**F2F – Families-To-Families Sharing Care of Children  
MIUR FISR 2020**

**Progetto di Ingegneria del Software 2021/22  
Università Ca’ Foscari Venezia**

**Documento dei requisiti**

versione 1.0  
 **Black Buffalos**

25 ottobre 2021

*Informazioni sul Documento*

| MUR FISR 2020 |  | Acronimo | | | F2F | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Coordinatore Progetto | Prof. Agostino Cortesi | | | | | | | |
| Deliverable | Documento dei requisiti | | | | | | | |
| Data di Consegna | 25/10/2021 | |  | | |  | |  |
| Team Leader | Giovanni Costa | | |  | | | 880892@stud.unive.it | |
| Team members | Massimo Cailotto 880873@stud.unive.it  Filippo Di Gennaro 882795@stud.unive.it Matteo Minardi 880895@stud.unive.it  Andrea Munarin 879607@stud.unive.it | | | | | | | |

*Document History*

| Version | Issue Date | Stage | Changes | Contributors |
| --- | --- | --- | --- | --- |
| 1.0 | 25/10/2021 | Draft | First version | All members of the team |
| 1.1 |  | Draft |  |  |
| 2.0 |  | Final |  |  |

**Indice**

[***1.***](#bookmark=id.liwzxjwaoeaf) ***Introduzione*** **5**

[1.1 Scopo del documento](#bookmark=id.7t4o4375som6) 5

[1.2 Descrizione del documento](#bookmark=id.t3uwrvsyrs35) 5

[1.3 Descrizione delle funzionalità del sistema](#bookmark=id.5is6lvft3axp) 6

[***2.***](#bookmark=id.qh4lqwl98ra5) ***Glossario*** **6**

[***3.***](#bookmark=id.a3znp9vtiex5) ***Modelli del sistema* 8**

[3.1 Modelli dei casi d’uso](#bookmark=id.jg9xul4ejaxd) 8

3.1.1 Creazione utente 10

3.1.2 Creazione bambini 10

3.1.3 Visione informazioni utenti, attività, gruppi 11

3.1.4 Creazione e gestione gruppo 12

3.1.5 Creazione attività 13

3.1.6 Contact Tracing 14

3.1.7 Gestione attività 14

3.1.8 Rimozione attività 15

3.1.9 Annullamento attività 16

3.1.10 Invito, aggiunta o rimozione partecipanti 16

3.1.11 Richiesta green pass 17

3.1.12 Gestione bacheca 18

3.1.13 Gestione mercatino digitale 18

[3.2 Modello di collaborazione](#bookmark=id.wzb2ig5qh4n) 20

[***4.***](#bookmark=id.k2fwhwd5frkw) ***Classi di requisiti*** **21**

[4.1 Requisiti durevoli e volatili](#bookmark=id.13bp823kq3du) 21

[4.2 Requisiti funzionali e non funzionali](#bookmark=id.kvnbtzwu38cf) 21

[***5.***](#bookmark=id.c36t29cr8xlf) ***Definizione dei requisiti funzionali*** **21**

[5.1 Requisiti](#bookmark=id.gpp3fuo7irpt) 22

[***6.***](#bookmark=id.ij0r7tuq013q) ***Definizione dei requisiti non funzionali*** **30**

[6.1 Requisiti di prodotto](#bookmark=id.j17bl5x3bryf) 31

6.1.1 Sicurezza 31

6.1.2 Efficienza 33

6.1.3 Adattabilità 33

6.1.4 Usabilità 34

[6.2 Requisiti di processo](#bookmark=id.cuaj8x4xp12j) 35

[6.3 Requisiti esterni](#bookmark=id.y5gbosoywgma) 36

[6.4 Misure dei requisiti](#bookmark=id.8p6nw9co1o9f) 37

[**7. Evoluzione del sistema**](#bookmark=id.5punqtel7sg6) **40**

[**8. Specifica dei requisiti**](#bookmark=id.2frxmt8yoxs7) **41**

[8.1 Specifiche](#bookmark=id.74cxt6nnttoj) 42

[8.2 Tabelle di tracciabilità dei requisiti](#bookmark=id.sdcplqk2xpbs) 50

[**9. Appendici**](#bookmark=id.v70vdnox5y1j) **52**

[9.1 Requisiti dell’app](#bookmark=id.6zs8zxq75u7x) 52

[9.2 Requisiti del database](#bookmark=id.w0rfwnzgyovg) 52

[9.3 Requisiti del server](#bookmark=id.k1blj3w6f4n7) 52

[9.4 Requisiti dei dispositivi](#bookmark=id.l4tw0iwpygfc) 52

[**10. Riferimenti**](#bookmark=id.2onjfg249odv) **53**

1. **Introduzione**

* 1. **Scopo del documento**

Lo scopo di questo documento è fornire tutte le informazioni riguardanti lo sviluppo delle funzionalità aggiuntive per il progetto “Families Share”. Verranno illustrati nel dettaglio tutti i servizi che il team intende fornire, in modo conforma a quanto dichiarato e specificato nel Piano di Progetto

Inoltre, sarà presente una descrizione ed una valutazione dei vari requisiti funzionali e non funzionali necessari alla realizzazione dell’applicazione. Dopo l’analisi e la definizione di questi requisiti, verrà fornita anche la loro specifica dettagliata.

* 1. **Descrizione del documento**

Il seguente documento è composto da 10 sezioni, inclusa l’Introduzione:

**Glossario**: Contiene la definizione di tutti i termini tecnici utilizzati nel documento. Utile per la sua comprensione anche per utenti meno esperti.

**Modelli del sistema**: Modellazione della struttura dell’applicazione, fornendone una rappresentazione in UML.

**Classi dei requisiti:** Contiene la descrizione di tutti i requisiti necessari per il funzionamento e il mantenimento dell’applicazione.

**Definizione dei requisiti funzionali**: Definizione di tutti i requisiti necessari alla realizzazione dei servizi forniti dall’applicazione.

**Definizioni dei requisiti non funzionali**: Contiene tutti i vincoli che l’applicazione finale deve rispettare.

**Evoluzione del sistema**: Descrizione di tutte le possibilità per evolvere ed espandere l’applicazione finale.

**Specifica dei requisiti**: Analisi nel dettaglio di tutti i requisiti presentati nei capitoli precedenti.

**Appendice**: Contiene vari requisiti per il progetto, nello specifico i requisiti dell’app, del database, del server e dei dispositivi

**Riferimenti**: Riferimenti utilizzati per la realizzazione di questo documento

* 1. **Descrizione delle funzionalità del sistema**

L’obiettivo del team, come descritto nel Piano di progetto, è implementare funzionalità aggiuntive per un progetto già esistente, Families Share.   
In particolare, si intende realizzare nuovi servizi basati sul mutuo-aiuto e la solidarietà tra famiglie, con funzioni ad hoc per problematiche derivanti dalla pandemia di Covid-19.

I servizi principali nello specifico sono:

* Tracing dei contagi: fondamentale per informare gli utenti, monitorando la partecipazione alle attività, che un partecipante è risultato positivo a Covid-19 per arginare la diffusione della pandemia
* Bacheca per la condivisione: condividere proprie esperienze, pensieri o materiale multimediale vario ed utile al gruppo
* Mercatino digitale: possibilità di creare un mercato di scambi e prestiti, che possono aiutare gli utenti in diversi motivi di necessità, anche economica

1. **Glossario**

Qui sono elencati i vari termini tecnici e parole chiave che si utilizzano all’interno di questa documentazione di progetto, correlati da una breve descrizione:

* **Front-end** : sezione dell’applicazione con cui l’utente interagisce direttamente. Si occupa della gestione dei flussi di dati in input da parte degli utilizzatori e della visualizzazione dei risultati delle operazioni effettuate.
* **Back-end** : questa parte salva ed elabora gli input di dati che riceve e fondamentalmente svolge tutte quelle operazioni che sono necessarie per un corretto funzionamento dell’applicazione, al termine delle quali invia i risultati al front-end in modo che possa mostrare ciò che è stato richiesto
* **Android** : Android è un sistema operativo mobile basato su una versione modificata del [kernel Linux](https://it.wikipedia.org/wiki/Linux_(kernel)) e altri software open source, progettato principalmente per dispositivi mobili touchscreen come smartphone e tablet.

È un software gratuito e open source ed il suo codice sorgente è noto come Android Open Source Project (AOSP)

Fonte: [Wikipedia](https://it.wikipedia.org/wiki/Android)

