



Laurea Triennale in Informatica - Università degli Studi di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba



REPORT.IT
ITALIAN DISCRIMINATION COMMUNITY

System Design Document

Report.it

Riferimento	Report.it_SDD_V2.0
Versione	2.0
Data	12/02/2023
Destinatario	F. Ferrucci
Presentato da	Riccardo Kevin Ferraris, Nicola Frugieri, Alberto Genovese, Arturo Gentile, Marisa La Sorda, Leonardo Schiavo
Approvato da	Simona Grieco, Maria Concetta Schiavone



Laurea Triennale in Informatica - Università degli Studi di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba

Team composition

Ruolo	Nome	Acronimo	Contatti
Top Management	Filomena Ferrucci	FF	f.ferrucci@unisa.it
Project Manager	Simona Grieco	SG	s.grieco13@studenti.unisa.it
Project Manager	Maria Concetta Schiavone	MCS	m.schiavone29@studenti.unisa.it
Team Member	Riccardo Kevin Ferraris	RKF	r.ferraris1@studenti.unisa.it
Team Member	Nicola Frugieri	NF	n.frugieri@studenti.unisa.it
Team Member	Alberto Genovese	ALG	a.genovese42@studenti.unisa.it
Team Member	Arturo Gentile	ARG	a.gentile38@studenti.unisa.it
Team Member	Marisa La Sorda	MLS	m.lasorda@studenti.unisa.it
Team Member	Leonardo Schiavo	LS	l.schiavo15@studenti.unisa.it



Sommario

Team composition	2
Revision History	4
1. Introduzione	5
1.1 Scopo del Sistema.....	5
1.2 Design Goals	6
1.3 Trade-offs	9
1.4 Definizioni, acronimi e abbreviazioni	10
1.5 Riferimenti	10
1.6 Panoramica	11
2. Architettura del sistema corrente	12
3. Architettura del sistema proposto	12
3.1 Panoramica	12
3.2 Decomposizione in sottosistemi.....	14
3.3 Mapping Hardware/Software.....	16
3.4 Gestione dati persistenti	18
3.5 Controllo degli accessi e sicurezza.....	21
3.6 Controllo del flusso globale del sistema	22
3.7 Condizioni Boundary.....	22
4. Servizi dei sottosistemi	24
5. Glossario	30



Revision History

Data	Versione	Descrizione	Autori
26/11/2022	0.1	Prima stesura del documento	[TEAM MEMBER]
27/11/2022	0.2	Introduzione e Architettura del Sistema Corrente	[TEAM MEMBER]
30/11/2022	0.3	Mapping Hardware/Software, Condizioni Boundary, Servizi dei sottosistemi	NF
30/11/2022	0.4	Panoramica, Condizioni Boundary	MLS
30/11/2022	0.5	Decomposizione in sottosistemi, Controllo degli accessi e sicurezza	ALG e LS
30/11/2022	0.6	Gestione dati persistenti, Servizi dei sottosistemi	ARG
30/11/2022	0.7	Controllo del flusso globale del sistema	RKF
01/12/2022	0.8	Diagramma Architettuale	ALG e LS
01/12/2022	0.9	Servizi dei sottosistemi	[TEAM MEMBER]
02/12/2022	0.10	Revisione Design Goals, aggiunta di un Trade-Off	ARG e LS
03/12/2022	0.11	Revisione e Glossario	LS
09/12/2022	1.0	Revisione documento	[TEAM MEMBER]
12/02/2023	2.0	Revisione finale documento	[TEAM MEMBER]



1. Introduzione

1.1 Scopo del Sistema

Al giorno d'oggi, per le vittime di discriminazione, effettuare una denuncia può essere un procedimento lungo ed ostacolato che può implicare anche un peggioramento dello stato emotivo e psicologico della vittima. A tal proposito il sistema proposto, Report.it, ha lo scopo di fornire una piattaforma digitale ai servizi della Pubblica Amministrazione per snellire l'iter di denuncia per discriminazioni di ogni tipo, supportato dalla possibilità di aver accesso anche allo sportello psicologico, recentemente introdotto nel SSN, ma ostacolato dai tempi lunghi di gestione delle richieste di aiuto.

Ad oggi è presente una forte discrepanza tra denunce effettuate e numero di discriminazioni commesse giornalmente sul territorio italiano ed un motivo per cui è presente questa forte differenza, risiede nel fatto che le vittime provano vergogna per il gesto subito e spesso non si sentono in una posizione sicura per poter emettere una denuncia. Inoltre, dato lo stato confusionale in cui la vittima può ritrovarsi inizialmente, è possibile avvalersi anche dell'utilizzo di una mappa al fine di trovare la caserma e/o il centro ASL più vicino.

Il sistema Report.it si propone come ponte comunicativo tra la Pubblica Amministrazione ed il cittadino, snellendo il carico di lavoro per il primo e mettendo a maggior agio il secondo, permettendogli di avere, in un unico luogo, accesso sia alla denuncia che allo sportello psicologico al fine di avere un sostegno non solo giuridico ma anche psicologico.

Il sistema si propone anche come luogo di informazione e di scambio di opinioni tramite l'utilizzo di una sezione di aiuto ed una sezione forum dove i cittadini possono interagire, confrontarsi e poter discutere dell'argomento.

Ad oggi un sistema simile non è ancora presente nell'ambito della Pubblica Amministrazione Italiana, se non per:

- reati minori come piccoli furti e smarrimenti (<https://denunceviaweb.poliziadistato.it/>) oppure <https://extranet.carabinieri.it/DenunciaWeb/>), rendendo ancora oggi obbligatorio il recarsi fisicamente in caserma per poter confermare e concludere la pratica di denuncia.
- segnalazioni (non denuncia) di sole discriminazioni razziali (<https://www.unar.it/portale/home>) e atti di bullismo, violenza domestica e spaccio



(<https://play.google.com/store/apps/details?id=it.poliziadistato.youpol&hl=it&gl=US&pli=1>)

1.2 Design Goals

In questo paragrafo vengono esposti i Design Goals (“Obiettivi di Design”), che descrivono le qualità chiave del sistema (quelle che devono essere ottimizzate) e stabiliscono le loro priorità. Sviluppando in modo corretto i Design Goal sarà possibile stabilire un percorso di sviluppo ben delineato, solido e dettagliato.

