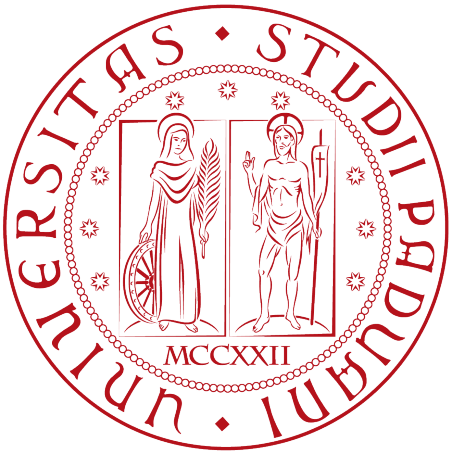
Università degli Studi di Padova

Dipartimento di Matematica "Tullio Levi Civita"

Corso di Laurea in Informatica



Autenticazione biometrica in ambiente Android

Tesi di laurea triennale

Relatore

Prof. Gilberto Filè

Laureando

Giulia Petenazzi

Anno Accademico 2016/2017

Sommario

Il presente documento descrive il lavoro svolto durante il periodo di stage, della durata di circa trecento ore, dalla laureanda Giulia Petenazzi presso l'azienda Wintech S.p.A.. L'obbiettivo principale dello stage era il confronto delle soluzioni di autenticazione biometrica presenti sul mercato, e la scelta di una di esse, nell'ottica di sviluppo di una applicazione Android integrabile con i sistemi di Wintech S.p.A..

Il mercato di oggi offre varie soluzioni di autenticazione biometrica: riconoscimento dell'iride, riconoscimento dell'impronta digitale, riconoscimento facciale, riconoscimento vocale. Per questo stage era quindi richiesta una figura dotata di competenze tecniche informatiche che si documentasse sulle varie possibilità in maniera autonoma e che individuasse punti di forza e problematiche di ciascuna soluzione.
Era inoltre necessario che lo stagista si sapesse interfacciare con altre aziende e che fosse in grado di effettuare attente analisi costi-benefici tra le soluzioni, al fine di trovare gli strumenti necessari per implementare una soluzione che effettuasse l'autenticazione biometrica integrabile con i sistemi aziendali.

In secondo luogo, era richiesto lo sviluppo di un prototipo di applicazione Android che utilizzasse gli strumenti scelti nel primo periodo di stage. Il prototipo avrebbe dovuto consentire l'autenticazione biometrica di un utente.

Nell'ultimo periodo lo stagista si sarebbe occupato di creare un esempio di integrazione del prototipo con il sistema aziendale. Opzionalmente si sarebbe dovuto mostrare che un certo video contenuto all'interno della piattaforma multimediale aziendale si sarebbe sbloccato in caso di autenticazione avvenuta con successo. Nel fare ciò, in particolare, si sarebbe dovuto rispettare quanto prescritto dal brevetto che l'azienda possiede riguardante l'autenticazione biometrica.
Al termine dello stage inoltre, l'azienda ha richiesto di mostrare i risultati al presidente e ad altre figure di rilievo in azienda attraverso due presentazioni (commerciale e tecnica).

Riguardo la stesura del testo sono state adottate le seguenti convenzioni tipografiche:
- i termini ambigui o di uso non comune vengono definiti nel glossario (in azzurro);
- termini facenti parti del gergo tecnico sono evidenziati con il carattere *corsivo*;
- termini indicanti parti di codice sono evidenziati con il carattere `monospace`;

Ringraziamenti

Innanzitutto, vorrei esprimere la mia gratitudine al Prof. Gilberto Filè, relatore della mia tesi, per l'aiuto e il sostegno fornitomi durante la stesura del lavoro.

Desidero ringraziare i miei genitori Giuseppe e Clara per il sostegno, i consigli e gli aiuti che mi hanno offerto durante gli anni di studio, e mio fratello Manuel per la sua vicinanza in questi anni e per aver ravvivato la quotidianità vissuta in famiglia.

Ringrazio i miei amici e amiche per tutti i bellissimi anni passati insieme e le mille avventure vissute, assolutamente indimenticabili, insieme ai miei compagni di corso, per le giornate in università e i momenti di festa trascorsi insieme.

Ringrazio il "Buso della Giaretta" che mi ha insegnato a farmi spazio nel mondo del lavoro e tutto il suo staff con cui mi sono confrontata e con cui ho passato molte serate in allegria negli ultimi anni.

Ringrazio Jordan per le piccole e sincere attenzioni che mi dedica ogni giorno, per gli aiuti tecnici, per l'affetto dimostratomi in questi mesi.

Ringrazio tutte le altre persone a me care che conosco e ho conosciuto e che hanno contribuito, anche solo per un momento, a farmi diventare la persona che sono oggi.

Padova, Settembre 2017 Giulia Petenazzi

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

L'azienda: Wintech S.p.a

1.1 Contesto aziendale

figura 1.1: Logo di Wintech S.p.A.

Wintech S.p.A è un'azienda nata nel 1987 e che si occupa di system integration. Ha una sede principale a Padova e due filiali a Milano e Bassano del Grappa, e conta circa 80 dipendenti suddivisi su sei business unit. Grazie alla propria esperienza, competenza e creatività, trasforma le complessità tecnologiche in soluzioni informatiche innovative, efficienti e dal facile utilizzo. Wintech fornisce consulenza personalizzata e soluzioni IT, per ottimizzare i processi aziendali e raggiungere gli obiettivi definiti assieme al cliente. Grazie all'esperienza maturata negli anni, Wintech vanta numerose partnership a livello mondiale tra cui Microsoft e IBM.

1.1.1 Servizi

L'azienda offre ai clienti servizi quali:

- analisi e progettazione di soluzioni su misura;
- installazione ed avviamento delle soluzioni;
- formazione per l'uso delle applicazioni proposte;
- consulenza;
- assistenza, grazie all'ausilio di software quali Team Viewer.

1.1.2 Prodotti

Vantando una vasta rete di clienti, Wintech è in grado di fornire prodotti ad hoc per ognuno di essi:

- per i professionisti: applicativi per la consulenza amministrativa e fiscale;
- per le aziende di moda: applicativi per la gestione di preventivi, vendite, logistica;
- per le piccole e medie imprese: applicativi per attività gestionali e finanziarie;
- per le Grandi Aziende: mirati al ripristino dei sistemi e dei dati;
- per Banche e assicurazioni: in particolare viene proposta la firma grafometrica;
- per la Pubblica amministrazione: mirati in particolare alla gestione documentale;

1.2 Organizzazione interna

1.2.1 Strumenti organizzativi

L'organizzazione del lavoro all'interno di Wintech è, allo stato attuale, supportata da applicazioni distinte l'una dall'altra tra le quali riportiamo **TdB** ("Tableau de Bord", gestisce anagrafiche, offerte, fatture per cliente) e **GeCo** (software di consuntivazione sulle attività svolte). Le diverse business unit utilizzano poi inoltre le proprie applicazioni specifiche.
Wintech ha sentito il bisogno di disporre di un'unica piattaforma di Digital Workplace (denominata **WOW**, World of Wintech) che consentisse di gestire varie applicazioni (calendario, consuntivazione, anagrafiche, ordini, dashboard...) e processi in maniera integrata, condividendo informazioni di utilità comune. WOW in un futuro potrebbe diventare un prodotto rivendibile e personalizzabile in base alle necessità del cliente.

1.2.2 Approccio Agile

Con l'obiettivo di focalizzarsi sempre sul client,e Wintech adotta nello sviluppo software, la metodologia agile *Scrum*. Nella metodologia *Scrum*, il carico di lavoro viene suddiviso in task che vengono inseriti in cicli di produzione chiamati *sprint*. Ciascuno *sprint* dura circa due settimane, durante le quali vengono sviluppate, funzionalità aggiuntive rispetto allo *sprint* precedente al termine del quale dovrebbe essere presentato al cliente un incremento potenzialmente rilasciabile.

1.2.3 Tecnologie di supporto

1.2.3.1 Gestione di progetto

Come strumento di supporto al project management Wintech utilizza **Jira 5** . Si tratta di un'applicazione web che permette, oltre le classiche funzionalità di uno strumento di gestione di progetto, di visualizzare una Scrum Board in cui si ha una visibilità immediata sullo stato dello *sprint*.

1.2.3.2 Ambiente di sviluppo

Per la maggior parte i computer di Wintech ospitano **Windows**, in varie versioni, ed in alcuni casi Linux. Per quanto riguarda l'ambiente di sviluppo viene utilizzato Eclipse 6, tuttavia, ogni sviluppatore è libero di utilizzare ciò che meglio crede nello svolgimento del proprio lavoro.

1.2.3.3 Versionamento, integrazione continua e test

Wintech, per il versionamento del codice, utilizza **Subversion 7**. Per l'integrazione continua l'azienda utilizza Jenkins 8. Per l'esecuzione dei test automatici vengono utilizzati, a backend, JUnit 9 e Mokito 10 , a frontend invece, Karma 11 e Jasmine 12. Ad ogni commit da parte di un membro del team parte automaticamente, sul server di integrazione, un processo che esegue la build del codice e notifica in chat.

1.2.4 Certificazioni

Riguardo i processi aziendali, Wintech ha di recente (2017) investito nella sicurezza ottenendo **certificazione UNI ISO 27001:2014** per i servizi sistemistici e di network forniti in outsourcing, offrendo soluzioni ancora più complete ed affidabili ai propri clienti.
Inoltre il sistema di qualità di Wintech è stato certificato secondo la norma ISO 9001:2008 per le attività di:

- progettazione, implementazione e fornitura di sistemi informativi integrati;
- progettazione e sviluppo di soluzioni software;
- erogazione di servizi di assistenza tecnica, sistematica e applicativa.

Le colonne portanti di questa certificazione sono la riservatezza dei dati, l'integrità e la disponibilità delle informazioni.

figura 1.2: Elementi caratteristici di Wintech S.p.A.

1.2.5 Ricerca e innovazione

Wintech, anche se nata vent'anni fa, ha sempre cercato di mantenersi costantemente aggiornata in modo da essere competitiva sul mercato. Cerca di essere innovativa nei prodotti, utilizzando tecnologie all'avanguardia e aggiornate, nel personale, spendendo risorse per la loro formazione continua e accogliendo stagisti, e nell'organizzazione, e utilizzando la metodologia Agile.

1.3 Collaborazione con Vision Learning

1.3.1 L'azienda: Vision Learning

figura 1.3: Logo di Vision Learning

Vision Learning è una azienda partner di Wintech che si occupa di videoformazione. Offre una piattaforma che consente la formazione a distanza, oltre che lo streaming in diretta di eventi live. In questo modo i clienti possono proporre eventi in diretta video in rete, piuttosto che un meeting, una lezione oppure uno spettacolo, sempre in modo partecipato e interattivo.

1.3.2 I servizi

1.3.2.1 Interactive Web Live Streaming

E' il servizio principale offerto dall'azienda: grazie infatti ad una particolari tecniche di regìa, l'utente può assistere via web all'esposizione di un relatore o docente accompagnata dalla grafica della sua presentazione, in alta qualità, pur con un modesto impegno di banda. Questo servizio è disponibile anche, tramite registrazione, *on-demand*.

1.3.2.2 e-Learning on demand

Attraverso questo servizio la pratica dell'e-Learning diventa strumento di valorizzazione e trasmissione delle competenze dal docente al discente, mantenendo viva l'attenzione e l'interazione, focalizzandosi sul risultato formativo.

1.3.2.3 Learning Management System

Si tratta dell'ambiente che ospita la formazione on-line in cui viene creato l'ambiente virtuale personale dotato di funzionalità come il caricamento o il download di materiale didattico-documentale e controllo delle valutazioni e dell'apprendimento dei discenti.

1.3.2.4 e-Learning certificato

Si tratta di avere formazione a distanza che possedendo garanzie sull'effettiva presenza del destinatario dell'azione formativa, al quale andranno poi attribuiti i crediti formativi, gli attestati od altri titoli formali. Vision Learning ha avviato una approfondita ricerca che si è conclusa con l'ideazione brevettuale e la realizzazione di un sistema probatorio di presenza del discente durante la fruizione di corsi on-line.

Capitolo 2

Il progetto di stage

2.1 Motivazioni e vincoli dello stage

La scelta di stage è ricaduta in Wintech principalmente perché l'argomento di stage, l'autenticazione biometrica in ambiente *Android*, era particolarmente interessante ai miei occhi, ma anche perché, in seguito a una visita in azienda, ho notato una struttura aziendale solida, l'uso di tecnologie all'avanguardia e un ambiente collaborativo, elementi che avrebbero potuto farmi crescere sia a livello tecnico-lavorativo sia personale.
Per quanto riguarda i vincoli, la durata massima dello stage era prevista di 312 ore, ed era richiesto di occuparsi dell'autenticazione biometrica in ambiente *Android*, pur avendo riguardo per la portabilità della soluzione in dispositivi *iOS*, e in caso si fosse arrivati all'integrazione con il server di Vision Learning si sarebbe dovuto usare lo stile architetturale REST.

2.2 Obbiettivi dello stage

Il mio stage sarebbe servito a Wintech come *proof of concept* per il progetto che l'azienda stava ipotizzando, ovvero quello di utilizzare l'autenticazione biometrica nei sistemi di Vision Learning.
Lo stage si componeva quindi di un'importante parte di ricerca e, se e solo se tale ricerca avesse portato qualche soluzione attuabile, anche in termini di costi-benefici, si sarebbe passati alla creazione di un prototipo funzionante, ed eventualmente alla sua integrazione nella piattaforma aziendale.
Proprio a causa della natura del progetto gli obbiettivi obbligatori riguardavano studio e confronto dei sistemi di autenticazione biometrica presenti sul mercato, gli obbiettivi riguardanti il prototipo erano desiderabil, mentre erano facoltativi gli obbiettivi riguardanti l'integrazione.

Gli obbiettivi dello stage definiti in principio erano i seguenti:

* **Obbligatori**
  - *Ob1*: studio dell'attuale stato di Vision Learning, in particolare:
    * *Ob1.1*: studio del contesto e orientamento aziendale di Vision Learning;
    * *Ob1.2*: studio del sistema attuale di Vision Learning;
    * *Ob1.3*: studio delle modalità attuali di autenticazione all'interno del sistema di Vision Learning;
  - *Ob2*: analisi dei costi e benefici dei sistemi di autenticazione biometrica, in particolare:
    * *Ob2.1*: confronto dei sistemi presenti sul mercato e scelta di uno tra essi;
    * *Ob2.2*: stesura resoconto;
  - *Ob3*: studio del sistema di autenticazione scelto di cui al punto *Ob2.1*;
  - *Ob4*: studio di *Android*;
* **Desiderabili**
  - *De1*: progettazione di un prototipo di applicazione *Android* che consenta l'autenticazione biometrica, in particolare:
    * *De1.1*: il prototipo deve consentire l'autenticazione biometrica;
    * *De1.2*: il prototipo deve essere integrabile nel sistema di Vision Learning;
    * *Fa1.3*: il prototipo deve essere integrabile con i player multimediali di Vision Learning.
* **Facoltativi**
  - *Fa1*: realizzazione e testing del prototipo di cui al punto *De1*;
  - *Fa2*: stesura di documentazione minimale del prototipo di cui al punto *De1*;
  - *Fa3*: integrazione del prototipo di cui al punto *De1* nel sistema di Vision Learning;
  - *Fa4*: integrazione del prototipo di cui al punto *De1* nei player multimediali di Vision Learning.