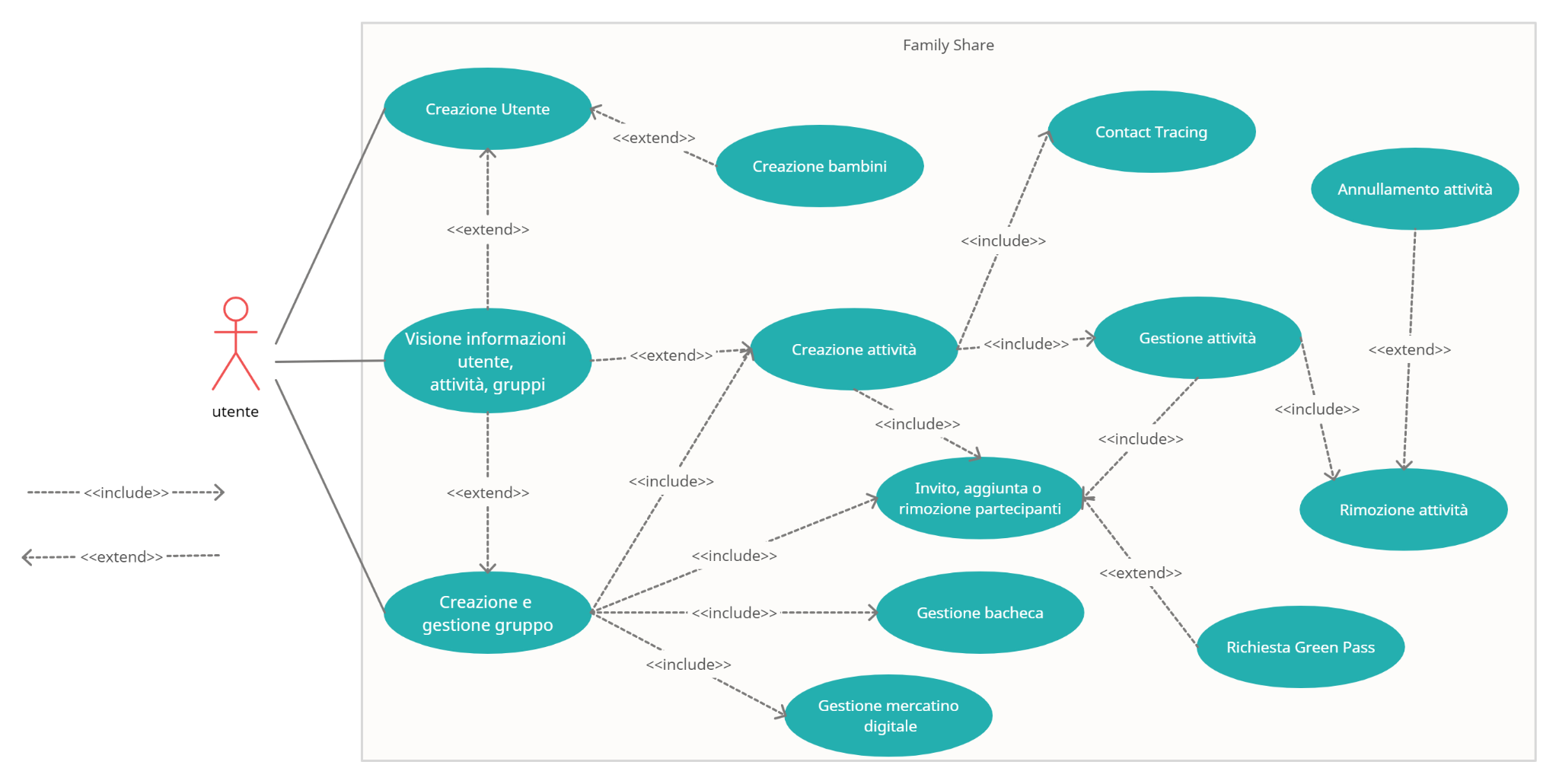
* **JavaScript** : JavaScript è un [linguaggio di programmazione](https://it.wikipedia.org/wiki/Linguaggio_di_programmazione) [orientato agli oggetti](https://it.wikipedia.org/wiki/Programmazione_orientata_agli_oggetti) e agli [eventi](https://it.wikipedia.org/wiki/Programmazione_a_eventi), comunemente utilizzato nella [programmazione Web](https://it.wikipedia.org/wiki/Programmazione_Web) [lato client](https://it.wikipedia.org/wiki/Lato_client) (esteso poi anche al [lato server](https://it.wikipedia.org/wiki/Lato_server)) per la creazione, in [siti web](https://it.wikipedia.org/wiki/Sito_web) e [applicazioni web](https://it.wikipedia.org/wiki/Applicazione_web), di effetti dinamici [interattivi](https://it.wikipedia.org/wiki/Interattivit%C3%A0) tramite [funzioni](https://it.wikipedia.org/wiki/Funzione_(informatica)) di [script](https://it.wikipedia.org/wiki/Script) invocate da eventi innescati a loro volta in vari modi dall'utente sulla [pagina web](https://it.wikipedia.org/wiki/Pagina_web) in uso ([mouse](https://it.wikipedia.org/wiki/Mouse), [tastiera](https://it.wikipedia.org/wiki/Tastiera_(informatica)), caricamento della pagina ecc...).  
  Fonte: [Wikipedia](https://it.wikipedia.org/wiki/JavaScript)
* **Node.Js** : Node.js è un [runtime system](https://it.wikipedia.org/wiki/Runtime_system) [open source](https://it.wikipedia.org/wiki/Open_source) multipiattaforma [orientato agli eventi](https://it.wikipedia.org/wiki/Programmazione_a_eventi) per l'esecuzione di codice [JavaScript](https://it.wikipedia.org/wiki/JavaScript), costruita sul [motore JavaScript V8](https://it.wikipedia.org/wiki/V8_(motore_JavaScript)) di [Google Chrome](https://it.wikipedia.org/wiki/Google_Chrome).  
  In origine JavaScript veniva utilizzato principalmente [lato client](https://it.wikipedia.org/wiki/Lato_client). In questo scenario gli script JavaScript, generalmente incorporati all'interno dell'HTML di una pagina web, vengono interpretati da un motore di esecuzione incorporato direttamente all'interno di un [Browser](https://it.wikipedia.org/wiki/Browser). Node.js consente invece di utilizzare JavaScript anche per scrivere codice da eseguire [lato server](https://it.wikipedia.org/wiki/Lato_server), ad esempio per la produzione del contenuto delle pagine web dinamiche prima che la pagina venga inviata al [Browser](https://it.wikipedia.org/wiki/Browser) dell'utente. Node.js in questo modo permette di implementare il cosiddetto paradigma "JavaScript everywhere" (JavaScript ovunque), unificando lo sviluppo di applicazioni Web intorno ad un unico linguaggio di programmazione (JavaScript).  
  Fonte: [Wikipedia](https://it.wikipedia.org/wiki/Node.js)
* **Client-Server** : In [informatica](https://it.wikipedia.org/wiki/Informatica) il termine sistema client-server (letteralmente cliente-serviente) indica un'[architettura di rete](https://it.wikipedia.org/wiki/Architettura_di_rete) nella quale genericamente un [computer](https://it.wikipedia.org/wiki/Computer) [client](https://it.wikipedia.org/wiki/Client) o [terminale](https://it.wikipedia.org/wiki/Terminale_(informatica)) si [connette](https://it.wikipedia.org/wiki/Connessione_(informatica)) ad un [server](https://it.wikipedia.org/wiki/Server) per la fruizione di un certo servizio, quale ad esempio la condivisione di una certa [risorsa](https://it.wikipedia.org/wiki/Risorsa_informatica) [hardware](https://it.wikipedia.org/wiki/Hardware)/[software](https://it.wikipedia.org/wiki/Software) con altri client, appoggiandosi alla sottostante architettura [protocollare](https://it.wikipedia.org/wiki/Protocollo_di_rete).  
  Più semplicemente, i sistemi client/server sono un'evoluzione dei sistemi basati sulla condivisione semplice delle [risorse](https://it.wikipedia.org/wiki/Risorsa_informatica): la presenza di un [server](https://it.wikipedia.org/wiki/Server) permette ad un certo numero di client di condividerne le risorse, lasciando che sia il server a gestire gli accessi alle risorse per evitare conflitti di utilizzazione tipici dei primi [sistemi informatici](https://it.wikipedia.org/wiki/Sistema_informatico)  
  Fonte: [Wikipedia](https://it.wikipedia.org/wiki/Sistema_client/server)
* **Android Studio** : Android Studio è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android.  
  Fonte: [Wikipedia](https://it.wikipedia.org/wiki/Android_Studio)
* **MongoDB** : MongoDB è un [DBMS](https://it.wikipedia.org/wiki/Database_management_system) non relazionale, [orientato ai documenti](https://it.wikipedia.org/wiki/Base_di_dati_orientata_al_documento). Classificato come un database di tipo [NoSQL](https://it.wikipedia.org/wiki/NoSQL), MongoDB si allontana dalla struttura tradizionale basata su tabelle dei [database relazionali](https://it.wikipedia.org/wiki/Database_relazionali) in favore di documenti in stile [JSON](https://it.wikipedia.org/wiki/JSON) con schema dinamico (MongoDB chiama il formato [BSON](https://it.wikipedia.org/wiki/BSON)), rendendo l'integrazione di dati di alcuni tipi di applicazioni più facile e veloce.  
  Fonte: [Wikipedia](https://it.wikipedia.org/wiki/MongoDB)
* **MQL** : è il linguaggio di query (interrogazione) del database MongoDB. É molto ricco ed inoltre supporta tutte le operazioni CRUD (Create, Read, Update, Delete) sui dati
* **GitHub** : GitHub è un servizio di hosting per progetti software,, ma offre molte più funzionalità. GitHub fornisce un’interfaccia grafica basata sul web, il controllo dell'accesso e diverse funzionalità di collaborazione, come strumenti di base per la gestione delle attività per ogni progetto
* **Repository** : può essere considerato un archivio o un deposito di informazioni, accessibili più o meno liberamente e spesso raggiungibili tramite Internet
* **Entry** : tupla del database

1. **Modelli del sistema**

* 1. **Modelli dei casi d’uso**

In questa sezione descriveremo nel dettaglio i modelli del sistema (Use-Case), evidenziando i componenti del sistema e le relazioni tra esse e con l’ambiente. Per caso d’uso (use-case) si intende un possibile metodo di utilizzo del sistema.   
Ogni caso d’uso descrive l’interazione tra attori e sistema: cattura chi (attori) fa cosa (interazione) con il sistema. Con quale scopo (goal), senza considerare nel dettaglio ciò che è dentro il sistema

Qui di seguito si rappresenta l’UML che fornisce una visione astratta delle relazioni tra utente (attore) e i casi d’uso. Successivamente, ogni Use-Case verrà descritto con maggiore dettaglio.



Il team si occupa di estendere le funzionalità già presenti all’interno dell’applicazione Family Share. Per questo sono stati considerati solo i casi d’uso ritenuti più importanti (essendo un’app complessa e già sviluppata). In particolar modo ci concentreremo nell’analizzare i casi ritenuti fondamentali per il funzionamento dell’applicazione e quelli da noi realizzati.   
Infine, è importante sottolineare come per tutti gli use cases sia necessaria una connessione alla rete sia (precondizione), dato che l’applicazione è basata fortemente sulla comunicazione client-server (il server comunica a sua volta con il database).

Qui di seguito è rappresentata la struttura delle schede utilizzate:

| Titolo | Descrizione |
| --- | --- |
| **Nome caso d’uso** | Ogni caso d’uso deve avere un nome; il nome esprime lo scopo dell’utente nell’utilizzo del sistema. |
| **Scopo** | Descrizione della funzionalità fornita dal sistema e che soddisfa una necessità dell’utente, ossia che è percepita dallo stesso utente come “valore”. |
| **Attori** | Persona, dispositivo o altra entità esterna al sistema che interagisce con il sistema. Per ogni caso d’uso esiste sempre un attore primario che è colui che inizia il caso d’uso stesso. |
| **Pre-condizioni** | Condizioni che devono essere soddisfatte all’inizio del caso d’uso. Rappresentano le “garanzie minime” che devono essere soddisfatte per poter attivare lo scenario di utilizzo del sistema. |
| **Trigger** | Evento trigger che attiva il caso d’uso. |
| **Descrizione** | Descrizione della sequenza di interazioni più comune tra gli attori e il sistema. In particolare viene descritta la sequenza principale che porta alla conclusione del caso d’uso con successo. La descrizione è definita in termini di input forniti dall’attore e di risposta del sistema. Il sistema è trattato secondo un modello di tipo black-box, concentrandosi su cosa esso fa in risposta agli input, e non su come internamente queste risposte vengano prodotte. |
| **Alternative** | Descrizioni delle variazioni della sequenza di passi tipica del main success scenario. Tali alternative estendono lo scenario principale. La gestione delle eccezioni è un esempio tipico di tali estensioni. Non tutte le alternative portano necessariamente ad un fallimento del caso d’uso. |
| **Post-condizioni** | Condizioni sempre soddisfatte al termine del caso d’uso. |
| **Questioni aperte** | Eventuali questioni lasciate in sospeso al termine del caso d’uso |