Nella tabella che segue, i design goal sono descritti in base ai seguenti campi, ognuno corrispondente a una colonna:

- **Rank:** ne indica la priorità rispetto agli altri design goal;
- **ID Design Goal:** un identificatore;
- **Descrizione:** ne descrive le caratteristiche;
- **Categoria:** raggruppa i design goal che descrivono qualità che rientrano nella stessa macroarea;
- **Origine:** il requisito non funzionale da cui è stato generato;
- **Trade off:** rappresenta una scelta tra due o più parametri, dove la diminuzione di uno costituisce l'aumento dell'altro.

In particolare, le categorie in cui sono stati suddivisi i design goal sono:

- **Performance:** indica, in maniera quantificabile, il livello di prestazioni del sistema software;
- **Dependability:** relativo all'affidabilità del sistema (gestione degli errori, resistenza ai crash...);
- **Cost:** indice dell'impatto economico dell'implementazione di una qualità sui costi di sviluppo;
- **Maintenance:** indica quanto effort è necessario per apportare modifiche al sistema dopo la prima release;
- **End user:** relativo ai canoni dell'interazione uomo-macchina e di ingente importanza per valutare la compatibilità di un sistema software con l'utenza.



Rank / Priorità	ID Design Goal	Descrizione design goal	Categoria	Origine	Trade-off
1	DG_1 Funzionalità Richieste	Requisiti funzionali richiesti dal committente con priorità alta saranno forniti nella prima release del prodotto.	Dependability	RNF_F.1	Tempo di rilascio vs Funzionalità
2	DG_2 Osservanza della Normativa per la privacy	Tutti i dati sensibili gestiti dal sistema sono tutelati ai sensi del Regolamento UE n. 679/2016 del 27/04/2016	Dependability	RNF_L.1	Tempi di risposta vs Sicurezza
3	DG_3 Sicurezza dati sensibili	Il sistema garantisce la sicurezza dei dati da accessi non autorizzati tramite un sistema di autenticazione basato su SPID	Dependability	RNF_A.2	Tempi di risposta vs Sicurezza
4	DG_4 Riservatezza	Capacità di impedire accessi ad utenti non autorizzati al sistema e a funzionalità a loro negate.	Dependability	RNF_A.6	Tempi di risposta vs Sicurezza
5	DG_5 Disponibilità (Reliability)	Il livello target dell'affidabilità del sistema rispetto alla sua "disponibilità" (ETA in cui il sistema dovrebbe essere reperibile), stabilito dallo	Dependability	RNF_A.1	Costi vs Affidabilità



Laurea Triennale in Informatica - Università degli Studi di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba

		SLO (Obiettivo del Livello di Servizio), deve essere di almeno il 99%. In tal modo, il budget di errore, in un mese commerciale, sarà pari a 43 minuti.			
6	DG_6 Portabilità	Il sistema deve poter essere utilizzato su hardware diversi sotto un unico codice sorgente.	Performance	RNF_I.3	Prestazioni vs Compatibilità
7	DG_7 Multiplatforma	Il sistema deve essere compatibile con i seguenti OS: <ul style="list-style-type: none"> • Android 4.1+ • IOS 11+ 	Maintenance	RNF_PA.1	Prestazioni vs Compatibilità
8	DG_8 Manutenibilità	Il sistema sarà facilmente manutenibile grazie all'elevata modularità del suo codice.	Maintenance	RNF_S.1	Tempo di rilascio vs Qualità
9	DG_9 User Friendly	Il sistema deve essere di facile utilizzo per tutti gli utenti e deve rispettare le euristiche dell'interazione uomo-macchina di J. Nielsen. Le azioni dell'utente saranno eseguite in modo chiaro e	End-user	RNF_U.1	Tempo di rilascio vs Qualità



		senza ambiguità.			
10	DG_10 Tempo di risposta	Il sistema deve rispondere alle azioni dell'utente entro e non oltre i 3 secondi.	Performance	RNF_P.2	Tempi di risposta vs Sicurezza
11	DG_11 Efficienza di spazio	Per garantire una maggiore diffusione e praticità dell'app (anche su device con una memoria di massa non elevata), questa non deve superare i 70 MB di grandezza.	Performance	RNF_P.1	Prestazioni vs Compatibilità
12	DG_12 Restrittività	Il sistema è stato creato per la Pubblica Amministrazione Italiana ed è destinato ai cittadini residenti in Italia ("region restricted to Italy").	Maintenance	RNF_L.2	Prestazioni vs Compatibilità

1.3 Trade-offs

Trade-off	Descrizione
Tempi di risposta vs Sicurezza	Poiché il sistema deve gestire dati sensibili, si può ricorrere a sistemi di sicurezza - come l'autenticazione con SPID - che offrono una maggiore sicurezza ma dilatano i tempi di risposta essendo il sistema esterno all'applicazione, portando quindi il tempo di accesso ad almeno 15 secondi.
Tempo di rilascio vs Qualità	Essendo la deadline invalicabile, si è deciso di dare più importanza ai tempi di consegna; nonostante questo la qualità del software rimarrà entro determinati standard.



Tempo di rilascio vs Funzionalità	Le funzionalità a più bassa priorità potrebbero essere non implementate in caso di mancanza di tempo.
Costi vs Affidabilità	Per permettere un'operatività h24 all'app, il sistema usa un sistema di server di alta qualità, che però risulta più costoso.
Prestazioni vs Compatibilità	Per permettere ad una platea maggiore di utenti di usufruire della nostra app, e per rientrare nella deadline, si è deciso di dare priorità maggiore alla compatibilità piuttosto che allo sviluppo di più applicativi differenti per più sistemi operativi (risultando anche in costi maggiori di sviluppo e manutenzione) che avrebbero avuto performace superiori.

1.4 Definizioni, acronimi e abbreviazioni

Acronimi:

- RAD = Requirement Analysis Document
- SDD = System Design Document
- COTS = Commercial Off-the-Shelf

Definizioni:

- Sottosistema: corrisponde alla parte di lavoro che può essere svolta autonomamente da un singolo sviluppatore o da un gruppo di sviluppatori. Si ottiene decomponendo il sistema.
- Design Goal: proprietà del sistema sulla quale ci si concentra maggiormente.
- Trade-off: possibilità di ridurre una certa qualità per aumentare il valore di un'altra qualità e viceversa. Il termine è espresso talvolta come costo opportunità, riferendosi a più alternative alle quali si è preferito rinunciare a vantaggio di un'altra scelta.
- Mapping Hardware-Software: descritto per indicare i vari device hardware utilizzati dal sistema e la loro interazione con le componenti software.
- Dati Persistenti: dati che devono sopravvivere all'esecuzione singola dell'applicazione. Sono i dati che vengono salvati nel database.