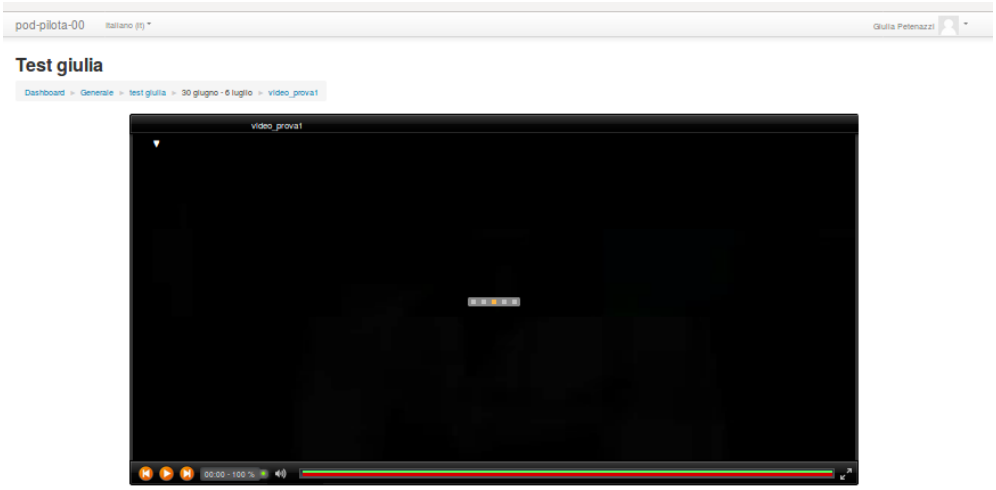


figura 2.1: Player da sbloccare al termine dell'autenticazione

2.3 Pianificazione

In accordo con l'azienda è stata stabilita una pianificazione a granularità settimanale dello stage che avrebbe compreso un primo periodo dedicato alla ricerca della libreria da utilizzare e un secondo periodo dedicato allo sviluppo del prototipo. La pianificazione è stata estesa da 8 a 10 settimane, in modo da tener conto preventivamente di alcuni giorni di assenza dovuti a motivi personali/universitari.

	Settimana	1°	2°	3°	4°	5°	6°	7°	8°	9°	10°
	Mese	Maggio			Giugno				Luglio		
	Date (dal - al)	17-19	22-26	29-01	05-09	12-17	19-23	26-30	03-07	10-14	17-21
Durata	Attività										
32	Analisi del sistema di Vision Learning										
72	Analisi dei sistemi di autenticazione biometrica										
48	- confronto tra le soluzioni										
24	- scelta di una soluzione, stesura resoconto										
68	Studio del sistema scelto										
56	Studio di Android										
60	Realizzazione prototipo Android										
24	- progettazione										
20	- codifica										
8	- testing										
8	- stesura documentazione										
24	Integrazione del prototipo nella piattaforma aziendale										
312	Totale										

figura 2.2: Pianificazione settimanale dello stage

Capitolo 3

Ricerca

3.1 Studio del contesto

3.1.1 Dominio e utilizzatori finali

Nei primi giorni in Wintech mi sono dedicata alla configurazione del mio ambiente di lavoro e allo studio del contesto nel quale andava a inserirsi il mio stage. Come già spiegato nel primo capitolo, Wintech è un system integrator e ha una collaborazione con Vision Learning che offre, tra i vari servizi, una piattaforma e-learning. Durante il mio progetto di stage mi sarei dovuta interfacciare con Vision Learning in quanto lo stage avrebbe anche toccato il mondo della formazione professionale dei professionisti. Secondo il *Regolamento della formazione professionale continua*, approvato dal Consiglio Nazionale il 3 dicembre 2015 viene specificato, all'articolo 4 comma 2 che "ogni iscritto nell'Albo è tenuto ad acquisire in un triennio formativo 90 crediti formativi professionali". Il comma 4 dello stesso articolo parla della formazione a distanza, in particolare, in caso di e-Learning tradizionale, "gli iscritti possono acquisire un massimo di 20 crediti formativi annui". Quello che è rilevante è che

"tramite le attività di formazione a distanza che utilizzano tecnologie di identificazione biometrica, gli iscritti possono acquisire senza alcun limite crediti formativi"

Questa è un'importante svolta per Vision Learning: potrebbe rilasciare l'innovativo servizio dell'e-learning certificato, a cui potranno essere interessati anche i numerosi clienti di Wintech appartenenti agli *ordini*.

3.1.2 Analisi dei rischi di soluzioni biometriche

Come punto di partenza ho fatto un'analisi dei rischi per capire quali problematiche avrei potuto incontrare in relazione all'autenticazione biometrica.

3.1.2.1 Furto d'identità biometrica

E' la prima problematica legata all'autenticazione biometrica. Il furto di identità biometrica può causare infatti effetti lesivi rilevanti nei confronti degli interessati in quanto non può essere fornita una nuova identità biometrica che utilizzi la stessa tipologia di dato biometrico, diversamente dai sistemi di riconoscimento tradizionali. Le caratteristiche biometriche, infatti, poiché normalmente non modificabili e inscindibilmente legate all'individuo (seppur soggette in misura variabile a deterioramento in base all'età, al tipo di caratteristica, alle attività e agli stili di vita dell'interessato), costituiscono una sorta di credenziale di autenticazione non revocabile e non sostituibile la cui appropriazione da parte di soggetti non legittimati può prestarsi alla realizzazione di azioni fraudolente e compromettere l'efficacia di sistemi di sicurezza basati sul riconoscimento biometrico.

3.1.2.2 Inaccuratezza del riconoscimento biometrico

Il riconoscimento biometrico avviene generalmente su base statistica e non deterministica, ed è dunque suscettibile di errore. Due dei più importanti parametri tecnici da considerare, connessi a un sistema biometrico, sono:

* **FRR**: il tasso dei falsi rigetti (false rejection rate) o "falsi negativi" ;
* **FAR**: il tasso delle false accettazioni (false acceptance rate) o "falsi positivi".

Per questo motivo, le prestazioni del sistema biometrico vanno attentamente valutate in funzione delle finalità d'uso: un alto tasso di falsi positivi, abilita, erroneamente, l'accesso a utenti non autorizzati creando situazioni di pericolo per persone, cose o informazioni.

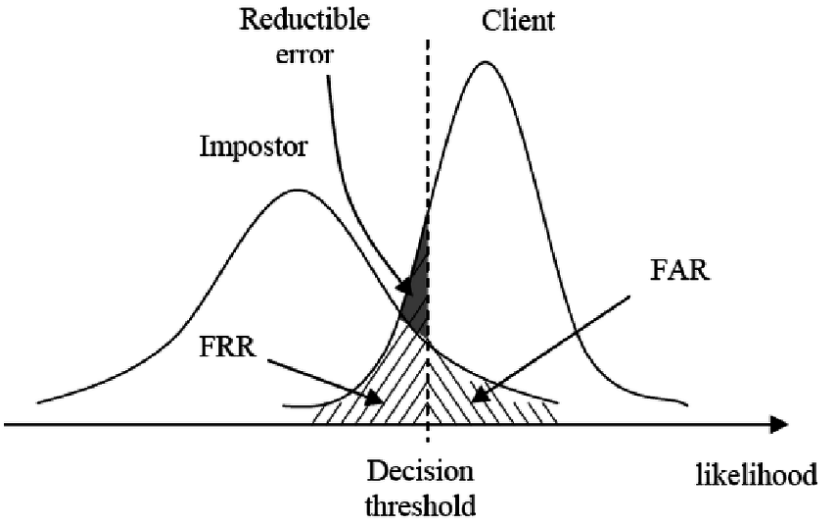


figura 3.1: FAR e FFR

3.1.2.3 Spoofing biometrico

C'è da considerare anche il rischio della falsificazione di alcune caratteristiche biometriche derivante dalla creazione di una caratteristica biometrica artificiale a partire da tracce rilevate al di fuori del sistema biometrico. L'esempio tipico è quello delle impronte digitali, mediante creazione di una sorta di "dito artificiale" che riproduca le sembianze anatomiche del polpastrello, contro cui sono ben efficaci misure tecniche in grado di garantire la genuinità della caratteristica rilevata dal dispositivo di acquisizione (come le funzioni di liveness test presenti in alcuni sensori per il rilevamento dell'impronta digitale). La creazione del modello biometrico dovrebbe essere sempre, qualora tecnicamente possibile, un processo univoco e non reversibile: non dovrebbe essere possibile, infatti, ricreare il campione biometrico a partire dal modello, dando luogo a una "ricostruzione" non autorizzata di una caratteristica biometrica.

3.1.3 Adempimenti giuridici

E' importante che Wintech, in caso di adozione di tali sistemi, eserciti le attività di autenticazione in conformità alle disposizioni del Codice, e a condizione che non si determini un'ingerenza ingiustificata e sproporzionata nei confronti degli interessati. In linea generale Wintech prima di iniziare il trattamento, dovrebbe, di regola, acquisire il consenso informato dell'interessato, e i sistemi informativi e i programmi informatici devono essere configurati riducendo al minimo l'utilizzazione di dati personali e di dati identificativi. Possono essere trattati i soli dati pertinenti e non eccedenti in relazione alle finalità perseguite. I dati oggetto di trattamento per mezzo di sistemi biometrici devono essere raccolti in maniera accurata e trattati per le sole finalità che il titolare intende legittimamente perseguire. Oltre a ciò si doveva contare che la applicazione che io sarei andata a creare avrebbe dovuto soddisfare il brevetto che Vision Learning possiede per la gestione dell'autenticazione biometrica all'interno dei loro sistemi.

3.2 Requisiti della libreria

Sulla base dello studio del contesto ho iniziato a cercare la soluzione più adeguata al problema di Wintech. Attraverso incontri più specifici con il mio tutor aziendale e con alcuni membri del personale di Vision Learning, ho individuato i requisiti che la soluzione doveva avere, nell'ottica della creazione di un prototipo di applicazione Android che consentisse tale tipo di autenticazione. Nella tabella seguente riporto alcuni dei requisiti che erano richiesti (per la notazione si veda Analisi dei requisiti).

Requisito	Descrizione
ROF1	La soluzione deve avere la forma di API o di SDK
ROF1.1	La soluzione deve essere utilizzabile in *Android Studio*
RDF1.2	La soluzione deve essere disponibile per possibili sviluppi futuri in *iOS*
ROF1.3	La soluzione deve avere la documentazione di metodi e campi dati con esempi minimali di utilizzo
ROF2	La soluzione deve consentire l'enrollment e la bio-autenticazione
ROF2.1	La soluzione deve consentire il completo controllo delle procedure di enrollment e di bio-autenticazione
ROF2.1.1	La soluzione deve poter consentire il blocco degli enrollment una volta effettuato il primo
RFF2.1.2	La soluzione deve consentire il completo controllo dell'interfaccia grafica della procedura di enrollment e di bio-autenticazione
ROF2.2	Il dato biometrico dell'utente deve essere memorizzato solo e unicamente nel dispositivo dell'utente
ROF2.3	La soluzione deve effettuare il liveness test
ROF2.4	La soluzione deve implementare al suo interno il meccanismo di matching tra riferimento biometrico e istanza biometrica
ROF2.5	La soluzione deve consentire l'autenticazione e l'enrollment di un utente utilizzando solamente il suo dispositivo, senza l'ausilio di ulteriori sensori aggiuntivi o altro
ROF3	In caso di software a pagamento, la soluzione deve essere *congruo alle aspettative di Wintech* (non citato per questioni di riservatezza)

ROF4	In caso di software a pagamento, l'azienda fornitrice della soluzione dovrà fornire dei prototipi funzionanti e una documentazione minimale prima di procedere alla stipula di contratti
RDF5	L'azienda fornitrice della soluzione (a meno di tecnologie open-source) deve rispondere alle nostre richieste di chiarimento entro una settimana (limitatamente al tempo di stage)
RDQ6	Il FAR (vedi 3.1.2.2) deve essere 0 su 10 prove
RDP7	La soluzione deve effettuare un enrollment con successo entro 10 secondi, altrimenti dare errore
RDP8	La soluzione deve effettuare una bio-autenticazione con successo entro 3 secondi, altrimenti dare errore
RFQ9	Un numero minore o uguale a 2 persone su 5 può fallire l'enrollment al primo tentativo di utilizzo dell'applicazione

tabella 3.1: Requisiti della libreria

3.3 Studio di Android

Android è un sistema operativo sviluppato per dispositivi mobili sviluppato da *Google Inc.* basato su *kernel Linux*. Nella tipica applicazione per *Android* tutto è pensato per essere un *componente* pilotato dagli eventi (*Event Driven*) dell'hardware o di altri componenti. Questi ultimi, inoltre, possono condividere le loro funzionalità con altre applicazioni: se abbiamo installato nel dispositivo un'applicazione che dà la possibilità di modificare le foto per esempio possiamo sfruttarla attraverso un'altra applicazione che non dispone di tale funzionalità ma ne necessiterebbe. Android mette a disposizione quattro tipologie di componenti:

* **activity**: rappresenta una singola schermata con un'interfaccia utente;
* **service**: esso non dispone di un'interfaccia grafica ed ha il compito di eseguire in background operazioni di lunga durata, come per esempio la riproduzione musicale.
* **content provider**: neanche tale componente ha una interfaccia grafica, ma ha il compito di incapsulare i dati che devono esser condivisi tra le varie applicazioni.

* **broadcast receiver**: nemmeno questa componente ha una interfaccia grafica, ma è responsabile di catturare eventi *broadcast* provenienti dall'intero sistema, come per esempio l'evento "batteria scarica".

