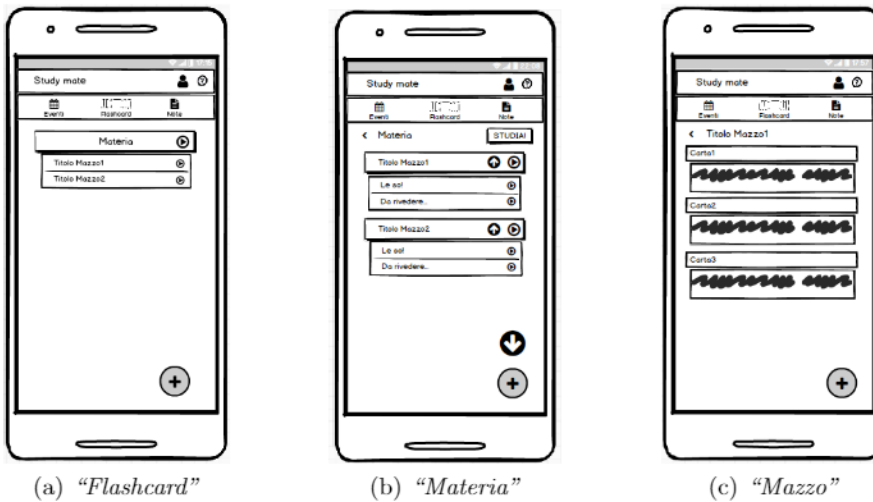


Figura 3.8: Wireframe che mostrano la schermata delle flashcard, quella della materia selezionata e quella del singolo mazzo

Schermata “Eventi”

Questa schermata è rappresentata dai wireframe in Figura 3.9(a) e (b). La prima immagine mostra la schermata con i soli eventi salvati, mentre la seconda mostra la stessa schermata quando viene attivata la funzione che permette di evidenziare il calendario. Entrambe presentano, infatti, il pulsante che permette di cambiare da una modalità all'altra e, ovviamente, il pulsante che permette di aggiungere nuovi eventi.

Schermata “Aggiungi evento”

Il wireframe in Figura 3.9(c) mostra la schermata che appare quando l'utente intende aggiungere un nuovo evento. Tale “pagina” sarà identica sia nel caso in cui l'opzione

per visualizzare il calendario sia attivata sia nel caso in cui essa sia disattivata, con l'unica differenza che, se il calendario è visibile, non sarà presente il campo “data”, in quanto questa viene selezionata direttamente sul calendario, prima di aggiungere l'evento stesso.

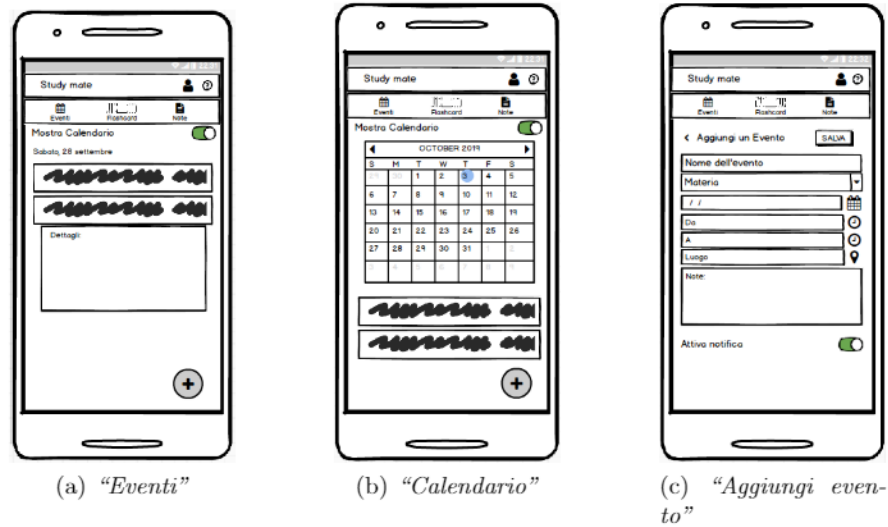


Figura 3.9: Wireframe che mostrano la schermata degli eventi, quella degli eventi nel calendario e quella per l'aggiunta di un nuovo evento

Schermata “Note”

Il wireframe in Figura 3.10(a) mostra la schermata che contiene le note. Gli utenti possono aggiungerne di nuove tramite il pulsante corrispondente.

Schermata “Profilo”

Il wireframe in Figura 3.10(b) mostra i dati personali dell'utente, se questo ha effettuato il login. Da questa schermata si possono modificare i dati stessi o eseguire il logout.

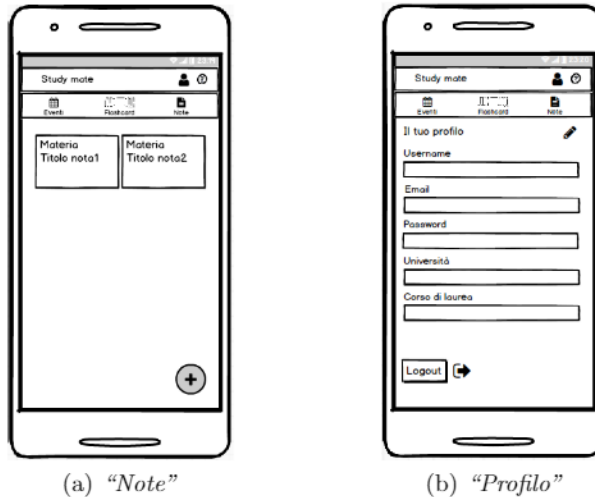


Figura 3.10: Wireframe che mostrano la schermata delle note e quella del profilo

3.2.3 Diagrammi UML

Per integrare e rendere più completa la documentazione relativa al progetto, soprattutto per ciò che riguarda la parte più squisitamente informatica, si è deciso di utilizzare i diagrammi UML. UML è un linguaggio di modellazione unificato, utilizzato per creare modelli che spieghino il funzionamento di prodotti, ma che siano anche comprensibili a tutti. Non ci sono delle regole precise che stabiliscano come creare un diagramma UML, ma esistono dei costrutti convenzionali e con significato univoco e comprensibile.

Si allega lo “use case diagram” (in italiano “diagramma dei casi d’uso”) che serve per descrivere le funzioni o i servizi offerti da un sistema, così come vengono percepiti dagli utenti che interagiscono con esso. Lo use case diagram relativo a *Study mate* è rappresentato in Figura 3.11. Qui si vedono le azioni che ogni utente può compiere, ovvero creare nuovi mazzi, creare nuove carte e studiare tali mazzi, creare delle note aggiungere degli eventi, modificare il proprio profilo e caricare o scaricare altri mazzi. Queste ultime due azioni sono possibili solo se prima si è effettuato il login.

Un’altra tipologia di diagramma UML molto utile è l’“activity diagram”, o diagramma delle attività, che spiega, in maniera dettagliata, come funziona ogni processo o attività di un prodotto. Nel caso specifico, non sono stati realizzati i diagrammi di ogni processo, in quanto molte possibili azioni sono piuttosto semplici. Si è deciso di realizzare solo il diagramma relativo alle azioni che si possono eseguire dalla sezione “flashcard”, mostrato in Figura 3.12.

Come si può notare da tale diagramma, ci sono delle azioni che sono state contraddistinte con un colore diverso; queste consistono, infatti, nelle operazioni che possono essere effettuate dall’utente solo se ha eseguito il login. Tutte le altre azioni si possono eseguire sia che l’utente si sia autenticato, sia che non l’abbia fatto.

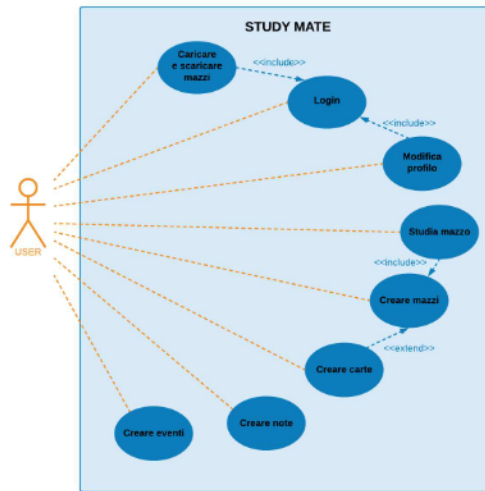


Figura 3.11: Use Case Diagram

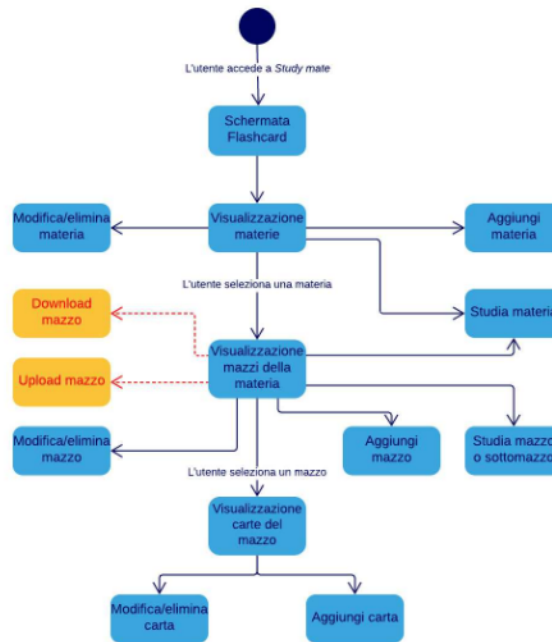


Figura 3.12: Activity Diagram

Implementazione della Componente Front-end e Manuale Utente

Questo capitolo è dedicato all'implementazione della componente front-end dell'applicazione Study mate. In particolare, sarà riportato e discusso il codice XML relativo alle interfacce utente. Seguirà il manuale utente, con lo scopo di illustrare nel dettaglio il funzionamento dell'app.

4.1 Implementazione della componente Java

In questo progetto, l'attenzione si è posta maggiormente sull'aspetto grafico, in quanto è l'ambito che, secondo gli utenti, risultava piuttosto carente ma che, invece, se curato, garantisce un'esperienza utente più completa e gradevole. Nonostante ciò, è comunque necessaria una componente Java, utile per garantire la navigazione tra le varie schermate dell'applicazione.

