



## PrenotaBiblio Unifi

Progetto Human Computer Interaction [9CFU]  
A.A. 2021/2022

Davide Pucci

davide.pucci@stud.unifi.it

### Abstract

*A seguito dell'introduzione di Kairos - il sistema informativo che consente la prenotazione dei servizi bibliotecari dell'ateneo di Firenze - ogni studente che voglia accedere a una Biblioteca o ad un'Aula Studio è costretto a prenotarsi online appena il periodo di prenotazioni si apre, per evitare il rischio di trovare tutti i posti già occupati. PrenotaBiblio Unifi è un'applicazione che consente la prenotazione automatica al servizio tramite smartphone, così da svincolare lo studente dal dover utilizzare Kairos all'ora di apertura delle prenotazioni e assicurargli un posto negli spazi studio.*

*Il report si articola seguendo i passi effettuati per sviluppare l'applicazione partendo dal Needfinding: il processo in cui si delineano le esigenze che motivano lo sviluppo del prodotto e i requisiti che questo deve rispettare. Viene quindi presentato il prototipo dell'applicazione nelle sue interfacce e vengono infine mostrati i risultati dell'Usability Testing della stessa.*

### 1. Introduzione

Nel settembre del 2020 le biblioteche dell'ateneo fiorentino sono state riaperte dopo la pausa forzata data dall'emergenza Covid-19. Al fine di regolamentare gli ingressi negli spazi studio e consentire il tracciamento dei contagi dopo la riapertura, è stato introdotto *Kairos*: una piattaforma online che consente agli studenti di prenotare i servizi bibliotecari.

Questo sistema di prenotazioni tuttavia non soddisfa completamente le esigenze degli studenti, che spesso si dimenticano di prenotare o non trovano posti liberi qualora

decidano troppo tardi di prenotare un posto. Qui riportiamo le caratteristiche principali del servizio:

- Per le **Biblioteche** è possibile prenotare una postazione studio nelle 48h precedenti alla data richiesta: se, ad esempio, uno studente volesse andare in biblioteca il mercoledì, si può prenotare a partire dalla mezzanotte di lunedì fino alla fine della giornata di martedì.
- Per quanto riguarda le **Aule-Studio** è possibile prenotare una postazione nei 14 giorni precedenti alla data richiesta.
- In entrambi i casi non è dunque possibile prenotarsi il giorno stesso.
- È possibile annullare una prenotazione solamente quando il periodo di prenotazione è ancora aperto, dunque cancellarsi per il giorno stesso non è possibile.
- È prevista inoltre una divisione tra mattina e pomeriggio, dunque per passare una giornata in biblioteca/aula-studio sono necessarie due prenotazioni, una per turno.

L'insieme di queste caratteristiche, unite al numero di posti limitati che sono presenti nei vari spazi studi dell'ateneo, hanno creato delle dinamiche spiacevoli per chi è un assiduo frequentatore delle biblioteche.

Dal momento che i posti a disposizione sono pochi, uno studente che voglia assolutamente prenotarsi è costretto a farlo appena il periodo di prenotazione si apre, dunque allo scoccare della mezzanotte di 2 (o 14 nel caso delle aule-studio) giorni precedenti alla data selezionata.

Inoltre, il fatto di non poter cancellare una prenotazione il giorno stesso, fa sì che molti posti prenotati non vengano

effettivamente utilizzati, dato che molti studenti fino all'ultimo minuto possono essere indecisi se andare o meno, dunque non annulleranno la prenotazione. Un ulteriore aspetto è dato dalla estrema ripetitività delle prenotazioni: molti studenti nell'arco della settimana possono volersi prenotare nella stessa biblioteca per più giorni consecutivi, rendendo le dimenticanze molto probabili.

Nasce in questo contesto l'idea di un'applicazione che permetta di automatizzare il processo di prenotazione, così da svincolare lo studente dall'effettuare le prenotazioni alla mezzanotte e da garantirgli sempre un posto. Nelle seguenti sezioni viene sviluppata questa idea, partendo da un processo di Needfinding per delineare gli obiettivi e i requisiti dell'applicazione, poi passando alla realizzazione di un prototipo funzionante della app, infine mostrando i risultati ottenuti durante l'Usability Testing.

## 2. Needfinding

Il Needfinding consiste nell'osservare le abitudini degli utenti nel contesto di interesse, al fine di estrarre quelle che sono le loro necessità. In questo processo si ricavano le **Personas**: i personaggi che rappresentano le varie tipologie di utenti nel nostro contesto applicativo, su cui costruiamo degli **Scenari**: contesti idealizzati in cui le varie personas si trovano ad utilizzare il prodotto. Questo lavoro preliminare ci permette di ricavare i **Requisiti**: tutte le caratteristiche che devono far parte del nostro prodotto finale.