Dal momento in cui lanciamo l'applicazione fino al momento in cui la mandiamo in background, l'*activity* passa attraverso vari stati, fasi che rappresentano il ciclo di vita di un'app *Android*. Capire il ciclo di vita di un'*activity* è fondamentale per assicurare un corretto funzionamento dell'applicazione. Gli stati che l'*activity* assume sono:

* **Active**: l'*activity* è visibile e può ricevere degli input dall'utente;
* **Paused**: l'*activity* è parzialmente visibile ma non può ricevere alcun genere di input, per esempio quando è presente in *foreground* un messaggio di avviso;
* **Stopped**: viene assunto tale stato quando l'applicazione non è visibile;
* **Destroyed**: rappresenta l'ultimo stato di una certa istanza dell'*activity*, ovvero quando essa è rimossa dalla memoria da *Android* in seguito ad un lungo periodo di inattività oppure in caso di necessità di risorse.

Ogni transizione di stato mette a disposizione un metodo che può esser *overridato* dallo sviluppatore nel caso in cui egli volesse aggiungerci un comportamento additivo a quello standard. Inoltre, quando viene creato un nuovo progetto, viene generata una struttura a cartelle e sottocartelle. Citiamo ad esempio:

* **il file AndroidManifest.xml**, nel quale vengono descritti il comportamento e la configurazione dell'applicazione, le risorse;
* **la cartella src**, che contiene il codice sorgente dell'applicazione, il codice per l'interfaccia grafica, le classi etc;
* **la cartella res**, che contiene tutti i file e le risorse necessarie del progetto: immagini, video, layout xml etc. Può inoltre essere strutturata in sottocartelle per una migliore organizzazione delle risorse, come ad esempio drawable, layout etc.

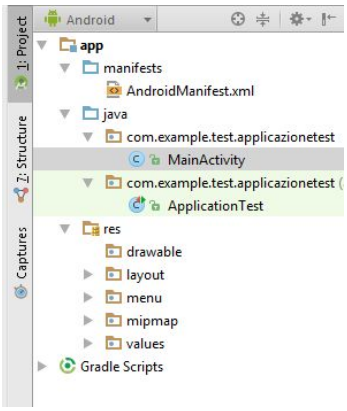


figura 3.2: Struttura di un progetto Android

3.4 Confronto tra soluzioni

3.4.1 Introduzione

Ho iniziato a cercare una *libreria* che rispettasse i requisiti (in questo capitolo utilizzerò il termine *libreria* o *soluzione* per indicare l'SDK o API da ricercare). Ho letto molti articoli in rete, per capire quale fosse la soluzione più adatta, e poi sono partita a valutare concretamente soluzioni gratuite per poi arrivare a contattare aziende che fornivano queste soluzioni professionalmente. In questo paragrafo riporterò alcune soluzioni ipotizzate con alcuni punti che hanno influito nella mia scelta. Per motivi di riservatezza, utilizzerò nomi fittizi di soluzioni e aziende (Prima soluzione, Seconda soluzione ecc...).

3.4.2 Overview sui metodi di autenticazione

3.4.2.1 Fingerprint

I sistemi biometrici basati sul fingerprint (impronte digitali) costituiscono attualmente, in termini di valore di mercato, più della metà dell'intero settore di riferimento in quanto questo tipo di riconoscimento rappresenta comunque una delle tecniche biometriche più valide. Il trattamento biometrico delle impronte digitali prevede il rilevamento, tramite dispositivi di acquisizione ottica, di un campione biometrico che riproduca la disposizione delle creste di Galton e delle valli cutanee presenti sui polpastrelli delle dita fin dalla fase prenatale. Da esse è possible ottenere un modello biometrico che fornisca una rappresentazione sintetica numerica dell'impronta di partenza
La grandezza del template, in funzione del tipo di acquisizione, varia fra le poche centinaia di bytes fino a superare il migliaio.

Vantaggi

* sul mercato esistono efficienti algoritmi di confronto
* i sistemi basati sul riconoscimento di impronta digitali, se di classe elevata, sono particolarmente robusti, in grado, ad esempio, di determinare se l'impronta digitale rilevata è quella di una persona viva;

* le impronte digitali sono stabili nel tempo: le impronte rimangono generalmente inalterate per tutta la vita (a meno di cause esterne come ferite o simili).
* molti smartphone dei giorni d'oggi contengono già un lettore di impronta digitale.
* non risente di luci e suoni dell'ambiente circostante.

Svantaggi

* gode di bassa accettazione da parte degli utenti, dovuta a una percezione negativa di tipo "storico" in quanto la rilevazione delle impronte è stata sempre generalmente associata a settori giudiziari ed investigativi;
* l'univocità del modello in una base dati biometrica non è garantita poiché (soprattutto in grandi archivi dattiloscopici che peraltro non sarebbero stato il mio contesto d'utilizzo), a più di una impronta può corrispondere un medesimo modello.

3.4.2.2 Emissione vocale

Le caratteristiche dell'emissione della voce sono, strettamente legate all'anatomia del tratto vocale, alla sua lunghezza, alle risonanze, alla morfologia della bocca e delle cavità nasali. Il riconoscimento dell'individuo viene usualmente realizzato non solo tramite l'elaborazione e l'analisi dei segnali vocali *(signal processing)*, ma anche tramite procedure di sfida dipendenti dalle modalità con le quali l'interessato viene invitato a ripetere delle frasi, nomi o numeri. E' possibile realizzare il riconoscimento anche senza sfida, nel caso in cui l'interessato è invitato a parlare senza uno schema prefissato. Normalmente il riconoscimento biometrico consiste in una verifica d'identità (confronto uno a uno, il mio caso), in cui è previsto che venga comunque fornita dall'utente un'informazione aggiuntiva nella sua disponibilità cognitiva (codice identificativo, codice utente...).

Vantaggi

* tutti i telefoni cellulari contengono un microfono al loro interno;
* l'accettazione da parte degli utenti è alta, soprattutto grazie alla mancanza di sensori da toccare o guardare e non ci sono particolari problemi nella fase di enrollment.

Svantaggi

* l'emissione vocale risente dello stato fisico dell'utente (ad esempio, la voce durante gli stati di raffreddamento può essere diversa dal normale);
* risente di eventuali qualità bassa del dispositivo di ingresso vocale o di rumori di fondo.

3.4.2.3 Forma dell'iride

Il procedimento di lettura dell'iride è una tecnica biometrica che consente la rilevazione della forma della pupilla e della parte anteriore dell'occhio mediante immagini ad alta risoluzione. E' quindi richiesta una fotocamera ad alte prestazioni. La scansione dell'iride avviene mediante una video-camera ad una distanza (per la maggior parte delle unità in commercio) di circa 40 centimetri e non richiede alcun contatto con il sensore. Nonostante le dimensioni dell'iride varino in funzione dell'illuminazione ambientale (una forte luce fa restringere la pupilla e di conseguenza aumentare il raggio della corona dell'iride), un apposito algoritmo tiene conto di queste modifiche e della eventuale copertura superiore ed inferiore dell'iride dovuta alle palpebre.
La dimensione del modello è approssimativamente di 500 bytes. I settori di applicazione concernono soprattutto applicazioni ove è richiesta una grande sicurezza.

Vantaggi

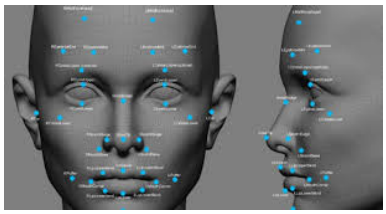
* alta velocità di comparazione;
* il FAR (vedi 3.1.2.2) è piuttosto basso rispetto ad altre caratteristiche biometriche;
* l'iride è praticamente stabile lungo tutta la vita dell'individuo e, normalmente, non è soggetta a incidenti (a differenza ad esempio della mano e delle impronte);
* le lenti a contatto e gli occhiali non interferiscono in linea di massima con il processo di acquisizione;
* l'iride ha un numero di caratteristiche di identificazione sei volte più alto delle impronte digitali, il sistema presenta ottimi dati in termini di accuratezza oltre ad essere molto resistente a tentativi di frode.

Svantaggi

* tassi elevati di falsi negativi che comporterebbero il mancato riconoscimento dell'individuo da parte del sistema;
* risente della luce presente nell'ambiente circostante al soggetto da identificare;
* è possibile che il riconoscimento dell'iride non funzioni con persone non vedenti o per le quali si siano perse in maniera massiccia le proprietà morfologiche o geometriche dell'iride.
* quasi tutti i telefoni di oggi ancora non dispongono di questa tecnologia a livello hardware.

3.4.2.4 Caratteristiche del volto

Il riconoscimento automatico di un individuo tramite l'analisi delle sue sembianze facciali è un procedimento complesso che utilizza immagini video in luce visibile o "termiche" a *infrarosso*. I confronti biometrici sono resi complicati dalla presenza di capigliatura, di occhiali e dalla posizione assunta dalla testa durante la ripresa, nonché dalle condizioni di illuminazione. Le stesse tecniche basate su riprese a *infrarosso* non sono invece influenzate dall'illuminazione e sono efficaci anche al buio. Possono essere ottenute anche immagini di tipo tridimensionale, per fusione di più immagini o con tecniche di *computer graphics* basate sull'elaborazione dell'ombreggiatura. Dal campione biometrico facciale tramite algoritmi, talvolta basati su reti neurali, vengono estratti un certo numero di tratti, quali la posizione degli occhi, del naso, delle narici, del mento, delle orecchie, al fine di costruire un modello biometrico. Laddove il procedimento avvenga in un contesto cooperativo il riconoscimento facciale può essere molto accurato, al punto da poter essere utilizzato in funzione di controllo di accesso logico o fisico. Le sembianze facciali possono fornire indicazioni sui dati sensibili. Un modello tridimensionale del viso, in funzione delle varie modalità di acquisizione, presenta dimensioni variabili da 100 a 3.500 byte. Il riconoscimento del volto avviene attraverso una serie di tecniche che vanno dal cosiddetto *eigenface*, ad algoritmi basati su reti neurali, al processamento automatico del volto basato sulla misura di alcune distanze misurabili tra punti di riferimento del viso.



Vantaggi

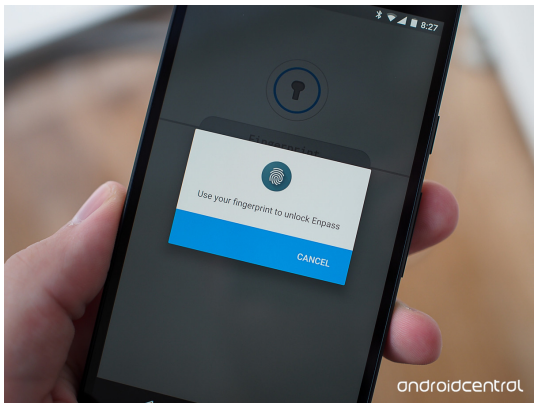
* l'accettazione da parte degli utenti del riconoscimento biometrico basato sui tratti somatici è generalmente alta, data la natura non invasiva e "naturale" del metodo di acquisizione;
* il riconoscimento facciale può essere molto accurato se eseguito con la collaborazione dell'utente;
* esistono molte librerie sul mercato che si occupano di riconoscimento del volto;

Svantaggi

* ha una bassa stabilità, dal momento che i tratti del viso di un individuo variano immancabilmente sia con l'età che a causa di incidenti o ferite;
* risente delle condizioni ambientali di illuminazione;
* presenta caratteristiche, in termini di accuratezza, meno pregevoli di quelli basati sul riconoscimento dell'impronta digitale o dell'iride;

3.4.3 La soluzione nativa di Android

La prima soluzione che ho ipotizzato, è stata quella di utilizzare quanto offerto dalla maggior parte degli smartphone odierni in maniera nativa: il sensore di impronta digitale. In particolare ero interessata, a livello implementativo, alla classe di Android `FingerprintManager`, che è la classe che coordina l'accesso all'hardware del `fingerprint` presente in molti dispositivi Android oggigiorno.
E' possibile infatti, a livello di codice, nel metodo `onCreate`, richiedere l'istanza del `FingerprintManager, che` si occuperà di svolgere l'autenticazione tramite impronte digitali tramite il metodo `authenticate` e inizializzare un oggetto definito come `CryptoObject`, che sarà passato al `FingerprintManager` per rendere sicura la comunicazione con lo scanner di impronte. E' possibile anche far sì che l'autenticazione abbia realmente inizio solo se le *permission* necessarie sono state assegnate, e se almeno un'impronta digitale è stata registrata. Il metodo `authenticate` del `FingerprintManager`, al momento dell'invocazione, riceverebbe il `CryptoObject` prima citato ed un riferimento ad un *listener* i cui metodi saranno invocati all'esito del riconoscimento delle impronte. Il metodo `onAuthenticationSucceeded` verrebbe invocato quando viene riconosciuta una impronta digitale valida.



Insomma, sarebbe stata una soluzione assolutamente semplice, ma questa soluzione è stata scartata perché nascondeva un caso particolare che avrebbe portato al mancato soddisfacimento del requisito ROF2.1. Utilizzando questa soluzione, l'utente, diciamo U, potrebbe memorizzare nel suo dispositivo un'impronta non propria, diciamo di una certa altra persona P. Tale soggetto potrebbe seguire i corsi al posto di U, e al termine del corso, ripristinare l'impronta di P, in modo da cancellare, in caso di controlli, ogni traccia di autenticazione falsificata. Questo scenario è da escludere. Riassumendo, con questa soluzione non si avrebbe pieno controllo sul processo di enrollment.

3.4.4 Soluzioni FIDO

Successivamente ho iniziato a interessarmi a soluzioni certificate FIDO. FIDO è un insieme di protocolli che usa le techiche di crittografia a chiave pubblica per fornire un'autenticazione più forte rispetto a una autenticazione standard. I protocolli FIDO erano interessanti ai miei occhi perchè in caso di utilizzo di tecniche di autenticazione biometrica, il dato biometrico non avrebbe mai lasciato il device dell'utente, come da requisiti; inoltre questi protocolli sono progettati appositamente per proteggere la privacy dell'utente.
Durante la registrazione ad un servizio online, il device dell'utente crea un nuovo paio di chiavi. Il device tiene la chiave privata e registra la chiave pubblica nel server del servizio online. L'autenticazione è fatta dal device provando il possesso della chiave privata al servizio firmando una richiesta. La chiave privata dell'utente può essere usata solo dopo che sia stata sbloccata localmente nel dispositivo dall'utente. Lo sblocco è fatto tramite una azione user-friendly e sicura come inserire un PIN, usare l'impronta digitale, parlare al microfono, abbinando un device ausiliario o premere un bottone.
FIDO offre due specifiche:

* **UAF - Passwordless Experience** Universal Authentication Framework (UAF) è il protocollo che regola la *passwordless experience*. In questo caso, l'utente registra il proprio dispositivo al servizio online selezionando un meccanismo di autenticazione locale. Una volta registrato, l'utente ripete semplicemente

l'azione di autenticazione locale ogni volta che hanno bisogno di autenticarsi al servizio. L'utente non ha più bisogno di inserire la propria password durante l'autenticazione da tale dispositivo.