1.5 Riferimenti

I riferimenti sono:



- Libro, “Object Oriented Software Engineering using UML, Patterns and Java”
Edizione: 3rd Edition
Anno: 2014
Autori: Bernd Bruegge, Allen H. Dutoit
- [Report.it RAD](#)
- [Report.it SOW](#)
- [Wikiwand](#)
- [Red Hat](#)

1.6 Panoramica

Il presente documento è strutturato in quattro sezioni principali:

- **Introduzione:** in questa sezione del documento è possibile trovare una descrizione dello scopo del sistema, i vari design goals ordinati per priorità ed arricchiti di descrizione, i trade-offs ed informazioni circa il linguaggio ed i riferimenti utilizzati.
- **Architettura del sistema corrente:** in questa sezione del documento è possibile analizzare l'architettura del sistema attualmente esistente.
- **Architettura del sistema proposto:** in questa sezione del documento è possibile analizzare tramite una descrizione accurata l'architettura del sistema da noi proposto. Tale sezione è a sua volta suddivisa in:
 - decomposizione dei sottosistemi;
 - mapping hardware/software;
 - gestione dati persistenti;
 - controllo degli accessi e sicurezza, del flusso globale del sistema;
 - condizioni boundary.
- **Servizi dei sottosistemi:** in questa sezione del documento è possibile leggere la descrizione dei servizi di ogni sottosistema proposto.
- **Glossario:** sezione del documento che associa ad ogni sigla/termine menzionato/a all'interno del documento una definizione/descrizione per evitare che il lettore vada a ricercare informazioni all'esterno del documento, rendendolo autosufficiente sulla materia trattata.



2. Architettura del sistema corrente

In questo momento non esiste nessun sistema che offre i servizi di Report.it, ossia tracciabilità delle denunce direttamente dal proprio smartphone, creazione di una denuncia completamente in app, ricerca delle ASL e delle Stazioni di Polizia. Se volessimo individuare dei possibili concorrenti, potremmo indicare la possibilità di denunciare solo alcuni reati online, quindi, non esiste una reale architettura a cui è possibile confrontare il sistema.

3. Architettura del sistema proposto

3.1 Panoramica

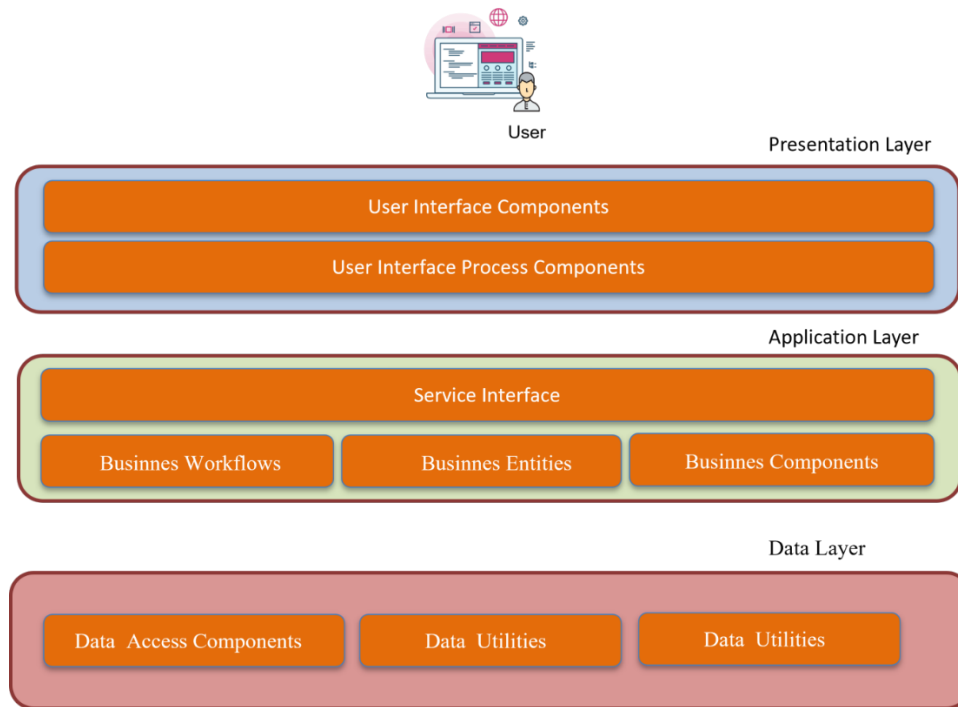
Il sistema da noi proposto è basato sull'architettura **Three-Layer** utilizzando come linguaggio di programmazione Dart ed il suo framework Flutter. Tale modello architetturale è stato scelto in quanto ha il vantaggio di fornire una separazione logica delle componenti software così da aumentarne la manutenibilità, la scalabilità ed il riutilizzo del codice.

Come linguaggi e framework adottati per lo sviluppo del sistema si annoverano:

- **Dart** per lo sviluppo del back-end;
- **Flutter** per lo sviluppo dell'interfaccia grafica;
- **Flutter Fire** per il collegamento al database;
- **Firebase** per il database su cloud.



Laurea Triennale in Informatica - Università degli Studi di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba





3.2 Decomposizione in sottosistemi

Il sistema è stato decomposto nei seguenti sottosistemi:

- **Autenticazione:** il sottosistema è garante dell'accesso al sistema degli attori (per l'Utente attraverso la funzionalità di Login SPID e per l'Operatore CUP e UffPolGiud attraverso il Login con credenziali) e della funzionalità di Logout.
- **Denunce:** il sottosistema si interessa delle funzionalità relative all'inoltro della denuncia per l'Utente e delle varie funzionalità relative alla gestione di queste da parte dell'UffPolGiud.
- **Forum:** il sottosistema si interessa delle funzionalità relative all'uso del forum, in particolare quelle per apertura, eliminazione, visualizzazione, risposta e supporto di una discussione.
- **Prenotazione e Supporto Psicologico:** il sottosistema si interessa delle funzionalità relative all'inoltro di richieste di prenotazione di sedute psicologiche per l'Utente e di approvazione di tali richieste per l'Operatore CUP.
- **Geolocalizzazione:** il sottosistema si interessa delle funzionalità relative alla visualizzazione della mappa e dei suoi marker per l'Utente.
- **Informativa e Contatti:** il sottosistema è garante delle funzionalità che offrono all'Utente l'accesso ai contatti delle FFOO e ASL e l'informativa sulle leggi.
- **Chat-bot:** il sottosistema è garante delle funzionalità collegate al chat-bot per gli attori Utente, UffPolGiud e Operatore CUP.
- **Flutterfire:** il sottosistema collega i sottosistemi Autenticazione, Denunce, Forum, Prenotazione e Supporto Psicologico, Geolocalizzazione, Chat-bot e Informativa e Contatti al database. Questo costituisce un componente COTS che rappresenta un framework di Flutter che collega i sottosistemi al database Firebase con dati persistenti.
- **Firebase:** database che garantisce di conservare dati in modo persistente. Questo costituisce un componente COTS che rappresenta un database cloud-hosted (i dati vengono archiviati come JSON e sincronizzati in tempo reale con ogni client connesso).