In questo caso il Needfinding è iniziato proponendo ad alcuni studenti dell'Università di Firenze un **questionario**, finalizzato a comprendere quali fossero le loro abitudini nel prenotarsi ai servizi bibliotecari, il gradimento di uno strumento di prenotazione automatico e quale fosse la piattaforma più adatta a questo scopo.

Un'applicazione che prenoti per me in modo automatico tutti i miei appuntamenti in biblioteca/aula-studio mi semplificherebbe la vita

11 risposte

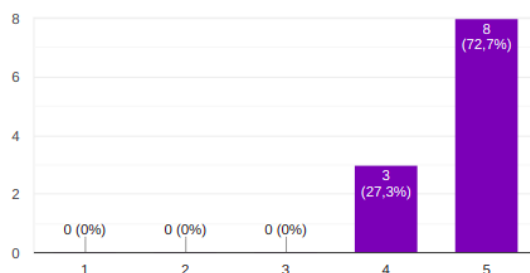


Figura 1. Risultati ottenuti alla domanda del questionario sul gradimento di uno strumento di prenotazione automatica ai servizi bibliotecari. 5 indica il completo assenso alla dichiarazione in alto, 1 il completo disaccordo.

Da questa prima indagine, eseguita su un campione di 12 studenti dell'ateneo fiorentino, sono emerse le seguenti osservazioni:

- Spesso gli studenti dimenticano di prenotare nonostante abbiano già programmato di andare in biblioteca.
- Il fatto che la prenotazione apra a mezzanotte ha modificato le abitudini di molti studenti, costringendoli ad aspettare dopo la mezzanotte per andare a dormire.
- L'eventualità di avere uno strumento di prenotazione automatica piace agli studenti.
- La piattaforma più adatta per uno strumento di questo tipo risulta lo smartphone, che è stato preferito al PC/Mac dalla quasi totalità degli studenti che hanno risposto al quiz.

A seguire sono stati intervistati 7 degli studenti che hanno risposto al questionario, tutti assidui frequentatori delle biblioteche dell'ateneo, in modo da ricevere un feedback più mirato e approfondito. L'intervista segue uno schema prefissato e si apre rivolgendo le seguenti domande:

- *Come ti prenoti in biblioteca? Quali sono le tue abitudini per prenotarti? Fai uso di sveglie?*
- *Cosa non ti piace riguardo alla gestione delle prenotazioni? Cosa dovrebbe cambiare? Hai qualche idea?*

All'intervistato viene dunque presentata l'idea di un'applicazione di prenotazione automatica per le biblioteche ed aula studio dell'ateneo. In particolare viene proposta l'idea di un calendario su cui inserire e modificare le prenotazioni richieste. Si procede con ulteriori domande:

- *Ti piace l'idea di un calendario? Come ti immagini di poter gestire tutte le tue prenotazioni?*
- *Cosa dovrebbe essere possibile fare in questa applicazione?*

Vengono inoltre poste alcune domande di carattere *etico*:

- *Secondo te un'applicazione di questo tipo, se tutti ne facessero uso, porterebbe ad un aumento delle persone che si prenotano ma non vengono? Oppure no?*
- *È facile cancellare una prenotazione su Kairos? Tu ti cancelli se non vai?*

L'intervista si chiude domandando se ci sono ulteriori osservazioni o suggerimenti, dunque si ringrazia l'intervistato. I punti principali emersi da questa analisi sono riportati di seguito:

- L'idea del calendario sembra la migliore per organizzare e gestire le prenotazioni.
- Una possibile soluzione al fatto che gli studenti si prenotino ma non vengano in biblioteca può essere quella di aggiungere una notifica la sera prima della prenotazione, così che sia semplice liberare il posto qualora si decida di non andare.
- Alcuni studenti hanno riportato uno stato di ansia dato dal doversi prenotare a tutti i costi con il corrente sistema di prenotazione.

L'insieme di queste interviste hanno aiutato alle successive fasi di progettazione, fornendo spunti fondamentali alla delineazione delle Personas e degli Scenari, quindi dei Requisiti fondamentali dell'applicazione.