4.1.1 Implementazione della classe MainActivity.java

La classe MainActivity.java, mostrata nel Listato 4.1, istanzia la schermata principale. Una delle funzioni principali di questa classe è quella che crea un nuovo PageAdapter, che viene poi settato al viewPager. Inoltre viene creato un listener, settato al tabLayout; in questo modo, l'applicazione resta in ascolto per capire se l'utente si sposta lungo il tabLayout. In tal caso, il metodo setCurrentItem() converte la posizione del tab selezionato in un numero e, in base a questo numero, verrà richiamato il file XML giusto per lo specifico fragment.

```
1 public class MainActivity extends AppCompatActivity {
2
3     Toolbar toolbar;
4     TabLayout tabLayout;
5     ViewPager viewPager;
6     PageAdapter pageAdapter;
7     TabItem tabEventi;
8     TabItem tabFlashcard;
9     TabItem tabNote;
10
11     @Override
12     protected void onCreate(Bundle savedInstanceState) {
13         super.onCreate(savedInstanceState);
```



```

14     setContentView(R.layout.activity_main);
15
16     toolbar = findViewById(R.id.toolbar);
17     toolbar.setTitle(R.string.app_name);
18     setSupportActionBar(toolbar);
19
20     tabLayout = findViewById(R.id.tablayout);
21     tabEventi = findViewById(R.id.tabEventi);
22     tabFlashcard = findViewById(R.id.tabFlashcard);
23     tabNote = findViewById(R.id.tabNote);
24     viewPager = findViewById(R.id.viewPager);
25
26     pageAdapter = new PageAdapter(getSupportFragmentManager(), tabLayout.
        getTabCount());
27     viewPager.setAdapter(pageAdapter);
28
29
30     tabLayout.addOnTabSelectedListener(new TabLayout.OnTabSelectedListener
        () {
31         @Override
32         public void onTabSelected(TabLayout.Tab tab) {
33             viewPager.setCurrentItem(tab.getPosition());
34         }
35
36         @Override
37         public void onTabUnselected(TabLayout.Tab tab) {
38         }
39     }
40
41     @Override
42     public void onTabReselected(TabLayout.Tab tab) {
43     }
44 }
45 });
46 viewPager.addOnPageChangeListener(new TabLayout.
        TabLayoutOnPageChangeListener(tabLayout));
47 }
48
49 }

```

Listato 4.1: Implementazione della classe MainActivity

4.1.2 Implementazione dei fragment

Le tre sezioni principali dell'applicazione, ovvero “eventi”, “calendario” e “note”, non vengono implementate ognuna come un'activity separata. Dal momento che la parte superiore della schermata è sempre uguale, si è deciso di lasciarla invariata, e di implementare ciascuna sezione come un fragment, che viene richiamato quando l'utente preme il tab corrispondente.

Questi fragment hanno struttura analoga tra loro, per cui si riporta un solo codice (Listato 4.2). Ciascun fragment estende la classe `Fragment` già esistente; è, quindi, stato effettuato l'*override* del metodo `onCreateView()`, essenziale in quanto esso ritorna un `inflater` che richiama il file XML corretto in base al fragment chiamato, così dal poter costruire la schermata giusta. È, anche, presente il metodo `onCreateOptionsMenu()`, utilizzato per far sì che ogni schermata contenga, nella parte superiore, i pulsanti nella toolbar.

```

1     public class EventiFragment extends Fragment {
2
3         @Override
4         public View onCreateView(LayoutInflater inflater, ViewGroup container,

```

```

5         Bundle savedInstanceState) {
6             setHasOptionsMenu(true);
7             return inflater.inflate(R.layout.fragment_eventi, container, false);
8         }
9
10        @Override
11        public void onCreateOptionsMenu(Menu menu, MenuInflater inflater) {
12            inflater.inflate(R.menu.menu_toolbar, menu);
13        }
14    }
15 }

```

Listato 4.2: Implementazione del fragment EventiFragment

I fragment, per essere visualizzati all'interno del ViewPager, devono essere istanziati tramite il PageAdapter (Listato 4.3); questa è una classe che estende FragmentPagerAdapter, permettendo di istanziare il fragment corretto. Infatti, il metodo getItem(), di cui viene effettuato l'override, richiama il file Java del giusto fragment, basandosi sulla posizione del tab cliccato dall'utente.

```

1     public class PageAdapter extends FragmentPagerAdapter {
2
3         private int numOfTabs;
4
5         public PageAdapter(FragmentManager fm, int numOfTabs) {
6             super(fm);
7             this.numOfTabs = numOfTabs;
8         }
9
10        @Override
11        public Fragment getItem(int position) {
12            switch (position) {
13                case 0:
14                    return new EventiFragment();
15                case 1:
16                    return new FlashcardFragment();
17                case 2:
18                    return new NoteFragment();
19                default:
20                    return null;
21            }
22        }
23
24        @Override
25        public int getCount() {
26            return numOfTabs;
27        }
28 }

```

Listato 4.3: Implementazione dell'adapter PageAdapter

4.2 Implementazione della componente XML

In questa sezione vengono mostrati i codici XML che sono stati creati per implementare le interfacce grafiche dell'applicazione. Si specifica che, nelle schermate che verranno illustrate, sono presenti dei componenti statici che vengono utilizzati per mostrare il layout dell'applicazione e per aiutare l'utente che utilizza *Study mate* per la prima volta a comprenderne meglio il funzionamento. Ad esempio, infatti, quando l'utente apre per la prima volta la schermata "note", vedrà delle note d'esempio,

impostate di default, che servono per far comprendere come verranno visualizzate le note che saranno aggiunte. Ovviamente, questi componenti statici potranno essere eliminati dall'utente. Tutto quello che egli aggiungerà durante l'utilizzo dell'applicazione non sarà, però, un componente statico, creato tramite XML, ma verrà costruito in Java a runtime.

4.2.1 Implementazione del file `activity_main.xml`

Nel Listato 4.4 è mostrato il codice che prepara la grafica della pagina principale. Essenzialmente, questo file crea la toolbar, con i suoi pulsanti, prepara il tablayout, con i tre tab che l'utente utilizzerà per la navigazione, e istanzierà il contenitore che ospiterà i diversi fragment.

```

1  <androidx.appcompat.widget.Toolbar
2  android:id="@+id/toolbar"
3  android:layout_width="match_parent"
4  android:layout_height="wrap_content"
5  android:background="?attr/colorPrimary"
6  android:minHeight="?attr/actionBarSize"
7  android:theme="?attr/actionBarTheme"
8  app:popupTheme="@style/ThemeOverlay.AppCompat.Light"
9  app:theme="@style/ThemeOverlay.AppCompat.Dark"/>
10
11 <com.google.android.material.tabs.TabLayout
12  android:id="@+id/tablayout"
13  android:layout_width="match_parent"
14  android:layout_height="wrap_content"
15  android:background="@color/colorPrimary"
16  app:tabSelectedTextColor="@android:color/white"
17  app:tabTextColor="@android:color/black">
18
19  <com.google.android.material.tabs.TabItem
20  android:id="@+id/tabEventi"
21  android:layout_width="wrap_content"
22  android:layout_height="wrap_content"
23  android:text="EVENTI" />
24
25  <com.google.android.material.tabs.TabItem
26  android:id="@+id/tabFlashcard"
27  android:layout_width="wrap_content"
28  android:layout_height="wrap_content"
29  android:text="FLASHCARD" />
30
31  <com.google.android.material.tabs.TabItem
32  android:id="@+id/tabNote"
33  android:layout_width="wrap_content"
34  android:layout_height="wrap_content"
35  android:text="NOTE" />
36
37 </com.google.android.material.tabs.TabLayout>
38
39 <androidx.viewpager.widget.ViewPager
40  android:id="@+id/viewPager"
41  android:layout_width="match_parent"
42  android:layout_height="wrap_content" />

```

Listato 4.4: File `activity_main.xml`

4.2.2 Implementazione dei fragment

In questa sezione si illustrano i codici XML delle principali schermate dell'applicazione. Si ricorda che sono stati inseriti dei componenti statici per mostrare la

disposizione degli oggetti all'interno della schermata e per facilitare l'utilizzo dell'app stessa. Ad esempio, sono state inserite due materie, ovvero Analisi 1 e Fisica 1, con alcuni mazzi e alcune carte, così da rendere più immediato l'utilizzo di *Study mate*.

Si specifica che tutti i fragment che verranno mostrati nei seguenti listati contengono solo la parte più importante del codice. Infatti, tutti i fragment sono contenuti all'interno del *FrameLayout*, ma si è evitato di riportare ogni volta le stesse righe di codice.

Fragment “Eventi”

Il Listato 4.5 contiene il codice XML che implementa il layout del fragment “eventi”. Sono stati istanziati uno `switch`, che permette di cambiare la modalità di visualizzazione degli eventi, e la `CalendarView`, ovvero un componente di *Android* che permette la gestione del calendario. Inoltre, sono presenti le `CardView`, ovvero componenti che contengono orario e nome dell'evento. Nel codice sono stati istanziati due eventi d'esempio, utilizzando delle `TextView`. Infine, è presente il `FloatingActionButton`, che serve per aggiungere nuovi eventi.

```

1  <LinearLayout
2  android:layout_width="match_parent"
3  android:layout_height="wrap_content"
4  android:orientation="vertical"
5  android:layout_gravity="center_horizontal"
6  android:layout_margin="@dimen/margin_a">
7
8  <Switch
9  android:id="@+id/switch_id"
10 android:layout_width="match_parent"
11 android:layout_height="wrap_content"
12 android:layout_marginLeft="@dimen/margin_b"
13 android:layout_marginRight="@dimen/margin_b"
14 android:layout_weight="1"
15 android:checked="true"
16 android:text="Mostra calendario"
17 android:textSize="20dp"/>
18
19 <CalendarView
20 android:id="@+id/calendarView"
21 android:layout_width="wrap_content"
22 android:layout_height="wrap_content"
23 android:layout_margin="@dimen/margin_b"/>
24
25 <androidx.cardview.widget.CardView
26 app:cardBackgroundColor="@color/colorAccent"
27 app:cardCornerRadius="20dp"
28 app:cardUseCompatPadding="false"
29 app:cardPreventCornerOverlap="false"
30 android:layout_width="match_parent"
31 android:layout_height="50dp"
32 android:layout_marginRight="80dp"
33 android:layout_marginLeft="@dimen/margin_a"
34 android:layout_marginBottom="@dimen/margin_a">
35
36 <TextView
37 android:id="@+id/impegno_1"
38 android:layout_width="match_parent"
39 android:layout_height="match_parent"
40 android:layout_weight="1"
41 android:paddingStart="@dimen/margin_b"
42 android:text="9:30-12.30 Esame pratico"
43 android:gravity="center_vertical|left"