* + 1. **Creazione utente**

| **Nome caso d’uso** | Creazione utente |
| --- | --- |
| **Scopo** | L’utente crea il proprio profilo, registrando tutti i dati personali in modo da permettere più facilmente l’organizzazione delle attività. |
| **Attori** | Utente. |
| **Pre-condizioni** | L’app deve essere stata installata sul dispositivo. Inoltre deve essere presente una connessione internet. |
| **Trigger** | Al primo avvio dell’applicazione sarà necessario registrarsi creando il proprio profilo. |
| **Descrizione** | 1. L’applicazione permette l’inserimento di tutti i dati personali.    1. numero di telefono    2. indirizzo    3. contatto personale(mail)    4. una breve descrizione 2. L’utente tramite l’API del server Node viene memorizzato nel database. 3. L’app permette poi di eseguire tutte le funzioni principali. |
| **Alternative** | 1. Se l’utente è già registrato è possibile effettuare il login e poi accedere a tutte le funzioni principali |
| **Post-condizioni** | Viene inserita una nuova entry nel database attraverso l’interfaccia del server Node. |
| **Questioni aperte** | - |

* + 1. **Creazione bambini**

| **Nome caso d’uso** | Creazione bambini |
| --- | --- |
| **Scopo** | Permette l’inserimento dei propri bambini all’interno del profilo utente. Il bambino potrà partecipare nelle varie attività. |
| **Attori** | Utente |
| **Pre-condizioni** | L’app deve essere stata installata sul dispositivo e l’utente deve essere creato. |
| **Trigger** | L’utente al momento della creazione del profilo (o successivamente) potrà scegliere se inserire un bambino. |
| **Descrizione** | 1. Inserimento delle informazioni riguardanti uno o più bambini all’interno del proprio profilo utente:    1. sesso    2. compleanno    3. genitore    4. colore che lo rappresenta |
| **Alternative** | 1. E’ possibile modificare le informazioni di un bambino già inserito. 2. E’ possibile eliminare le informazioni di un bambino già inserito. 3. Essendo una scelta arbitraria è inoltre possibile non inserire alcun bambino e terminare la registrazione dell’utente. |
| **Post-condizioni** | Inserimento all’interno del database delle informazioni del bambino correlate alle entry dell’utente che sta operando e di un eventuale ulteriore genitore. |
| **Questioni aperte** | - |

* + 1. **Visione informazioni utente, attività, gruppi**

| **Nome caso d’uso** | Visione informazioni utente, attività, gruppi |
| --- | --- |
| **Scopo** | Permette di visualizzare:   * Utente: tutte le info personali. * Attività: tutti i partecipanti, gli orari, i luoghi e le ulteriori info presenti in attività (vedi 3.1.6). * Gruppi: tutti i partecipanti e le varie info (vedi 3.1.5). |
| **Attori** | Utente |
| **Pre-condizioni** | E’ necessario che l’utente sia registrato. Inoltre per visualizzare le informazioni riguardanti:   * Attività: Necessario che l’utente partecipi ad almeno un gruppo e ad almeno un attività. * Gruppi: Necessario che l’utente partecipi ad almeno un gruppo. |
| **Trigger** | L’utente sceglie se accedere o no a tale funzionalità. |
| **Descrizione** | 1. Permette di visualizzare tutte le informazioni inserite durante la registrazione dell’utente (compresi eventuali bambini e/o ulteriori genitori). 2. Permette di elencare tutti i membri e le attività presenti in un determinato gruppo. 3. Permette di osservare i partecipanti (suddivisi tra genitori che si rendono disponibili e bambini), l’orario, il luogo ed eventuali ulteriori descrizioni relative ad un’attività. |
| **Alternative** | 1. La scelta di utilizzare questa particolare funzione è a discrezione dell’utente, è quindi possibile che non venga utilizzata |
| **Post-condizioni** | Fornisce informazioni all’utente. Queste informazioni riassumono le diverse funzionalità dell’applicazione permettendo all’utente di visualizzare rapidamente e intuitivamente tutti i dati di cui necessità. |
| **Questioni aperte** | - |

* + 1. **Creazione e gestione gruppo**

| **Nome caso d’uso** | Creazione e gestione gruppo |
| --- | --- |
| **Scopo** | Permette di creare o modificare un gruppo all’interno dell’applicazione. |
| **Attori** | Utente, in particolar modo l’utente creatore del gruppo. |
| **Pre-condizioni** | E’ necessario che l’utente sia registrato nell’applicazione. |
| **Trigger** | La scelta di accedere a tale funzionalità dipende dall’utente. |
| **Descrizione** | 1. Permette la creazione di un nuovo gruppo inserendo il nome e descrizione del gruppo, eventuali limitazioni all’accesso, l’area di competenza del gruppo e le informazioni di contatto. 2. Nella gestione del gruppo è possibile modificare le info principali inserite durante la fase di creazione. |
| **Alternative** | 1. Essendo una funzionalità principale l’utente può non creare nessun gruppo, ma può accedere ad uno esistente cercandolo oppure accettando un invito da parte di un altro utente. 2. L’utente, altrimenti, potrebbe visualizzare le proprie informazioni. |
| **Post-condizioni** | Inserimento nel database tramite le API del server Node di un entry per il gruppo memorizzandone il creatore e i partecipanti. |
| **Questioni aperte** | gestione dei partecipanti (vedi 3.1.11) |

* + 1. **Creazione attività**

| **Nome caso d’uso** | Creazione attività |
| --- | --- |
| **Scopo** | Permette di creare una nuova attività. |
| **Attori** | Utente, in particolar modo l’utente creatore dell’attività. |
| **Pre-condizioni** | L’utente deve essere all’interno di un gruppo per poter creare un’attività. L’amministratore del gruppo deve permettere la creazione dell’attività |
| **Trigger** | E’ una scelta dell’utente creare o no un’attività. |
| **Descrizione** | 1. Nella creazione di un’attività va specificato il nome dell’attività, è inoltre possibile inserire una descrizione, una posizione, un link, un colore o un eventuale costo. Vanno poi inserite le date di inizio e fine attività con i vari orari 2. Si possono inoltre porre dei limiti nel numero di bambini e genitori partecipanti 3. E’ necessario inserire orario, luogo e giorno dell’attività |
| **Alternative** | 1. L’utente invece di creare un'attività può decidere di partecipare ad una già esistente (vedi 3.1.11) 2. E’ possibile utilizzare altre funzioni presenti all’interno di un gruppo |
| **Post-condizioni** | Vengono creati tutti i relativi dati all’interno del database MongoDb utilizzato per la memorizzazione dei dati dell’applicazione |
| **Questioni aperte** | Gestione dei partecipanti (vedi 3.1.11) |

* + 1. **Contact Tracing**

| **Nome caso d’uso** | Contact Tracing |
| --- | --- |
| **Scopo** | Tenere traccia di tutti i contatti avvenuti tra le persone che hanno partecipato alle stesse attività in modo da poter fornire una lista di persone potenzialmente infette da Covid-19. |
| **Attori** | Utenti che hanno partecipato ad almeno un’attività. |
| **Pre-condizioni** | Necessario che tutti gli utenti siano registrati ed abbiamo partecipato ad almeno un’attività. |
| **Trigger** | - |
| **Descrizione** | 1. Questa funzionalità offre la possibilità di tracciare tutti gli spostamenti e gli incontri tra le persone che utilizzano l’applicazione. 2. Nel caso venga identificato un utente positivo al virus è possibile segnalarlo all’interno dell’applicazione. In questo modo tutti gli utenti che sono entrati in contatto con lui riceveranno una notifica dall’app in cui viene consigliato di effettuare un tampone. 3. L’amministratore del gruppo avrà inoltre accesso alla lista di tutti i potenziali contagiati. |
| **Alternative** | - |
| **Post-condizioni** | Tutti gli utenti vengono informati del rischio di essere stati contagiati, in modo da agire preventivamente e arginare la diffusione del virus. |
| **Questioni aperte** | - |

* + 1. **Gestione attività**

| **Nome caso d’uso** | Gestione attività |
| --- | --- |
| **Scopo** | Poter accedere e modificare tutte le opzioni registrare durante l’evento “creazione attività” vedi (3.1.6) |
| **Attori** | Utente creatore dell’attività |
| **Pre-condizioni** | L’utente deve essere il creatore dell’attività. |
| **Trigger** | Decisione dell’utente modificare oppure no le informazioni dell’attività |
| **Descrizione** | 1. Si tratta di modificare tutti gli elementi descrittivi di un’attività come:    1. data    2. nome    3. colore    4. descrizione e posizione    5. inserire o modificare nuovi orari 2. Consente la gestione dei partecipanti ad un’attività: sia bambini che genitori. 3. E’ inoltre possibile esportare i dati dell’attività in un foglio Excel o in formato PDF. |
| **Alternative** | 1. E’ possibile non gestire un’attività già presente e creata 2. E’ possibile accedere o creare un altra attività 3. E’ possibile visualizzare la bacheca o il mercatino digitale |
| **Post-condizioni** | Vengono aggiornate tutte le informazioni nel database, e di conseguenza nei dispositivi dei partecipanti all’attività. |
| **Questioni aperte** | Gestione partecipanti all’attività (vedi 3.1.11) |

* + 1. **Rimozione attività**

| **Nome caso d’uso** | Rimozione attività |
| --- | --- |
| **Scopo** | Permette di rimuovere un’attività a cui si partecipa dalla propria lista |
| **Attori** | Utente |
| **Pre-condizioni** | L’utente deve partecipare ad almeno un’attività |
| **Trigger** | L’utente decide se eliminare o no un attività oppure l’attività è stata segnalata come terminata |
| **Descrizione** | 1. L’utente decide di eliminare l’attività dalla propria lista cancellando la sua partecipazione all’evento. 2. Una volta che un’attività è terminata essa viene rimossa dalla lista delle attività dei partecipanti. |
| **Alternative** | 1. E’ possibile che l’utente non voglia rimuovere nessuna attività. 2. E’ possibile accedere o creare o gestire un’altra attività. 3. E’ possibile visualizzare la bacheca o il mercatino digitale. |
| **Post-condizioni** | L’attività non sarà più presente all’interno dell’applicazione dell'utente a cui è stata rimossa, quindi anche nel database sarà stata effettuata la correzione delle relazioni. |
| **Questioni aperte** | - |

* + 1. **Annullamento attività**

| **Nome caso d’uso** | Annullamento attività |
| --- | --- |
| **Scopo** | Permette di annullare un’attività creata, cancellando tutti i dati sia dal proprio dispositivo che da tutti i partecipanti (poiché tutti connessi al server Node). |
| **Attori** | Utente creatore dell’attività |
| **Pre-condizioni** | L’utente deve essere il creatore dell’attività che vuole annullare. |
| **Trigger** | A causa di un evento esterno l'utente decide di annullare un’attività da lui creata |
| **Descrizione** | 1. Permette la rimozione di un’attività come creatore, annullando l’iscrizione di tutti i partecipanti |
| **Alternative** | 1. E’ possibile che l’utente non voglia annullare nessuna attività 2. E’ possibile accedere o creare o gestire un’altra attività 3. E’ possibile visualizzare la bacheca o il mercatino digitale |
| **Post-condizioni** | Tutti i dati dell’attività vengono correttamente rimossi dal database. |
| **Questioni aperte** | - |