* **U2F - Second Factor Experience** Universal Second Factor (U2F) è il protocollo tramite il quale l'utente si connette con un nome utente e una password, ma in aggiunta l'utente presenta il secondo fattore utilizzzando su un dispositivo USB o NFC. Non è il caso di interesse, anche sulla base dei requisiti citati nello scorso paragrafo.

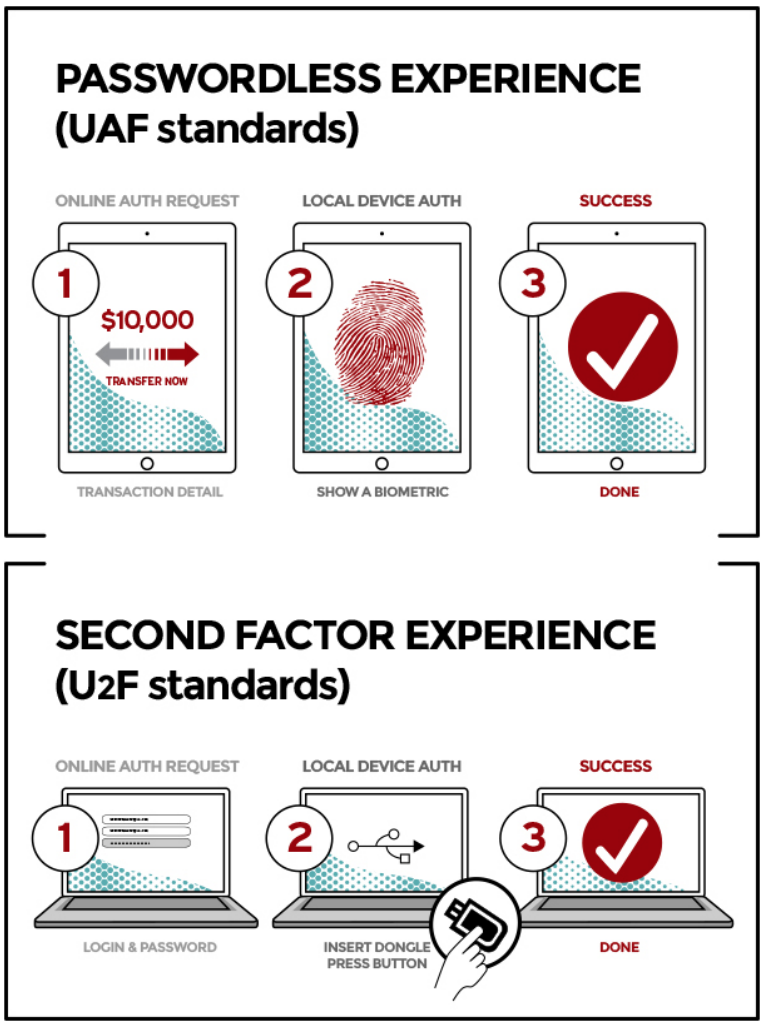


figura 3.3: FIDO - UAF U2F

In entrambi i casi la registrazione e l'autenticazione avvengono come specificato di seguito.

3.4.4.1 Registrazione

* l'utente sceglie un autenticatore FIDO in base alla polizza di accettazione del servizio online;
* l'utente sblocca l'autenticatore FIDO usando il lettore d'impronta digitale, inserendo un pin o utilizzando la USB di U2F;
* il device dell'utente crea una coppia di *chiavi pubblica/privata* unica per il *device*, per il servizio e account dell'utente;
* la chiave pubblica è mandata al servizio online e associata all'account dell'utente. La chiave privata e qualsiasi informazione riguardo l'autenticazione locale (compresi i dati biometrici) non lasciano mai il device dell'utente.

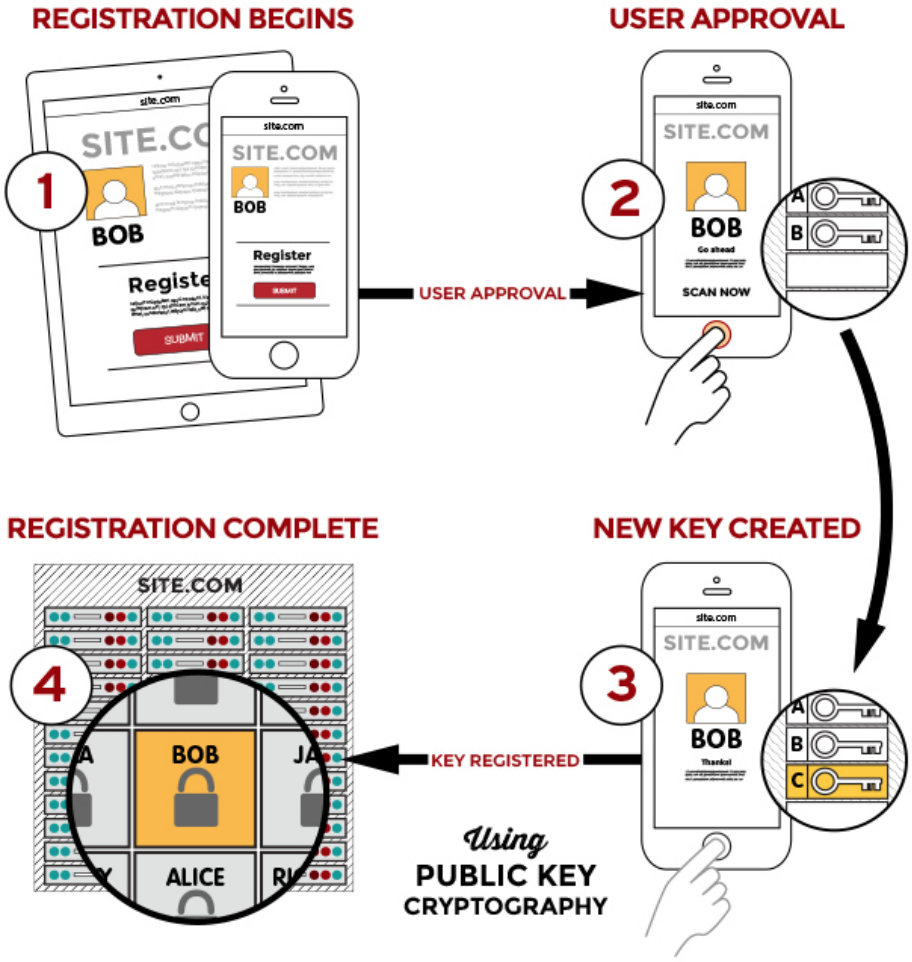


figura 3.4: FIDO - Registrazione

3.4.4.2 Autenticazione

* il servizio online chiede all'utente di effettuare la login con il device precedentemente utilizzato;
* l'utente sblocca l'autenticatore FIDO usando lo stesso metodo scelto durante la registrazione;
* il device usa l'identificatore dell'account dell'utente fornito dal servizio per selezionare la corretta chiave e firmare la richiesta del servizio online;
* il device manda la richiesta firmata al servizio online, che la verifica con la chiave pubblica immagazzinata, autenticando l'utente.

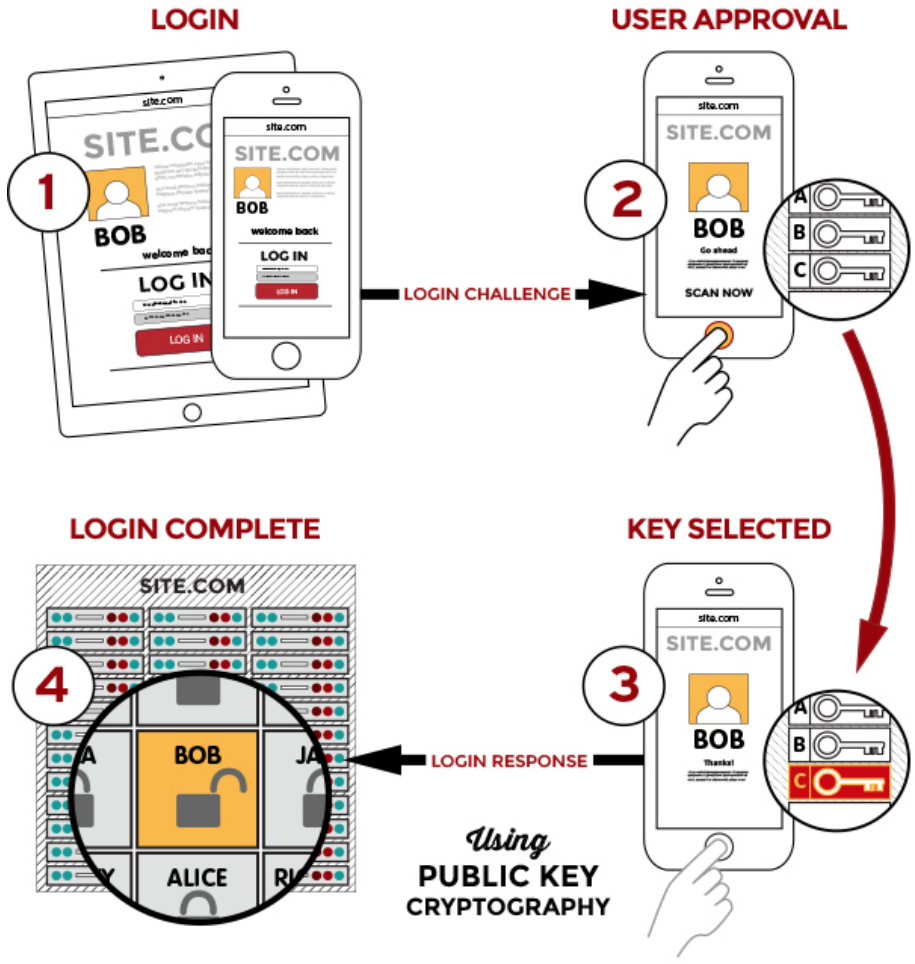


figura 3.5: FIDO - Autenticazione

3.4.5 Prima Soluzione: ZoOm Login

Una soluzione possibile è stata quella offerta da ZoOm Login. E' una soluzione di riconoscimento facciale certificata FIDO. Il link al sito è il seguente: https://zoomlogin.com/, mentre il link alla documentazione è il seguente: https://dev.zoomlogin.com/zoomsdk/#/android-sdk-reference-classes.
E' gratuita per:

* piccole aziende con fatturato minore ai 10 milioni;
* istituzioni educative;
* istituzioni non-profit;
* start-up e sviluppatori.

In virtù di ciò ho ottenuto l'SDK, che forniva un'ottima documentazione e degli esempi di utilizzo dell'SDK. Erano disponibili versioni per Android e iOS. Per quanto riguarda la versione Android ho studiato la documentazione e ho creato un piccolo prototipo funzionante. Le classi più importanti dell'SDK sono `ZoomEnrollmentActivity` e `ZoomAuthenticationActivity` che avviano l'enrollment e la bio-autenticazione `ZoomAuthenticationResult` e `ZoomEnrollmentResult` che gestiscono i risultati dell' enrollment e della bio-autenticazione. E' un SDK che offre queste funzionalità, quindi, molto ad alto livello, perlomeno secondo il lato di chi va ad utilizzare l'SDK. All'interno invece, l'algoritmo di ZoOm traccia e processa migliaia di punti del volto dei circa 30 *frames* di video e li analizza in pochi secondi, effettuando anche il liveness test.
I problemi di questa soluzione erano che innanzitutto non si aveva controllo su buona parte dell'interfaccia grafica (che spesso mostrava messaggi in inglese, poco user friendly per gli utenti a cui si sarebbe rivolta l'applicazione). Inoltre, in caso di sviluppi commerciali, poiché sia Vision Learning sia Wintech hanno più di 10 milioni all'anno di fatturato, avrebbe avuto un costo eccessivo rispetto a quanto le due aziende erano disposte a pagare. Infine la procedura di enrollment era troppo lunga (più di 20 secondi) e poco user-friendly perché, proprio a causa del liveness test, l'utente era tenuto a muovere il dispositivo allontanandolo e avvicinandolo più volte al volto, portando spesso al fallimento della procedura la prima volta di utilizzo della applicazione.

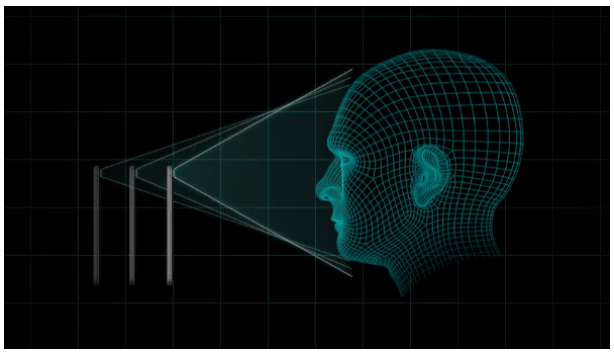


figura 3.6: Prima Soluzione - 3D

3.4.6 Seconda Soluzione

La Seconda Soluzione aveva la forma di API, usava il riconoscimento facciale, e di essa ci è stata fornito un prototipo Android e una documentazione assolutamente minimale (a mio parere insufficiente). Il prezzo era alto ma accettabile secondo Wintech. Il prototipo da loro fornito effettuava il liveness test con una esperienza utente alquanto insoddisfacente. L'utente era tenuto a muovere la testa più volte portando spesso al fallimento non solo al primo tentativo di utilizzo, ma anche ai successivi.

3.4.7 Terza Soluzione

La Terza Soluzione aveva sia la forma di API o, in alternativa di SDK e usava il riconoscimento facciale. Ci è stato fornito l'SDK. Implementava le funzioni molto a basso livello che non era quello che Wintech cercava, violando quindi il requisito ROF2.4, senza contare che il prezzo secondo Wintech era assolutamente esagerato.

3.4.8 Quarta Soluzione

La Quarta soluzione era certificata FIDO UAF. L'azienda ci ha fornito una documentazione introduttiva dalla quale si desumeva che il processo in se di enrollment e bio-autenticazione avveniva attraverso terze parti (riconocimento vocale, fingerprint). La Quarta Soluzione quindi andava a certificare che la terza parte effettuasse l'autenticazione secondo lo standard UAF, senza fornirlo. Anche in questo caso il prezzo era assolutamente esagerato per le esigenze di Wintech.

3.4.9 Quinta Soluzione: la scelta

Questa soluzione è stata quella su cui è ricaduta la mia scelta. L'azienda poteva fornirci sia il motore di autenticazione facciale sia quello vocale. Per motivi di facilità d'uso lato utente e visto che la soluzione vocale non disponeva di algoritmi di liveness test, ho scelto la soluzione facciale. Era un SDK ed era utilizzabile in Android Studio ed era disponibile una versione in caso di possibili sviluppi futuri in *iOS*. Ci è stata offerta una buona documentazione, grazie anche alla quale si riesce ad avere un buon controllo sulle procedure di enrollment e di bio-autenticazione. Queste funzionalità sono utilizzate dal programmatore ad alto livello e il dato biometrico resta memorizzato nel device dell'utente come da requisiti. I tempi di enrollment e di bio-autenticazione sono brevi (6 secondi mediamente per l'enrollment e autenticazione in 1 secondo). Il prezzo è accessibile. Per tutti questi motivi la mia scelta è ricaduta su questa libreria.