```

```

44         android:textColor="@color/colorPrimaryText"
45         android:textStyle="italic" />
46
47 </androidx.cardview.widget.CardView>
48
49 <androidx.cardview.widget.CardView
50     app:cardBackgroundColor="@color/colorAccent"
51     app:cardCornerRadius="20dp"
52     app:cardElevation="2dp"
53     app:cardUseCompatPadding="false"
54     app:cardPreventCornerOverlap="false"
55     android:layout_width="match_parent"
56     android:layout_height="50dp"
57     android:layout_marginRight="80dp"
58     android:layout_marginLeft="@dimen/margin_a"
59     android:layout_marginBottom="@dimen/margin_a">
60
61     <TextView
62         android:id="@+id/impegno_2"
63         android:layout_width="match_parent"
64         android:layout_height="match_parent"
65         android:layout_weight="1"
66         android:paddingStart="@dimen/margin_b"
67         android:text="14:30-16.30 Esame orale"
68         android:gravity="center_vertical|left"
69         android:textColor="@color/colorPrimaryText"
70         android:textStyle="italic" />
71
72 </androidx.cardview.widget.CardView>
73
74 </LinearLayout>
75
76 <com.google.android.material.floatingactionbutton.FloatingActionButton
77     android:id="@+id/fab"
78     android:layout_width="wrap_content"
79     android:layout_height="wrap_content"
80     android:layout_gravity="end|bottom"
81     android:layout_margin="16dp"
82     android:src="@drawable/ic_add" />

```

Listato 4.5: Implementazione del file XML di FragmentEventi

Fragment “Flashcard”

La schermata delle flashcard è costruita a partire dal file del Listato 4.6. Come si nota dal codice, la schermata è basata su un `LinearLayout` verticale che contiene le `CardView`. Ciascuna di esse contiene, a sua volta, un `LinearLayout`, questa volta orizzontale, con al proprio interno una `TextView`, con il nome della materia, e un `Button`, che permette di studiarla. Sotto ogni materia sono presenti altre `CardView`, che rappresentano i mazzi, che hanno la stessa struttura delle cardview precedenti, ciascuno con un `Button` per studiare lo specifico mazzo. Anche qui è presente il `FloatingActionButton`, questa volta per aggiungere nuove materie.

```

1 <LinearLayout
2     android:layout_width="match_parent"
3     android:layout_height="wrap_content"
4     android:orientation="vertical"
5     android:layout_margin="@dimen/margin_b">
6
7 <androidx.cardview.widget.CardView
8     android:id="@+id/cardview_titolo_materia"
9     android:layout_width="match_parent"
10    android:layout_height="75dp"

```

```

11     android:layout_marginTop="@dimen/margin_b"
12     android:layout_marginBottom="@dimen/margin_a"
13     app:cardBackgroundColor="@color/colorAnalisi1"
14     app:cardCornerRadius="10dp"
15     app:cardElevation="5dp"
16     app:cardPreventCornerOverlap="false"
17     app:cardUseCompatPadding="false">
18
19     <LinearLayout
20         android:layout_width="match_parent"
21         android:layout_height="match_parent"
22         android:layout_margin="@dimen/margin_a"
23         android:gravity="center_vertical"
24         android:orientation="horizontal">
25
26         <TextView
27             android:id="@+id/materia_1"
28             android:layout_width="match_parent"
29             android:layout_height="wrap_content"
30             android:layout_weight="1"
31             android:gravity="center_vertical|left"
32             android:paddingStart="@dimen/margin_a"
33             android:text="@string/titolo_materia_1_flashcard"
34             android:textColor="@color/colorPrimaryText"
35             android:textSize="@dimen/font_titolo_carta"
36             android:textStyle="italic" />
37
38         <Button
39             android:id="@+id/btn_materia"
40             style="@style/Widget.AppCompat.Button.Colored"
41             android:layout_width="@dimen/button_icon"
42             android:layout_height="wrap_content"
43             android:layout_weight="1"
44             android:background="@drawable/ic_play_circle" />
45
46     </LinearLayout>
47
48 </androidx.cardview.widget.CardView>
49
50 <androidx.cardview.widget.CardView
51     android:id="@+id/cardview_titolo_mazzo"
52     android:layout_width="match_parent"
53     android:layout_height="50dp"
54     android:layout_marginLeft="@dimen/margin_a"
55     android:layout_marginRight="@dimen/margin_a"
56     android:layout_marginBottom="@dimen/margin_a"
57     app:cardBackgroundColor="@color/colorAnalisi1"
58     app:cardCornerRadius="8dp"
59     app:cardElevation="2dp"
60     app:cardPreventCornerOverlap="false"
61     app:cardUseCompatPadding="false">
62
63     <LinearLayout
64         android:layout_width="match_parent"
65         android:layout_height="match_parent"
66         android:layout_margin="@dimen/margin_a"
67         android:gravity="center_vertical"
68         android:orientation="horizontal">
69
70         <TextView
71             android:id="@+id/mazzo_1"
72             android:layout_width="match_parent"
73             android:layout_height="wrap_content"
74             android:layout_weight="1"
75             android:gravity="center_vertical|left"
76             android:paddingStart="@dimen/margin_a"
77             android:text="@string/titolo_mazzo_1_flashcard"
78             android:textColor="@color/colorIcons"
79             android:textSize="16dp"

```



```
81         android:textStyle="italic" />
82
83     <Button
84         android:id="@+id/btn_mazzo_1"
85         style="@style/Widget.AppCompat.Button.Colored"
86         android:layout_width="@android:dimen/notification_large_icon_height"
87         android:layout_height="wrap_content"
88         android:layout_weight="1"
89         android:background="@drawable/ic_play_circle" />
90
91 </LinearLayout>
92
93 </androidx.cardview.widget.CardView>
94
95 <androidx.cardview.widget.CardView
96     android:layout_width="match_parent"
97     android:layout_height="50dp"
98     android:layout_marginLeft="@dimen/margin_a"
99     android:layout_marginRight="@dimen/margin_a"
100    android:layout_marginBottom="@dimen/margin_a"
101    app:cardBackgroundColor="@color/colorAnalisi1"
102    app:cardCornerRadius="8dp"
103    app:cardElevation="2dp"
104    app:cardPreventCornerOverlap="false"
105    app:cardUseCompatPadding="false">
106
107     <LinearLayout
108         android:layout_width="match_parent"
109         android:layout_height="match_parent"
110         android:layout_margin="@dimen/margin_a"
111         android:gravity="center_vertical"
112         android:orientation="horizontal">
113
114         <TextView
115             android:id="@+id/mazzo_1_2"
116             android:layout_width="match_parent"
117             android:layout_height="wrap_content"
118             android:layout_weight="1"
119             android:gravity="center_vertical|left"
120             android:paddingStart="@dimen/margin_a"
121             android:text="@string/titolo_mazzo_1_2_flashcard"
122             android:textColor="@color/colorIcons"
123             android:textSize="16dp"
124             android:textStyle="italic" />
125
126         <Button
127             android:id="@+id/btn_mazzo_1_2"
128             style="@style/Widget.AppCompat.Button.Colored"
129             android:layout_width="@android:dimen/notification_large_icon_height"
130             android:layout_height="wrap_content"
131             android:layout_weight="1"
132             android:background="@drawable/ic_play_circle" />
133
134     </LinearLayout>
135
136 </androidx.cardview.widget.CardView>
137
138 </androidx.cardview.widget.CardView>
139
140 <androidx.cardview.widget.CardView
141     android:id="@+id/cardview_titolo_materia_2"
142     android:layout_width="match_parent"
143     android:layout_height="75dp"
144     android:layout_marginTop="@dimen/margin_b"
145     android:layout_marginBottom="@dimen/margin_a"
146     app:cardBackgroundColor="@color/colorFisical"
147     app:cardCornerRadius="10dp"
148     app:cardElevation="5dp"
149     app:cardPreventCornerOverlap="false"
150     app:cardUseCompatPadding="false">
```



```

151
152 <LinearLayout
153     android:layout_width="match_parent"
154     android:layout_height="match_parent"
155     android:layout_margin="@dimen/margin_a"
156     android:gravity="center_vertical"
157     android:orientation="horizontal">
158
159     <TextView
160         android:id="@+id/materia"
161         android:layout_width="match_parent"
162         android:layout_height="wrap_content"
163         android:layout_weight="1"
164         android:gravity="center_vertical|left"
165         android:paddingStart="@dimen/margin_a"
166         android:text="@string/titolo_materia_2_flashcard"
167         android:textColor="@color/colorPrimaryText"
168         android:textSize="@dimen/font_titolo_carta"
169         android:textStyle="italic" />
170
171     <Button
172         android:id="@+id/btn_materia_2"
173         style="@style/Widget.AppCompat.Button.Colored"
174         android:layout_width="@dimen/button_icon"
175         android:layout_height="wrap_content"
176         android:layout_weight="1"
177         android:background="@drawable/ic_play_circle" />
178
179 </LinearLayout>
180
181
182 </androidx.cardview.widget.CardView>
183
184 <androidx.cardview.widget.CardView
185     android:id="@+id/cardview_titolo_mazzo_2"
186     android:layout_width="match_parent"
187     android:layout_height="50dp"
188     android:layout_marginLeft="@dimen/margin_a"
189     android:layout_marginRight="@dimen/margin_a"
190     android:layout_marginBottom="@dimen/margin_a"
191     app:cardBackgroundColor="@color/colorFisical"
192     app:cardCornerRadius="8dp"
193     app:cardElevation="2dp"
194     app:cardPreventCornerOverlap="false"
195     app:cardUseCompatPadding="false">
196
197     <LinearLayout
198         android:layout_width="match_parent"
199         android:layout_height="match_parent"
200         android:layout_margin="@dimen/margin_a"
201         android:gravity="center_vertical"
202         android:orientation="horizontal">
203
204         <TextView
205             android:id="@+id/mazzo_2"
206             android:layout_width="match_parent"
207             android:layout_height="wrap_content"
208             android:layout_weight="1"
209             android:gravity="center_vertical|left"
210             android:paddingStart="@dimen/margin_a"
211             android:text="@string/titolo_mazzo_2_flashcard"
212             android:textColor="@color/colorIcons"
213             android:textSize="16dp"
214             android:textStyle="italic" />
215
216         <Button
217             android:id="@+id/btn_mazzo_2"
218             style="@style/Widget.AppCompat.Button.Colored"
219             android:layout_width="@android:dimen/notification_large_icon_height"
220             android:layout_height="wrap_content"