Component Diagram UML

Nel seguente Component Diagram UML sono mostrati i sottosistemi e le relative dipendenze tra essi.

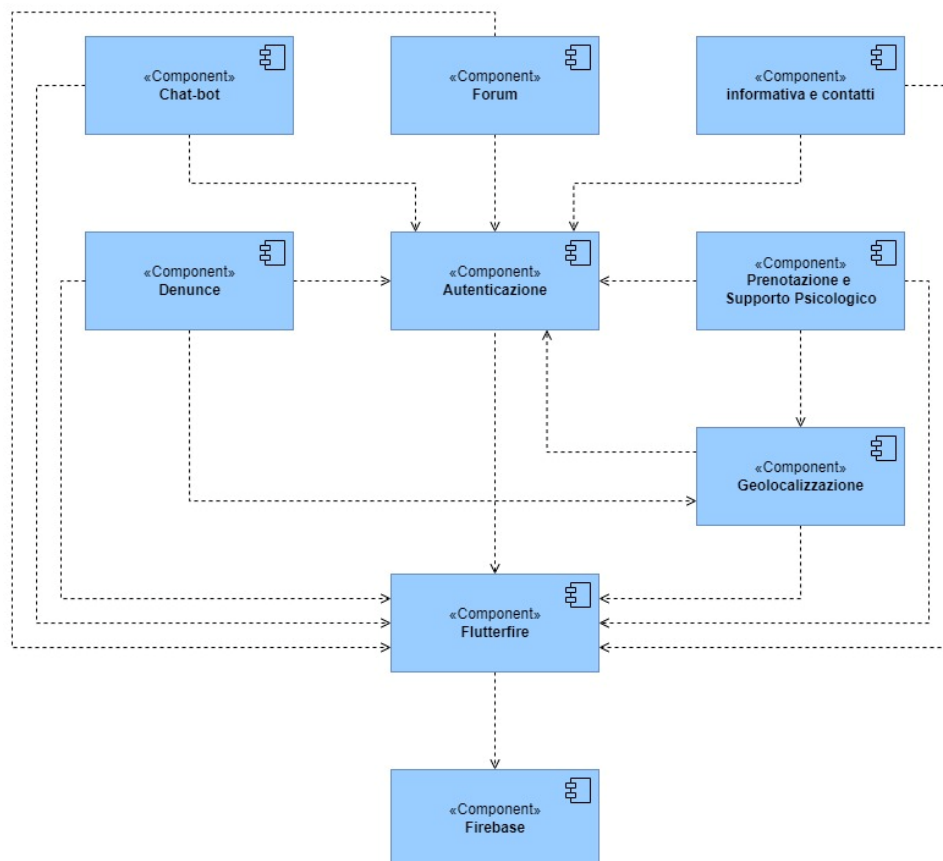
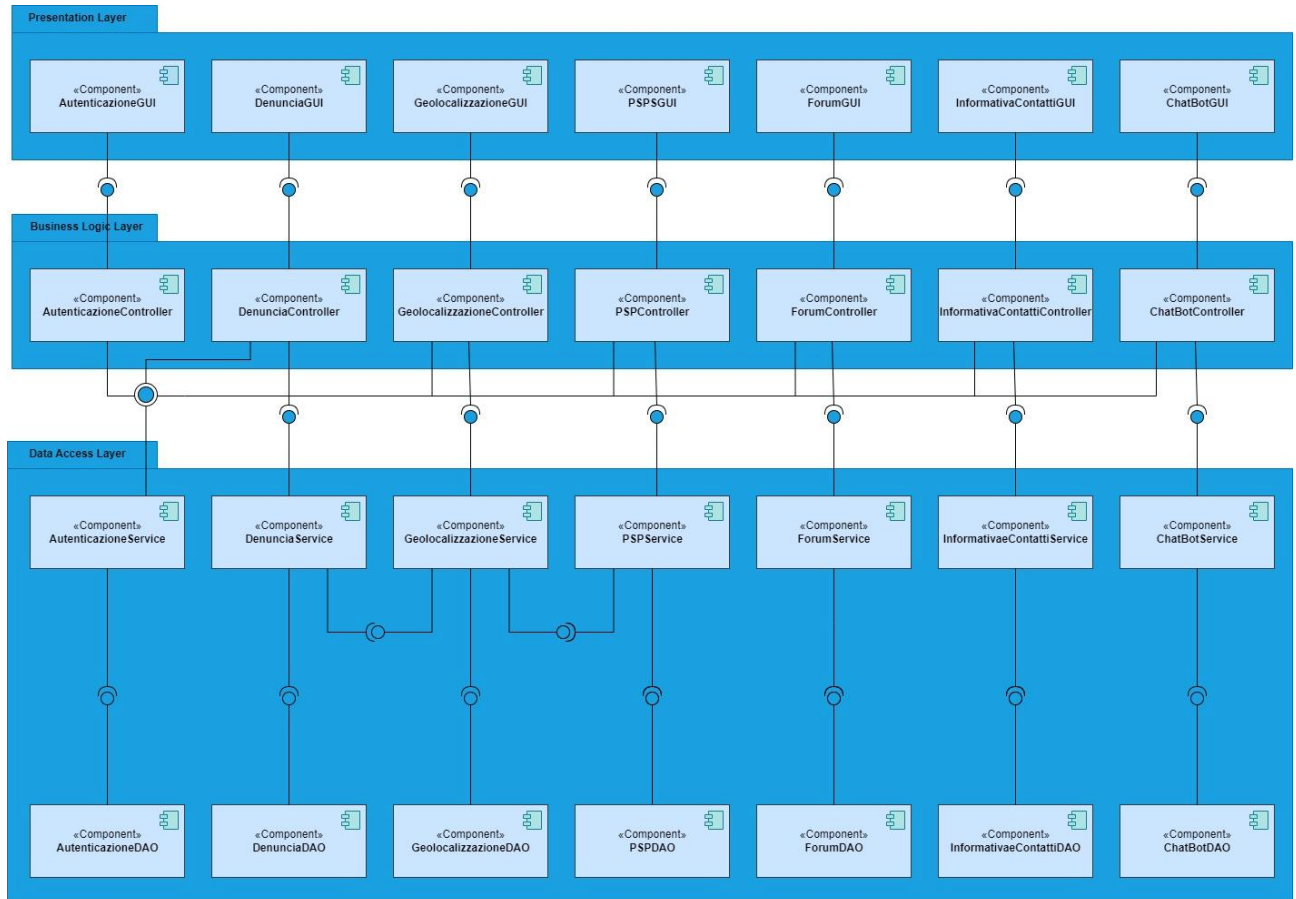