3.4.10 Tabella riassuntiva

Segue una tabella riassuntiva che mostra quali soluzioni soddisfano (V) o non soddisfano (X) un dato requisito. I numeri (1, 2, 3, 4, 5) rappresentano rispettivamente la Prima Soluzione, la Seconda Soluzione ecc... Quando il dato non è disponibile, la casella viene riempita da un "-". Ricordo che la Quinta soluzione è quella scelta.

Requisito	**Descr. requisito**	**1**	**2**	**3**	**4**	**5**
ROF1	La soluzione deve avere la forma di API o di SDK	V	V	V	V	V
ROF1.1	La soluzione deve essere utilizzabile in Android Studio	V	-	X	-	V
RDF1.2	La soluzione deve essere disponibile per possibili sviluppi futuri in iOS	V	-	V	X	V
ROF1.3	La soluzione deve avere la documentazione di classi, metodi e campi dati con esempi minimali di utilizzo	V	V	V	V	V
ROF2	La soluzione deve consentire l'enrollment e la bio-autenticazione	V	V	V	V	V
ROF2.1	La soluzione deve consentire il completo controllo delle procedure di enrollment e di bio-autenticazione	X	X	V	X	V
ROF2.1.1	La soluzione deve poter consentire il blocco degli enrollment una volta effettuato il primo	V	V	V	V	V
RFF2.1.2	La soluzione deve consentire il completo controllo dell'interfaccia grafica della procedura di enrollment e di bio-autenticazione	X	X	V	-	V
ROF2.2	Il dato biometrico dell'utente deve essere memorizzato solo e unicamente nel dispositivo dell'utente	V	V	V	V	V
ROF2.3	La soluzione deve effettuare il liveness test	V	V	-	X	V

ROF2.4	La soluzione deve implementare al suo interno il meccanismo di matching tra riferimento biometrico e istanza biometrica	V	V	X	X	V
ROF2.5	La soluzione deve consentire l'autenticazione e l'enrollment di un utente utilizzando solamente il suo dispositivo, senza l'ausilio di ulteriori sensori aggiuntivi o altr	V	V	V	V	V
ROF3	In caso di software a pagamento, la soluzione deve essere *congruo alle aspettative di Wintech* (non citato per questioni di riservatezza)	X	V	X	X	V
ROF4	In caso di software a pagamento, l'azienda fornitrice della soluzione dovrà fornire dei prototipi funzionanti e una documentazione minimale prima di procedere alla stipula di contratti	V	V	V	V	V
RDF5	L'azienda fornitrice della soluzione (a meno di tecnologie open-source) deve rispondere alle nostre richieste di chiarimento entro una settimana (limitatamente al tempo di stage)	V	X	V	V	V
RDQ6	Il FAR (vedi 3.1.2.2)deve essere 0 su 10 prove	V	V	-	V	V
RDP7	La soluzione deve effettuare un enrollment con successo entro 10 secondi, altrimenti dare errore	X	V	-	V	V
RDP8	La soluzione deve effettuare una bio-autenticazione con successo entro 3 secondi, altrimenti dare errore	X	X	-	X	V
RFQ9	Un numero minore o uguale a 2 persone su 5 può fallire l'enrollment al primo tentativo di utilizzo dell'applicazione	V	X	V	V	V

tabella 3.2: Tabella comparativa di alcune librerie studiate

Capitolo 4

Prototipo

4.1 Modello di sviluppo adottato

Come modello per la gestione del progetto ho adottato la metodologia agile. Tale modello mi ha permesso di adottare una certa flessibilità e velocità nel rispondere alle esigenze di Wintech e di Vision Learning riguardo al prototipo.

Nel mio caso, erano previste iterazioni continue, della durata di sei giorni lavorativi, entro le quali si sarebbero susseguite attività di:

* analisi dei requisiti (nel mio caso dialogando con il mio tutor e alcuni membri del personale di Vision Learning);
* pianificazione delle funzionalità da implementare nell'iterazione corrente;
* progettazione, codifica, rilascio delle funzionalità dell'iterazione corrente.

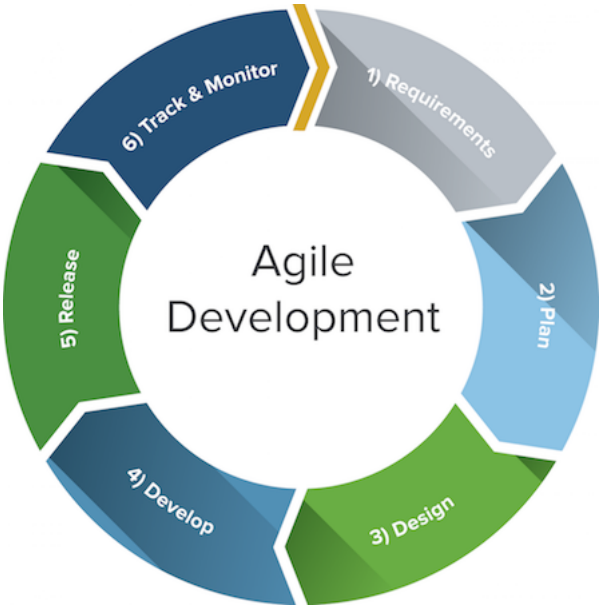


figura 4.1: Grafico a sinistra

Il punto di partenza di ogni ciclo è il *feedback* ricevuto tramite la presentazione della soluzione raggiunta fin'ora al tutor.

Gli *sprint* erano tre e riguardavano:

* enrollment e bio-autenticazione;
* predisposizione della app per l'integrazione col server di Vision Learning;
* integrazione col server di Vision Learning.

4.2 Analisi dei requisiti

4.2.1 Casi d'uso

4.2.1.1 Introduzione

Durante la prima parte dello stage mi sono occupata della definizione dei requisiti software che il prodotto avrebbe dovuto soddisfare. Attraverso una successione di incontri con il tutor aziendale e alcuni membri del personale di Vision Learning si è riusciti a delineare l'idea del prodotto, che inizialmente non era molto chiara.
Per facilitare l'emersione dei requisiti ho steso dei diagrammi dei casi d'uso (in inglese Use Case Diagram), conformi allo standard UML 2.0. e sono diagrammi dedicati alla descrizione delle funzioni o servizi offerti da un sistema, così come sono percepiti e utilizzati dagli attori che interagiscono col sistema stesso. Per ogni use case sono stati inoltre riportati:

* Attori
* Descrizione
* Pre-condizioni (se presenti)
* Post-condizioni
* Scenario principale
* Estensioni (se presenti)

L'unico attore dell'applicazione mobile sviluppata è il semplice discente del corso di e-Learning. Ogni caso d'uso ha un identificativo univoco strutturato in questo modo: UC Codice dove Codice è un numero che identifica il caso d'uso in modo gerarchico, tale che se il caso d'uso UC3 viene suddiviso in altri casi d'uso, questi saranno chiamati UC3.1 UC3.2 ecc. . . I casi d'uso sono inoltre definiti da un nome che deve spiegare in breve quale funzionalità viene rappresentata.

4.2.1.2 Panoramica dei casi d'uso

Per facilitare la consultazione dei casi d'uso, viene fornita una immagine riassuntiva (non in notazione UML) che descrive le principali funzionalità offerte dall'applicazione e un elenco dei casi d'uso riportati in questa sezione.

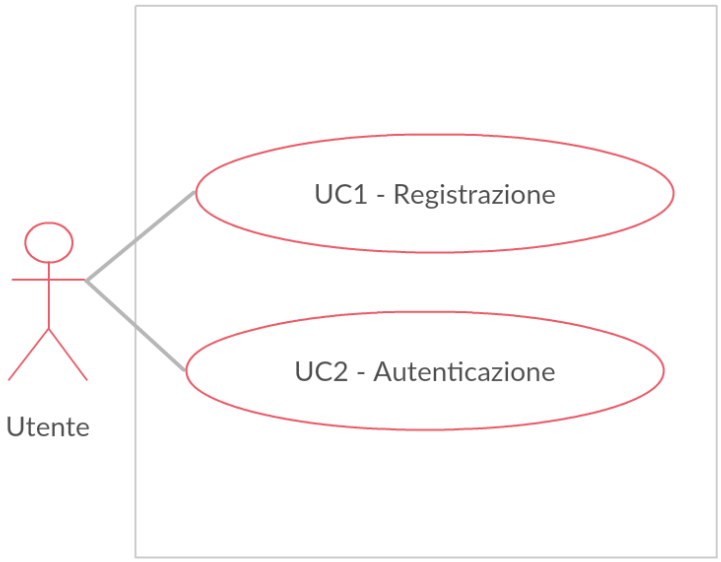


figura 4.2: Panoramica dei casi d'uso

UC1 Registrazione
- UC1.1 Enrollment
- UC1.2 Inserimento credenziali
  - UC1.2.1 Compilazione codice A
  - UC1.2.2 Compilazione codice B
  - UC1.2.3 Compilazione codice C
- UC1.3 Conferma credenziali
- UC1.4 Visualizzazione messaggio di non avvenuta registrazione

UC2 Autenticazione
- UC2.1 Bio-autenticazione
- UC2.2 Inserimento PIN
- UC2.3 Conferma PIN
- UC2.4 Visualizzazione messaggio di non avvenuta autenticazione

Per motivi di riservatezza non sono specificati esattamente i codici di cui agli UC1.2.1, 1.2.2, 1.2.3. Seguono i diagrammi riguardanti l'applicativo lato *smartphone*, quello di cui mi sono occupata.

UC1 Registrazione

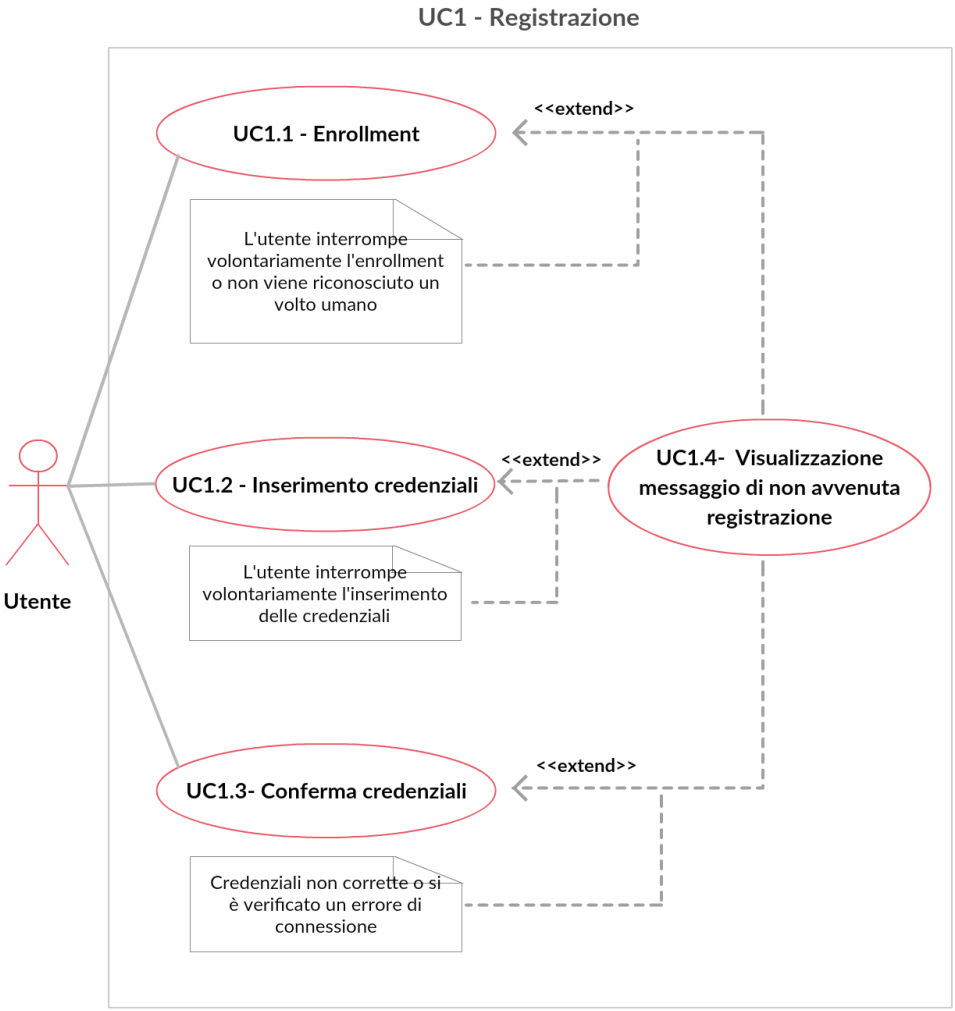


figura 4.3: UC1 Registrazione

Attori	Utente
Descrizione	l'utente si registra
Pre-condizione	l'utente ha aperto l'applicazione
Post-condizione	l'utente si è registrato
Scenario principale	UC1.1 Enrollment; UC1.2 Inserimento credenziali; UC1.3 Conferma credenziali.

UC1.1 Enrollment

Attori	Utente
Descrizione	l'utente esegue l'enrollment
Pre-condizione	l'applicazione permette di eseguire l'enrollment
Post-condizione	l'utente ha eseguito l'enrollment
Scenario principale	UC1.1 Enrollment;
Estensioni	UC1.4 Visualizzazione messaggio di non avvenuta registrazione.