```

```

221         android:layout_weight="1"
222         android:background="@drawable/ic_play_circle" />
223
224     </LinearLayout>
225
226
227 </androidx.cardview.widget.CardView>
228 <androidx.cardview.widget.CardView
229     android:layout_width="match_parent"
230     android:layout_height="50dp"
231     android:layout_marginLeft="@dimen/margin_a"
232     android:layout_marginRight="@dimen/margin_a"
233     android:layout_marginBottom="@dimen/margin_a"
234     app:cardBackgroundColor="@color/colorFisical"
235     app:cardCornerRadius="8dp"
236     app:cardElevation="2dp"
237     app:cardPreventCornerOverlap="false"
238     app:cardUseCompatPadding="false">
239
240     <LinearLayout
241         android:layout_width="match_parent"
242         android:layout_height="match_parent"
243         android:layout_margin="@dimen/margin_a"
244         android:gravity="center_vertical"
245         android:orientation="horizontal">
246
247         <TextView
248             android:id="@+id/mazzo_2_2"
249             android:layout_width="match_parent"
250             android:layout_height="wrap_content"
251             android:layout_weight="1"
252             android:gravity="center_vertical|left"
253             android:paddingStart="@dimen/margin_a"
254             android:text="@string/titolo_mazzo_2_2_flashcard"
255             android:textColor="@color/colorIcons"
256             android:textSize="16dp"
257             android:textStyle="italic" />
258
259         <Button
260             android:id="@+id/btn_mazzo_2_2"
261             style="@style/Widget.AppCompat.Button.Colored"
262             android:layout_width="@android:dimen/notification_large_icon_height"
263             android:layout_height="wrap_content"
264             android:layout_weight="1"
265             android:background="@drawable/ic_play_circle" />
266
267     </LinearLayout>
268
269 </androidx.cardview.widget.CardView>
270 </LinearLayout>
271
272
273
274
275 <com.google.android.material.floatingactionbutton.FloatingActionButton
276     android:id="@+id/fab"
277     android:layout_width="wrap_content"
278     android:layout_height="wrap_content"
279     android:layout_gravity="end|bottom"
280     android:layout_margin="16dp"
281     android:src="@drawable/ic_add" />

```

Listato 4.6: Implementazione del file XML di FragmentFlashcard

Fragment “Materia”

Questo fragment (Listato 4.7) viene richiamato quando l’utente preme su una `CardView` che contiene il nome di una materia. La struttura della schermata è

rappresentata da un `LinearLayout` verticale. In alto è presente un `LinearLayout`, questa volta orizzontale, che contiene un `ImageView`, una `TextView`, ovvero l'etichetta della materia selezionata, e un `Button`, che permette di studiare la materia. Ogni mazzo della materia selezionata è rappresentato tramite una `CardView`, formata da un altro `LinearLayout` orizzontale, che contiene il titolo del mazzo, sotto forma di `TextView`, e due `Button`; il primo serve per caricare online il mazzo, il secondo per studiarlo. Infine, sono presenti due `FloatingActionButton`. Quello inferiore serve per aggiungere nuovi mazzi; quello superiore serve per scaricare mazzi creati da altri utenti.

```

1  <LinearLayout
2  android:layout_width="match_parent"
3  android:layout_height="wrap_content"
4  android:orientation="vertical"
5  android:layout_margin="@dimen/margin_b">
6
7  <LinearLayout
8  android:layout_width="match_parent"
9  android:layout_height="match_parent"
10 android:orientation="horizontal"
11 android:layout_gravity="center_horizontal">
12
13  <ImageView
14  android:id="@+id/imageButton"
15  style="@android:style/Widget.Holo.Light.ImageButton"
16  android:layout_width="@dimen/margin_a"
17  android:layout_height="@dimen/card_small_height"
18  android:layout_weight="1"
19  android:background="@drawable/ic_keyboard_arrow_left_black_24dp"
20  app:srcCompat="@drawable/ic_keyboard_arrow_left_black_24dp" />
21
22  <TextView
23  android:id="@+id/textView"
24  android:layout_width="wrap_content"
25  android:layout_height="match_parent"
26  android:layout_weight="1"
27  android:gravity="center_vertical|left"
28  android:textColor="@color/colorPrimaryDark"
29  android:text="@string/titolo_materia_2_flashcard"
30  android:textSize="@dimen/font_titolo_carta" />
31
32  <Button
33  android:id="@+id/button"
34  style="@style/Widget.AppCompat.Button.Colored"
35  android:layout_width="@dimen/card_small_height"
36  android:layout_height="wrap_content"
37  android:layout_weight="1"
38  android:text="@string/btn_studia" />
39 </LinearLayout>
40
41  <androidx.cardview.widget.CardView
42  android:layout_width="match_parent"
43  android:layout_height="75dp"
44  android:layout_marginTop="@dimen/margin_b"
45  android:layout_marginBottom="@dimen/margin_a"
46  app:cardBackgroundColor="@color/colorAccent"
47  app:cardCornerRadius="10dp"
48  app:cardElevation="5dp"
49  app:cardPreventCornerOverlap="false"
50  app:cardUseCompatPadding="false">
51
52  <LinearLayout
53  android:layout_width="match_parent"
54  android:layout_height="match_parent"
55  android:layout_margin="@dimen/margin_a"
56  android:gravity="center_vertical"

```

```

57         android:orientation="horizontal">
58
59         <TextView
60             android:id="@+id/materia_1"
61             android:layout_width="match_parent"
62             android:layout_height="wrap_content"
63             android:layout_weight="1"
64             android:gravity="center_vertical|left"
65             android:paddingStart="@dimen/margin_a"
66             android:text="@string/titolo_mazzo_2_flashcard"
67             android:textColor="@color/colorPrimaryText"
68             android:textSize="@dimen/font_titolo_carta"
69             android:textStyle="italic" />
70
71         <Button
72             android:id="@+id/btn_materia_carica"
73             style="@style/Widget.AppCompat.Button.Colored"
74             android:layout_width="@dimen/btn_width"
75             android:layout_height="wrap_content"
76             android:layout_weight="1"
77             android:background="@drawable/ic_file_upload_black_24dp" />
78
79         <Button
80             android:id="@+id/btn_materia"
81             style="@style/Widget.AppCompat.Button.Colored"
82             android:layout_width="@dimen/btn_width"
83             android:layout_height="wrap_content"
84             android:layout_weight="1"
85             android:background="@drawable/ic_play_circle" />
86
87     </LinearLayout>
88
89 </androidx.cardview.widget.CardView>
90
91 <androidx.cardview.widget.CardView
92     android:id="@+id/cardview_titolo_mazzo"
93     android:layout_width="match_parent"
94     android:layout_height="50dp"
95     android:layout_marginLeft="@dimen/margin_a"
96     android:layout_marginRight="@dimen/margin_a"
97     android:layout_marginBottom="@dimen/margin_a"
98     app:cardBackgroundColor="@color/colorAccent"
99     app:cardCornerRadius="8dp"
100    app:cardElevation="2dp"
101    app:cardPreventCornerOverlap="false"
102    app:cardUseCompatPadding="false">
103
104     <LinearLayout
105         android:layout_width="match_parent"
106         android:layout_height="match_parent"
107         android:layout_margin="@dimen/margin_a"
108         android:gravity="center_vertical"
109         android:orientation="horizontal">
110
111         <TextView
112             android:layout_width="match_parent"
113             android:layout_height="wrap_content"
114             android:layout_weight="1"
115             android:gravity="center_vertical|left"
116             android:paddingStart="@dimen/margin_a"
117             android:text="@string/da_rivedere"
118             android:textColor="@color/colorIcons"
119             android:textSize="16dp"
120             android:textStyle="italic" />
121
122         <Button
123             style="@style/Widget.AppCompat.Button.Colored"
124             android:layout_width="@android:dimen/notification_large_icon_height"
125

```

```

126         android:layout_height="wrap_content"
127         android:layout_weight="1"
128         android:background="@drawable/ic_play_circle" />
129
130     </LinearLayout>
131
132
133 </androidx.cardview.widget.CardView>
134
135 <androidx.cardview.widget.CardView
136     android:id="@+id/cardview_titolo_materia"
137     android:layout_width="match_parent"
138     android:layout_height="75dp"
139     android:layout_marginTop="@dimen/margin_b"
140     android:layout_marginBottom="@dimen/margin_a"
141     app:cardBackgroundColor="@color/colorAccent"
142     app:cardCornerRadius="10dp"
143     app:cardElevation="5dp"
144     app:cardPreventCornerOverlap="false"
145     app:cardUseCompatPadding="false">
146
147     <LinearLayout
148         android:layout_width="match_parent"
149         android:layout_height="match_parent"
150         android:layout_margin="@dimen/margin_a"
151         android:gravity="center_vertical"
152         android:orientation="horizontal">
153
154         <TextView
155             android:id="@+id/materia_2"
156             android:layout_width="match_parent"
157             android:layout_height="wrap_content"
158             android:layout_weight="1"
159             android:gravity="center_vertical|left"
160             android:paddingStart="@dimen/margin_a"
161             android:text="@string/titolo_mazzo_2_2_flashcard"
162             android:textColor="@color/colorPrimaryText"
163             android:textSize="@dimen/font_titolo_carta"
164             android:textStyle="italic" />
165
166         <Button
167             android:id="@+id/btn_materia_carica_2"
168             style="@style/Widget.AppCompat.Button.Colored"
169             android:layout_width="@dimen/btn_width"
170             android:layout_height="wrap_content"
171             android:layout_weight="1"
172             android:background="@drawable/ic_file_upload_black_24dp" />
173
174         <Button
175             android:id="@+id/btn_materia"
176             style="@style/Widget.AppCompat.Button.Colored"
177             android:layout_width="@dimen/btn_width"
178             android:layout_height="wrap_content"
179             android:layout_weight="1"
180             android:background="@drawable/ic_play_circle" />
181
182     </LinearLayout>
183
184 </androidx.cardview.widget.CardView>
185
186 <androidx.cardview.widget.CardView
187     android:id="@+id/cardview_titolo_mazzo"
188     android:layout_width="match_parent"
189     android:layout_height="50dp"
190     android:layout_marginLeft="@dimen/margin_a"
191     android:layout_marginRight="@dimen/margin_a"
192     android:layout_marginBottom="@dimen/margin_a"
193     app:cardBackgroundColor="@color/colorAccent"
194     app:cardCornerRadius="8dp"