## 2.1. Personas e Scenari

Grazie alle interviste e al questionario abbiamo un quadro molto preciso di quelle che sono le caratteristiche degli utenti a cui è destinata l'applicazione, e che possiamo riassumere in una singola *persona*:

- **Erika:** 19-26 anni, è una studentessa di psicologia dell'Università di Firenze, e centrale della squadra di pallavolo dove gioca sin da bambina. Durante la sessione di esami trascorre giornate intere nella biblioteca dell'ateneo che frequentano i suoi amici dove, oltre allo studio, può concedersi un po' di chiacchiere davanti ad un buon caffè. Dato che i suoi allenamenti finiscono tardi, di solito si prenota in biblioteca prima di andare a letto, così da essere sicura di trovare un posto. A volte però, soprattutto dopo gli allenamenti più provanti, se ne dimentica ed è così costretta a restare a casa, dove però non riesce a concentrarsi e si sente meno produttiva.

Al fine di contestualizzare la nostra persona Erika nella vita di tutti i giorni si immaginano qui degli scenari di utilizzo dell'applicazione, gli *scenari contestuali*:

- Al bar accanto alla biblioteca che frequenta, Erika viene a sapere dal suo gruppo di amici dell'applicazione PrenotaBiblio Unifi: una app che consente la prenotazione automatica alle biblioteche ed aule studio dell'ateneo. Decide di installarla, apre dunque la app e inserisce i suoi dati: **matricola e password che usa per accedere a Kairos**.
- Una volta tornata in biblioteca Erika riapre la app: dato che ha il lunedì libero questo semestre, vorrebbe sfruttarlo per venire in biblioteca. Dunque nel calendario della app seleziona lunedì, aggiunge i **dati della prenotazione (nome della biblioteca/aula-studio, piano/aula di preferenza, turno mattina/pomeriggio)**

e decide di ripetere questa prenotazione per le settimane seguenti fino alla fine del semestre. Una volta confermato, nel calendario le appaiono il giallo le prenotazioni che ha programmato per tutti questi lunedì.

- È domenica sera, Erika sta guardando una puntata della sua serie preferita, quando riceve una **notifica sullo smartphone**: è l'app PrenotaBiblio Unifi che le ricorda che domani ha una prenotazione in biblioteca. Dato che è sicura di andare domani, ignora la notifica, e non procede all'annullamento della prenotazione.
- È lunedì mattina, Erika è appena uscita di casa diretta verso la biblioteca. Apre la app PrenotaBiblio Unifi, dove nel calendario, selezionando il giorno corrente, può vedere **l'aula e la postazione che le è stata assegnata**.
- Appena arrivata in biblioteca Erika apre nuovamente la app per mostrare al tornello il **QR di accesso**, che trova comodamente nella schermata principale.

In grassetto sono stati evidenziati alcuni punti salienti dell'applicazione, che consentono di delineare quelli che sono i requisiti e le funzionalità fondamentali dell'applicazione.

## 2.2. Requisiti

Dall'analisi approfondita degli scenari sono stati estratti i *data and functional requirements* dell'applicazione, che riportiamo di seguito:

- **Account:** memorizzazione di matricola e password di Kairos.
- **Prenotazione:** memorizzazione dei dati della prenotazione, sia dal lato utente (nome della biblioteca/aula-studio, piano/aula di preferenza, turno mattina/pomeriggio), che dal lato di Kairos (codice prenotazione, numero postazione assegnata, orari di apertura).
- **Calendario:** per gestire e visualizzare le prenotazioni.
- **Creazione prenotazione:** interfaccia di creazione della prenotazione, con selezione di biblioteca/aula-studio, piano/aula di preferenza e turno mattina/pomeriggio. Possibilità di iterare la prenotazione e shortcut per prenotare sia mattina che pomeriggio.
- **Gestione prenotazione:** interfaccia di riepilogo prenotazione con possibilità di annullamento della stessa.
- **QR di accesso:** per accedere alla biblioteca.

- **Reminder prenotazione:** notifica che ricordi la prenotazione il giorno successivo, e che consenta la cancellazione.
- **Bot di prenotazione notturno:** sistema automatico che prenoti gli appuntamenti dell'utente durante la notte, così da svincolarlo dalla prenotazione.

Tutti questi elementi sono stati implementati nella applicazione finale, così che tutte le esigenze di *Erika*, e quindi degli utenti, siano soddisfatte.

### 2.3. Mockup

A seguito della definizione dei requisiti dell'applicazione possiamo procedere con l'implementazione della app vera e propria. In primo luogo vengono costruiti dei *mockup* così da delineare le caratteristiche dell'applicativo finale e da guidare nelle scelte di design.