Diagramma Architeturale



3.3 Mapping Hardware/Software

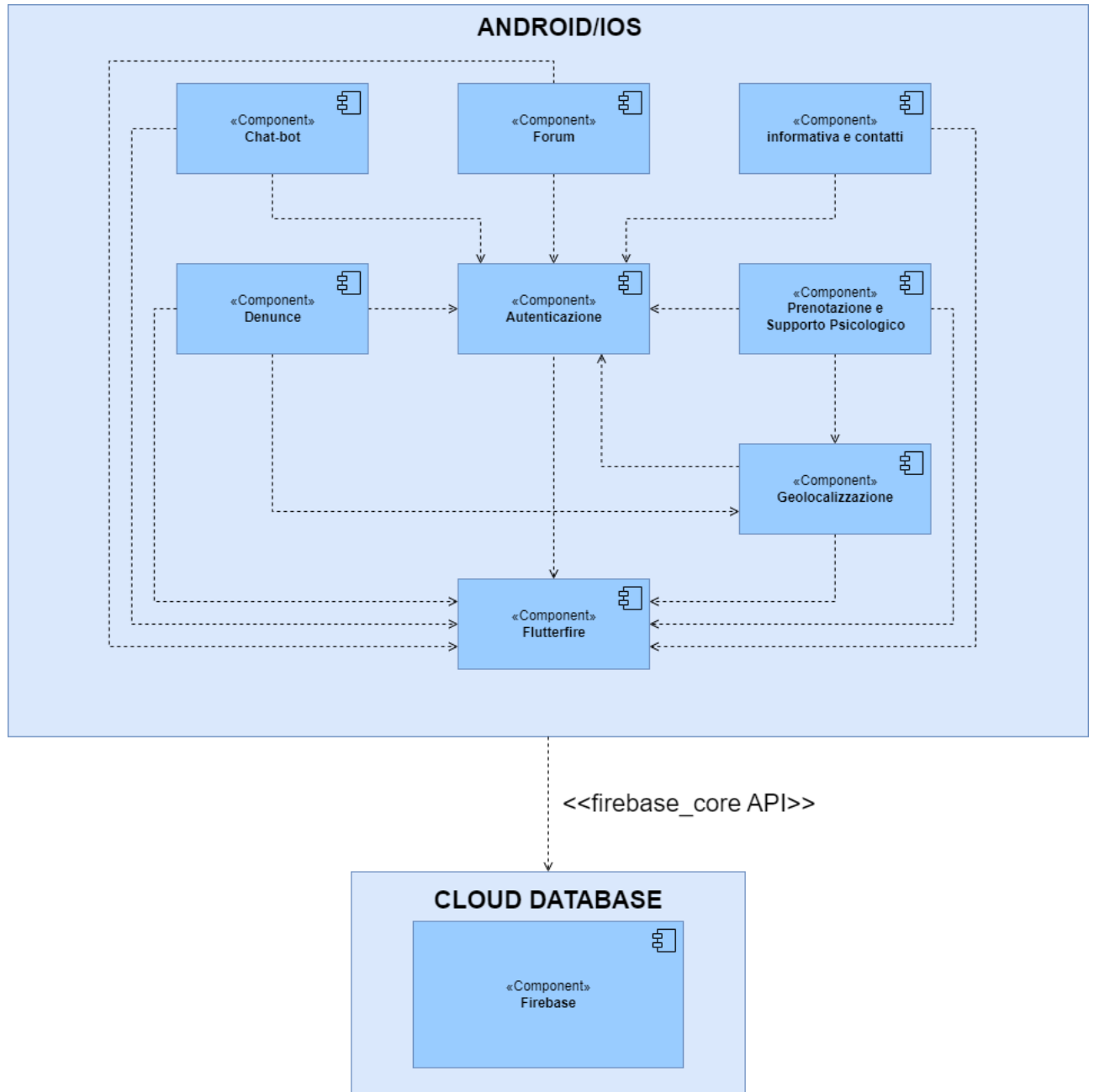
La piattaforma che verrà sviluppata sarà una mobile app scaricabile dagli utenti su dispositivi mobile Android e iOS con una connessione ad Internet. I dati sono memorizzati su un cloud database con cui l'applicazione comunica direttamente tramite appositi plugin forniti dal cloud provider.

A differenza di una web-application, un sistema basato su cloud database utilizza delle API per comunicare con il database, senza utilizzare un webserver come intermediario.

Di seguito un UML Deployment Diagram che descrive il mapping hardware/software:



UML Deployment Diagram





3.4 Gestione dati persistenti

Per la gestione dei dati persistenti del sistema si è deciso di utilizzare un database non relazionale, al fine di avere una maggiore flessibilità rispetto ai DB SQL che sono più rigidi nella struttura.