* + 1. **Invito, aggiunta o rimozione partecipanti**

| **Nome caso d’uso** | Invito, aggiunta o rimozione partecipanti |
| --- | --- |
| **Scopo** | Permette la gestione dei partecipanti. Generalizza la gestione nei casi specifici dei gruppi, delle attività. |
| **Attori** | Utente creatore (gruppo/attività) |
| **Pre-condizioni** | L’utente deve aver creato almeno un gruppo o un’attività. |
| **Trigger** | La creazione di un gruppo/attività consente l’invito o aggiunta di partecipanti. E’ possibile comunque farlo in seguito. |
| **Descrizione** | 1. E’ possibile inviare una richiesta di partecipazione ad un qualunque utente di cui si conosca lo username (se l’utente è visibile) 2. Ogni utente ha la possibilità di aggiungersi ad un gruppo/attività se il suo creatore ne permette l’accesso tramite le impostazioni. 3. L’utente creatore può rimuovere dei partecipanti se lo desidera essendo l’amministratore del gruppo/attività. |
| **Alternative** | 1. E’ possibile non inserire nessun partecipante (facendo così, però, l’applicazione perde il suo scopo). 2. E’ possibile comunque accedere a tutte le funzionalità all’interno di un gruppo (attività, bacheca e mercatino digitale). |
| **Post-condizioni** | Modifica le relazioni tra gruppi/attività e i vari utenti partecipanti all’interno del database. |
| **Questioni aperte** | - |

* + 1. **Richiesta green pass**

| **Nome caso d’uso** | Richiesta green pass. |
| --- | --- |
| **Scopo** | Permettere agli utenti che creano un’attività di richiedere che tutti i partecipanti abbiamo il green pass. |
| **Attori** | Utente creatore dell’attività |
| **Pre-condizioni** | L’utente deve essere il creatore dell’attività. |
| **Trigger** | Durante la creazione o la gestione dell’attività l’utente abilita l’opzione della richiesta del green pass. |
| **Descrizione** | 1. Verifica che, una volta inserita la richiesta di green pass, tutti i partecipanti abbiano la certificazione. Questa verifica può essere fatta sia tramite l’inserimento di un documento pdf che lo verifichi (tampone negativo), sia in automatico se nel profilo del partecipante è presente l’informazione relativa al green pass (ottenuto dopo 1 o 2 dosi di vaccino). |
| **Alternative** | 1. l’utente può abilitare oppure no quest’impostazione. 2. Nel caso un partecipante non abbia la certificazione verde verrà considerato come non idoneo a partecipare all’attività. |
| **Post-condizioni** | Annullamento dell’inserimento di tutti i partecipanti “non idonei" dalle attività che richiedono il green pass. |
| **Questioni aperte** | - |

* + 1. **Gestione bacheca**

| **Nome caso d’uso** | Gestione bacheca |
| --- | --- |
| **Scopo** | Condividere contenuti multimediali all’interno del gruppo. |
| **Attori** | Utente |
| **Pre-condizioni** | L’utente deve partecipare ad almeno un gruppo. |
| **Trigger** | L’utente una volta che entra all’interno del gruppo può selezionare la visualizzazione della bacheca. |
| **Descrizione** | 1. Permette di inserire, condividere e visualizzare poesie, appunti, testi, foto e video |
| **Alternative** | 1. Possibilità di eliminare elementi condivisi se amministratore o utente che ha postato il contenuto. |
| **Post-condizioni** | Ogni elemento viene memorizzato all’interno del database in modo da essere visualizzabile da tutti i membri del gruppo |
| **Questioni aperte** | - |

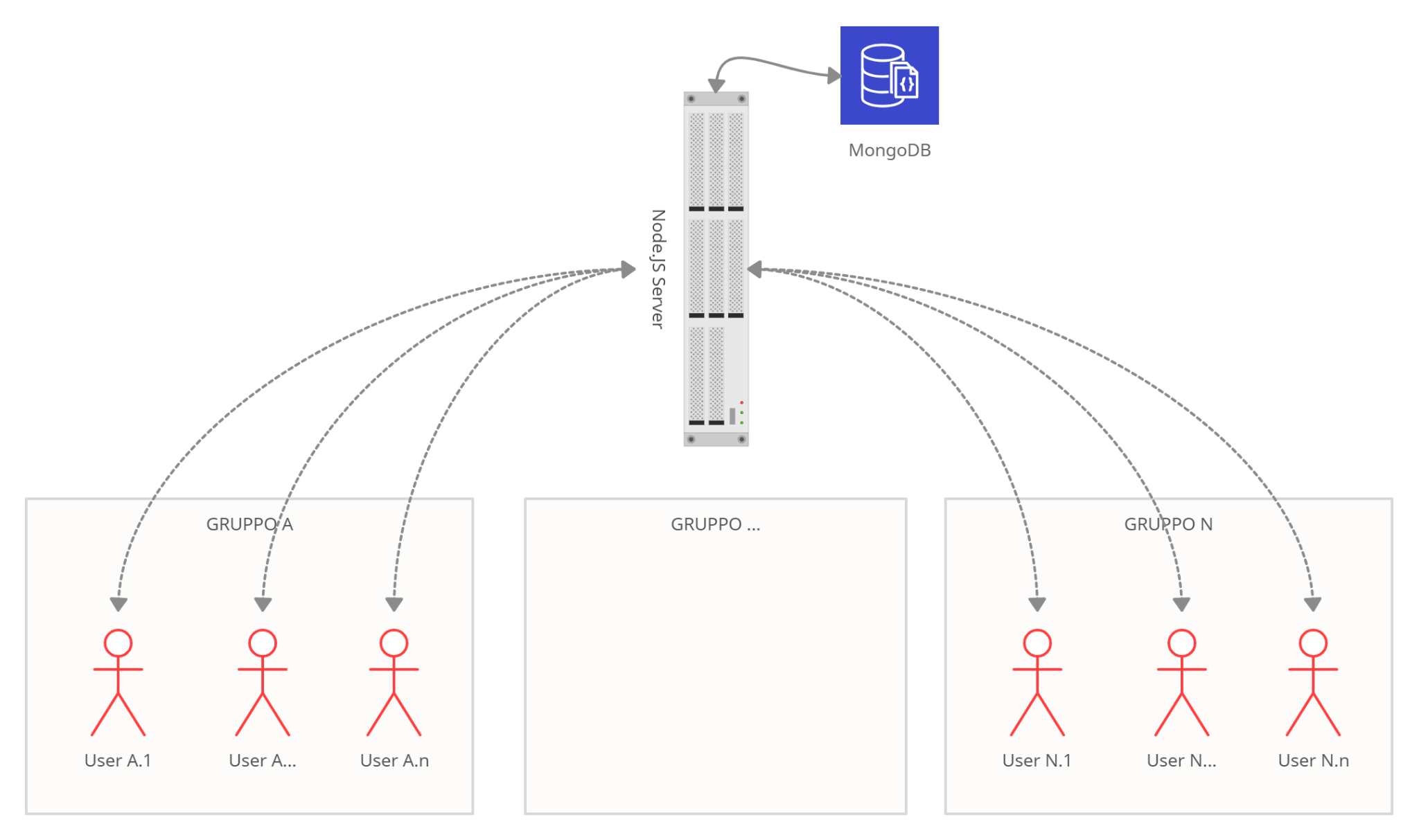
* + 1. **Gestione mercatino digitale**

| **Nome caso d’uso** | Gestione mercatino digitale |
| --- | --- |
| **Scopo** | Ogni utente può condividere un oggetto (giocattolo, libro, alimento, ecc) o decidere di cambiarlo con un'altro che richiede, anche come prestito. |
| **Attori** | Utente. |
| **Pre-condizioni** | L’utente deve partecipare ad almeno un gruppo. |
| **Trigger** | L’utente una volta che entra all’interno del gruppo può selezionare la visualizzazione del mercatino digitale. |
| **Descrizione** | 1. Permette l’inserimento di elementi nel mercatino digitale e la richiesta di prodotti per lo scambio 2. Permette di selezionare e confermare lo scambio di un contenuto presente nel mercatino (se l’utente possiede il prodotto richiesto da chi ha postato l’offerta). 3. E’ possibile ricercare un elemento del mercatino tramite search bar. |
| **Alternative** | 1. E’ possibile rimuovere i propri contenuti postati. |
| **Post-condizioni** | Tutti i post sono salvati all’interno del database e accessibili e visibili da tutti i membri del gruppo. |
| **Questioni aperte** | - |

* 1. **Modello di collaborazione**

Il modello di collaborazione dell’applicazione sarà il seguente:

* Ogni utente partecipante ad un gruppo avrà accesso alle stesse informazioni sulle attività presenti nel database grazie alla connessione al server Node.JS
* Tutte le operazioni sul database saranno effettuate grazie alle API fornite dal server Node.JS organizzato nello stile architetturale REST.
* Solo gli utenti creatori avranno accesso a delle funzionalità aggiuntive garantite da un controllo delle autorizzazioni fornite sempre dal Server in modo che il database sia protetto e non possa essere modificato da ogni utente.



1. **Classi di requisiti**

* 1. **Requisiti durevoli e volatili**

**Requisiti durevoli**

Sono tutti quei requisiti che non variano per l’intera durata del progetto.

Per utilizzare l’applicazione e di conseguenza poter comunicare con server e database, il dispositivo utilizzato deve essere dotato di una connessione ad internet stabile. Le attività e i gruppi devono necessariamente avere un utente “creatore” che avrà successivamente la possibilità di gestire il tutto. Inoltre, la funzionalità GPS deve essere attiva per permettere il tracciamento dei luoghi di interesse.

**Requisiti volatili**

Sono i requisiti che possono variare nel tempo e durante lo sviluppo del progetto. Solitamente sono requisiti più “leggeri” e spesso determinano dei casi limite. Nel nostro contesto ad esempio il numero dei gruppi e delle attività non è definito. Infatti ogni utente può creare arbitrariamente i propri gruppi e/o le proprie attività, inserendo o invitando partecipanti fino ad un massimo di 256 persone. Nella gestione del contact tracing, invece, i dati riportati risalgono fino ai 30 giorni precedenti rispetto alla data presa in considerazione. Per poter effettuare scambi all’interno del mercatino digitale devono essere presenti almeno 2 oggetti, condivisi da 2 utenti differenti. Infine il database deve restituire le informazioni con un ritardo non superiore a 3 secondi.

* 1. **Requisiti funzionali e non funzionali**

I requisiti possono essere suddivisi anche in requisiti funzionali e non funzionali. In questo contesto le due tipologie descrivono le funzionalità ed i vincoli del sistema.

I **requisiti funzionali** comprendono l’insieme di tutti i servizi che vengono offerti dal sistema.

I **requisiti non funzionali**, invece, rappresentano i vincoli che devono essere soddisfatti rispetto al sistema e alle operazioni effettuate su di esso. Solitamente sono più “rigidi” dei precedenti, poiché devono essere soddisfatti per non rendere il sistema inutile.