UC1.2 Inserimento credenziali

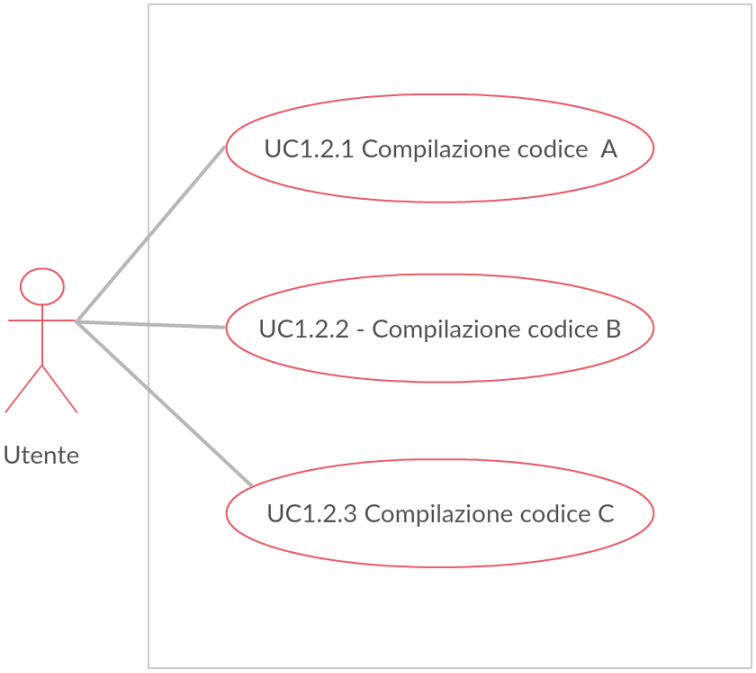


figura 4.4: UC1.2 Inserimento credenziali

Attori	Utente
Descrizione	l'utente inserisce le credenziali
Pre-condizione	l'enrollment è terminato con successo e l'applicazione consente di inserire le credenziali
Post-condizione	l'utente ha inserito le credenziali
Scenario principale	UC1.2.1 Compilazione codice A; UC1.2.2 Compilazione codice B; UC1.2.3 Compilazione codice C.
Estensioni	UC1.4 Visualizzazione messaggio di non avvenuta registrazione.

UC1.2.1 Compilazione codice A

Attori	Utente
Descrizione	l'utente compila il campo del codice A
Post-condizione	l'utente ha compilato il campo del codice A
Scenario principale	UC1.2.1 Compilazione codice A.

UC1.2.2 Compilazione codice B

Attori	Utente
Descrizione	l'utente compila il campo codice B
Post-condizione	l'utente ha compilato il campo del codice B
Scenario principale	UC1.2.1 Compilazione codice A.

UC1.2.3 Compilazione codice C

Attori	Utente
Descrizione	l'utente compila il campo codice C
Post-condizione	l'utente ha compilato il campo del codice C
Scenario principale	UC1.2.3 Compilazione codice C.

UC1.3 Conferma credenziali

Attori	Utente
Descrizione	l'utente conferma le credenziali
Pre-condizione	l'enrollment è terminato con successo e l'applicazione permette di confermare le credenziali
Post-condizione	l'utente ha confermato le credenziali
Scenario principale	UC1.3 Conferma credenziali.
Estensioni	UC1.4 Visualizzazione messaggio di non avvenuta registrazione.

UC1.4 Visualizzazione messaggio di non avvenuta registrazione

Attori	Utente
Descrizione	l'utente visualizza un messaggio di non avvenuta registrazione
Pre-condizione	l'utente ha interrotto volontariamente l'enrollment o l'inserimento delle credenziali oppure durante l'enrollment non viene riconosciuto un volto umano oppure le credenziali inserite non sono corrette
Post-condizione	l'utente visualizza un messaggio di non avvenuta registrazione; l'applicazione ritorna alla schermata iniziale con enrollment e credenziali annullate.
Scenario principale	UC1.4 Visualizzazione messaggio di non avvenuta registrazione.

UC2 Autenticazione

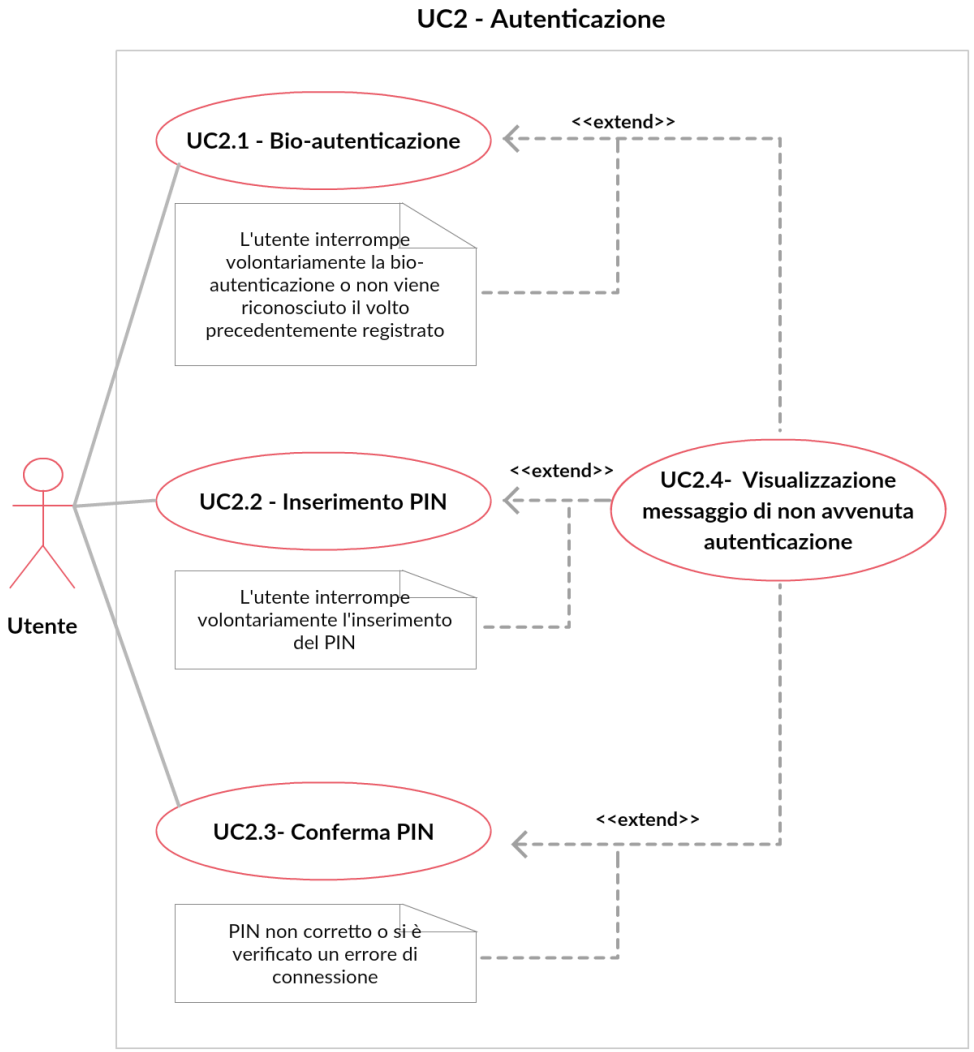


figura 4.5: UC2 Autenticazione

Attori	Utente
Descrizione	l'utente si autentica
Pre-condizione	l'utente ha aperto l'applicazione
Post-condizione	l'utente si è autenticato
Scenario principale	UC2.1 Bio-autenticazione; UC2.2 Inserimento PIN; UC2.3 Conferma PIN.

UC2.1 Bio-autenticazione

Attori	Utente
Descrizione	l'utente esegue la bio-autenticazione
Pre-condizione	l'applicazione permette di eseguire la bio-autenticazione
Post-condizione	l'utente ha eseguito la bio-autenticazione
Scenario principale	UC2.1 Bio-autenticazione.
Estensioni	UC2.4 Visualizzazione messaggio di non avvenuta autenticazione;

UC2.2 Inserimento PIN

Attori	Utente
Descrizione	l'utente inserisce il PIN
Pre-condizione	la bio-autenticazione è terminata con successo e l'applicazione consente di inserire il PIN
Post-condizione	l'utente ha inserito il PIN
Scenario principale	UC2.2 Inserimento PIN.
Estensioni	UC2.4 Visualizzazione messaggio di non avvenuta autenticazione;

UC2.3 Conferma PIN

Attori	Utente
Descrizione	l'utente conferma il PIN
Pre-condizione	la bio-autenticazione è terminata con successo e l'applicazione permette di confermare le credenziali
Post-condizione	l'utente ha confermato il PIN
Scenario principale	UC2.3 Conferma PIN.
Estensioni	UC2.4 Visualizzazione messaggio di non avvenuta autenticazione;

UC2.4 Visualizzazione messaggio di non avvenuta autenticazione

Attori	Utente
Descrizione	l'utente visualizza un messaggio di non avvenuta autenticazione
Pre-condizione	l'utente ha interrotto volontariamente l'autenticazione o l'inserimento del PIN oppure durante la bio-autenticazione non viene riconosciuto un volto umano oppure il PIN inserito non è corretto
Post-condizione	l'utente visualizza un messaggio di non avvenuta autenticazione; l'applicazione ritorna alla schermata iniziale.
Scenario principale	UC2.4 Visualizzazione messaggio di non avvenuta autenticazione.

4.2.2 Requisiti

Grazie anche ai casi d'uso e grazie a discussioni con il tutor aziendale o il personale di Vision Learning ho stilato una lista di requisiti, molti dei quali vengono riportati in questa sezione. Non vengono forniti dettagli riguardo il rapporto tra applicazione e sistemi di Vision Learning per motivi di riservatezza. I requisiti sono classificati secondo la seguente notazione:

R[Importanza][Tipologia][Codice]

dove:

* importanza: può assumere questi valori:
  - O: indica un requisito obbligatorio;
  - D indica un requisito desiderabile;
  - F: indica un requisito facoltativo (opzionale).
* tipologia: può assumere questi valori:
  - F: indica un requisito funzionale. Rappresenta le funzionalità che il prodotto deve fornire;
  - Q: indica un requisito di qualità. Rappresenta gli elementi per aumentare la qualità del prodotto finale;
  - P: indica un requisito prestazionale. Rappresenta un valore misurabile delle performance raggiunte dal prodotto terminato;
  - V: indica un requisito di vincolo. Rappresenta le limitazioni principalmente inerenti alle tecnologie che il prodotto deve rispettare.
* codice: codice numerico che identifica il requisito; deve essere univoco ed indicato in forma gerarchica, da sinistra a destra, nella notazione X.Y.Z.

Le fonti dei requisiti sono una (o più) tra le seguenti:

* caso d'uso: requisito emerso da un caso d'uso;
* interno: requisito derivato da una discussione con il tutor aziendale o il personale di Vision Learning.

Requisito	Descrizione
ROF1	L'utente può registrarsi
ROF1.1	L'utente può effettuare l'enrollment
RDF1.2	L'utente può inserire le credenziali
RDF1.2.1	L'utente può compilare il campo codice A
RDF1.2.2	L'utente può compilare il campo codice B
RDF1.2.3	L'utente può compilare il campo codice C
RDF1.3	L'utente può confermare le credenziali
RDF1.4	L'utente può interrompere l'enrollment
RFF1.4.1	Al momento dell'interruzione dell'enrollment viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF1.5	L'utente può interrompere l'inserimento delle credenziali
RFF1.5.1	Al momento dell'interruzione dell'inserimento delle credenziali viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF1.6	In caso non venga riconosciuto alcun volto umano durante l'enrollment viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF1.7	In caso di credenziali errate viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF1.8	In caso di errore di connessione viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
ROF2	L'utente può autenticarsi
ROF2.1	L'utente può effettuare la bio-autenticazione
RDF2.2	L'utente può inserire il PIN
RDF2.3	L'utente può confermare il PIN

RDF2.4	L'utente può interrompere la bio-autenticazione
RFF2.4.1	Al momento dell'interruzione della bio-autenticazione viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
ROF2.5	L'utente può interrompere l'inserimento del PIN
RFF2.5.1	Al momento dell'interruzione dell'inserimento del PIN viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF2.6	In caso non venga riconosciuto alcun volto umano durante la bio-autenticazione e in caso di fallimento di bio-autenticazione viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF2.7	In caso di PIN errato viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale
RFF3	L'applicazione deve essere integrabile con i sistemi di Vision Learning
RFF3.1	Al termine della procedura di registrazione l'utente è registrato nei sistemi di Vision Learning
RFF3.2	Al termine della procedura di autenticazione il video presente nei sistemi di Vision Learning che l'utente aveva scelto viene sbloccato
RDP4	L'enrollment deve durare meno di 15 secondi
RDP5	La bio-autenticazione deve durare meno di 4 secondi
RDV6	L'applicazione deve funzionare su dispositivi mobile aventi come sistema operativo Android dalla versione 6 in avanti
RDV7	L'applicazione deve funzionare su dispositivi aventi un accesso al sito indicato da Vision Learning
RDV8	L'applicazione deve funzionare su dispositivi aventi una connessione ad internet attiva

tabella 4.14: Requisiti del prototipo

4.3 Progettazione e codifica

4.3.1 Tecnologie utilizzate

Per lo sviluppo del progetto mi sono avvalsa delle seguenti tecnologie:

* Java come linguaggio di sviluppo;
* Android come framework di sviluppo;
* Android Studio come IDE per lo sviluppo;
* Gradle come sistema di building;
* Git per il versionamento del codice, per adattarmi ai sistemi adottati da Vision Learning;
* Javadoc per la documentazione del codice;
* l'SDK fornitomi nella Quinta Soluzione (per la descrizione consultare: Quinta Soluzione: la scelta).

figura 4.6: Loghi di alcune tecnologie utilizzate nell'applicazione

4.3.2 Pattern utilizzati

Per lo sviluppo del progetto mi sono avvalsa dei seguenti pattern:

* architetturale: REST come stile architetturale;
* creazionale: Builder (Design Pattern).

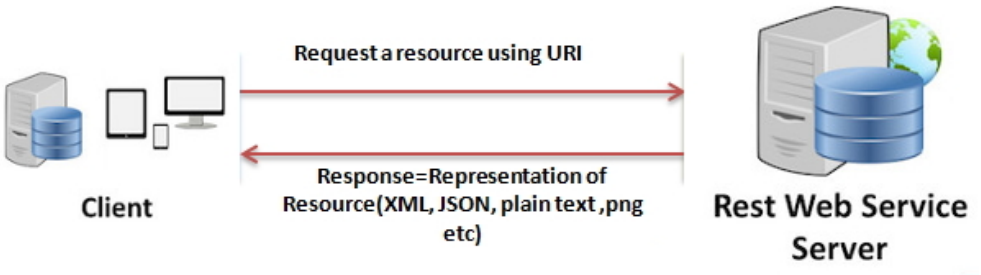


figura 4.7: Architettura utilizzata nell'applicazione

4.3.3 Progettazione architetturale

L'architettura generale del prototipo è costituita da un frontend e da un back-end. Quest'ultimo fornisce una serie di API REST per l'accesso alle risorse dell'applicazione che il primo consuma attraverso le richieste HTTP. Un database, poi, si occupa di mantenere la persistenza dei dati.
La comunicazione tra il back-end e il frontend avviene tramite un insieme di API che il primo mette a disposizione del secondo seguendo i principi REST (REpresentational State Transfer). Mi sono state fornite le interfacce di comunicazione col server di Vision Learning. Avrei dovuto interfacciarmi con esso tramite due chiamate di tipo POST da loro indicate: una per la registrazione e una per l'autenticazione. A queste chiamate ricevevo una risposta di tipo JSON contenente l'esito della registrazione o autenticazione.