```

```

196     app:cardElevation="2dp"
197     app:cardPreventCornerOverlap="false"
198     app:cardUseCompatPadding="false">
199
200     <LinearLayout
201         android:layout_width="match_parent"
202         android:layout_height="match_parent"
203         android:layout_margin="@dimen/margin_a"
204         android:gravity="center_vertical"
205         android:orientation="horizontal">
206
207         <TextView
208             android:id="@+id/mazzo_1"
209             android:layout_width="match_parent"
210             android:layout_height="wrap_content"
211             android:layout_weight="1"
212             android:gravity="center_vertical|left"
213             android:paddingStart="@dimen/margin_a"
214             android:text="@string/da_rivedere"
215             android:textColor="@color/colorIcons"
216             android:textSize="16dp"
217             android:textStyle="italic" />
218
219         <Button
220             android:id="@+id/btn_mazzo_1"
221             style="@style/Widget.AppCompat.Button.Colored"
222             android:layout_width="@android:dimen/notification_large_icon_height"
223
224             android:layout_height="wrap_content"
225             android:layout_weight="1"
226             android:background="@drawable/ic_play_circle" />
227     </LinearLayout>
228
229 </androidx.cardview.widget.CardView>
230
231 </LinearLayout>
232
233 <com.google.android.material.floatingactionbutton.FloatingActionButton
234     android:id="@+id/download"
235     android:layout_width="wrap_content"
236     android:layout_height="wrap_content"
237     android:layout_gravity="end|bottom"
238     android:layout_marginRight="16dp"
239     android:layout_marginBottom="90dp"
240     android:src="@drawable/ic_file_download_black_24dp" />
241
242 <com.google.android.material.floatingactionbutton.FloatingActionButton
243     android:id="@+id/fab"
244     android:layout_width="wrap_content"
245     android:layout_height="wrap_content"
246     android:layout_gravity="end|bottom"
247     android:layout_margin="16dp"
248     android:src="@drawable/ic_add" />
249

```

Listato 4.7: Implementazione del file XML di FragmentMateria

Fragment “Mazzo”

Il fragment del Listato 4.8 viene richiamato quando l’utente preme su una **CardView** contenente il nome di un mazzo. La struttura della schermata è sempre un **LinearLayout** verticale. Anche qui è presente un **LinearLayout** orizzontale, che contiene un **ImageView** e una **TextView**, ovvero l’etichetta del mazzo selezionato. Ogni carta di tale mazzo è rappresentata come una **CardView**, contenente un **LinearLayout** verticale, ma centrato, con al proprio interno due **TextView**, una con il titolo della carta,

e una con il suo contenuto. Infine, è sempre presente un `FloatingActionButton`, per aggiungere nuove carte.

```

1  <LinearLayout
2  android:layout_width="match_parent"
3  android:layout_height="wrap_content"
4  android:orientation="vertical"
5  android:layout_margin="@dimen/margin_b">
6
7  <LinearLayout
8  android:layout_width="match_parent"
9  android:layout_height="match_parent"
10 android:orientation="horizontal"
11 android:layout_gravity="center_horizontal">
12
13  <ImageView
14  android:id="@+id/imageButton"
15  style="@android:style/Widget.Holo.Light.ImageButton"
16  android:layout_width="@dimen/button_icon"
17  android:layout_height="@dimen/card_small_height"
18  android:layout_weight="1"
19  android:background="@drawable/ic_keyboard_arrow_left_black_24dp"
20  app:srcCompat="@drawable/ic_keyboard_arrow_left_black_24dp" />
21
22  <TextView
23  android:id="@+id/textView"
24  android:layout_width="match_parent"
25  android:layout_height="match_parent"
26  android:layout_weight="1"
27  android:gravity="center_vertical|left"
28  android:textColor="@color/colorPrimaryDark"
29  android:text="@string/titolo_mazzo_2_2_flashcard"
30  android:textSize="@dimen/font_titolo_carta" />
31
32 </LinearLayout>
33
34 <androidx.cardview.widget.CardView
35 android:layout_width="match_parent"
36 android:layout_height="wrap_content"
37 android:layout_marginTop="@dimen/margin_a"
38 android:layout_marginBottom="@dimen/margin_a"
39 app:cardBackgroundColor="@color/colorIcons"
40 app:cardCornerRadius="10dp"
41 app:cardElevation="5dp"
42 app:cardPreventCornerOverlap="false"
43 app:cardUseCompatPadding="false">
44
45  <LinearLayout
46  android:layout_width="match_parent"
47  android:layout_height="match_parent"
48  android:layout_margin="@dimen/margin_a"
49  android:gravity="center_vertical"
50  android:orientation="vertical">
51
52  <TextView
53  android:id="@+id/domanda_1"
54  android:layout_width="match_parent"
55  android:layout_height="wrap_content"
56  android:layout_weight="1"
57  android:gravity="center_vertical|left"
58  android:paddingStart="@dimen/margin_a"
59  android:text="@string/titolo_domanda_1"
60  android:textColor="@color/colorPrimaryText"
61  android:textSize="@dimen/font_titolo_carta"
62  android:backgroundTint="@color/colorPrimary"
63  android:textStyle="italic" />
64
65  <TextView
66  android:id="@+id/testo_domanda_1"

```



```

67         android:layout_width="match_parent"
68         android:layout_height="wrap_content"
69         android:layout_weight="1"
70         android:gravity="center_vertical|left"
71         android:paddingStart="@dimen/margin_a"
72         android:text="@string/testo_domanda_1"
73         android:textColor="@color/colorSecondaryText"
74         android:textSize="18sp"
75         android:textStyle="italic" />
76
77     </LinearLayout>
78
79 </androidx.cardview.widget.CardView>
80
81 <androidx.cardview.widget.CardView
82     android:id="@+id/cardview_titolo_materia"
83     android:layout_width="match_parent"
84     android:layout_height="wrap_content"
85     android:layout_marginTop="@dimen/margin_a"
86     android:layout_marginBottom="@dimen/margin_a"
87     app:cardBackgroundColor="@color/colorIcons"
88     app:cardCornerRadius="10dp"
89     app:cardElevation="5dp"
90     app:cardPreventCornerOverlap="false"
91     app:cardUseCompatPadding="false">
92
93     <LinearLayout
94         android:layout_width="match_parent"
95         android:layout_height="match_parent"
96         android:layout_margin="@dimen/margin_a"
97         android:gravity="center_vertical"
98         android:orientation="vertical">
99
100         <TextView
101             android:id="@+id/domanda_2"
102             android:layout_width="match_parent"
103             android:layout_height="wrap_content"
104             android:layout_weight="1"
105             android:gravity="center_vertical|left"
106             android:paddingStart="@dimen/margin_a"
107             android:text="@string/titolo_domanda_2"
108             android:textColor="@color/colorPrimaryText"
109             android:textSize="@dimen/font_titolo_carta"
110             android:backgroundTint="@color/colorPrimary"
111             android:textStyle="italic" />
112
113         <TextView
114             android:id="@+id/testo_domanda_2"
115             android:layout_width="match_parent"
116             android:layout_height="wrap_content"
117             android:layout_weight="1"
118             android:gravity="center_vertical|left"
119             android:paddingStart="@dimen/margin_a"
120             android:text="@string/testo_domanda_2"
121             android:textColor="@color/colorSecondaryText"
122             android:textSize="18sp"
123             android:textStyle="italic" />
124
125     </LinearLayout>
126
127 </androidx.cardview.widget.CardView>
128
129 </LinearLayout>
130
131 </LinearLayout>
132
133 <com.google.android.material.floatingactionbutton.FloatingActionButton
134     android:id="@+id/fab"

```

```

137     android:layout_width="wrap_content"
138     android:layout_height="wrap_content"
139     android:layout_gravity="end|bottom"
140     android:layout_margin="16dp"
141     android:src="@drawable/ic_add" />

```

Listato 4.8: Implementazione del file XML di FragmentMazzo

Fragment “Note”

Nel Listato 4.9 è mostrato il codice che viene richiamato quando l’utente preme sul tab che contiene l’etichetta “Note”. Questa schermata si basa su un `LinearLayout` orizzontale, contenente due `CardView`, entrambe con la stessa struttura. Queste sono formate da un `LinearLayout` verticale, con due `TextView`, una per il nome della materia e una per la descrizione. Infine, è sempre presente un `FloatingActionButton`, per aggiungere nuove note.

```

1     <LinearLayout
2         android:layout_width="match_parent"
3         android:layout_height="match_parent"
4         android:orientation="horizontal"
5         android:layout_margin="@dimen/margin_b">
6
7     <androidx.cardview.widget.CardView
8         app:cardBackgroundColor="@color/colorAccent"
9         app:cardCornerRadius="3dp"
10        app:cardUseCompatPadding="false"
11        app:cardPreventCornerOverlap="false"
12        android:layout_width="match_parent"
13        android:layout_height="120dp"
14        android:layout_weight="1"
15        android:layout_marginRight="@dimen/margin_a"
16        android:layout_marginLeft="@dimen/margin_a"
17        android:layout_marginBottom="@dimen/margin_a">
18
19     <LinearLayout
20         android:layout_width="match_parent"
21         android:layout_height="match_parent"
22         android:gravity="center_vertical"
23         android:orientation="vertical"
24         android:layout_margin="@dimen/margin_a">
25
26     <TextView
27         android:id="@+id/titolo_nota_1"
28         android:layout_width="match_parent"
29         android:layout_height="wrap_content"
30         android:layout_weight="1"
31         android:paddingStart="@dimen/margin_a"
32         android:text="@string/nota_1"
33         android:gravity="center_vertical|left"
34         android:textColor="@color/colorPrimaryText"
35         android:textSize="18dp"
36         android:textStyle="italic" />
37
38     <TextView
39         android:id="@+id/input_nota_1"
40         android:layout_width="match_parent"
41         android:layout_height="wrap_content"
42         android:layout_weight="1"
43         android:paddingStart="@dimen/margin_a"
44         android:text="@string/questo_il_testo_della_nota_1"
45         android:textColor="@color/colorPrimaryText"
46         android:textSize="14dp"
47         android:inputType="text|textMultiLine"