1. **Definizione dei requisiti funzionali**

I requisiti funzionali descrivono le funzionalità del software come: servizi che devono essere forniti dal software stesso, risposte che l’utente si aspetta in specifiche condizioni e output che il software deve produrre in risposta a determinati input.

(vedi 4.2)

Il piano di attuazione dei requisiti funzionali è dettagliato nella progettazione del

sistema.

Di seguito la descrizione dettagliata di ogni sezione delle schede che verranno utilizzate.

| Titolo | Descrizione |
| --- | --- |
| **ID** | Identificativo univoco (F = requisito funzionale) |
| **Caso d’uso** | Si fa riferimento ad uno dei casi d’uso definiti nella sezione 3.1. |
| **Definizione** | Un requisito funzionale definisce la funzione di un sistema per un determinato componente. Definisce la tipologia degli ingressi e delle uscite, nonché il comportamento. Un requisito funzionale è tale per soddisfare una specifica richiesta, senza la richiesta non è effettuata nel modo richiesto. |
| **Motivazione** | Il motivo che ha portato alla scelta del requisito. |
| **Influisce** | I cambiamenti che il requisito comporta. Descrive brevemente l’influenza sulle caratteristiche del progetto che verranno alterate per far si che il vincolo sia rispettato. |
| **Priotità** | Un requisito funzionale definisce le priorità che devono essere rispettate in modo da far proseguire correttamente e in maniera lineare il progetto. |

* 1. **Requisiti**

Qui verranno elencate tutte le schede per i requisiti funzionali.

| **ID** | RF01 |
| --- | --- |
| **Caso d’uso** | Creazione utente |
| **Definizione** | L’applicazione deve permettere la creazione di un nuovo utente nel caso sia la prima volta che utilizza l’app. Verranno richieste le informazioni personali e il sistema genererà in automatico un ID univoco. |
| **Motivazione** | Per utilizzare l’applicazione si deve essere in possesso di un profilo utente, che attraverso il proprio ID può prelevare le proprie informazioni, come attività e gruppi, oppure aggiungerne di nuove. |
| **Influisce** | L’utente deve identificarsi prima di poter accedere alle funzionalità nell’applicazione. |
| **Priotità** | Alta |

| **ID** | RF02 |
| --- | --- |
| **Caso d’uso** | Creazione utente |
| **Definizione** | L’applicazione deve garantire l’accesso ad essa tramite delle credenziali nel caso in cui l’utente sia già registrato. |
| **Motivazione** | Nel caso in cui l’utente sia già registrato all’applicazione deve avere la possibilità di accedere alla stessa tramite le credenziali in suo possesso. Un utente registrato possiede ovviamente un profilo utente identificato da un ID univoco. |
| **Influisce** | L’utente ha bisogno di loggarsi per accedere alle funzionalità dell’applicazione. |
| **Priotità** | Alta |

| **ID** | RF03 |
| --- | --- |
| **Caso d’uso** | Creazione bambino |
| **Definizione** | L’applicazione deve dare la possibilità di creare un profilo bambino, chiedendo le informazioni personali e creando un ID univoco che lo identifica. |
| **Motivazione** | Per la gestione dei bambini ognuno di essi deve essere in possesso di un profilo (identificato da un id) in modo che si possano prelevare/aggiungere/aggiornare/eliminare informazioni. |
| **Influisce** | L’utente deve identificarsi prima di poter accedere alle funzionalità nell’applicazione. |
| **Priotità** | Media |

| **ID** | RF04 |
| --- | --- |
| **Caso d’uso** | Visione informazioni utente, attività, gruppi |
| **Definizione** | L’utente deve conoscere le proprie informazioni personali, come ad esempio username, nome, cognome, insieme alle attività e i gruppi in cui è presente. |
| **Motivazione** | L’utente deve avere accesso alle informazioni che le altre persone possono vedere, come ad esempio lo username; inoltre, deve poter vedere le proprie attività e i propri gruppi in modo da poter monitorare e controllare la propria situazione. |
| **Influisce** | -- |
| **Priotità** | Bassa |

| **ID** | RF05 |
| --- | --- |
| **Caso d’uso** | Creazione e gestione gruppo |
| **Definizione** | L’applicazione deve poter permettere la creazione di un nuovo gruppo da parte di un utente. Se questo è il creatore può gestirlo in diversi modi |
| **Motivazione** | Per poter avere dei gruppi e la loro conseguente gestione, essi devono poter essere creati da un determinato utente. |
| **Influisce** | Se un utente non può creare un gruppo, non ci possono essere gruppi. |
| **Priotità** | Alta |

| **ID** | RF06 |
| --- | --- |
| **Caso d’uso** | Creazione e gestione gruppo |
| **Definizione** | L’applicazione deve dare la possibilità di accedere ad un gruppo già esistente. |
| **Motivazione** | L’app deve permettere la ricerca attraverso il nome dei gruppi già creati e la creazione di link di invito ad un gruppo (in modo che degli utenti possano invitarne altri). |
| **Influisce** | Se non si ha la possibilità di accedere ad un gruppo, esso non può avere altri membri oltre al creatore stesso. |
| **Priotità** | Alta |

| **ID** | RF07 |
| --- | --- |
| **Caso d’uso** | Creazione attività |
| **Definizione** | L’applicazione deve poter permettere la creazione di un nuova attività da parte di un utente. |
| **Motivazione** | Per poter avere delle attività e la loro conseguente gestione, esse devono poter essere create da un determinato utente. |
| **Influisce** | Se un utente non può creare un’attività, non ci possono essere attività. |
| **Priotità** | Alta |

| **ID** | RF08 |
| --- | --- |
| **Caso d’uso** | Creazione attività |
| **Definizione** | L’applicazione deve dare la possibilità di partecipare ad un’attività già esistente. |
| **Motivazione** | L’app deve permettere la partecipazione a delle attività che sono state precedentemente create da altri utenti. |
| **Influisce** | Se non si ha la possibilità di accedere ad un’attività, essa non può avere altri membri oltre al creatore stesso. |
| **Priotità** | Alta |

| **ID** | RF09 |
| --- | --- |
| **Caso d’uso** | Contact Tracing |
| **Definizione** | L’applicazione deve permettere il tracciamento delle persone che hanno partecipato alla stessa attività. |
| **Motivazione** | Per garantire una corretta gestione dello scenario pandemico l’app deve garantire il tracing di tutte le persone che hanno avuto un contatto (che hanno quindi partecipato ad una stessa attività). |
| **Influisce** | Se non si tiene conto dei contatti avuti dalle persone, nel caso di un contagiato non si potrebbe sapere chi altro deve testare la sua condizione. |
| **Priotità** | Alta |

| **ID** | RF10 |
| --- | --- |
| **Caso d’uso** | Gestione attività |
| **Definizione** | L’applicazione garantisce la corretta gestione delle attività da parte di un utente che le ha create. |
| **Motivazione** | L’utente che è creatore dell’attività deve avere la possibilità di gestirla modificando le opzioni specificate in “creazione attività” vedi (3.1.6) |
| **Influisce** | -- |
| **Priotità** | Media |

| **ID** | RF11 |
| --- | --- |
| **Caso d’uso** | Rimozione attività |
| **Definizione** | L’applicazione deve dare la possibilità di eliminare un’attività dalla lista delle attività di un utente. |
| **Motivazione** | L’utente che non vuole più partecipare ad un’attività deve poter eliminare la stessa dalla propria lista delle attività (che verrà aggiornata di conseguenza) |
| **Influisce** | -- |
| **Priotità** | Alta |

| **ID** | RF12 |
| --- | --- |
| **Caso d’uso** | Annullamento dell’attività |
| **Definizione** | L’applicazione deve permettere l’annullamento di una determinata attività da parte dell’utente che l’ha creata. |
| **Motivazione** | L’utente che per determinati motivi vuole annullare un’attività, cancellando così i dati per tutti i partecipanti, deve poterlo fare direttamente. |
| **Influisce** | -- |
| **Priotità** | Media |

| **ID** | RF13 |
| --- | --- |
| **Caso d’uso** | Invito, aggiunta o rimozione partecipanti |
| **Definizione** | L’applicazione deve permettere la gestione dei partecipanti di un gruppo. |
| **Motivazione** | Per la corretta gestione di un gruppo devono essere possibili le operazioni di invito, aggiunta e rimozione di partecipanti. |
| **Influisce** | Nel caso in cui non si possano gestire i partecipanti, un gruppo perderebbe il suo significato. |
| **Priotità** | Alta |

| **ID** | RF14 |
| --- | --- |
| **Caso d’uso** | Invito, aggiunta o rimozione partecipanti |
| **Definizione** | L’applicazione deve permettere la gestione dei partecipanti ad un’attività. |
| **Motivazione** | Per la corretta gestione di un’attività devono essere possibili le operazioni di invito, aggiunta e rimozione di partecipanti. |
| **Influisce** | Nel caso in cui non si possano gestire i partecipanti, un’attività perderebbe il proprio senso. |
| **Priotità** | Alta |

| **ID** | RF15 |
| --- | --- |
| **Caso d’uso** | Richiesta green pass. |
| **Definizione** | L’applicazione deve dare la possibilità di settare l’obbligo di green pass (per i partecipanti) agli utenti creatori di attività. |
| **Motivazione** | L’utente che ha creato una certa attività deve avere la possibilità di vincolare i partecipanti attraverso l’obbligo del green pass che verrà controllato in presenza. |
| **Influisce** | -- |
| **Priotità** | Alta |

| **ID** | RF16 |
| --- | --- |
| **Caso d’uso** | Gestione bacheca |
| **Definizione** | L’applicazione deve permettere la condivisione di contenuti all’interno di un determinato gruppo. |
| **Motivazione** | Un utente che partecipa ad un certo gruppo deve poter visualizzare la bacheca e poter condividere vari tipi di contenuti multimediali su di essa, in modo da renderli accessibili anche agli altri partecipanti. |
| **Influisce** | Se un utente non può condividere materiali sulla bacheca di un suo gruppo, la bacheca non potrebbe funzionare correttamente. |
| **Priotità** | Media |

| **ID** | RF17 |
| --- | --- |
| **Caso d’uso** | Gestione bacheca |
| **Definizione** | L’applicazione deve permettere l’eliminazione di contenuti dalla bacheca. |
| **Motivazione** | Un utente che partecipa ad un certo gruppo, nel caso voglia eliminare dei contenuti che ha precedentemente condiviso sulla bacheca, deve poterlo fare direttamente. |
| **Influisce** | -- |
| **Priotità** | Bassa |

| **ID** | RF18 |
| --- | --- |
| **Caso d’uso** | Gestione mercatino digitale |
| **Definizione** | L’applicazione deve permettere la condivisione o lo scambio di oggetti. |
| **Motivazione** | Ogni utente deve avere la possibilità di condividere un proprio oggetto all’interno del mercatino digitale oppure di effettuare uno scambio con un oggetto che è già presente all’interno del mercatino, condiviso quindi da un altro oggetto. |
| **Influisce** | Se un utente non può condividere materiali sulla bacheca di un suo gruppo, la bacheca non potrebbe funzionare correttamente. |
| **Priotità** | Media |

| **ID** | RF19 |
| --- | --- |
| **Caso d’uso** | Gestione mercatino digitale |
| **Definizione** | L’applicazione deve permettere l’eliminazione degli scambi che sono stati effettuati. |
| **Motivazione** | Ogni utente deve avere la possibilità di eliminare uno o più scambi effettuati con altri utenti. I motivi possono essere svariati, ad esempio l’oggetto scambiato non va bene per una delle due parti. |
| **Influisce** | -- |
| **Priotità** | Bassa |

1. **Definizione dei requisiti non funzionali**

I requisiti non funzionali specificano i vincoli, i criteri e le proprietà che definiscono la corretta pianificazione di un progetto. Riguardano diverse fasi del progetto, dall’organizzazione allo sviluppo, e influenzano sia caratteristiche e standard generali che feature specifiche.   
È possibile che alcuni requisiti non funzionali facciano riferimento a requisiti funzionali.  
  