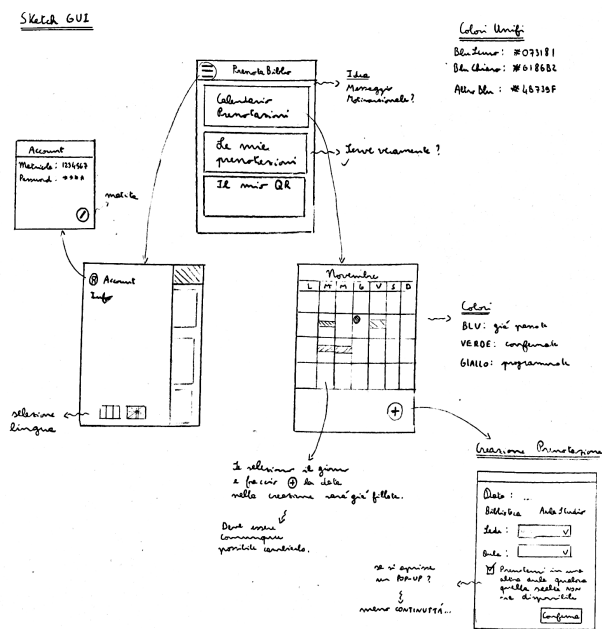


Figura 2. Primo sketch di mockup con carta e penna.

Il primo mockup ha contribuito a formalizzare l'idea del **calendario** come vista principale per gestire le prenotazioni. Con ulteriori mockup è stata aggiunta l'idea di utilizzare i **colori** per indicare intuitivamente lo stato delle prenotazioni, come mostrato in figura 3.

I mockup più avanzati sono stati costruiti utilizzando *diagrams.net* [2]. Il mockup finale nella sua interezza è mostrato in figura 4, assieme alla logica di navigazione delle viste. L'applicazione finale si basa su questa modellazione.

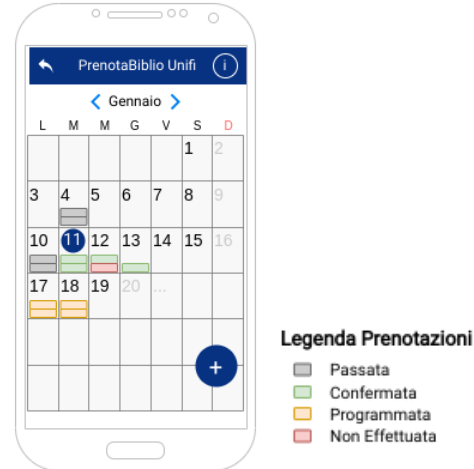


Figura 3. Mockup della vista a calendario con l'uso di colori per indicare lo stato delle prenotazioni.

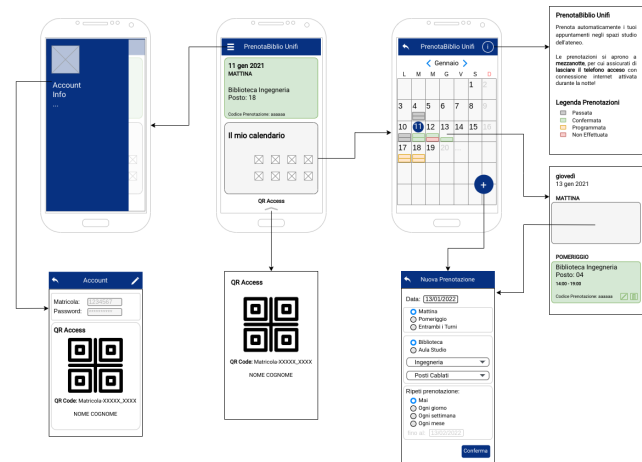


Figura 4. Mockup finale.

## 3. Implementazione

Una volta conclusa la fase di needfinding, il lavoro si è concentrato sull'implementazione vera e propria dell'applicazione, ponendo particolare attenzione al rispetto dei requisiti e seguendo le scelte di design date dai mockup.

### 3.1. Tecnologie utilizzate

L'applicazione è stata implementata con Flutter [1]: un framework open source che consente di creare applicazioni multi-platform da un unico sorgente. Questa scelta è stata effettuata nell'ottica di offrire un'applicazione funzionante sia per *Android* che per *iOS*, dato che in generale *Erika* può avere uno smartphone di una qualsiasi tipologia.

L'applicazione segue il pattern architetturale **model-view-controller**: è costituita da un insieme di viste, chiamate *pages*, che vengono navigate tramite un *navigation*

*controller* intrinseco a Flutter. Tutte le informazioni dell'utente sono contenute in un *model*, in cui grazie al plugin `flutter_secure_storage` [3] tutti i dati sono mantenuti in modo sicuro e persistente, e vengono forniti alle viste quando necessario.