```

```

48         android:gravity="top"/>
49     </LinearLayout>
50
51
52
53 </androidx.cardview.widget.CardView>
54
55
56 <androidx.cardview.widget.CardView
57     app:cardBackgroundColor="@color/colorAccent"
58     app:cardCornerRadius="3dp"
59     app:cardUseCompatPadding="false"
60     app:cardPreventCornerOverlap="false"
61     android:layout_width="match_parent"
62     android:layout_height="wrap_content"
63     android:layout_weight="1"
64     android:layout_marginRight="@dimen/margin_a"
65     android:layout_marginLeft="@dimen/margin_a"
66     android:layout_marginBottom="@dimen/margin_a">
67
68     <LinearLayout
69         android:layout_width="match_parent"
70         android:layout_height="match_parent"
71         android:gravity="center_vertical"
72         android:orientation="vertical"
73         android:layout_margin="@dimen/margin_a">
74
75         <TextView
76             android:id="@+id/titolo_nota_2"
77             android:layout_width="match_parent"
78             android:layout_height="wrap_content"
79             android:layout_weight="1"
80             android:paddingStart="@dimen/margin_a"
81             android:text="@string/nota_2"
82             android:gravity="center_vertical|left"
83             android:textColor="@color/colorPrimaryText"
84             android:textSize="18dp"
85             android:textStyle="italic" />
86
87         <TextView
88             android:id="@+id/input_nota_2"
89             android:layout_width="match_parent"
90             android:layout_height="wrap_content"
91             android:layout_weight="1"
92             android:paddingStart="@dimen/margin_a"
93             android:text="@string/questo_il_testo_della_nota_2"
94             android:gravity="center_vertical|left"
95             android:textColor="@color/colorPrimaryText"
96             android:textSize="14dp"
97             android:textStyle="italic" />
98
99     </LinearLayout>
100
101
102 </androidx.cardview.widget.CardView>
103
104 </LinearLayout>
105
106 <com.google.android.material.floatingactionbutton.FloatingActionButton
107     android:id="@+id/fab"
108     android:layout_width="wrap_content"
109     android:layout_height="wrap_content"
110     android:layout_gravity="end|bottom"
111     android:layout_margin="16dp"
112     android:src="@drawable/ic_add" />

```

Listato 4.9: Implementazione del file XML di FragmentNote

Fragment “Profilo”

Il fragment del Listato 4.10 viene richiamato quando l’utente preme sul tasto, presente nella toolbar, per accedere ai propri dati personali. La schermata di base su un `LinearLayout` verticale, composto da varie `TextView` e da varie `EditText`: le prime contengono i dati che vengono richiesti mentre le seconde permettono di modificarli. Infine, in un `LinearLayout` orizzontale, sono presenti due pulsanti, uno per tornare indietro e uno per aggiornare i dati.

```

1  <LinearLayout
2  android:orientation="vertical"
3  android:layout_width="match_parent"
4  android:layout_height="match_parent">
5  <TextView
6      android:id="@+id/register_text_profile"
7      android:layout_width="wrap_content"
8      android:layout_height="wrap_content"
9      android:layout_marginTop="40dp"
10     android:text="Aggiorna i tuoi dati"
11     android:textSize="25dp"
12     android:textStyle="bold"
13     android:layout_gravity="center"/>
14  <TextView
15     android:id="@+id/username_text_profile"
16     android:layout_width="wrap_content"
17     android:layout_height="wrap_content"
18     android:layout_marginLeft="100dp"
19     android:layout_marginTop="20dp"
20     android:text="Username"/>
21  <EditText
22     android:id="@+id/username_input_profile"
23     android:layout_width="wrap_content"
24     android:layout_height="wrap_content"
25     android:layout_marginLeft="100dp"
26     android:ems="10"/>
27
28  <TextView
29     android:id="@+id/email_text_profile"
30     android:layout_width="wrap_content"
31     android:layout_height="wrap_content"
32     android:layout_marginLeft="100dp"
33     android:text="Email"/>
34  <EditText
35     android:id="@+id/email_input_profile"
36     android:layout_width="wrap_content"
37     android:layout_height="wrap_content"
38     android:layout_marginLeft="100dp"
39     android:inputType="textEmailAddress"
40     android:ems="10"/>
41
42  <TextView
43     android:id="@+id/password_text_profile"
44     android:layout_width="wrap_content"
45     android:layout_height="wrap_content"
46     android:text="Password"
47     android:layout_marginLeft="100dp" />
48
49  <EditText
50     android:id="@+id/password_input_profile"
51     android:layout_width="wrap_content"
52     android:layout_height="wrap_content"
53     android:layout_marginLeft="100dp"
54     android:inputType="textPassword"
55     android:ems="10" />
56
57  <TextView
58     android:id="@+id/universita_text_profile"

```

```

59     android:layout_width="wrap_content"
60     android:layout_height="wrap_content"
61     android:layout_marginLeft="100dp"
62     android:text="Università"/>
63 <EditText
64     android:id="@+id/universita_input_profile"
65     android:layout_width="wrap_content"
66     android:layout_height="wrap_content"
67     android:layout_marginLeft="100dp"
68     android:ems="10"/>
69
70 <TextView
71     android:id="@+id/corso_text_profile"
72     android:layout_width="wrap_content"
73     android:layout_height="wrap_content"
74     android:layout_marginLeft="100dp"
75     android:text="Corso di laurea"/>
76 <EditText
77     android:id="@+id/corso_input_profile"
78     android:layout_width="wrap_content"
79     android:layout_height="wrap_content"
80     android:layout_marginLeft="100dp"
81     android:ems="10"/>
82 <LinearLayout
83     android:layout_width="match_parent"
84     android:layout_height="wrap_content"
85     android:orientation="horizontal">
86
87     <Button
88         android:id="@+id/btn_logout"
89         style="@style/Widget.AppCompat.Button.ButtonBar.AlertDialog"
90         android:layout_width="wrap_content"
91         android:layout_height="wrap_content"
92         android:layout_marginLeft="100dp"
93         android:layout_marginTop="20dp"
94         android:text="indietro" />
95
96     <Button
97         android:id="@+id/btn_update"
98         style="@style/Widget.AppCompat.Button.Colored"
99         android:layout_width="wrap_content"
100        android:layout_height="wrap_content"
101        android:layout_marginLeft="50dp"
102        android:layout_marginTop="20dp"
103        android:text="Aggiorna" />
104
105 </LinearLayout>
106
107
108 </LinearLayout>

```

Listato 4.10: Implementazione del file XML di FragmentProfilo

4.3 Manuale Utente

Concludiamo questo capitolo sull'implementazione del front-end dell'applicazione *Study mate* con il manuale utente, che contiene gli screenshot delle schermate più importanti dell'app con la relativa spiegazione.

Per prima cosa, è importante far notare che l'applicazione è composta da una toolbar e da un tablayout, posti nella parte alta dello schermo, che rimangono sempre costanti, mentre la parte inferiore della schermata si differenzia in base alla sezione

in cui l'utente si trova. In particolare, il tablayout è composto da tre tab che permettono di navigare da una sezione all'altra, per cui questi conterranno le etichette “eventi”, “flashcard” e “note”. La toolbar, invece, contiene il nome dell'applicazione e due pulsanti. Quello più a destra, indicato dal simbolo “?”, permette all'utente di ricevere un aiuto sul funzionamento dell'app, mentre quello caratterizzato dal simbolo del profilo permette, appunto, di visualizzare le proprie informazioni personali. Questo avviene se l'utente ha effettuato il login; altrimenti, invece che il simbolo del profilo, sarà presente un tasto che permette di eseguire l'accesso.

4.3.1 Home Page

La schermata iniziale di *Study mate*, mostrata in Figura 4.1, coincide con la sezione “flashcard” dell'app, per cui la barra sarà impostata di default su questa sezione. L'utente, però, ha la possibilità di cambiare sezione, premendo sul tasto corrispondente. Una volta aperta l'applicazione, quindi, l'utente visualizzerà le materie che avrà salvato, con, sotto ciascuna di esse, i titoli dei mazzi creati. L'utente, dunque, potrà premere sul tasto “play”, posto di fianco al nome della materia, per studiare tutte le flashcard relative a quella determinata materia; in alternativa, può premere sempre sul tasto “play”, ma posto a lato del nome del mazzo, per visualizzare solo una porzione delle carte. Inoltre, egli può cliccare sulla materia per aprire una nuova schermata che mostrerà solo i mazzi della materia selezionata. Dalla home page è, anche, possibile aggiungere nuove materie, premendo sul pulsante che contiene il simbolo “+”, posto in basso a destra. Invece, per eliminare una materia, è sufficiente tenere premuto sul nome e scegliere l'opzione corrispondente. Eventualmente, sempre tenendo premuto sul nome, è possibile modificare il nome della materia.

4.3.2 Schermata “Materia”

La schermata in Figura 4.2 mostra ciò che l'utente visualizza quando, dalla home page, preme sul nome di una materia. In alto a sinistra è presente un tasto con una freccia che permette di tornare alla schermata iniziale, di fianco è riportato il nome della materia selezionata; a destra, invece, è situato il pulsante “studia”, che permette di ripassare tutte le carte della materia stessa. In questa schermata vengono visualizzati i titoli di ogni mazzo; sotto ciascuno di essi, sono presenti i corrispondenti sottomazzi. Infatti ogni mazzo può essere diviso in due parti, ovvero si possono distinguere le carte di cui si sa il contenuto da quelle che ancora devono essere studiate. Per studiare uno di questi sottomazzi, è necessario premere sul tasto “play” corrispondente, mentre, per studiare l'intero mazzo, bisogna premere il pulsante a lato del corrispondente nome. Inoltre è possibile creare un nuovo mazzo premendo sul tasto che contiene il simbolo “+”. Tenendo premuto su un mazzo, si possono modificare i dettagli o lo si può eliminare. Se l'utente ha eseguito l'accesso, ci sono altre azioni che può eseguire. Infatti, egli può condividere un proprio mazzo con altri utenti, premendo sul pulsante di upload posto a lato del nome del mazzo, oppure può scaricare mazzi di altri utenti, premendo sul tasto di download, situato in basso a sinistra. Ovviamente, questi ultimi due pulsanti non saranno presenti nella schermata se l'utente non ha eseguito il login.

