# Documentazione Progetto Give & Take

Indice

1. **Analisi**
   1. Analisi preliminare della realtà
   2. Definizione di una soluzione del progetto
2. **Premesse del progetto**
   1. Obiettivi e scopo del prodotto
   2. Stakeholder
3. **Servizi del sistema**
   1. Scopo del sistema
   2. Requisiti funzionali
   3. Requisiti informativi
4. **Vincoli di sistema**
   1. Requisiti di interfaccia
   2. Requisiti di sicurezza
   3. Requisiti operativi

**1. Analisi**

**1.1 Analisi preliminare della realtà**

Nella realtà attuale vi sono moltissime associazioni di volontariato a cui una persona può prendere parte e mettere a disposizione il suo tempo per aiutare, tramite piccole iniziative, persone in difficoltà. Gestire tutte queste attività ed i relativi dati coinvolti risulta difficile, e necessita dunque il coinvolgimento di personale qualificato che sfrutti particolari metodi di archiviazione dell’enorme quantità di informazioni. Si reputa quindi necessaria una migliore memorizzazione e digitalizzazione dei dati.

**1.2 Definizione di una soluzione del progetto**

Una possibile soluzione è quella di sviluppare un’applicazione giovanile per avere sotto controllo tutte le eventuali attività di volontariato e permette ad ogni partecipante di gestire gli incarichi che ha accettato. Tramite il progetto “Give & Take” si permette agli utenti sia di offrire che ricevere aiuto per svolgere alcune mansioni quotidiane. Al fine di avere un servizio il più organizzato ed efficiente possibile, la soluzione migliore è quella di unire le conoscenze apprese nel corso degli anni.

**2. Premesse del progetto**

**2.1 Obiettivi e scopo del prodotto**

Gestire le attività di volontariato gratuite delle persone iscritte alla community.

**2.2 Stakeholder**

Gli attori coinvolti sono gli utenti registrati al servizio fornito dall’applicazione.

**3. Servizi del sistema**

**3.1 Scopo del sistema**

Il sistema dovrà occuparsi di gestire le attività di volontariato. In particolare: la registrazione dell’utente, l’accesso dell’utente iscritto, la creazione di una richiesta di volontariato, la ricerca di attività al quale aderire, la ricerca di un utente, la cancellazione della richiesta e dell’iscrizione ed, infine, la modifica dei dati della richiesta e dei dati personali.

**3.2 Requisiti funzionali**

3.2.1 Permette l’iscrizione alla community delle persone volontarie.

Il sistema dovrà mostrare all’utente una particolare view per la registrazione al servizio.

3.2.1.1 Il sistema dovrà mostrare inizialmente all’utente una finestra dalla quale è possibile inserire i dati anagrafici richiesti.

3.2.1.2 Il sistema dovrà mostrare, infine, il riepilogo dei dati inseriti e le scelte di privacy prese dall’utente e permettere tramite il pulsante “Registrati” di inviare la registrazione e tramite il pulsante “Annulla” di tornare indietro all’activity precedente.

3.2.2 Permette la modifica dei dati anagrafici inseriti dall’utente.

Il sistema dovrà mostrare all’utente una particolare view per la modifica dei dati dell’utente.

3.2.2.1 Il sistema dovrà mostrare all’utente una finestra dalla quale è possibile modificare i propri dati anagrafici.

3.2.2.2 Il sistema dovrà mostrare il riepilogo dei dati inseriti e le scelte di privacy prese dall’utente e permettere tramite il pulsante “Modifica” di inviare la modifica dei dati e tramite il pulsante “Annulla” di tornare indietro per annullare l’operazione di modifica.

3.2.3 Permette la ricerca di un utente o di una attività particolare.

Il sistema dovrà mostrare all’utente una particolare view per la ricerca di un utente o di una attività.

3.2.3.1 Il sistema dovrà mostrare all’utente una finestra dalla quale è possibile inserire l’username dell’utente o il nome dell’attività da ricercare. Vi saranno due bottoni, “Ricerca” per procedere alla ricerca di ciò che si vuole cercare, ed “Annulla” che permette di tornare alla schermata precedente.

3.2.3.2 In caso si prema il pulsante “Ricerca”, il sistema dovrà mostrare l’utente trovato o l’attività trovata. In caso non si fossero riscontrati utenti o attività, verrà visualizzato un messaggio per avvisare del risultato ottenuto.

3.2.3.3 In caso si prema il pulsante “Annulla”, il sistema dovrà mostrare la activity precedente all’operazione.

3.2.4 Permette la creazione di un’attività di volontariato/richiesta di aiuto.

Il sistema dovrà mostrare all’utente una particolare view per la creazione di una nuova attività, che sia di richiesta o di volontariato.

3.2.4.1 Il sistema dovrà mostrare inizialmente all’utente una finestra dalla quale è possibile inserire i dati richiesti per la creazione di un’attività di volontariato/richiesta di aiuto.