Un altro elemento fondamentale della app è quello denominato `kairos_utils`, che contiene l'insieme di funzionalità per interfacciarsi a Kairos e che permette di effettuare una prenotazione, cancellarla, verificare le credenziali, e così via. L'insieme di queste funzioni è stato implementato facendo uso di una *WebView* che viene automatizzata con *JavaScript* per portare a termine la richiesta.

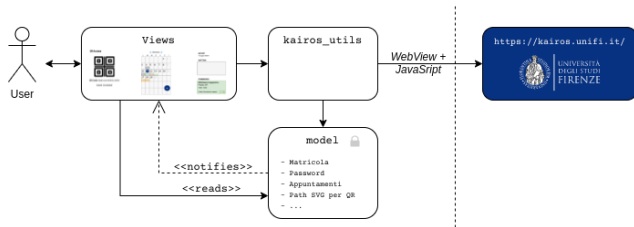


Figura 5. Schema dell'applicazione.

Il tutto funziona completamente in locale, non è necessario alcun lato server.

### 3.2. L'applicazione

Al primo avvio della app si apre la schermata di login, mostrata in figura 6. Qui sono richieste le credenziali di accesso a Kairos dell'utente: una volta inserite, `kairos_utils` si occupa di verificarne la validità; se queste risultano corrette vengono aggiunte al *model* per poterle utilizzare successivamente.

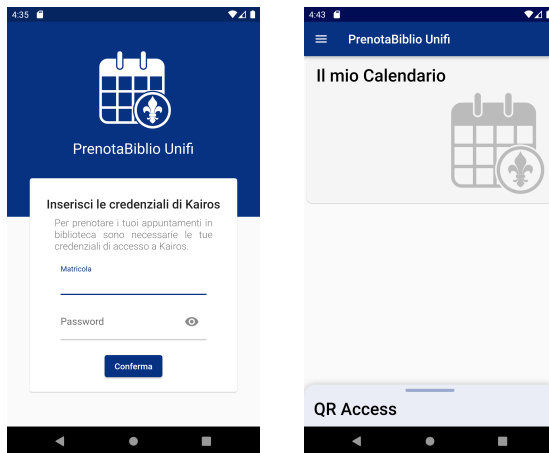


Figura 6. A sinistra la schermata di login mostrata alla prima apertura della app, mentre a destra la home page.

La home page è la pagina principale dell'applicazione; qui con uno *swipe-up* è possibile visualizzare il QR di ac-

cesso, mentre con un *tap* su "Il mio Calendario" si passa alla vista contenente tutte le prenotazioni, organizzate in un calendario con vista mensile. Nel caso in cui vi siano per la data odierna delle prenotazioni, nella home page appaiono i loro dettagli.

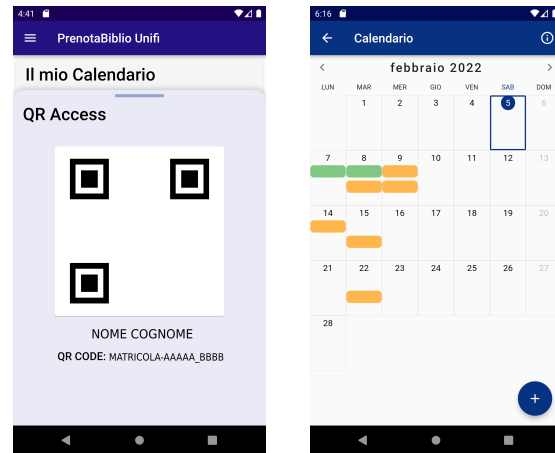


Figura 7. A sinistra il QR, a destra il calendario delle prenotazioni.

Nel calendario sono mostrate in verde le prenotazioni confermate, in arancione quelle programmate, in grigio quelle precedenti ed infine in rosso quelle in cui la prenotazione non è stata effettuata a causa di un errore. La posizione del blocchetto colorato permette di comprendere subito se una prenotazione si riferisce al turno di mattina o a quello di pomeriggio. Facendo *tap* su una data si accede alla pagina del giorno selezionato. Con un *tap* sul bottone in basso a destra si accede invece all'interfaccia di creazione di una nuova prenotazione.