4.3.3.1 SSL

Ho utilizzato SSL come protocollo di comunicazione tra *client* e *server*. *SSL* o "*Secure Sockets Layer*" è un protocollo in cui le applicazioni che utilizzano i certificati *SSL* sono in grado di gestire l'invio e la ricezione di chiavi di protezione e di criptare/decriptare le informazioni trasmesse utilizzando le stesse chiavi, comunicando in modo sicuro e protetto.

4.3.3.2 Certificato self-signed

Nella crittografia e nella sicurezza informatica, un certificato auto-firmato è un certificato di identità firmato dalla stessa entità la cui identità è certificata. In termini tecnici un certificato auto-firmato è firmato con la propria chiave privata. Nelle tipiche infrastrutture di chiave pubblica (PKI), una firma digitale da un'autorità di certificazione (CA) attesta che un particolare certificato di chiave pubblica è valido. Nel mio caso, avevo a che fare con un server che aveva un certificato self-signed. Per comunicare con esso, tra i parametri della chiamata ho inserito una speciale stringa che mi era stata comunicata per la chiamata REST.

4.3.3.3 REST

Lo stile architetturale REST prescrive che lo stato in un'applicazione e le sue funzionalità siano interpretati come risorse web univoche e accessibili tramite un URL e, di solito, un protocollo HTTP. L'approccio architetturale REST è definito dai seguenti vincoli applicati ad un'architettura:

* **Client-server**: un insieme di interfacce uniformi separa il client dal server in modo da ridurre l'accoppiamento tra le componenti del sistema. In questo modo, ad esempio, il client non deve occuparsi della persistenza dei dati e il server dell'interfaccia grafica;
* **Stateless**: la comunicazione client–server è ulteriormente vincolata in modo che nessun contesto client venga memorizzato sul server tra le richieste. Ogni richiesta da ogni client contiene tutte le informazioni necessarie per richiedere il servizio, e lo stato della sessione è contenuto sul client;
* **Cacheable**: i client possono fare caching delle risposte;
* **Layered system**: i client non sono tenuti a conoscere quale livello del sistema server stanno interrogando.

Tale architettura oltre ad essere molto diffusa nelle applicazioni web, è semplice e riduce notevolmente l'accoppiamento tra *client* e *server*.

4.3.4 Progettazione di dettaglio e codifica

Per quanto riguarda il prototipo *Android*, ho creato delle *activity* per la gestione di interfaccia e comportamento, cercando di dividere per quanto possibile presentazione da logica di *business*, creando classi ad hoc per la gestione del rapporto client-server. Per quanto riguarda la codifica, ho scritto il codice in maniera da rispettare uno stile di codifica coerente, tracciando a volte il comportamento della applicazione in un *logger*. Riporto un estratto di classi, metodi e proprietà che ho creato per la applicazione. In questa sezione la notazione in *corsivo* indicherà una descrizione di metodo o proprietà.

4.3.4.1 `WelcomeActivity`

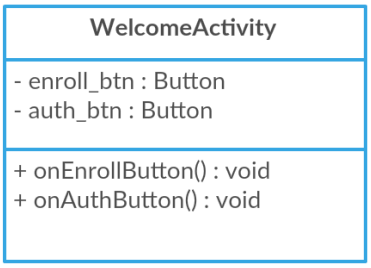


figura 4.8: Diagramma della classe WelcomeActivity

Descrizione	Activity mostrata all'apertura dell'applicazione
Proprietà	`- enroll_btn : Button` » *Bottone che fa partire l'enrollment* `- auth_btn : Button` » *Bottone che fa partire la bio-autenticazione*
Metodi	`+ onEnrollButton() : void` » *Metodo che viene invocato al click del bottone di registrazione* `+ onAuthButton() : void` » *Metodo che viene invocato al click del bottone di autenticazione*

4.3.4.2 `BiometricCore`

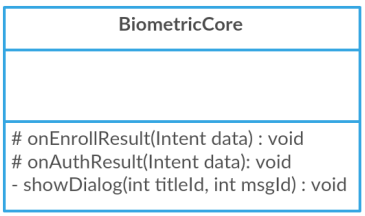


figura 4.9: Diagramma della classe BiometricCore

Descrizione	Classe che gestisce l'enrollment e la bio-autenticazione.
Metodi	`# onErollResult(Intent data): void` » *Metodo che viene invocato una volta terminato l'enrollment e ne gestisce i risultati, sia positivi sia negativi* `# onAuthResult(Intent data): void` » *Metodo che viene invocato una volta terminata la bio-autenticazione e ne gestisce i risultati, sia positivi sia negativi* `- showDialog(int titleId, int msgId): void` » *Metodo di utilità per mostrare una dialog contentente il risultato dell'enrollment o della bio-autenticazione*

4.3.4.3 `CredentialActivity`

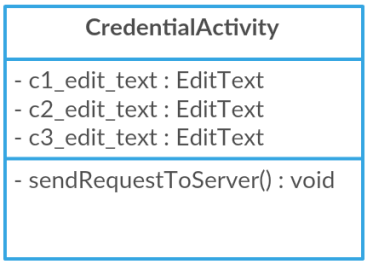


figura 4.10: Diagramma della classe CredentialActivity

Descrizione	Activity che gestisce l'inserimento delle credenziali e la risposta che viene ricevuta dal server di Vision Learning al momento del loro invio
Proprietà	`- c1_edit_text : EditText` » *Campo che contiene la prima credenziale* `- c2_edit_text : EditText` » *Campo che contiene la seconda credenziale* `- c3_edit_text : EditText` » *Campo che contiene la terza credenziale*
Metodi	`- sendRequestToServer() : void` » *Metodo che viene invocato al momento della conferma delle credenziali, delegando la chiamata REST alla classe AsyncCallTask*

4.3.4.4 `PinActivity`

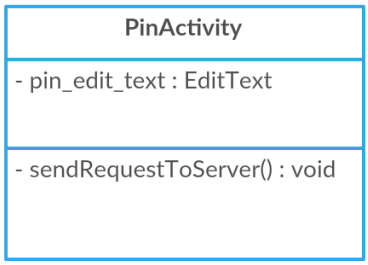


figura 4.11: Diagramma della classe PinActivity

Descrizione	Activity che gestisce l'inserimento del PIN e la risposta che viene ricevuta dal server di Vision Learning al momento del suo invio
Proprietà	`- pin_edit_text : EditText` » *Campo che contiene il PIN*
Metodi	`- sendRequestToServer() : void` » *Metodo che viene invocato al momento della conferma delle credenziali, delegando la chiamata REST alla classe AsyncCallTask*

4.3.4.5 `AsyncCallTask`

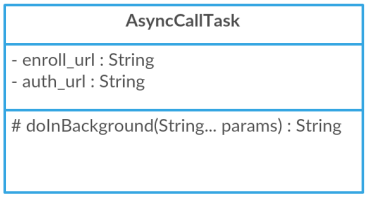


figura 4.12: Diagramma della classe AsyncCallTask

Descrizione	Classe che gestisce la chiamata REST al server di Vision Learning
Proprietà	`- enroll_url : String` » *Stringa che contiene l'URL per fare la chiamata REST di enrollment al server di Vision Learning* `- auth_url : String` » *Stringa che contiene l'URL per fare la chiamata REST di enrollment al server di Vision Learning*
Metodi	`# doInBackground(String... params) : String` » *Metodo che crea ed esegue le chiamate di enroll o di autenticazione al server di Vision Learning*

4.3.5 Documentazione

Durante il mio stage ho prodotto la documentazione del prototipo in *Javadoc*. Per quanto riguarda il periodo di ricerca, ho catalogato quanto raccolto dalle aziende in maniera ordinata, e steso un foglio per comparare le varie soluzioni. Alla fine ho inoltre preparato due presentazioni: una commerciale a cui hanno partecipato anche membri come il presidente di Wintech, la responsabile umane, il direttore marketing, il direttore commerciale e una più tecnica a cui hanno partecipato alcuni membri del personale di Vision Learning.

4.4 Prodotto finale

Segue una serie di screenshot che mostrano l'applicazione nelle sue parti più importanti.

4.4.1 Schermata principale

La schermata iniziale dell'applicazione è la seguente, e da questa è possibile passare alla procedura di registrazione o di autenticazione:

figura 4.13: Schermata principale della applicazione

4.4.2 Registrazione

La procedura di registrazione inizia con l'enrollment. Se l'enrollment ha avuto successo si passa alla schermata di inserimento credenziali (sfuocate per motivi di riservatezza).

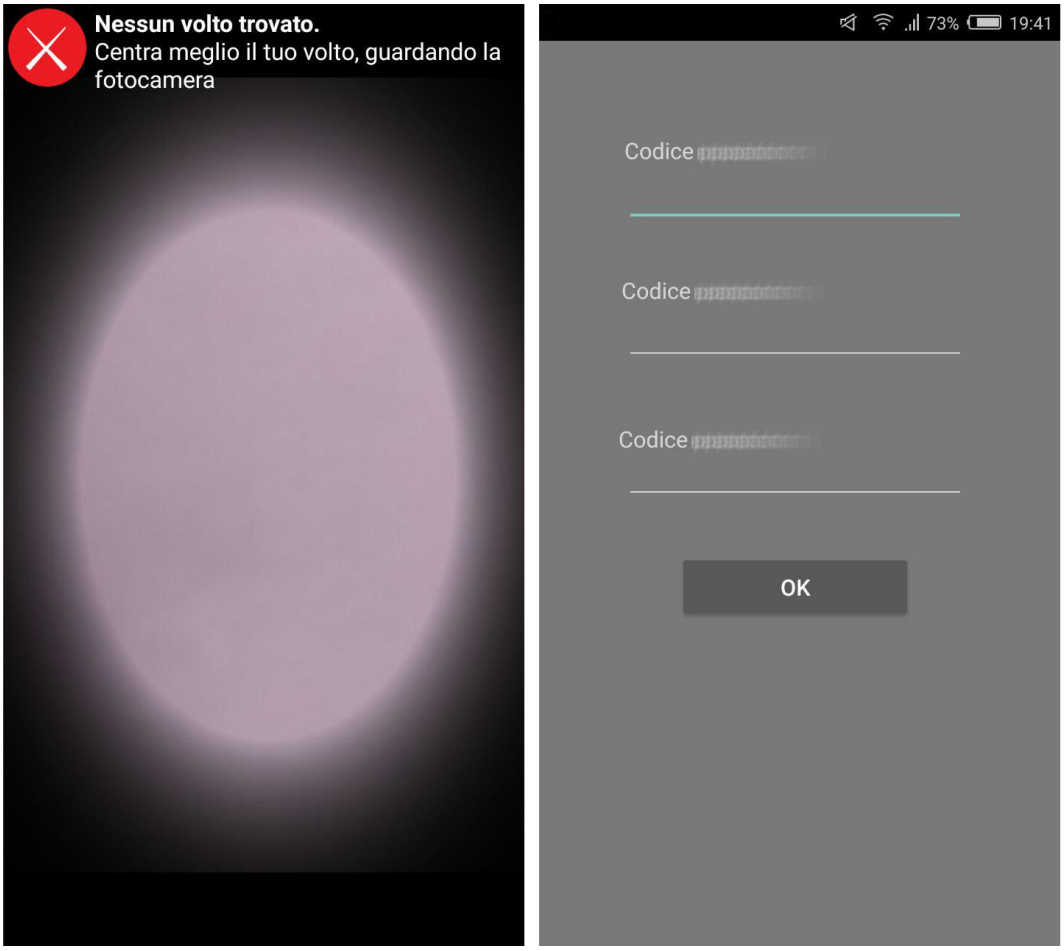


figura 4.14: Schermata di enrollment e di inserimento credenziali

Al termine della procedura si è registrati nel server di Vision Learning e viene mostrato all'utente un messaggio di avvenuta registrazione.

4.4.3 Autenticazione

La procedura di autenticazione inizia con la bio-autenticazione. Questa schermata è del tutto simile alla schermata di enrollment.
Se la bio-autenticazione ha avuto successo si passa alla schermata di inserimento PIN:

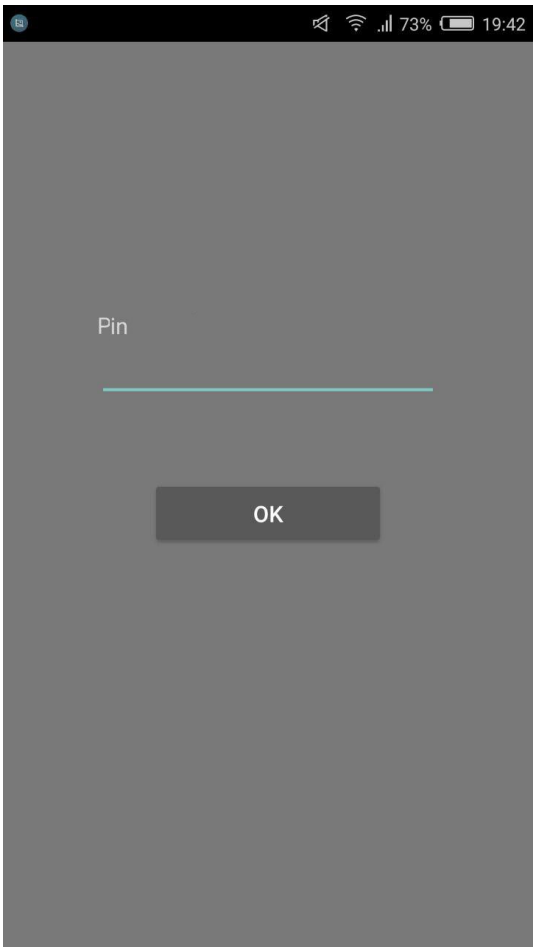


figura 4.15: Schermata di inserimento PIN

Al termine della procedura di autenticazione il video scelto dall'utente viene sbloccato.

4.5 Qualifica

4.5.1 Verifica

Il processo di verifica serve ad accertare che il prodotto (o le sue parti) soddisfi gli obiettivi ed i requisiti precedentemente fissati, e che tale prodotto venga costruito nel modo giusto.

Durante il mio stage, lo sviluppo dei nuovi servizi è avvenuto concorrentemente alle attività di analisi statica e dinamica.

4.5.1.1 Analisi statica

Utilizza tecniche che non richiedono l'esecuzione del prodotto software, e si basa su tecniche di lettura del codice.

Durante il mio stage, l'analisi statica è avvenuta tramite rilettura del codice al raggiungimento di ogni metodo o traguardo significativo di sviluppo.

4.5.1.2 Analisi dinamica

Utilizza tecniche che richiedono l'esecuzione del prodotto software. Spesso si avvale di test progettati per essere ripetibili ed utilizzabili ogni volta che viene effettuata un modifica sul software.