La scelta di un Database NoSQL (Firebase) è stata presa al fine di:

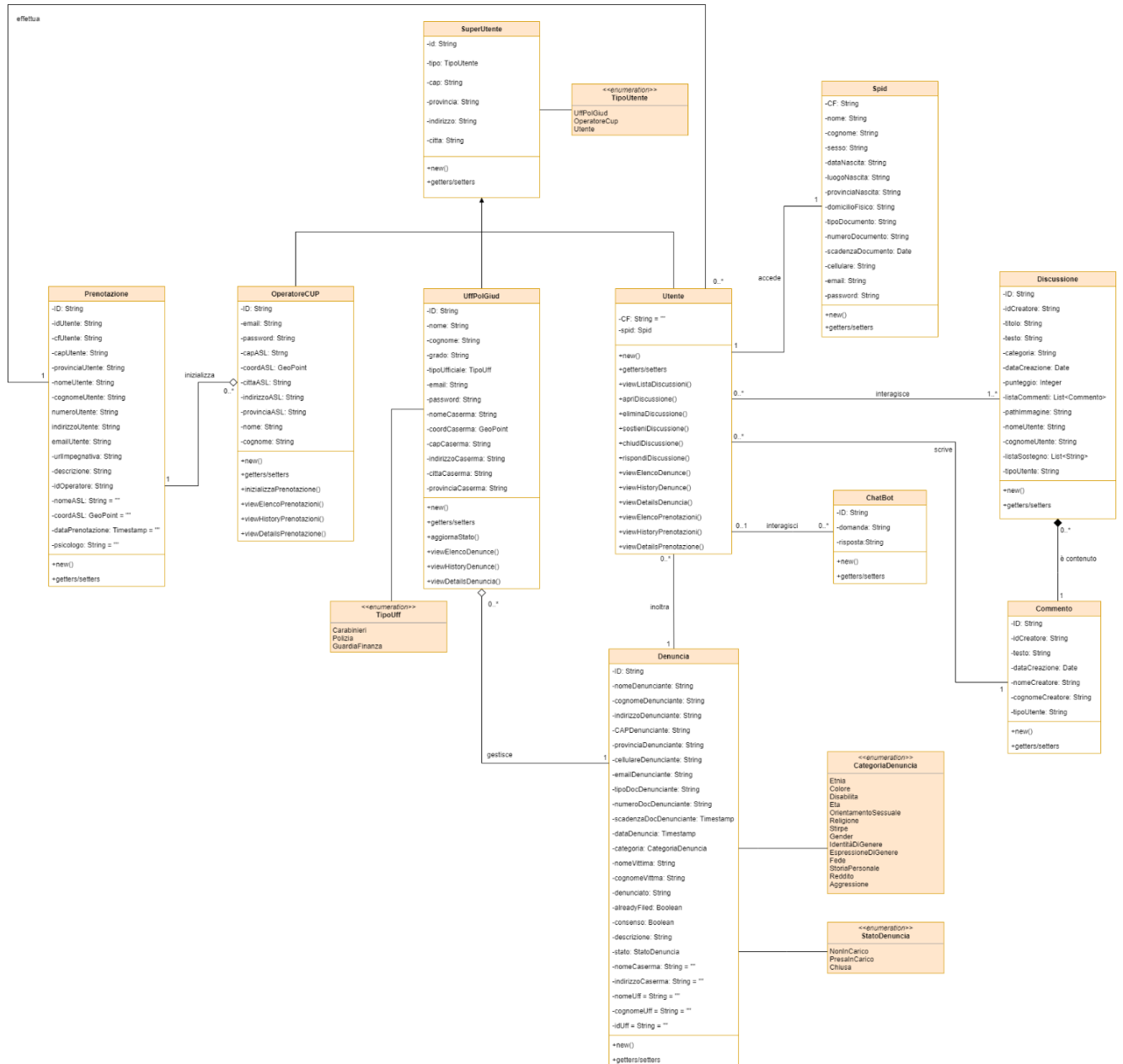
- 1) **Aumentare le prestazioni** – I Database NoSQL sono molto più efficienti rispetto a quelli tradizionali riuscendo a mantenere meglio le prestazioni con una mole di dati molto grande, infatti, è perfetto per il nostro caso visto che andremo a gestire una quantità di dati molto grande;
- 2) **Inserimento dei dati sicuro** – I Database NoSQL ci permettono l’inserimento di nuovi dati, in maniera facile, veloce e priva di rischi, a differenza dei Database SQL;
- 3) **Migliore gestione dei dati offline** – Firebase permette una gestione migliore dei dati quando si è offline, così da non perdere nessun dato in situazioni in cui non si ha la connessione ad Internet ed essendo la nostra piattaforma progettata per dispositivi mobile è un aspetto molto importante.
- 4) **App serverless** – Firebase permette di costruire app senza l’utilizzo di un server, così da semplificare di molto la gestione dei dati che verranno salvati semplicemente in Cloud.

Open source ma supportata da Google, Firebase sfrutta l'infrastruttura di Google e il suo cloud per fornire una suite di strumenti per scrivere, analizzare e mantenere applicazioni cross-platform.

Infatti, Firebase offre funzionalità come analisi, database (usando strutture NoSQL), messaggistica e segnalazione di arresti anomali per la gestione di applicazioni web, iOS e Android.