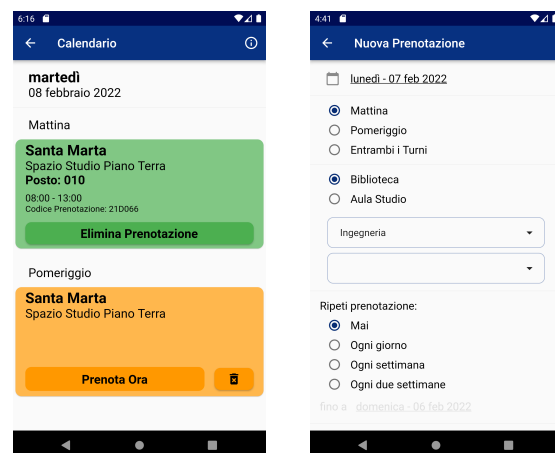


Figura 8. A sinistra la pagina relativa all'8 febbraio, a destra l'interfaccia di creazione di nuove prenotazioni.

Dalla vista giornaliera del calendario ci si può spostare da una data all'altra con uno *swipe* orizzontale. A seconda

dello stato delle prenotazioni presenti sono possibili diverse azioni, come mostrato in figura 8 a sinistra. Prima di procedere con azioni critiche, come ad esempio eliminare una prenotazione, si apre un *AlertDialog* per chiedere conferma all'utente.

L'interfaccia di creazione delle prenotazioni consente di scegliere la data di prenotazione, il turno, la sede ed eventualmente se questa vada ripetuta. Sono selezionabili tutti gli spazi studio dell'ateneo. Quando una prenotazione viene aggiunta con successo si apre un *AlertDialog* per dare feedback all'utente.

L'applicazione inoltre è abilitata all'uso delle **notifiche**. Tramite queste offre all'utente dei *reminder* per ricordare le prenotazioni per il giorno successivo, così che sia ancora possibile effettuare delle eliminazioni.

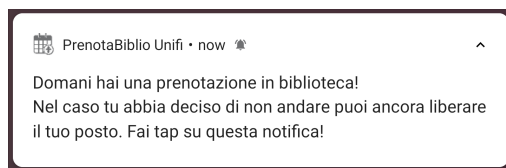


Figura 9. Reminder di prenotazione.

Sempre tramite le notifiche viene chiamato il bot di prenotazione notturno: tutte le volte che è possibile effettuare una prenotazione durante la notte, a mezzanotte si ha una notifica di tipologia *full-screen* che mostra all'utente l'interfaccia in figura 10 e che fa uso di *kairos\_utils* per portare a termine le prenotazioni. Questo funziona anche se lo smartphone è bloccato, purché la connessione Internet sia attiva.



Figura 10. Prenotazione notturna.

Su *iOS* sfortunatamente non sono disponibili le notifiche *full-screen*: è necessario un *tap* sulla notifica per attivare il bot di prenotazione, cosa che può essere fatta comodamente anche la mattina successiva.

## 4. Usability Testing

Al fine di valutare quanto l'applicazione finale fosse *user-friendly* è stato effettuato un *usability test*. Coloro che hanno partecipato al test si allineano a quella che è *Erika*: sono 10 studenti dell'università di Firenze, assidui frequentatori delle biblioteche dell'ateneo, con età compresa tra 20 e 27 anni.

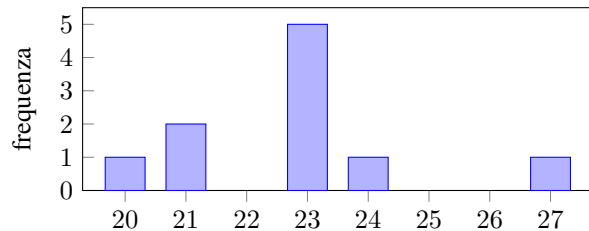


Figura 11. Densità dell'età dei partecipanti mostrata in frequenza.

Il test viene effettuato in un ambiente informale, riprendendo le caratteristiche di un *Test di Hallaway*, proponendo agli intervistati 5 task di tipo diretto, che si allineano agli scenari di utilizzo dell'applicazione e che permettono all'utente di interagire con le varie funzionalità:

1. *Apri la app e fai il login.*
2. *Scarica il tuo QR di accesso, così da poterlo utilizzare quando entri/esci in biblioteca.*
3. *Apri il calendario e programma una prenotazione per la tua biblioteca preferita questo lunedì mattina. Una volta programmata procedi a prenotarla (se è possibile).*
4. *Apri il calendario e programma delle prenotazioni per tutti i lunedì da qui al 21 marzo per la tua biblioteca preferita, per il turno pomeridiano. Prenota gli appuntamenti "prenotabili" qualora ve ne siano.*
5. *Apri il calendario e prenotati per questo martedì in aula-studio a Santa Marta per entrambi i turni (mattina e pomeriggio). Una volta confermate le prenotazioni annullale entrambe.*

Al termine di ciascuno dei task vi sono delle domande specifiche su di esso, che permettono di avere una valutazione istantanea di come l'utente si è trovato a dover svolgere il compito assegnato. Al termine di tutti i task sono riportate dunque delle domande di carattere generale, per acquisire le impressioni complessive sulla app.