Durante il mio stage, a causa della natura prototipale del prodotto e dato il breve tempo che ho avuto da dedicare al prototipo (circa quattro settimane), discutendo con il tutor aziendale abbiamo convenuto che sarebbe stato meglio non dedicarsi alla progettazione di un sistema automatizzato di verifica, ma piuttosto investire il tempo residuo per ultimare il prodotto con l'integrazione effettiva con il server di Vision Learning e la documentazione utile per il proseguio del prodotto.

Durante tutto lo svolgimento del progetto, ho sostenuto varie revisioni di avanzamento a livello informale, tra me e il tutor aziendale, in relazione ai progressi svolti. Queste revisioni sono state utili per dirigermi verso l'implementazione della soluzione più corretta del prototipo.
Inoltre nel mio progetto sono state inserite via codice (*hard-coded*) alcune stringhe. Tali stub rappresentavano l'input che permetteva di testare informalmente il comportamento di un certo metodo.
Infine ho verificato quanto prodotto tramite collaudo, effettuando i test di accettazione.

4.5.2 Validazione

Il processo di validazione ha invece lo scopo di accertare che il prodotto finale corrisponda alle attese soddisfando tutti i requisiti prefissati inizialmente. costruito nel modo giusto.

Durante il mio stage, la validazione del prodotto è avvenuta su due fronti: validazione interna e validazione esterna.

4.5.2.1 Test di accettazione

Segue un elenco dei test di accettazione effettuati durante il mio stage.

Requisito	Descrizione	Esito
TA1	Dopo aver aperto l'applicazione, l'utente può effettuare l'enrollment	Superato
TA2	L'utente può interrompere l'enrollment	Superato
TA3	Al momento dell'interruzione dell'enrollment viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato
TA4	In caso non venga riconosciuto alcun volto umano durante l'enrollment viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato
TA5	L'utente può inserire le credenziali al termine dell'enrollment se quest'ultimo è avvenuto con successo	Superato
TA6	L'utente, confermando le credenziali, se queste sono corrette e non ci sono errori di connessione, viene registrato nei sistemi di Vision Learning	Superato
TA7	L'utente può interrompere l'inserimento delle credenziali	Superato
TA8	Al momento dell'interruzione dell'inserimento delle credenziali viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato
TA9	In caso di credenziali errate viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato

TA10	In caso di errore di connessione viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato
TA11	Dopo aver aperto l'applicazione, l'utente può effettuare la bio-autenticazione	Superato
TA12	L'utente può interrompere la bio-autenticazione	Superato
TA13	Al momento dell'interruzione della bio-autenticazione viene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato
TA14	In caso non venga riconosciuto alcun volto umano e in caso di fallimento di bio-autenticazione viene mostrato un messaggio di non avvenuta autenticazione e la app ritorna alla schermata iniziale	Superato
TA15	L'utente può inserire il PIN se la procedura di bio-autenticazione è andata a buon fine	Superato
TA16	L'utente, confermando il PIN, se questo è corretto e non ci sono errori di connessione, viene autenticato nei sistemi di Vision Learning e il video bloccato nella piattaforma di e-learning viene sbloccato	Superato
TA17	L'utente può interrompere l'inserimento del PIN	Superato
TA18	Al momento dell'interruzione dell'inserimento del PIN viene mostrato un messaggio di non avvenuta autenticazione; la app ritorna alla schermata iniziale	Superato
TA19	In caso di PIN errato o in caso di errore di connessioneviene mostrato un messaggio di non avvenuta registrazione e la applicazione ritorna alla schermata iniziale	Superato
TA20	L'enrollment deve durare meno di 15 secondi	Superato
TA21	La bio-autenticazione deve durare meno di 4 secondi	Superato

tabella 4.20: Test effettuati sul prodotto

4.5.2.2 Validazione interna

Viene svolta da chi ha sviluppato il sistema/prodotto. Viene chiamata anche *precollaudo*.

Al termine del mio stage, ho effettuato un precollaudo in modo autonomo alla fine della verifica di tutti i servizi, eseguendo i test sopra elencati e definendo una simulazione del prodotto, che l'azienda aveva esplicitamente richiesto come demo da presentare in apposita occasione.

4.5.2.3 Validazione esterna

Viene svolta dal committente del prodotto o dall'utenza e va a dimostrare che il prodotto dello stage si comporti come specificato. Occorre provare che fornisce tutte le funzionalità, le prestazioni, l'affidabilità richieste e che non fallisca.

Al termine del mio stage, la validazione esterna sul progetto è avvenuta in seguito di una presentazione del lavoro svolto durante il periodo di stage qui in azienda eseguendo i test sopra elencati. Tutti i test in sede di validazione esterna sono stati superati con successo e la presentazione, valutata da alcune figure del personale aziendale interessato, è terminata con demo che dimostrava l'effettiva copertura di tutti i requisiti pianificati ed il soddisfacimento di tutte le funzionalità richieste dai servizi.

Capitolo 5

Conclusioni

5.1 Obbiettivi realizzati

Riporto delle tabelle che mostrano quali obbiettivi dello stage sono riuscita a portare a termine:

5.1.1 Obbiettivi obbligatori

	Obbiettivo	**Risultato**
Ob1	Studio dell'attuale stato di Vision Learning	Realizzato
Ob1.1	studio del contesto e orientamento aziendale di Vision Learning	Realizzato
Ob1.2	studio del sistema attuale di Vision Learning	Realizzato
Ob1.3	studio delle modalità attuali di autenticazione all'interno del sistema di Vision Learning	Realizzato
Ob2	Analisi dei costi e benefici dei sistemi di autenticazione biometrica	Realizzato
Ob2.1	confronto dei sistemi presenti sul mercato e scelta di uno tra essi	Realizzato
Ob2.2	stesura resoconto	Realizzato
Ob3	Studio del sistema di autenticazione scelto di cui al punto *Ob2.1*	Realizzato
Ob4	Studio di Android	Realizzato

tabella 5.1: Resoconto obbiettivi obbligatori realizzati durante lo stage

5.1.1.1 Obbiettivi desiderabili

De1	Progettazione di un prototipo di applicazione Android che consenta l'autenticazione biometrica	Realizzato
De1.1	il prototipo deve consentire l'autenticazione biometrica	Realizzato
De1.2	il prototipo deve essere integrabile nel sistema di Vision Learning	Realizzato
Fa1.3	il prototipo deve essere integrabile con i player multimediali di Vision Learning	Realizzato

tabella 5.2: Resoconto obbiettivi desiderabili realizzati durante lo stage

5.1.2 Obbiettivi facoltativi

Fa1	Realizzazione e testing del prototipo di cui al punto *De1*	Parziale: test automatici non eseguiti
Fa2	Stesura di documentazione minimale del prototipo di cui al punto *De1*	Realizzato
Fa3	Integrazione del prototipo di cui al punto *De1* nel sistema di Vision Learning	Realizzato
Fa4	Integrazione del prototipo di cui al punto *De1* nei player multimediali di Vision Learning	Realizzato

tabella 5.3: Resoconto obbiettivi facoltativi realizzati durante lo stage

5.2 Considerazioni personali

Lo stage prima della laurea porta un sensibile valore aggiunto al corso di studi e favorisce l'inserimento dello studente nel mondo del lavoro.
Per quanto mi riguarda, ho avuto modo di approfondire e comprendere meglio la metodologia Agile, ho imparato molto sul tema dell'autenticazione biometrica, ho approfondito il funzionamento dello stile architetturale REST, ho preso coscienza di cosa significhi system integration.
Per concludere, sono stata in grado di realizzare tutte le funzionalità previste e, considerando il lavoro nel complesso, ho sviluppato un prototipo che costituisce un'importante punto di partenza per la creazione della applicazione che Wintech vuole rilasciare.
Mi sento soddisfatta dell'esperienza, mi ha dato l'opportunità di entrare nel mondo del lavoro e confrontarmi con esso costituendo un'importante opportunità di crescita personale.

Glossario

API In informatica con il termine *Application Programming Interface API* (ing. interfaccia di programmazione di un'applicazione) si indica ogni insieme di procedure disponibili al programmatore, di solito raggruppate a formare un set di strumenti specifici per l'espletamento di un determinato compito all'interno di un certo programma. La finalità è ottenere un'astrazione, di solito tra l'hardware e il programmatore o tra software a basso e quello ad alto livello semplificando così il lavoro di programmazione. 12, 15, 25, 27, 43, 61

Back-end Il back-end è la parte di un software che elabora i dati generati dal front-end. Il back-end incapsula la logica di elaborazione dei dati, e non interagisce direttamente con l'utilizzatore.. 43, 61

Bio-autenticazione (oppure Verifica biometrica) è il confronto automatizzato tra un modello biometrico acquisito nel momento in cui l'interessato interagisce con il sistema biometrico e un modello biometrico previamente memorizzato e (presuntivamente) a lui corrispondente; questo tipo di verifica è detta confronto uno a uno (one-to-one comparison).. 12, 13, 24–29, 37, 38, 40, 41, 45, 51, 54, 61

Business unit Le unità strategiche di business, aree strategiche d'affari, sono comparti di un'impresa con mercati e strategie autonomi. Si tratta di segmenti prodotto-mercato fondamentali per l'impresa non solo a livello di marketing ma anche a livello strategico (business).. 1, 61

Commit Nei sistemi di controllo di versione, un commit è l'operazione atomica che aggiunge le modifiche più recenti a (parte del) codice sorgente nel repository, rendendo queste modifiche parte della revisione dell'head del repository.. 3, 61

Design pattern Soluzione progettuale generale a un problema ricorrente. Una descrizione o un modello da applicare per risolvere un problema che può presentarsi in diverse situazioni durante la progettazione e lo sviluppo del software.. 42, 61

Enrollment enrolment: iscrizione in un sistema, nel caso in oggetto, in un sistema biometrico. La fase di enrolment va dall'acquisizione del campione biometrico alla sua memorizzazione, all'estrazione dei tratti fino alla generazione del riferimento

biometrico da archiviare per i confronti successivi.. xi, 12, 13, 17, 20, 24–29, 33–35, 40, 41, 45, 47, 50, 51, 53, 54, 61

Framework Un framework, in informatica e specificatamente nello sviluppo software, è un'architettura logica di supporto su cui un software può essere progettato e realizzato, spesso facilitandone lo sviluppo da parte del programmatore.. 42, 62

Frontend Nel campo della progettazione software, il front-end è la parte di un sistema software che gestisce l'interazione con l'utente o con sistemi esterni che producono dati di ingresso.. 43, 62

IDE In informatica un ambiente di sviluppo integrato (in lingua inglese integrated development environment) è un software che, in fase di programmazione, aiuta i programmatori nello sviluppo del codice sorgente di un programma. Spesso l'IDE aiuta lo sviluppatore segnalando errori di sintassi del codice direttamente in fase di scrittura, oltre a tutta una serie di strumenti e funzionalità di supporto alla fase di sviluppo e debugging.. 42, 62

Istanza Biometrica modello biometrico generato ogni volta che l'interessato interagisce con il sistema biometrico.. 12, 28, 62

JSON (JavaScript Object Notation) Standard usato per trasmettere dati tramite lo scambio di oggetti nei quali le informazioni sono salvate in coppie chiave-valore.. 43, 62

Liveness Test Test che viene effettuato durante l'algoritmo di riconoscimento biometrico per capire se l'utente che sta per essere riconosciuto sia un individuo vivente o una sua immagine, un suo video.. 10, 12, 25–27, 62

Metodologia agile L'espressione metodologia agile si riferisce a un insieme di metodi di sviluppo del software fondati su un insieme di principi comuni, direttamente o indirettamente derivati dai princìpi del "Manifesto per lo sviluppo agile del software". I metodi agili si contrappongono al modello a cascata e altri processi software tradizionali, proponendo un approccio meno strutturato e focalizzato sull'obiettivo di consegnare al cliente, in tempi brevi e frequentemente software funzionante e di qualità.. 2, 29, 62

REST Representational State Transfer. Si tratta di un tipo di architettura software per i sistemi di ipertesto distribuiti come il World Wide Web. Un concetto importante in REST è l'esistenza di risorse (fonti di informazioni), a cui si può accedere tramite un identificatore globale (un URI). Per utilizzare le risorse, le componenti di una rete (componenti client e server) comunicano attraverso una interfaccia standard (ad es. HTTP) e si scambiano rappresentazioni di queste risorse.. 5, 42, 43, 46, 47, 62

Riferimento Biometrico modello biometrico utilizzato come termine di confronto e registrato in modo persistente e invariabile nel tempo (a meno di aggiornamenti resi necessari dalle variazioni anche naturali della caratteristica biometrica da cui è estratto). 12, 28, 63

SDK Acronimo di software development kit (SDK, traducibile in italiano come "pacchetto di sviluppo per applicazioni"), in informatica, indica genericamente un insieme di strumenti per lo sviluppo e la documentazione di software.. 12, 15, 24–27, 42, 63

System integration In informatica, system integration è il processo che collega assieme differenti sistemi informatici e applicazioni software, fisicamente o funzionalmente, allo scopo di creare un unico sistema con funzionalità superiori alle componenti singole di partenza.. 1, 63

System integrator Con il termine inglese system integrator viene indicata una azienda (o uno specialista) che si occupa dell'integrazione di sistemi. Il compito del system integrator è quello di far dialogare impianti diversi tra di loro allo scopo di creare una nuova struttura funzionale che possa utilizzare sinergicamente le potenzialità degli impianti d'origine e creando quindi funzionalità originariamente non presenti.. 9, 63

Teamviewer Software gratuito per il controllo remoto dei computer. Si tratta, dunque, di un programma che consente di dirigere i computer a distanza e di ricevere assistenza remota tramite Internet.. 1, 63

UML In ingegneria del software *UML, Unified Modeling Language* (ing. linguaggio di modellazione unificato) è un linguaggio di modellazione e specifica basato sul paradigma object-oriented. L'*UML* svolge un'importantissima funzione di "lingua franca" nella comunità della progettazione e programmazione a oggetti. Gran parte della letteratura di settore usa tale linguaggio per descrivere soluzioni analitiche e progettuali in modo sintetico e comprensibile a un vasto pubblico. 30, 63

Bibliografia

Siti consultati

Wikipedia, sistemi di riconoscimento biometrico:
URL: https://it.wikipedia.org/wiki/Sistema_di_riconoscimento_biometrico/

Sistemi biometrici:
URL: http://www.privacy.it/archivio/cnipabiometria.html

Regolamento per la formazione professionale continua:
URL: http://www.odcecviterbo.it/gesFiles/Filez/1450360763K145923.pdf

Stack Overflow:
URL: http://stackoverflow.com/

ZoOm Login:
URL: https://zoomlogin.com/

Android:
URL: https://www.android.com/