3.2.4.2 Il sistema dovrà mostrare quindi il riepilogo dei dati inseriti e permettere tramite il pulsante “Crea” di inviare i dati riguardanti la nuova attività/richiesta e tramite il pulsante “Annulla” di tornare indietro all’activity precedente.

3.2.5 Permette la modifica di un’attività di volontariato/richiesta di aiuto.

Il sistema dovrà mostrare all’utente una particolare view per la modifica di un’attività, che sia di richiesta o di volontariato.

3.2.5.1 Il sistema dovrà mostrare all’utente una finestra dalla quale è possibile modificare i dati riguardanti le singole attività di volontariato/richiesta.

3.2.5.2 Il sistema dovrà mostrare il riepilogo dei dati inseriti e permettere tramite il pulsante “Modifica” di inviare la modifica delle attività e tramite il pulsante “Annulla” di tornare alla precedente activity.

3.2.6 Permette la cancellazione di un’attività di volontariato/richiesta di aiuto.

Il sistema dovrà mostrare all’utente una particolare view per la cancellazione di un’attività, che sia di richiesta o di volontariato.

3.2.6.1 Il sistema dovrà mostrare all’utente una finestra dalla quale è possibile avere un resoconto dell’attività ed avere la possibilità di cancellare i dati inseriti per quella particolare richiesta tramite il bottone “Elimina”, oppure di annullare l’operazione e tornare alla activity precedente grazie al bottone “Annulla”.

3.2.7 Permette di gestire le attività alle quali un utente ha aderito.

Il sistema dovrà mostrare all’utente una particolare view per la gestione delle attività a cui ha aderito.

3.2.7.1 Il sistema dovrà mostrare all’utente una finestra dalla quale è possibile gestire le richieste a cui ha aderito l’utente.

3.2.7.2 Cliccando su una attività, il sistema dovrà mostrare il riepilogo di quella specifica attività. Per ogni attività, l’utente avrà la possibilità di modificarne i dati tramite il bottone “Modifica”, di eliminarla tramite il bottone “Elimina” o di tornare alla precedente activity tramite il bottone “Annulla”.

3.2.8 Permette la cancellazione dell’iscrizione.

Il sistema dovrà mostrare all’utente una particolare view per la cancellazione dell’iscrizione.

3.2.8.1 Il sistema dovrà mostrare all’utente una finestra dalla quale è possibile visualizzare il riepilogo del proprio profilo ed eliminarne l’iscrizione tramite il bottone “Elimina account”. Vi sarà inoltre il bottone “Annulla”, per tornare all’activity precedente.

**3.3 Requisiti informativi**

3.3.1 Tramite un DB è possibile gestire in modo ottimale le attività create e gli utenti tramite tabelle specifiche.

3.3.1.1 La tabella relativa agli UTENTI avrà i seguenti attributi: ID per identificare ogni utente, nome per sapere il nome di ciascun utente, cognome per sapere il cognome di ciascun utente, email per poter contattare un qualsiasi utente, password per avere un account sicuro per ciascun utente, nomeUtente per avere un nickname facile da cercare per ogni utente, telefono per avere un riferimento telefonico per ogni utente (sarà poi il singolo a scegliere se rendere visibile il proprio numero o meno).

3.3.1.2 La tabella relativa alle RICHIESTE avrà i seguenti attributi: ID per identificare ogni richiesta, utente per sapere l’utente correlato ad essa, ambito per sapere di cosa tratta generalmente la richiesta, durata per avere un limite di tempo di eseguimento della richiesta, oraReg per sapere a che ora è stata registrata la richiesta, completata per sapere se la richiesta è già stata completata o meno, incaricato per sapere che utente ha preso in incarico la richiesta, note per aggiungere una breve descrizione alla richiesta, titolo per avere una prima visualizzazione del tipo di richiesta, longitudine per avere le coordinate della longitudine del luogo in cui verrà effettuata la richiesta, latitudine per avere le coordinate della latitudine del luogo in cui verrà effettuata la richiesta.

3.3.1.3 La tabella relativa agli AMBITI avrà i seguenti attributi: IDAmbito per identificare ogni ambito, descrizione per avere una descrizione del relativo ambito.

3.3.1.1 La tabella relativa allo SPIDUTENTI avrà i seguenti attributi: IDU per identificare ogni utente, nome per sapere il nome di ciascun utente, cognome per sapere il cognome di ciascun utente, email per poter contattare un qualsiasi utente, password per avere un account sicuro per ciascun utente, username per avere un nickname facile da cercare per ogni utente, telefono per avere un riferimento telefonico per ogni utente.

**4. Vincoli di sistema**

**4.1 Requisiti di interfaccia**

4.1.1 Il sistema permette di tornare indietro in qualsiasi momento.

4.1.2 Il sistema permette di annullare in qualsiasi momento qualsiasi funzionalità.

**4.2 Requisiti di sicurezza**

4.2.1 L’utente, al primo utilizzo dell’applicativo, non ha il diritto di accedere ad alcune informazioni presenti nel db, di conseguenza non sarà sviluppata alcuna funzione che permetta di ottenere informazioni sensibili.