Tutte le domande rivolte agli intervistati sono di tipologia *Single Ease Question (SEQ)* con una scala da 1 a 7, in cui 1 indica forte dissenso con la frase in oggetto e 7 invece il completo assenso. Alcune domande vengono formulate con logica negativa per fare in modo che l'intervistato

		$\mu$	$\sigma$	% d'accordo
<i>Task 1: apri la app e fai login.</i>				
1.1	Sono riuscito a fare il login facilmente	6.80	0.632	100%
1.2	È stato chiaro capire come fare login	7.00	0.000	100%
1.3	Ci sono stati problemi di rete	2.30	2.163	20%
1.4	Nel caso abbia sbagliato la matricola/password, ho subito capito cosa non andava	7.00	0.000	1 persona
<i>Task 2: scarica il tuo QR di accesso.</i>				
2.1	Sono riuscito a scaricare il QR facilmente	6.90	0.316	100%
2.2	È difficile capire cosa fare per scaricare il QR	1.10	0.316	0%
2.3	La posizione del QR è pratica	6.90	0.316	100%
<i>Task 3: prenota per lunedì mattina.</i>				
3.1	Aggiungere la prenotazione è stato complicato	1.10	0.316	0%
3.2	L'uso dei colori sul calendario mi ha aiutato a capire in che stato fosse la prenotazione	6.80	0.422	100%
3.3	Ho avuto difficoltà a trovare la mia biblioteca preferita tra le varie opzioni	1.10	0.316	0%
3.4	Una volta aggiunta la prenotazione mi è stato chiaro che ancora dovevo procedere a prenotarla	4.80	2.044	60%
3.5	Prenotare è stato facile	6.60	0.699	100%
3.6	I dati della prenotazione confermata sono chiari ed esatti	6.50	1.581	90%
<i>Task 4: effettua una prenotazione che si ripete settimanalmente.</i>				
4.1	Aggiungere tutte queste prenotazioni è stato complicato	1.60	1.578	10%
4.2	Non era facile aggiungere tutte queste prenotazioni, ma adesso ho imparato e potrei farlo molto più rapidamente	4.10	2.726	50%
4.3	È facile capire se una prenotazione programmata può essere subito prenotata	6.30	0.675	100%
<i>Task 5: effettua due prenotazioni, confermale e poi annulla entrambe.</i>				
5.1	È stato complicato annullare le prenotazioni una volta aggiunte	1.00	0.000	0%
5.2	Ho avuto difficoltà a capire che le due prenotazioni fossero confermate	1.80	1.619	10%
5.3	Non ho capito se le prenotazioni sono state annullate con successo	1.40	1.265	10%
<i>Domande di Riepilogo</i>				
6.1	L'applicazione presenta pochi errori	6.50	0.707	100%
6.2	L'applicazione è reattiva e non si blocca	6.40	1.0750	90%
6.3	La navigazione del calendario è macchinosa	1.60	0.843	0%
6.4	Le icone rispecchiano il comportamento della app	6.80	0.422	100%
6.5	Risulta chiaro come procedere ad effettuare una prenotazione	6.20	0.789	100%
6.6	Risulta chiaro come procedere ad eliminare una prenotazione	6.70	0.675	100%
6.7	L'applicazione è difficile da utilizzare	1.10	0.316	0%
6.8	L'applicazione soddisfa le mie necessità	6.90	0.316	100%
6.9	Potrei fare uso esclusivamente di questa applicazione invece di usare Kairos	6.90	0.316	100%

Tabella 1. Domande rivolte agli intervistati durante l'*usability-test* e risultati ottenuti.

ponga attenzione alle sue risposte. Vi è infine una sezione a testo libero in cui l'intervistato può aggiungere note o suggerimenti.

In tabella 1 sono riportate in dettaglio le domande formulate per ciascun task e quelle di riepilogo, assieme ad una sintesi dei risultati ottenuti in termini di media, deviazione standard e percentuale di intervistati in accordo con la frase in oggetto.

Analizzando i risultati ottenuti sono state individuate le seguenti criticità dell'applicazione:

- Il *Task 1* era relativo alla schermata di login: nella sua interfaccia risulta chiara, tuttavia un po' lenta nella

fase di verifica delle credenziali.