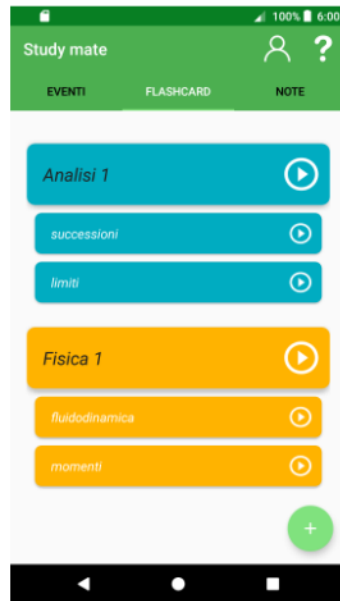


Figura 4.1: Screenshot della home page, ovvero la schermata “Flashcard”



Figura 4.2: Screenshot della schermata “Materia”

4.3.3 Schermata “Mazzo”

La schermata in Figura 4.3 mostra ciò che viene visualizzato quando l'utente preme sul titolo di un mazzo. Si apre, quindi, una nuova schermata che mostra tutte le carte

appartenenti a quel mazzo, disposte una sotto l'altra. In particolare, l'utente vedrà il titolo della carta e, subito sotto, il suo contenuto o la sua descrizione. L'utente può aggiungere nuove carte, premendo sul tasto che contiene il simbolo "+", posto in basso a destra. Se tiene premuto su una carta, inoltre, egli può modificarla o eliminarla.



Figura 4.3: Screenshot della schermata “Mazzo”

4.3.4 Schermata “Eventi”

Per raggiungere questa schermata (Figura ??), l'utente deve, semplicemente, utilizzare la toolbar, premendo il tab con l'etichetta “eventi”. Subito sotto la toolbar è presente uno switch, che permette di modificare la vista della pagina e decidere se si vogliono vedere solo gli eventi che sono stati salvati (opzione di default), oppure se si vuole visualizzare l'intero calendario. Quest'ultima opzione è quella mostrata in figura. L'utente può, quindi, spostarsi nel calendario e selezionare un determinato giorno per visualizzarne gli eventi salvati. Questi saranno mostrati uno sotto all'altro; i dettagli, invece, saranno evidenti solo se premuti. Inoltre, l'utente può selezionare un giorno sul calendario e poi premere il pulsante che contiene il simbolo “+” per creare un nuovo evento. Se, invece, l'opzione “mostra calendario” è disattivata, egli vedrà nello schermo solo gli eventi che ha salvato, in ordine temporale. Anche in questo caso, premendo il tasto “+”, l'utente potrà creare nuovi eventi, ma, a differenza dell'altra modalità, dovrà anche specificare il giorno dell'evento. In ogni caso, per modificare o eliminare un evento, bisognerà, semplicemente, tenere premuto su di esso e poi selezionare l'opzione desiderata.

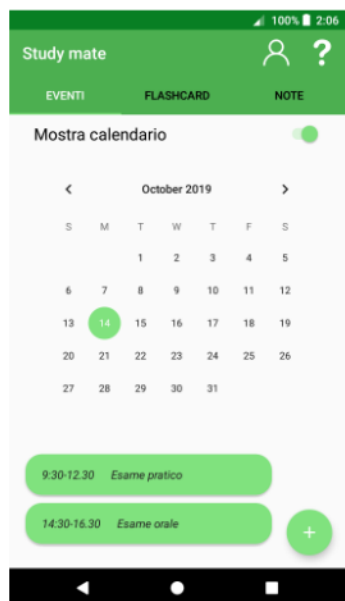


Figura 4.4: Screenshot della schermata “Eventi”

4.3.5 Schermata “Note”

Quando l’utente preme il tab che contiene l’etichetta “note”, viene mostrata la schermata in Figura 4.5. Questa sezione è piuttosto semplice; infatti, in essa, vengono mostrate le note che sono state salvate dall’utente, disposte due per riga. Ogni nota ha, come etichetta, il nome della materia e il contenuto della nota stessa, e risulta essere espandibile nella schermata, adattandosi al numero di caratteri utilizzati. Il pulsante che contiene il simbolo “+”, posto in basso a destra, permette di aggiungere nuove note, mentre, per modificare o eliminare delle note già esistenti, è necessario tenere premuto sulla nota e selezionare l’opzione corrispondente.

4.3.6 Schermata “Profilo”

La schermata “profilo”, mostrata in Figura 4.6, è disponibile soltanto se l’utente ha effettuato il login. In tal caso, in alto a destra, di fianco al nome dell’applicazione, sarà presente un pulsante con il simbolo del profilo, che, appunto, condurrà in questa schermata. Qui saranno visualizzati i dati dell’utente, ovvero lo username, l’email, la password, l’università e il corso di laurea. Da questa schermata, l’utente può modificare i propri dati, premendo sul tasto “aggiorna dati”, oppure può disconnettersi dall’applicazione, cliccando sul pulsante “logout”.



Figura 4.5: Screenshot della schermata “Note”



Figura 4.6: Screenshot della schermata “Profilo”

Discussione, conclusioni e sfide future

Quest'ultimo capitolo serve per trarre le conclusioni di quanto fatto in questa tesi. Cercheremo di capire quanto è stato imparato grazie a questo progetto, cosa potrà essere utile in futuro e cosa, invece, è bene non ripetere. Concluderemo, infine, con un bilancio di quanto fatto e con uno sguardo a cosa potrà essere migliorato in futuro.

5.1 Discussione

L'applicazione che è stata riprogettata in questa tesi, *Assistente di Studio*, ora *Study mate*, può rivelarsi un ottimo strumento per il supporto allo studio. È un'app piuttosto semplice e immediata, ma che può veramente offrire un valido aiuto agli studenti, incentivando un metodo per ripassare veloce e sempre a porta di mano. Le flashcard, infatti, sono una tipologia di studio ancora poco usata, ma che, invece, è fortemente sottovalutata, in quanto garantisce un apprendimento a lungo termine. Aiuterà gli studenti ad essere più preparati il giorno dell'esame, ma anche a ricordare le nozioni apprese più a lungo. Il tutto, poi, è condito con altre funzioni che possono, in ogni caso, supportare lo studente nella sua vita universitaria.

Questo progetto ha avuto come fine una riprogettazione dell'applicazione basata sul concetto di dare massima importanza all'utente che utilizzerà il prodotto. Non si è modificata l'app in base a gusti personali, ma tramite uno studio dei modelli mentali di un gruppo di volontari del target desiderato. Questo processo che si è attuato è basato, dunque, sulle tecniche di *User Experience Design*.

5.1.1 Ruolo della UXD nella realizzazione delle app

Affrontando questo progetto si è avuto modo di conoscere, almeno un po', il complesso mondo dello *User Experience Design*. Questo è un universo più complicato di quel che sembra, in quanto non si occupa di rendere un prodotto digitale semplicemente gradevole, ma ha come obiettivo quello di renderlo usabile, accattivante e in grado di suscitare una sensazione nell'utilizzatore.

Lo *User Experience Design* è un concetto che si sta diffondendo negli ultimi anni; basti pensare al fatto che non esiste, ancora, una sua definizione precisa. Negli ultimi anni il numero di *User Experience Designer* sta aumentando significativamente, proprio per soddisfare la richiesta, sempre crescente, di esperti del settore. Nonostante tutto, questo sviluppo dei prodotti basato sull'esperienza utente è ancora, molto spesso, sottovalutato.

Tramite questa tesi abbiamo potuto apprezzare tali tecniche e capire il valore e l'importanza della *User Experience*. In un mondo ormai digitalizzato, i prodotti informatici sono numerosissimi, di qualunque tipo e capaci di offrire tutti i servizi immaginabili. È, dunque, difficile produrre qualcosa di completamente nuovo e innovativo, in quanto, spesso, qualcuno ha già sviluppato la stessa idea. La chiave del successo del proprio prodotto informatico non è, quindi, l'originalità dei servizi offerti, ma l'offerta di un determinato servizio a condizioni migliori degli altri. Il sito o l'applicazione non deve, semplicemente, funzionare, ma deve funzionare bene. Ciò significa che il prodotto, oltre che utile, deve essere facile da usare, deve essere bello da vedere, deve essere accattivante e invogliare l'utente a utilizzarlo ancora. Se il prodotto non riesce a offrire tutto ciò, si rischia di perdere clienti, in quanto la concorrenza è tanta e, se l'utente non è soddisfatto, può facilmente disinstallare l'applicazione per provarne un'altra.

In quest'ottica, lo *User Experience Design* risulta essere la chiave vincente per creare un buon prodotto che possa avere successo. È un processo che può essere considerato l'avanguardia della progettazione di prodotti informatici, centrato sulla comprensione dei desideri e delle aspettative dell'utente finale e sullo studio dei suoi modelli mentali, così da progettare un prodotto creato su misura di chi lo utilizzerà.

5.1.2 Lezioni apprese

Nell'affrontare questo progetto ci siamo imbattuti in molti aspetti che ci hanno aiutato a crescere e a imparare nuove nozioni in vista di progetti futuri.

La progettazione di un'app all'insegna della *User Experience* non è semplice come può sembrare. Infatti, non esiste un iter progettuale preciso e ben definito, ma vengono definite delle tecniche e delle best practice, che, però, devono essere valutate in base allo specifico progetto. Il fatto che non ci siano dei passaggi del tutto stabiliti può essere un bene, in quanto lascia spazio al progettista di decidere a sua discrezione, basandosi su ciò che ritiene più giusto e corretto. Dall'altra parte, per chi non ha mai lavorato in questo contesto applicativo, e quindi non ha alcuna esperienza nell'ambito, può essere complesso capire quale tecnica è più appropriata per il progetto specifico.