4.2.2 L’utente, in seguito all’iscrizione, ha solamente il diritto di effettuare una ricerca degli utenti iscritti e visualizzare solamente i dati personali che l’utente ricercato vuole mostrare.

4.2.3 La sicurezza viene garantita tramite l’utilizzo di tecniche di criptaggio delle password.

4.2.4 La sicurezza viene garantita tramite l’utilizzo delle funzioni messe a disposizione dallo SPID (fac-simile al momento dello sviluppo dell’applicazione).

**4.3 Requisiti operativi**

Il dispositivo sul quale l’applicazione è installata deve essere necessariamente collegato in rete: dovrà quindi essere garantita una connessione ad Internet.

Il sistema operativo richiesto per il corretto funzionamento dell’applicazione sul dispositivo sarà il sistema operativo Android. Si sceglie questo sistema operativo perché essendo diffuso può essere più facilmente utilizzato.

Il DBMS scelto è MySQL. Tale DBMS consente minori tempi di inattività per le applicazioni, maggiori livelli di scalabilità e prestazioni e controlli di protezione rigorosi e flessibili; offre inoltre una infrastruttura di gestione estensibile, facile da programmare mediante l'utilizzo di oggetti. Servirà un DB che memorizzi i dati necessari per far funzionare al meglio la fluidità e la memorizzazione dell’applicazione.

Il linguaggio di programmazione utilizzato per la realizzazione dell’applicativo è il linguaggio Java, SQL ed il meta-linguaggio XML per la gestione dei dati.

|  |
| --- |
| Caso d’Uso: Iscrizione Nuovo Utente |
| **ID**: UC\_IscrizioneUtente |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente accede per la prima volta all’applicazione 2. Il sistema mostra una activity dove inserire tutti i dati anagrafici relativi a se stesso 3. Se l’utente conferma, i dati vengono inseriti nell’archivio |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. L’iscrizione è inserita nell’archivio |

|  |
| --- |
| Caso d’Uso: Modifica Dati Anagrafici Utente |
| **ID**: UC\_ModificaDatiUtente |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente va alla activity per la visualizzazione del suo profilo 2. Il sistema mostra la possibilità di modificare i propri dati anagrafici 3. Se l’utente conferma, i nuovi dati vengono inseriti nell’archivio |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. La modifica dei dati è inserita nell’archivio |

|  |
| --- |
| Caso d’Uso: Ricerca Utente O Attività |
| **ID**: UC\_RicercaUtenteAttivita |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente va all’activity di ricerca utente/attività 2. Se l’utente conferma la ricerca, il sistema mostra il risultato ottenuto |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. La ricerca è stata eseguita |

|  |
| --- |
| Caso d’Uso: Creazione Attività Volontariato O Richiesta |
| **ID**: UC\_CreaAttivitaORichiesta |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente va all’activity di creazione di una nuova attività o richiesta 2. Il sistema mostra la possibilità di creare una nuova attività di volontariato o di richiesta, inserendo i relativi dati 3. Se l’utente conferma, i dati vengono inseriti nell’archivio |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. I dati relativi alla creazione dell’attività sono inseriti nell’archivio |

|  |
| --- |
| Caso d’Uso: Modifica Attività Di Volontariato o Richiesta |
| **ID**: UC\_ModificaAttivita |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente va all’activity relativa alla modifica di una attività 2. Il sistema offre la possibilità di modificare i dati relativi a quella determinata attività 3. Se l’utente conferma, i nuovi dati vengono inseriti nell’archivio |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. I dati relativi alla modifica di un’attività sono inseriti nell’archivio |

|  |
| --- |
| Caso d’Uso: Cancellazione Attività Di Volontariato o Richiesta |
| **ID**: UC\_CancellaAttivita |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente va all’activity per la cancellazione di un’attività 2. Il sistema offre la possibilità di visualizzare un resoconto dell’attività prima di fare la scelta 3. Se l’utente conferma la cancellazione, i dati vengono inseriti nell’archivio |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. I dati relativi alla cancellazione dell’attività sono inseriti nell’archivio |

|  |
| --- |
| Caso d’Uso: Gestione Attività Di Volontariato o Richiesta |
| **ID**: UC\_GestioneAttivita |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente va all’activity di gestione delle attività 2. Il sistema offre la possibilità di gestire le varie attività tramite l’interfaccia Android |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. La gestione delle attività è stata eseguita |

|  |
| --- |
| Caso d’Uso: Cancellazione Iscrizione |
| **ID**: UC\_CancellazioneIscrizione |
| **Attori:**  Utente |
| **Precondizioni:**   1. Il sistema è stato avviato |
| **Sequenza eventi:**   1. Il caso d'uso inizia quando l’utente visualizza l’activity riguardante le sue informazioni personali 2. Il sistema offre la possibilità di cancellare l’account dell’utente 3. Se l’utente conferma, i dati vengono eliminati dall’archivio |
| **Sequenze alternative:**   1. L’utente può abbandonare la funzionalità in qualsiasi momento |
| **Postcondizioni:**   1. I dati dell’utente sono eliminati dall’archivio |