CD_SDD: Entity Class Diagram

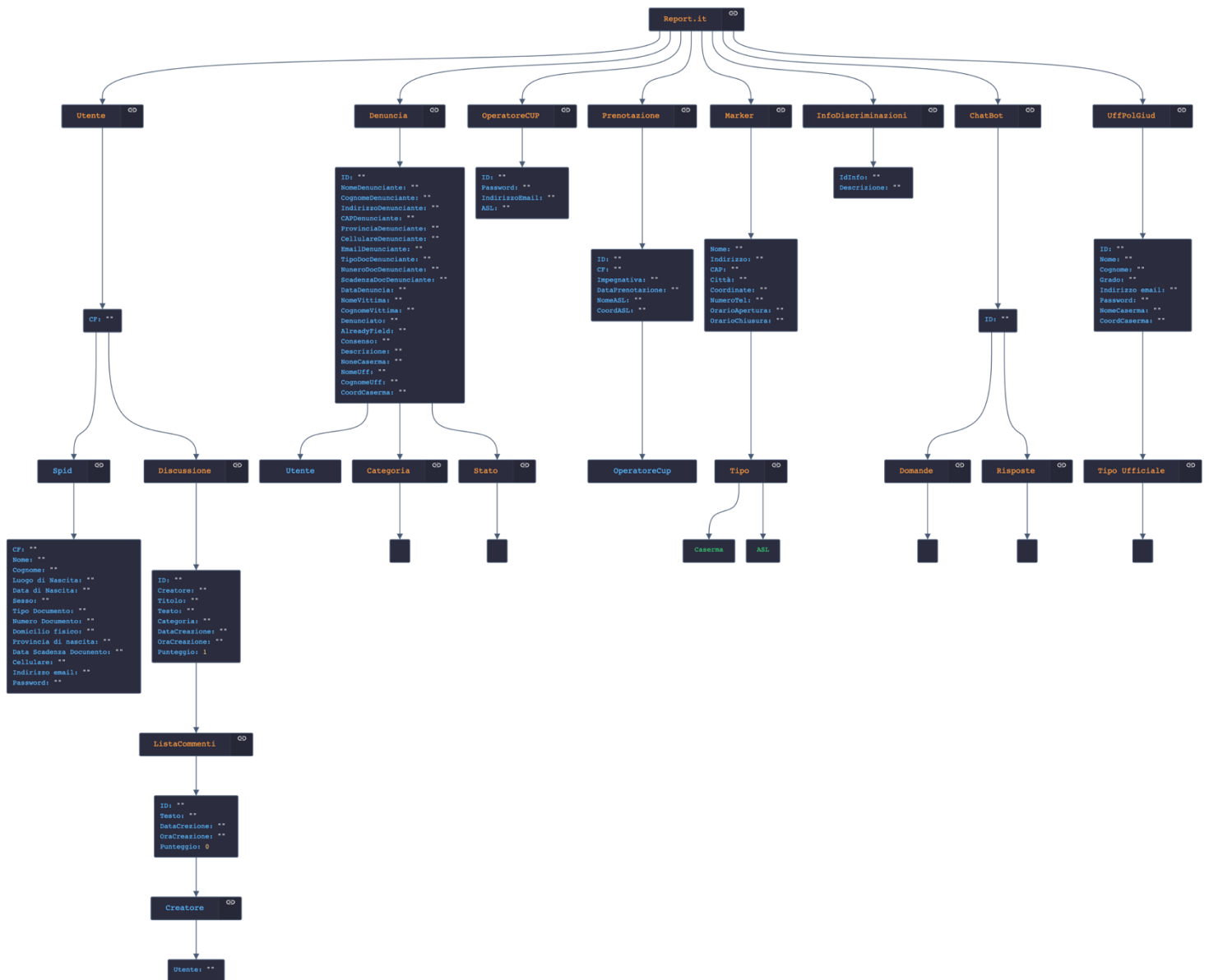




JS_SDD: JSON Structure

Di seguito il link della struttura JSON:

<https://jsoneditoronline.org/#left=cloud.b67adb4c42fb4cf5b3fbab3a0943d6a4>





3.5 Controllo degli accessi e sicurezza

Oggetti \ Attori	Utente	OperatoreCUP
Autenticazione	LoginSPID Logout	Login Logout
Denunce	InoltraDenuncia VisualizzaElencoDenunce VisualizzaDettagliDenunce VisualizzaStatoDenunce	-
Geolocalizzazione	VisualizzaMappaConMarker VisualizzaInformazioniSuMarker	-
Informativa e Contatti	ContattiFFOOeASL InformativaLeggiDiscriminazioni	-
Prenotazione e Supporto Psicologico	InoltraRichiestaDiPrenotazione VisualizzaListaPrenotazioneAttive VisualizzaStoricoPrenotazioni VisualizzaDettagliPrenotazione	VisualizzaListaPrenotazioniAttive VisualizzaStoricoPrenotazioni VisualizzaDettagliPrenotazione InizializzaPrenotazione
Forum	VisualizzaDiscussioneSuForum ApriDiscussioneSuForum EliminaDiscussioneSuForum RispondiDiscussioneSuForum SostieniDiscussione ChiudiDiscussione NotificheForum	-
Chat-bot	AvviaChat-Bot	AvviaChat-Bot

Oggetti \ Attori	UffPolGiud
Autenticazione	Login Logout



Denuce	VisualizzaElencoDenunce VisualizzaDettagliDenunce VisualizzaStatoDenuncia CambiaStatoDenuncia <u>VisualizzaStoricoDenunceInCarico</u>
Geolocalizzazione	-
Informativa e Contatti	-
Prenotazione e Supporto Psicologico	-
Forum	-
Chat-bot	AvviaChat-Bot

3.6 Controllo del flusso globale del sistema

Il sistema Report.it è un sistema software interattivo; ciò significa che, tramite l'utilizzo di un'interfaccia grafica, l'utente impartisce dei comandi che avviano una funzionalità specifica. L'utente invia un comando tramite interfaccia grafica, il sistema selezionerà quindi il controllo corrispondente. Questo porterà il flusso allo strato logico contenente i moduli software che erogano dei servizi implementati in maniera modulare. Infine, si va allo strato logico che contiene la struttura dati utilizzata per immagazzinare, processare e condividere informazioni.

3.7 Condizioni Boundary

In questo paragrafo verranno illustrati le condizioni boundary riguardanti il fallimento del sistema ed errore di accesso a dati persistenti.

Avvio del sistema e spegnimento del sistema non sono presi in considerazione in quanto l'applicazione si affida ad un database in cloud, per cui avvio e spegnimento non sono previsti, essendo il servizio di sua natura sempre disponibile.



Fallimento del sistema

Identificativo	UCBC_1 – Fallimento del sistema	Data	30/11/2022
		Versione	1.0
		Autori	Marisa La Sorda
Descrizione	L'UC descrive il comportamento dell'utente in caso di fallimento da parte del sistema.		
Attore principale	L'utente		
Attori secondari	NA		
Entry condition	Il sistema non risponde alla richiesta di connessione.		
Exit condition On success	L'utente ritenta la connessione riavviando l'applicazione.		
Exit condition On failure	L'utente non può utilizzare l'applicazione.		
Flusso di eventi principale			
1	Sistema	Notifica all'utente che non è stato possibile stabilire la connessione.	
2	Utente	Ritenta la connessione riavviando l'applicazione.	

Errore di accesso ai dati persistenti

Identificativo	UCBC_2 – Errore di accesso ai dati persistenti	Data	30/11/2022
		Versione	1.0
		Autori	Nicola Frugieri
Descrizione	L'UC descrive il comportamento dell'utente in caso di errore di accesso ai dati persistenti da parte del sistema.		
Attore principale	Sistema		
Attori secondari	NA		
Entry condition	Il sistema non riesce ad accedere ai dati persistenti OR I dati risultano corrotti.		



Exit condition On success		Il Sistema riprende il normale funzionamento
Exit condition On failure		Il Sistema non riprende il normale funzionamento
Flusso di eventi principale		
1	Sistema	Restituisce un messaggio di errore e invita l'utente a riprovare più tardi

4. Servizi dei sottosistemi

In questa sezione vengono descritti i servizi di ogni sottosistema precedentemente elencati.