- Il *Task 2* relativo al QR presenta un'interfaccia chiara e pratica. Alcuni studenti tuttavia si sono domandati perché sia necessario re-immettere le proprie credenziali di accesso per scaricare il QR, dato che già nel *Task 1* erano state aggiunte. Purtroppo il sito web da cui si estrae il QR<sup>1</sup> gestisce gli eventi di input un carattere alla volta, triggerando molteplici *EventListeners*. Questo ha reso difficoltoso la creazione di un bot.

<sup>1</sup><https://qraccess.unifi.it/>

- Nel *Task 3* è emerso come il processo per aggiungere e poi confermare una prenotazione sia poco chiaro: con l'interfaccia di creazione in figura 8 si aggiungono delle prenotazioni che sono **programmate**, e quindi mostrate in arancione. Poi, qualora si possano prenotare, si può fare *tap* su "*Prenota Ora*" e tramite `kairos_utils` verranno confermate. Questo processo non è stato chiaro agli utenti - come si nota dai risultati ottenuti alla domanda 3.4 - per due motivi principali che sono di immediata correzione:

- Una volta premuto il tasto "*Conferma*" nell'interfaccia di creazione prenotazione, appare un *AlertDialog* che notifica all'utente che le prenotazioni sono state aggiunte correttamente. Chiuso il *dialog* però la navigazione resta sulla pagina di creazione prenotazione e non torna al calendario, così che l'utente veda in arancione le nuove prenotazioni appena aggiunte.
- A livello lessicale è stato utilizzato "*Aggiungi prenotazione*" e "*Nuova prenotazione*" per indicare le prenotazioni programmate, facendo sì che l'utente non si trovi mai di fronte al fatto che "*programmata*" sia una cosa diversa da "*confermata*" se non per i colori distinti.

Si può dunque scegliere con più attenzione il lessico utilizzato e curare cosa accade una volta aggiunta una prenotazione programmata, così da offrire feedback più chiaro all'utente. In ogni caso, una volta superata questa difficoltà, tutti gli utenti sono riusciti a completare il task e, come mostrato dai risultati ottenuti alle domande 3.5 e 6.5, hanno trovato il processo di prenotazione più che soddisfacente. Si può dunque concludere che il processo di prenotazione sia di facile apprendibilità, anche perché nei task successivi gli utenti non hanno avuto difficoltà a procedere con le prenotazioni.

- Il *Task 4* ha permesso agli utenti di testare la funzionalità di prenotazioni programmate che si ripetono. Molti di essi, come si evince dai risultati ottenuti alla domanda 4.1, hanno trovato il task molto semplice e quindi alla domanda 4.2 hanno risposto con un 1, altri invece lo hanno trovato un po' più complicato e dunque hanno risposto 6 o 7 alla domanda 4.2. Questo spiega i risultati ottenuti a quest'ultima, che presentano una media molto vicina al 4 e grande deviazione standard.

Per quanto riguarda il *Task 5* non sono state evidenziate particolari criticità. Nel complesso, come mostrato dalle domande di riepilogo, PrenotaBiblio Unifi è di facile utilizzo e soddisfa ampiamente le necessità degli intervistati, al netto di qualche difficoltà nel processo di creazione delle prenotazioni programmate. L'uso del calendario è intuitivo

e i colori sono utili a capire lo stato delle prenotazioni. Infine le risposte ottenute alla domanda 6.9 dimostrano ancora quanto questa applicazione sia apprezzata dagli studenti, tanto da preferirla a Kairos.

## 5. Ulteriori Sviluppi

Come notato nella sezione precedente l'applicazione presenta un buon grado di maturità, è di facile utilizzo e può migliorare risolvendo le criticità che sono emerse. La diffusione di uno strumento di questo tipo fa sorgere tuttavia degli interrogativi: un prodotto che si interfaccia ad un sito web dell'ateneo per offrire dei vantaggi a chi lo usa potrebbe sollevare critiche di carattere etico.

Resta in ogni caso evidente quanto gli studenti in questo momento non siano soddisfatti di Kairos e come PrenotaBiblio Unifi si allinei con le loro esigenze.

## Riferimenti bibliografici

- [1] Google. Flutter. <https://flutter.dev/>.
- [2] JGraph Ltd. diagrams.net. <https://github.com/jgraph/drawio>.
- [3] German Saprykin. flutter\_secure\_storage. [https://pub.dev/packages/flutter\\_secure\\_storage](https://pub.dev/packages/flutter_secure_storage).