Ad ogni requisito non funzionale sono stati assegnati degli attributi che lo descrivono ed è stata fatta una descrizione che mostri il motivo per cui è stato scelto.

| Titolo | Descrizione |
| --- | --- |
| **ID** | Identificativo univoco (N = requisito non funzionale) |
| **Riferimento funzionale** | Possibile riferimento ad uno o più requisiti funzionali descritti nella sezione 5 |
| **Definizione** | L’azione che il requisito necessita Descrive brevemente il comportamento da seguire per rispettare i vincoli, i criteri e le proprietà definite nel sistema |
| **Motivazione** | Il motivo che ha portato alla scelta del requisito |
| **Influisce** | I cambiamenti che il requisito comporta Descrive brevemente l’influenza sulle caratteristiche del progetto che verranno alterate per far si che il vincolo sia rispettato |
| **Priotità** | In generale ha la stessa priorità del requisito funzionale a cui fa riferimento Può avere altri livelli di priorità a seconda dell’importanza della funzione a cui fa riferimento |

Di seguito elenchiamo tutti i requisiti non funzionali che l’intero progetto dovrà rispettare.

* 1. **Requisiti di prodotto**

Sono elencati i requisiti che il prodotto deve soddisfare, ovvero i vari comportamenti che deve assumere nelle varie situazioni.

1. **SICUREZZA**Abilità di funzionare in caso di possibili malfunzionamenti o eventi inaspettati.

| **ID** | N01 |
| --- | --- |
| **Riferimento funzionale** | RF04, RF06, RF07, RF09, RF13, RF15 |
| **Definizione** | Non verranno mai rese pubbliche informazioni che possano essere ritenute sensibili senza aver prima ottenuto il consenso dell’utente specifico |
| **Motivazione** | Rispetto della privacy degli utenti |
| **Influisce** | Le funzionalità dovranno essere progettate in modo da non far trapelare informazioni sensibili |
| **Priotità** | Media |

| **ID** | N02 |
| --- | --- |
| **Riferimento funzionale** | RF04, RF05, RF06, RF07, RF08, RF09, RF10, RF11, RF13, RF14, RF15, RF16, RF17, RF18, RF19 |
| **Definizione** | Evitare operazioni che portino a crash del sistema bloccando il funzionamento dell’applicazione |
| **Motivazione** | Mantenere l’integrità dei risultati delle operazioni effettuate |
| **Influisce** | Prestazioni e user experience |
| **Priotità** | Media |

| **ID** | N03 |
| --- | --- |
| **Riferimento funzionale** | Tutti |
| **Definizione** | In caso di malfunzionamento del dispositivo personale dell’utente, esso deve poter comunque accedere ai servizi offerti dall’applicazione attraverso altri dispositivi |
| **Motivazione** | Necessità di garantire il servizio al verificarsi di imprevisti dal punto di vista degli utilizzatori |
| **Influisce** | Lo sviluppo dell’architettura deve tener conto la distribuzione dei dati tra client e server |
| **Priotità** | Alta |

| **ID** | N04 |
| --- | --- |
| **Riferimento funzionale** | Nessuno |
| **Definizione** | In caso l’applicazione venga disinstallata, non bisognerà lasciare dati o altre tracce sul dispositivo |
| **Motivazione** | Non è corretta prassi inquinare la memoria con dati non necessari |
| **Influisce** | Prestazioni del dispositivo |
| **Priotità** | Bassa |

| **ID** | N05 |
| --- | --- |
| **Riferimento funzionale** | RF11 |
| **Definizione** | All’interno del database, ogni qualvolta che un’attività sarà terminata, verrà cancellata la relativa chat e tutte le informazioni che non saranno più strettamente necessarie |
| **Motivazione** | Mantenimento nel database di soli dati effettivamente utili per diminuire la quantità di dati messi a rischio in caso di attacchi o malfunzionamenti |
| **Influisce** | Manutenzione del database |
| **Priotità** | Media |

1. **EFFICIENZA**  
   Capacità di funzionare in maniera efficace senza causare sprechi di spazio, tempo e risorse computazionali.

| **ID** | N06 |
| --- | --- |
| **Riferimento funzionale** | Nessuno |
| **Definizione** | È necessario che l’applicazione occupi la minor quantità possibile di spazio all’interno del dispositivo |
| **Motivazione** | Se lo spazio occupato dovesse essere eccessivo, molti utenti potrebbero decidere di disinstallare l’applicazione |
| **Influisce** | Manutenzione del database |
| **Priotità** | Media |

| **ID** | N07 |
| --- | --- |
| **Riferimento funzionale** | Tutti |
| **Definizione** | Evitare operazioni che rallentano eccessivamente l’utilizzo del sistema e cercare di sfruttare al massimo le capacità computazionali dei dispositivi attraverso processi in background, senza che l’utente se ne accorga |
| **Motivazione** | Porterebbe ad un peggioramento dell’esperienza da parte degli utenti, che potrebbero decidere di abbandonare l’applicazione per colpa della frustrazione |
| **Influisce** | Scelta meticolosa delle strategie di programmazione |
| **Priotità** | Alta |

1. **ADATTABILITÀ**Abilità di funzionare in sistemi caratterizzati da sistemi operativi di versioni differenti, anche a seguito di modifiche alla progettazione o alle tecnologie usate

| **ID** | N08 |
| --- | --- |
| **Riferimento funzionale** | Nessuno |
| **Definizione** | Utilizzo della versione di Android 6.0+ |
| **Motivazione** | Per far si che l’applicazione sia supportata da una vasta gamma di dispositivi e che quindi il numero di possibili utilizzatori non venga limitato Inoltre versioni assodate del software dispongono di documentazioni e test più soddisfacenti delle controparti più recenti |
| **Influisce** | Scelta di funzionalità su cui poter fare affidamento e quali dover rinunciare |
| **Priotità** | Alta |

1. **USABILITÀ**

Modifiche che permettano agli utenti di poter sfruttare l’app in maniera più comoda, efficiente e completa

| **ID** | N09 |
| --- | --- |
| **Riferimento funzionale** | Tutti |
| **Definizione** | Introduzione della modalità scura |
| **Motivazione** | Permette di utilizzare l’applicazione con più comfort in orari serali della giornata, e soddisfa i bisogni di coloro che la usano per semplici preferenze personali |
| **Influisce** | Modifica del design dell’interfaccia grafica |
| **Priotità** | Bassa |

| **ID** | N10 |
| --- | --- |
| **Riferimento funzionale** | Tutti |
| **Definizione** | Creazione di un’interfaccia intuitiva |
| **Motivazione** | Un’applicazione facile da capire e da usare migliora notevolmente l’opinione che i suoi utilizzatori hanno di essa e garantisce che non la abbandonino perché troppo confusionaria |
| **Influisce** | Progettazione dell’interfaccia grafica delle nuove funzionalità |
| **Priotità** | Media |

* 1. **Requisiti di processo**

| **ID** | N11 |
| --- | --- |
| **Riferimento funzionale** | Nessuno |
| **Definizione** | Bisognerà rispettare le date di consegna dei deliverables indicate durante il corso di Ingegneria del Software |
| **Motivazione** | La consegna entro i giorni prefissati permette una migliore organizzazione e distribuzione del carico di lavoro del progetto, inoltre consegnare in ritardo porterebbe a delle sanzioni da parte del docente |
| **Influisce** | Organizzazione del progetto |
| **Priotità** | Alta |

| **ID** | N12 |
| --- | --- |
| **Riferimento funzionale** | RF03, RF05, RF06, RF07, RF08, RF09, RF10, RF11, RF12, RF13, RF14, RF15, RF16, RF17, RF18, RF19 |
| **Definizione** | Utilizzo di MongoDB come infrastruttura per il database |
| **Motivazione** | È un vincolo imposto nella consegna del progetto da parte del professore  Permette di sfruttare un server remoto per il salvataggio delle informazioni, rendendole accessibili da diversi dispositivi e affidando la manutenzione ad un servizio affermato |
| **Influisce** | Organizzazione dei dati |
| **Priotità** | Alta |

| **ID** | N13 |
| --- | --- |
| **Riferimento funzionale** | Nessuno |
| **Definizione** | Utilizzo di Android come linguaggio di programmazione |
| **Motivazione** | È un vincolo imposto nella consegna del progetto da parte del professore  Permette di programmare le varie funzionalità utilizzando il linguaggio Java, e di strutturare la parte grafica tramite il linguaggio XML  È compatibile con una grande percentuale di dispositivi nel mondo |
| **Influisce** | Sviluppo del codice applicativo |
| **Priotità** | Alta |

* 1. **Requisiti esterni**

Insieme di requisiti e necessità che non sono intrinseci al sistema o alla sua organizzazione

| **ID** | N14 |
| --- | --- |
| **Riferimento funzionale** | Tutti |
| **Definizione** | Il telefono su cui è installata l’applicazione deve essere dotato di accesso ad internet |
| **Motivazione** | L’applicazione è fortemente basata sulla comunicazione attraverso la rete in ogni sua funzionalità, ed una mancata connessione non permetterebbe lo scambio di dati necessario per un corretto funzionamento del sistema |
| **Influisce** | Usabilità dell’applicazione |
| **Priotità** | Alta |

| **ID** | N15 |
| --- | --- |
| **Riferimento funzionale** | Nessuno |
| **Definizione** | Il servizio di MongoDB deve essere sempre attivo e correttamente funzionante |
| **Motivazione** | Il sistema si appoggia molto su tale tecnologia ed un suo malfunzionamento comprometterebbe l’utilità dell’applicazione |
| **Influisce** | Utilizzo dell’applicazione |
| **Priotità** | Alta |

| **ID** | N16 |
| --- | --- |
| **Riferimento funzionale** | RF07, RF08, RF10 |
| **Definizione** | Il dispositivo su cui verrà installata l’applicazione deve avere un servizio di GPS attivabile |
| **Motivazione** | È un’opzione necessaria all’API di Google Maps, per la ricerca dei punti di interesse, che verrà integrata all’interno dell’applicazione |
| **Influisce** | Funzionalità del sistema |
| **Priotità** | Alta |