Sottosistema Autenticazione

Servizio	Descrizione	Interfaccia
Login SPID	Questa funzionalità permette di effettuare l'accesso al sistema tramite le proprie credenziali SPID per sfruttare le funzionalità lato utente.	AutenticazioneService
Login standard	Questa funzionalità permette di effettuare l'accesso al sistema tramite e-mail e password per sfruttare le funzionalità lato UffPolGiud e Operatore CUP	AutenticazioneService



Logout	Questa funzionalità permette di disconnettersi dal sistema	AutenticazioneService
--------	--	-----------------------

Sottosistema Forum

Servizio	Descrizione	Interfaccia
Visualizza discussione sul forum	Questa funzionalità permette di vedere le discussioni sul forum	ForumService
Apri discussione sul forum	Questa funzionalità permette di aprire una discussione del forum	ForumService
Elimina discussione sul forum	Questa funzionalità permette di eliminare una discussione dal forum	ForumService
Rispondi ad una discussione sul forum	Questa funzionalità permette di rispondere ad una discussione aperta sul forum	ForumService
Sostieni discussione	Questa funzionalità permette di sostenere una discussione presente sul forum	ForumService
Chiudi discussione	Questa funzionalità permette di chiudere una discussione presente sul forum	ForumService



Sottosistema Denunce

Servizio	Descrizione	Interfaccia
Inoltra denuncia	Questa funzionalità permette di inoltrare una denuncia alle FFOO	DenunciaService
Visualizza elenco denunce	Questa funzionalità permette: <ul style="list-style-type: none">all'Utente di visualizzare l'elenco di tutte le denunce fatte.all'UffPolGiud di visualizzare l'elenco di tutte le denunce ricevute.	DenunciaService
Visualizza dettagli denuncia	Questa funzionalità permette: <ul style="list-style-type: none">all'Utente di visualizzare i dettagli di una denuncia con annesso stato di una denuncia e info sull'UffPolGiud che ha preso in carico la denunciaall'UffGiulPol di visualizzare i dettagli di una denuncia con annesso stato di denuncia	DenunciaService
Cambia stato denuncia	Questa funzionalità permette all'UffPolGiud di modificare lo stato di una	DenunciaService



	denuncia inoltrata da un Utente	
Visualizza storico denunce prese in carico	Questa funzionalità permette all'UffPolGiud di visualizzare un elenco di tutte le denunce precedentemente prese in carico	DenunciaService

Sottosistema Geolocalizzazione

Servizio	Descrizione	Interfaccia
Visualizza mappa con marker	Questa funzionalità permette di visualizzare una mappa geografica della propria zona contrassegnata da marker che rappresentano i centri ASL e stazioni di Polizia.	GeolocalizzazioneService
Visualizza informazioni su marker	Questa funzionalità permette di visualizzare le informazioni sui marker a tutti gli utilizzatori dell'app: utente, UffPolGiud e Operatore CUP.	GeolocalizzazioneService

Sottosistema Prenotazioni e Supporto Psicologico

Servizio	Descrizione	Interfaccia
Inoltra richiesta di prenotazione	Questa funzionalità permette di inoltrare una prenotazione per ricevere supporto psicologico	PSPService



Visualizza lista prenotazioni attive	Questa funzionalità permette di visualizzare le prenotazioni attive	PSPService
Visualizza storico prenotazioni	Questa funzionalità permette di visualizzare lo storico delle prenotazioni	PSPService
Visualizza dettagli prenotazione	Questa funzionalità permette di visualizzare i dettagli di una prenotazione	PSPService
Inizializza prenotazione	Questa funzionalità permette di inizializzare una prenotazione	PSPService

Sottosistema Informativa e Contatti

Servizio	Descrizione	Interfaccia
Visualizza Contatti FFOO e ASL	Questa funzionalità permette di Visualizzare i contatti delle ASL e delle FFOO.	InformativaContattiService
Visualizza Informativa sulle leggi riguardo discriminazioni	Questa funzionalità permette di visualizzare tutte le leggi attuali che riguardano qualsiasi tipo di discriminazione e abuso vigenti in Italia.	InformativaContattiervice



Laurea Triennale in Informatica - Università degli Studi di Salerno
Corso di *Ingegneria del Software* - Prof.ssa F. Ferrucci, Prof. F. Palomba

Sottosistema Chat-Bot

Servizio	Descrizione	Interfaccia
Avvia chat-bot	Questa funzionalità permette di avviare l'assistenza tramite chat-bot	<u>ChatBotService</u>



5. Glossario

Sigla/Termine	Definizione
Pubblica Amministrazione	La Pubblica Amministrazione è l'insieme degli enti pubblici che concorrono all'esercizio e alle funzioni di gestione, direzione e coordinazione di uno Stato nelle materie di sua competenza.
Forze dell'Ordine	Le forze dell'ordine sono dei corpi che sovrintendono allo svolgimento delle funzioni di polizia nella Repubblica Italiana.
Ufficiale di Polizia Giudiziaria	La posizione di Ufficiale di Polizia Giudiziaria è una posizione soggettiva di chi appartiene alle forze dell'ordine e svolge attività investigativa per conto di un magistrato.
Operatore CUP	Operatore che gestisce il flusso di clienti nelle sedi operative in grado di dare le prime informazioni sui servizi e prenotare gli appuntamenti anche telefonicamente.
ASL	Un'azienda sanitaria locale è un ente pubblico della pubblica amministrazione italiana, deputato all'erogazione di servizi sanitari in un determinato territorio, di solito provinciale.
SPID	Il Sistema Pubblico di Identità Digitale è un'identità digitale composta da una coppia di credenziali, strettamente personali, con le quali è possibile accedere ai servizi online della pubblica amministrazione e dei privati aderenti.
Componenti COTS	Con il termine componenti COTS ci si riferisce a componenti hardware e software disponibili sul mercato per l'acquisto da parte di aziende di sviluppo interessate a utilizzarli nei loro progetti (https://www.wikiwand.com/it/COTS).
Open Source	Con open source (in italiano sorgente aperto), in informatica, si indica un sistema di sviluppo software decentralizzato basato sulla condivisione dei file sorgenti (source). (https://www.wikiwand.com/it/Open_source)ser
Serverless	<p>Il serverless computing è un modello di sviluppo cloud native che consente agli sviluppatori di creare ed eseguire applicazioni senza gestire i server.</p> <p>Anche se in questo modello i server vengono utilizzati comunque, sono astratti dallo sviluppo delle app. Le attività di routine per il provisioning, la manutenzione e la scalabilità dell'infrastruttura server vengono gestite da un provider di servizi cloud. Gli sviluppatori devono semplicemente creare pacchetti di codice all'interno di container per il deployment.</p> <p>(https://www.redhat.com/it/topics/cloud-native-apps/what-is-serverless)</p>