In generale, abbiamo compreso che sarebbe sempre più opportuno applicare tecniche di progettazione che coinvolgano gli utenti di persona, in quanto non è solo importante cosa dicono, ma anche come reagiscono, il modo in cui si pongono e il linguaggio del corpo. Infatti, pareri di utenti non esperti del settore possono risultare ambigui e poco specifici, ma osservazioni fatte da un mediatore sugli utenti coinvolti possono integrare le conoscenze che vengono raccolte. Per problemi di tempo e di logistica, in questa tesi, sono state applicate tecniche che hanno coinvolto gli utenti da remoto, con il supporto di strumenti informatici. Tuttavia, vi è la consapevolezza di aver interpellato alcuni colleghi di corso, che hanno un buon bagaglio culturale di

base per poter fornire un contributo importante e tecnico. In ogni caso, riteniamo che, per progetti futuri, sia più professionale applicare tecniche per monitorare e studiare gli utenti dal vivo, al fine di raccogliere informazioni più complete.

Abbiamo anche imparato che è importante dividere bene il lavoro, soprattutto nei casi in cui è necessario coinvolgere gli utenti finali per stabilire i requisiti. Infatti, serve parecchio tempo per riuscire a raccogliere le informazioni necessarie e, dunque, va diviso bene il tempo da dedicare al coinvolgimento degli utenti, quello per l'analisi dei dati e, infine, quello per la realizzazione vera e propria dell'app.

Infine, abbiamo anche compreso l'importanza del parere delle altre persone e nel loro ascolto. Non sempre le idee del progettista corrispondono ai bisogni e alle aspettative di chi utilizzerà il prodotto ed è, quindi, bene, anzi necessario, coinvolgere altri individui, perchè, come ha affermato Dan Saffer, "Tu non sei l'utente".

5.2 Conclusioni e uno sguardo al futuro

Study mate non è ancora un'applicazione pronta da lanciare sul mercato, ma ha buone potenzialità per essere effettivamente utilizzata e per dare un contributo significativo alla vita universitaria dello studente. La riprogettazione affrontata in questa tesi ha avuto come obiettivo quello di stabilire l'architettura della nuova applicazione, in modo da renderla più intuitiva e facile da usare, e di progettare le nuove interfacce grafiche, che rendano l'app più intrigante, usabile e accattivante.

L'iter di vita di un prodotto digitale non finisce mai, ma è ciclico; infatti, non appena una versione del prodotto viene rilasciata, si volge subito lo sguardo a cosa si può fare per migliorarla ancora. Nel caso di questo progetto, dal momento che il prodotto non è completo, tale passaggio è di fondamentale importanza.

Per far sì che *Study mate* possa essere effettivamente utilizzata è necessario realizzare il back-end dell'applicazione, tra cui la gestione degli account e l'accesso degli utenti, garantendo, ovviamente, un ottimo grado di sicurezza per proteggere i dati personali. È, inoltre, necessario implementare il database che contiene i mazzi che gli utenti decidono di caricare e garantirne, poi, l'accesso a chi si è registrato nell'app e intende utilizzare anche mazzi altrui.

Inoltre, per migliorare l'applicazione, sarà importante riprogettare la grafica delle singole carte e aggiungere nuove funzioni per renderle più usabili. Per far ciò bisognerà implementare le seguenti funzionalità:

- Deve essere possibile inserire non solo testo, ma anche immagini, così da rendere l'app uno strumento utilizzabile per studiare qualunque materia. Per far ciò, bisogna gestire i permessi di accesso dell'applicazione alla galleria e alla fotocamera del dispositivo elettronico.
- Deve essere possibile inserire anche formule matematiche, per cui bisogna prevedere qualche tool per poter inserire in maniera comoda anche formule e simboli.
- Ogni carta deve avere anche una valutazione della sua difficoltà, calcolata come media delle valutazioni dei vari utenti che l'hanno utilizzata. Una volta associato a ogni carta un livello di difficoltà, è utile per gli studenti avere la possibilità di studiare le carte per livelli, e quindi avere un filtro che permetta di visualizzare solo una determinata tipologia di carte.

Si dovrà, anche, migliorare gli altri servizi dell'applicazione, tramite le seguenti modifiche:

- Per quanto riguarda la note, si deve prevedere la possibilità sia di inserire immagini sia di aggiungere note vocali, gestendo, quindi, i permessi d'accesso al microfono.
- Riguardo al calendario si deve implementare una funzione che permetta di specificare che l'evento che l'utente sta inserendo è ricorrente. In particolare, deve essere possibile indicare che si sta aggiungendo l'orario di una lezione, in modo tale che l'applicazione lo aggiunga in automatico anche per le settimane successive.

In poche parole, *Study mate* è un progetto appena iniziato, che ha molti possibili sviluppi e margini di miglioramento per far sì che diventi un'applicazione integrata nella vita degli studenti.

Ringraziamenti

Ogni traguardo necessita di un percorso per raggiungerlo e di una festa per celebrarlo. Non importa quanto possa esser stato difficile arrivarci; ogni passo che compiamo in avanti è un simbolo di crescita e di maturazione. Il merito di questo traguardo non è solo mio, ma anche di tutte quelle persone che mi sono state accanto in questi tre anni ed è per questo che desidero ringraziarle tutte, per l'affetto che mi hanno dato e il supporto di cui avevo bisogno.

Desidero ringraziare i miei genitori, che mi hanno permesso di intraprendere questo percorso, che non mi hanno mai ostacolata, che hanno sempre creduto in me, senza vacillare mai. Spero di avervi reso orgogliosi di me.

Ringrazio la mia sorellina Elisa che, nonostante le litigate, è sempre stata al fianco e mi ha sempre strappato un sorriso.

Ringrazio mia zia Paola, che nonostante la lontananza, nonostante il fatto che ci vediamo poco, non si è mai dimenticata di augurarmi buona fortuna prima di un esame.

Ringrazio i miei nonni, che mi hanno sempre sostenuta e hanno fatto sempre il tifo per me e che da un anno a questa parte continuavano a chiedermi quando mi sarei laureata, perchè non vedevano l'ora di vedermi raggiungere i miei traguardi.

Ringrazio mio cugino Michele, che dopo gli esami mi comprava la merenda per festeggiare insieme.

Un ringraziamento speciale va al mio ragazzo, Francesco, che da due anni mi supporta in questo percorso universitario e nella vita di tutti i giorni. È lui che mi ha aiutato a superare i momenti difficili ed è lui che mi ha dovuto sopportare quando avevo l'ansia per gli esami, ma è sempre stato al mio fianco senza abbandonarmi mai.

Vorrei anche ringraziare la mia "squadra", Pietro, Millo, Alkis, Lollo, KC, Gatto e Migheli, compagni di studi, di progetti, di esaurimenti prima degli esami e di festa subito dopo. Li ringrazio perchè le giornate all'università con loro hanno un sapore diverso, perchè affrontare le difficoltà insieme è stato più facile e perchè ho trovato degli amici su cui so di poter contare.

Ringrazio la mia migliore amica Sara, che conosco da quando è nata e con cui condivido tutto da più di vent'anni. La ringrazio per essere sempre dalla mia parte e per spronarmi sempre a dare il meglio di me.

Ringrazio tutti i miei amici, quelli che conosco da anni ma anche i più recenti, perchè mi hanno sempre incitato e perchè sono sempre pronti ad aiutarmi.

Per ultimo, ma non per importanza, ringrazio il Professore Domenico Ursino, che mi ha supportato nella stesura di questa tesi, perchè è sempre stato disponibile e presente, anche agli orari più improbabili, e perchè mi ha dato la possibilità di mettermi alla prova.

Grazie a tutti.

Riferimenti bibliografici

1. User Experience e Usabilità, per dare valore alle persone. <https://www.thismarketerslife.it/digital/user-experience-e-usabilita-per-dare-valore-alle-persone/>, 2016.
2. Card Sorting: cos'è e perchè è importante usarlo. <https://www.sinfoniaperformance.com/blog/card-sorting-cosa-e-perche-e-importante>, 2017.
3. Cos'è la User Experience? <https://www.tonifontana.it/user-experience-definizione/>, 2018.
4. La Storia della User Experience. <https://www.uxblog.it/la-storia-della-user-experience/>, 2018.
5. Come si fa un'Analisi SWOT: guida completa con esempio di progetto. <https://it.semrush.com/blog/come-si-fa-analisi-swot-con-esempio/>, 2019.
6. Flashcards: come studiare, ricordare e non dimenticare più. <https://www.gliudadacidellamemoria.com/flashcards/>, 2019.
7. Guida Android. <https://www.html.it/guide/guida-android/>, 2019.
8. Il processo creativo completo di UX Design in 14 fasi. <https://www.grafigata.com/processo-completo-ux-design/>, 2019.
9. The Fascinating History of UX Design: A Definitive Timeline. <https://careerfoundry.com/en/blog/ux-design/the-fascinating-history-of-ux-design-a-definitive-timeline/>, 2019.
10. What is UX design? 15 user experience design experts weigh in. <https://www.usertesting.com/blog/what-is-ux-design-15-user-experience-experts-weigh-in/>, 2019.
11. D. Bottà. *User eXperience design*. Hoepli, 2018.
12. M. Burton and D. Felker. *Android App Development For Dummies*. John Wiley Sons Inc, 2015.
13. M. Canducci. *XML. Conoscere il linguaggio XML signi ca poter comunicare veramente con tutti*. Apogeo, 2009.
14. M. Carli. *Android 6. Guida per lo sviluppatore*. Apogeo, 2016.
15. E. Cisotti and M. Giannino. *Android: Guida completa*. Edizioni LSWR, 2015.
16. I.G. Clifton. *Android User Interface Design: Implementing Material Design for Developers (Usability)*. Addison-Wesley Professional, 2015.
17. C. Horstmann. *Concetti di Informatica e fondamenti di Java (IV Edizione)*. Apogeo, 2007.
18. S. Krug. *Don't make me think*. Tecniche Nuove, 2014.
19. M. C. Lavazza. *Comunicare la User Experience*. Apogeo Education, 2012.

20. D. A. Norman. *La caffettiera del masochista*. Giunti Editore, 2015.
21. G. Nudelman. *Android Design Patterns: Interaction Design Solutions for Developers*. Wiley, 2013.
22. M. Di Pascale. *Manuale di sopravvivenza per UX designer*. Hoepli, 2019.
23. J. Pasquini S. Giomi and. *Web usability*. Hoepli, 2014.
24. M. C. Caratozzolo S. Giomi and, J. Pasquini and. *UX Designer*. Franco Angeli, 2018.
25. B. C. Zapata. *Android Studio Application Development*. Packt Publishing, 2013.