* 1. **Misure dei requisiti**

Permettono una valutazione dell’andamento del progetto nel tempo, aiutando la fase di monitoraggio dei progressi..  
Forniscono dunque una misura di qualità che i membri del gruppo potranno seguire per sapere se il proprio lavoro sta rispettando gli standard prefissati.

| **Proprietà** | Misura |
| --- | --- |
| **Velocità** | Capacità di ricevere, analizzare e rispondere a determinati input ricevuti dall’utente entro un breve periodo di tempo |
| **Facilità d’uso** | Livello che indica la facilità con cui gli utilizzatori dell’applicazione sono in grado di comprenderne il funzionamento per poterne sfruttare al meglio tutte le sue funzionalità. L’obiettivo è di rendere il sistema deve essere intuitivo in modo che il pubblico che lo possa utilizzare possa essere il più ampio possibile |
| **Robustezza** | Capacità del sistema di funzionare al verificarsi di imprevisti o particolari esigenze durante il proprio funzionamento. Indica quanti casi siano stati previsti e gestiti dai programmatori al momento della progettazione. Idealmente l’applicazione non dovrebbe crashare, generare errori o assumere comportamenti non predefiniti |
| **Portabilità** | Grado di facilità con cui il sistema si adatta ad un nuovo ambiente. I cambiamenti possono seguire aggiornamenti del sistema operativo del dispositivo in uso oppure un vero e proprio cambiamento dell’hardware utilizzato. Idealmente l’applicazione dovrebbe funzionare indipendentemente dalle tecnologie usate, purchè compatibili con le scelte effettuate in fase di progettazione |

Seguono le misure dei requisiti non funzionali:

| **Proprietà** | **RNF** | Misura |
| --- | --- | --- |
| **Velocità** | **07** | Il tempo necessario per il completamento delle operazioni svolte dall’applicazione non supererà i due secondi |
| **11** | I deliverables saranno sempre consegnati entro le date di scadenza |
| **Facilità d’uso** | **10** | La struttura grafica dell’applicazione non dovrebbe creare ambiguità per l’utilizzatore, che non sbaglierà le scelte delle interazioni |
| **04** | L’utilizzatore non dovrà preoccuparsi di eseguire l’eliminazione di eventuali dati salvati sul proprio dispositivo, perché la loro cancellazione avverrà in automatico |
| **09** | La modalità scura renderà l’applicazione più ergonomica per i suoi utilizzi in orari serali |
| **16** | Il servizio di GPS permette di trovare la propria posizione automaticamente, facilitando l’utilizzo della mappa all’interno dell’applicazione in caso non si conosca sufficientemente la zona in cui si deve svolgere l’attività |
| **Robustezza** | **01** | Non è possibile che qualche attaccate acceda ad informazioni che l’utente non ha deciso di inserire manualmente |
| **02** | Le operazioni non bloccano l’intero sistema |
| **05** | Le informazioni persistono solo per la durata dell’attività e dunque non potranno creare anomalie in futuro |
| **12** | Non bisogna occuparsi di reperire fisicamente lo spazio di archiviazione dei dati di funzionamento dell’applicazione |
| **15** | In caso di malfunzionamento di un server, il servizio offerto da MongoDB si occuperà automaticamente di tutto il necessario per continuare a fornire il servizio di archiviazione |
| **14** | È necessaria per una corretta comunicazione con i server per i trasferimenti di messaggi, contenuti multimediali e varie informazioni |
| **Portabilità** | **03** | All’utente basterà eseguire il login nell’applicazione per poterla utilizzare pienamente anche da altri dispositivi, e dunque non è vincolato al proprio hardware attuale |
| **06** | Le dimensioni necessarie per l’installazione e l’utilizzo dell’applicazione devono essere tali che essa non sia ingombrante nel dispositivo che la ospiterà |
| **08** | Si userà la versione di Android 0.6+, per rendere l’applicazione disponibile ad un numero elevato di persone |
| **13** | La programmazione del software Android avverrà in Java, con l’aggiunta di componenti di XML per la strutturazione dell’interfaccia grafica |

1. **Evoluzione del sistema**

Per lo sviluppo del piano si assume che l’applicazione di partenza sia funzionante e correttamente strutturata, e soprattutto che gli obiettivi prefissati siano effettivamente possibili da raggiungere con il tempo e le risorse a disposizione.  
Al momento della pianificazione non sono stati previsti possibili mutamenti riguardanti l’hardware che supporterà il prodotto finale.  
Il team si impegna a produrre codice modulare e scalabile, così da avvantaggiare ogni possibile evoluzione del progetto, compresa la manutenzione, l’aggiunta di funzionalità e la rimozione di quelle presenti.

Inoltre, sempre in vista di modifiche future, i vari sorgenti e il progetto stesso saranno opportunamente documentati.

Qui vengono elencate alcune possibili evoluzioni:

1. **Chat dedicata**

Creazione di una chat dedicata per ogni attività, accessibile e visibile esclusivamente dai responsabili e da chi decide di partecipare.

1. **Migliore integrazione dei luoghi**

Connessione mediante API di Google Maps per la ricerca dei punti di interesse, con annessa apertura di un’app di navigazione al momento della selezione del luogo.

1. **Orari di disponibilità**

Ogni utente può indicare i propri orari di disponibilità generali, in modo da informare tutte le persone presenti nei suoi stessi gruppi.

1. **Dark Mode**

Inserimento della modalità scura dell’app.

1. **Bug Fixes**

Modifiche da apportare a sezioni già implementate in Family Share, che si ritiene saranno utili per garantire che il risultato finale raggiunga il desiderato livello di qualità.

1. **Specifica dei requisiti**

In questa sezione verranno descritte e approfondite le specifiche dell’applicazione, ovvero le descrizione precise e dettagliate delle specifiche del sistema. In questa sezione il linguaggio naturale sarà meno presente rispetto al linguaggio tecnico per descrivere in totalità e massima accuratezza ogni specifica.

Qui sotto è descritta ciascuna sezione delle schede che verranno utilizzate.

| **Titolo** | Descrizione |
| --- | --- |
| **ID** | Identificativo univoco (S = specifica) |
| **Riferimento funzionale** | Riferimento ad uno dei requisiti funzionali descritti nella sezione 5 |
| **Specifica** | Descrizione precisa e dettagliata delle funzionalità del sistema. Serve come base da una parte per il contratto, e dall’altra per la progettazione e lo sviluppo. Ogni requisito funzionale viene qui descritto in modo più dettagliato e formale |
| **Pre-condizione** | Condizioni che devono essere soddisfatte all’inizio della specifica. Rappresentano le “garanzie minime” che devono essere soddisfatte per poter attivare la suddetta specifica. |
| **Post-condizione** | Condizioni sempre soddisfatte al termine della specifica. |
| **Side effect** | Effetto collaterale che può essere generato dalla specifica. |

* 1. **Specifiche**

| **ID** | S01 |
| --- | --- |
| **RF** | RF01 |
| **Specifica** | L’applicazione deve permettere la creazione di un nuovo utente nel caso sia la prima volta che utilizza l’app. Verranno richieste le informazioni personali e il sistema genererà in automatico un ID univoco. |
| **Pre-condizione** | L’applicazione deve essere scaricabile dallo store e installata correttamente nel telefono |
| **Post-condizione** | Al termine della registrazione l’utente sarà in grado di navigare nel menu principale |
| **Side effect** | Creazione nel database di una entry relativa all’utente |

| **ID** | S02 |
| --- | --- |
| **RF** | RF02 |
| **Specifica** | L’applicazione deve permettere l’accesso di un utente precedentemente registrato. L’identificazione di quest’ultimo avviene mediante un id univoco nel database. |
| **Pre-condizione** | L’utente è registrato all’applicazione |
| **Post-condizione** | L’utente ha accesso all’app contenente i suoi vari dati. Può inoltre svolgere determinate nuove azioni |
| **Side effect** | L’app verrà popolata con tutte le informazioni ad esso legate e verranno rese disponibili diverse nuove sezioni |

| **ID** | S03 |
| --- | --- |
| **RF** | RF03 |
| **Specifica** | É resa disponibile la possibilità di creare un profilo bambino con le relative informazioni personali. Ogni profilo è identificato nel database mediante un ID univoco |
| **Pre-condizione** | Accesso dell’utente all’app |
| **Post-condizione** | Il profilo del bambino è disponibile nell’app e nel database, è possibile svolgere varie operazioni |
| **Side effect** | Inserimento di di una entry nel database con le informazioni del bambino ed eventualmente del legame con un utente esistente |

| **ID** | S04 |
| --- | --- |
| **RF** | RF04 |
| **Specifica** | L’utente può visualizzare i gruppi a cui appartiene, le attività a cui ha scelto di partecipare e le sue informazioni personali. Tutti i dati sono ricavati mediante query nel database |
| **Pre-condizione** | Accesso dell’utente all’app |
| **Post-condizione** | Visualizzazione di tutti i dettagli dell’utente |
| **Side effect** | L’app verrà popolata con tutte le informazioni ad esso legate e verranno rese disponibili diverse nuove sezioni |

| **ID** | S05 |
| --- | --- |
| **RF** | RF05 |
| **Specifica** | Mediante l’applicazione è possibile creare un gruppo ed amministrarlo, se si è l’utente creatore. Nel database quindi verranno salvati i riferimenti tra utente e gruppo, e le informazioni di quest’ultimo |
| **Pre-condizione** | Accesso dell’utente all’app |
| **Post-condizione** | Visualizzazione di tutti i dettagli del gruppo |
| **Side effect** | L’app verrà popolata con tutte le informazioni legate al gruppo e verranno rese disponibili diverse nuove sezioni, come la possibilità per utenti registrati di entrare a farne parte |

| **ID** | S06 |
| --- | --- |
| **RF** | RF06 |
| **Specifica** | É possibile effettuare l’accesso a gruppi già esistenti, identificati da un ID nel database, o essere invitati a partecipare, mediante un codice o un link generato dall’app |
| **Pre-condizione** | Accesso dell’utente all’app |
| **Post-condizione** | Visualizzazione di tutti i dettagli del gruppo |
| **Side effect** | L’app verrà popolata con tutte le informazioni ad esso legate e verranno rese disponibili diverse nuove sezioni. |

| **ID** | S07 |
| --- | --- |
| **RF** | RF07 |
| **Specifica** | Attraverso l’applicazione è possibile creare nuove attività, specificando, ad esempio, luogo, data, ora, ripetizione dell’evento e referenti dell’attività |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Visualizzazione e modifica di tutti i dettagli dell’attività |
| **Side effect** | L’app verrà popolata con tutte le informazioni legate all’app e verranno rese disponibili diverse nuove sezioni.  Altri utenti del gruppo inoltre potranno partecipare all’attività creata |

| **ID** | S08 |
| --- | --- |
| **RF** | RF08 |
| **Specifica** | É possibile effettuare l’accesso ad attività già esistenti, identificate da un ID nel database, o essere invitati a partecipare, mediante un codice o un link generato dall’app |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Visualizzazione di tutti i dettagli dell’attività |
| **Side effect** | L’app verrà popolata con le informazioni inerenti all’attività come: gli utenti o i bambini che vi partecipano, il luogo, la data e l’ora |

| **ID** | S09 |
| --- | --- |
| **RF** | RF09 |
| **Specifica** | L’applicazione segnala tramite notifica e avviso su homepage se qualche membro di un gruppo in cui si è presenti, o qualche partecipante ad un’attività alla quale si è preso parte, ha segnalato di essere positivo a Covid19.  Queste operazioni devono essere fatte modificando o aggiungendo funzioni a server back-end e database utili per implementare il tracciamento |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Segnalazione di possibile positività a tutti gli utenti che possono essere entrati in contatto col positivo |
| **Side effect** | Generazione notifica e banner nell’applicazione.  Modifica di informazioni nel database inerenti al tracing |

| **ID** | S10 |
| --- | --- |
| **RF** | RF10 |
| **Specifica** | Le attività possono essere completamente modificate dell’utente che le ha create, inoltre alcuni dettagli, come la partecipazione dei propri bambini all’attività, possono essere modificati anche da altri utenti |
| **Pre-condizione** | Accesso dell’utente al gruppo |
| **Post-condizione** | L’attività viene modificata |
| **Side effect** | Modifica dei dati dell’attività nel database, con conseguente eventuale rimozione, aggiunta o modifica di informazioni per gli utenti che visualizzano i dettagli dell’attività |

| **ID** | S11 |
| --- | --- |
| **RF** | RF11 |
| **Specifica** | L’applicazione permette ad un certo utente di rimuoversi dall’attività, con conseguente aggiornamento della schermata con la lista degli eventi a cui partecipa e relativi collegamenti nel database.  Inoltre, se una persona è dichiarata positiva, è annullata in automatico la partecipazione a tutte le attività a cui è iscritto nei 14 giorni successivi alla data in cui sono stati rilevati i sintomi o la positività. Questo annullamento è anche segnalato con una notifica all’utente |
| **Pre-condizione** | Accesso dell’utente al gruppo |
| **Post-condizione** | Rimozione dalla schermata dell’app delle informazioni sull’attività. Notifica se la rimozione è relativa alla positività dell’utente |
| **Side effect** | Rimozione del collegamento nel database tra utente e attività. |

| **ID** | S12 |
| --- | --- |
| **RF** | RF12 |
| **Specifica** | L’utente creatore dell’attività può decidere di annullarla.  A tutti i partecipanti verrà segnalato da una notifica e da un avviso sulla home page |
| **Pre-condizione** | Accesso dell’utente creatore dell’attività all’app |
| **Post-condizione** | Notifica e rimozione dalla schermata dell’app delle informazioni sull’attività |
| **Side effect** | Rimozione del collegamento nel database tra utenti e attività. |

| **ID** | S13 |
| --- | --- |
| **RF** | RF13 |
| **Specifica** | L’app comprende la possibilità di aggiungere e rimuovere utenti da gruppi esistenti, o invitarli mediante link o codice.  Aggiunta e rimozione sono privilegi disponibili solo al creatore del gruppo.  Gli utenti invitati devono essere approvati |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Accesso o rimozione dell’utente ad un gruppo |
| **Side effect** | Modifiche nel database per aggiunta o rimozione dal gruppo, aggiornamento delle schermate inerenti nell’app |

| **ID** | S14 |
| --- | --- |
| **RF** | RF14 |
| **Specifica** | L’app comprende la possibilità per il creatore dell’attività di amministrare i partecipanti per attività esistenti.  Aggiunta e rimozione sono privilegi disponibili solo al creatore dell’attività. |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Accesso o rimozione dell’utente ad un gruppo |
| **Side effect** | Modifiche nel database per aggiunta o rimozione dal gruppo, aggiornamento delle schermate inerenti nell’app |

| **ID** | S15 |
| --- | --- |
| **RF** | RF15 |
| **Specifica** | Deve essere svolto un controllo sui partecipanti ad un'attività se per essa è richiesto il Green Pass.  Nel suo profilo l’utente deve aver segnalato di essere in possesso di un Green Pass valido che verrà controllato prima che inizi l’attività |
| **Pre-condizione** | Partecipazione ad un attività |
| **Post-condizione** | Partecipazione, o rifiuto, dell’utente all’attività se non ha dichiarato di avere un green pass valido nel suo profilo |
| **Side effect** | -- |

| **ID** | S16 |
| --- | --- |
| **RF** | RF16 |
| **Specifica** | L’applicazione deve permettere la condivisione di contenuti all’interno di un determinato gruppo.  I documenti vengono memorizzati nel database e gli utenti possono visualizzarli o scaricarli |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Visione con possibilità di download da parte degli utenti |
| **Side effect** | Caricamento nel database dei vari documenti |

| **ID** | S17 |
| --- | --- |
| **RF** | RF17 |
| **Specifica** | L’applicazione deve permettere l’eliminazione di contenuti dalla bacheca. |
| **Pre-condizione** | Accesso all’app da parte dell’utente che ha postato l’elemento |
| **Post-condizione** | Rimozione dell’elemento dalla schermata di bacheca dei vari utenti |
| **Side effect** | Rimozione dal database dei vari documenti |

| **ID** | S18 |
| --- | --- |
| **RF** | RF18 |
| **Specifica** | Ogni utente può condividere o scambiare un proprio oggetto o richiede quello di un altro utente  L’oggetto può essere marcato come occupato, scambiato o disponibile tra le sue informazioni  Gli oggetti possono essere ricercati per categoria o per altre informazioni, come ad esempio il nome  L’oggetto e le sue informazioni possono anche essere modificati dal creatore dell’annuncio |
| **Pre-condizione** | Accesso dell’utente ad un gruppo |
| **Post-condizione** | Visualizzazione dei vari oggetti da parte degli utenti e possibilità di scambio tra questi |
| **Side effect** | Modifica dello stato dell’oggetto o delle informazioni ad esso inerenti |

| **ID** | S19 |
| --- | --- |
| **RF** | RF19 |
| **Specifica** | Una volta effettuato uno scambio esso deve poter essere eliminato.  L’altro utente coinvolto deve essere notificato dell’annullamento dello scambio o della condivisione mediante notifica o avviso nella home page |
| **Pre-condizione** | Aver concordato lo scambio, la condivisione o la richiesta di un oggetto |
| **Post-condizione** | Rimozione dello scambio dalla relativa sezione dell’app dell’utente |
| **Side effect** | Modifica dello stato dell’oggetto o delle informazioni ad esso inerenti. |

* 1. **Tabelle di tracciabilità dei requisiti**

La tabella di tracciabilità permette di rappresentare i requisiti correlati che sono raggiungibili l’uno dall’altro. Se nelle sezioni precedenti sono stati indicati dei riferimenti incrociati a requisiti che sono correlati, ora verranno evidenziate tutte queste correlazioni in un’unica tabella.

La tracciabilità è suddivisa in due tabelle, una contenente quella inerente a requisiti funzionali e requisiti funzionali, l’altra relativa a requisiti funzionali e non funzionali

| RF | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RF01 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF02 | x |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF03 |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF04 |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF05 |  | x |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF06 |  | x |  | x | x |  |  |  |  |  |  |  |  |  |  | x |  | x |  |
| RF07 |  | x |  | x |  | x |  |  |  |  |  |  |  |  | x |  |  |  |  |
| RF08 |  | x | x | x |  | x | x |  |  |  |  |  |  |  | x |  |  |  |  |
| RF09 |  | x | x | x | x | x | x | x | x | x | x | x | x | x | x |  |  |  |  |
| RF10 |  | x |  |  |  | x | x | x |  |  |  |  |  |  |  |  |  |  |  |
| RF11 |  | x |  |  |  |  | x | x |  |  |  |  |  |  |  |  |  |  |  |
| RF12 |  | x |  |  |  |  | x | x |  |  |  |  |  |  |  |  |  |  |  |
| RF13 |  | x |  |  |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF14 |  | x |  |  |  |  | x | x |  |  |  |  | x | x |  |  |  |  |  |
| RF15 |  | x |  |  |  | x | x | x | x |  |  |  |  |  |  |  |  |  |  |
| RF16 |  | x |  | x |  | x |  |  |  |  |  |  |  |  |  |  | x |  |  |
| RF17 |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| RF18 |  | x |  | x |  | x |  |  |  |  |  |  |  |  |  |  |  |  | x |
| RF19 |  | x |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

| RF | 01 | 02 | 03 | 04 | 05 | 06 | 07 | 08 | 09 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N01 |  |  |  | x |  | x | x |  | x |  |  |  | x |  | x |  |  |  |  |
| N02 |  |  |  | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N03 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N04 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N05 |  |  |  |  |  |  |  |  |  |  | x |  |  |  |  |  |  |  |  |
| N06 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N07 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N08 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N09 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N10 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N11 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N12 |  |  | x |  | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N14 | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x | x |
| N15 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| N16 |  |  |  |  |  |  | x | x |  | x |  |  |  |  |  |  |  |  |  |

1. **Appendici**

* 1. **Requisiti dell’app**

Per il corretto funzionamento della parte front-end dell’app è necessario che la parte back-end sia operativa.

* 1. **Requisiti del database**

I requisiti del database resi necessari per il progetto sviluppato dal team di Families Share e da questo team, sono perfettamente soddisfatti dal database MongoDB, infatti questo offre i seguenti vantaggi:

* Un modello di dati basato su documenti. Si tratta di una ricca struttura di dati in grado di contenere matrici e altri documenti.
* Profonda capacità di interrogazione. MongoDB supporta query dinamiche su documenti utilizzando un linguaggio di query basato su documenti che è quasi potente quanto SQL.
* Nessuna migrazione dello schema. Dato che MongoDB è privo di schemi, il tuo codice definisce il tuo schema.
* Un percorso chiaro verso la scalabilità orizzontale.

La memorizzazione ed il recupero dei dati è quindi semplice, efficiente e conforme alla sintassi JSON per la condivisione di informazioni.

I dati sono schematizzati nel database secondo quanto indicato nella documentazione tecnica di Families Share.

* 1. **Requisiti del server**

La macchina ospitante la parte server dell’applicazione deve essere abbastanza performante da svolgere la maggior parte dei calcoli computazionali del sistema e permettere la connessione di molti dispositivi, anche simultaneamente.

* 1. **Requisiti dei dispositivi**
* I dispositivi devono essere provvisti di un sistema operativo pari ad Android 6.0 (Marshmallow) o superiore;
* Avere un GPS integrato e la possibilità di connettersi ad internet;
* Avere almeno 1 GB di RAM;
* Almeno 150 MB di spazio di archiviazione liberi per poter installare l’app;
* È necessario aver installato Google Play Services 11+.

1. **Riferimenti**

Per creare questo documento sono stati utilizzati come riferimento:

* Il Documento dei requisiti di alcuni gruppi degli anni passati
* Il Piano di Progetto stilato precedentemente
* Materiale messo a disposizione dal professore.