



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

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

**Piano di Progetto
v1.0**

cpqs_lock



15/10/2021

Informazioni sul Documento

MUR FISR 2020		Acronimo	F2F
Coordinatore Progetto	Prof. Agostino Cortesi		
Deliverable	Piano di Progetto		
Data di Consegna	15/10/2021		
Team Leader	Marco Pistollato	880889@stud.unive.it	
Team members	Giovanni Stevanato	880077@stud.unive.it	
	Marco Quarta	880789@stud.unive.it	
	Alexandru Condurache Petronel	880890@stud.unive.it	

Document History

Version	Issue Date	Stage	Changes	Contributors
1.0	15 / 10 / 2021	Draft	Piano di Progetto stilato in prima versione	All members of the team
1.1		Draft		
2.0		Final		

INDICE

INTRODUZIONE	5
Overview del progetto	5
Struttura del documento	5
Evoluzione del progetto	6
Materiale di riferimento	6
Definizioni ed abbreviazioni	6
ORGANIZZAZIONE DEL PROGETTO	9
Modello del processo	9
Struttura organizzativa	10
Interfacce organizzative	11
Responsabilità di progetto	12
DESCRIZIONE DEI PROCESSI GESTIONALI	13
Obiettivi e priorità	13
Assunzioni, dipendenze, vincoli	14
Gestione dei rischi	16
Meccanismi di monitoraggio e controllo	20
Pianificazione dello staff	20
DESCRIZIONE DEI PROCESSI TECNICI	21
Metodi, strumenti e tecniche	21
Documentazione del software	23
Funzionalità di supporto al progetto	23
PIANIFICAZIONE	24
WBS (Work Breakdown Structure)	24
Dipendenze	28
Risorse necessarie	29
Allocazione del budget e delle risorse	31
Pianificazione	32
RIFERIMENTI	35

1. INTRODUZIONE

1.1. Overview del progetto

Il progetto in questione prevede la definizione dei requisiti, la progettazione, l'implementazione, la messa in esercizio ed il testing di un'estensione di un sistema già sviluppato ed in uso denominato "Families Share", il quale si pone come obiettivo di supportare a gruppi di genitori nell'organizzazione condivisa di attività di cura dei figli. La funzionalità che è stata pensata per essere aggiunta a questo sistema consiste nel dare la possibilità a persone facenti parte di uno stesso gruppo di condividere diverse tipologie di oggetti quali, a titolo esemplificativo e non esaustivo, libri, videogiochi, vestiti, attrezzi per la casa. Questa nuova estensione consente agli utenti che hanno oggetti che non utilizzano più di liberarsene andando ad aiutare altre persone che per un qualsiasi motivo desiderano possedere quegli oggetti favorendo in questo modo la "Second Hand Economy", ovvero la riduzione di consumi attraverso il riutilizzo di oggetti non nuovi ma che sono ancora nelle condizioni di svolgere le funzionalità adibite ad esso.

Inoltre questa funzionalità risponde alle esigenze che le persone potrebbero avere nell'ipotesi in cui a causa della malattia infettiva Covid-19 debbano rimanere in quarantena o in isolamento. Tali soggetti, infatti, dovendo rimanere sempre a casa per un certo lasso di tempo potrebbero avere bisogno di oggetti che in quel momento non avrebbero ovviamente la possibilità di andare ad acquistare e quindi il prestito a tempo determinato o indeterminato di quei beni risulta essere una comoda soluzione. Da notare che questa nuova estensione si rivolge a persone di qualsiasi età, non solo a gruppi di genitori.

Il sistema Families Share, con estensione annessa, sarà un'applicazione disponibile attraverso l'uso di un terminale Android.

Nella sezione 3.1 si discuterà in modo più approfondito della funzionalità appena citata.

1.2. Struttura del documento

Di seguito vengono riportate tutte le scadenze previste per questo progetto:

- Definizione dei gruppi di lavoro (entro il 22/09/2021)
- Proposta iniziale (29/9/2021)

- Piano di Progetto (15/10/2021)
- Documento dei Requisiti (25/10/2021)
- Piano di Testing (14/11/2021)
- Documento di Progettazione (30/11/2021)
- Versione 1.0 del codice sorgente (15/12/2021)
- Versione 1.1. del codice e allineamento documentazione (15.01.2022)

1.3. Evoluzione del progetto

Il progetto adotta il modello di processo Agile per essere sempre reattivi e disponibili di fronte a necessità di cambiamento richiesta dagli utenti. La necessità di soddisfare questa comunicazione con l'utente finale porta il team a prevedere le seguenti possibilità di evoluzione del progetto:

- **Canale di Comunicazione User-Developers:** per poter consentire agli utenti finali del prodotto di segnalare eventuali bug, fare richiesta di feature non ancora implementate ma che ritengono essere utili alla piattaforma si pensa di attuare la creazione di un canale di comunicazione interno all'applicazione dove questo invio di richieste e necessità possa avvenire.
- **Richiesta di aggiunta feature:** se la community degli utenti finali, attraverso i canali di comunicazione predefiniti, chiede in maniera numerosa una certa feature il team di Sviluppo valuterà le opzioni richiesta e deciderà se e come implementarle.

1.4. Materiale di riferimento

Per lo svolgimento dell'intero progetto, in tutte le sue parti, il gruppo di lavoro fa particolare riferimento ai materiali presenti sulla piattaforma Moodle del corso.

Per quanto riguarda, invece, lo sviluppo del lato front-end dell'applicazione, il team consulta [Android for Developers](#), una guida che ha l'obiettivo di mostrare come scrivere codice in Android Studio.

1.5. Definizioni ed abbreviazioni

Di seguito vengono riportate definizioni ed eventuali abbreviazioni utilizzate in questo documento:

- **Android:** Android è un sistema operativo per dispositivi mobili sviluppato da Google, progettato principalmente per smartphone e tablet e basato sul kernel Linux.

Fonte: https://it.wikipedia.org/wiki/Android_Studio

- **Android Studio:** Android Studio è un ambiente di sviluppo integrato (IDE) per lo sviluppo per la piattaforma Android.

Fonte: <https://it.wikipedia.org/wiki/Android>

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

Fonte: https://it.wikipedia.org/wiki/Integrated_development_environment

- **GitHub:** GitHub è un servizio di hosting per progetti software. Con la parola “hosting” si intende un servizio di rete che consiste nell'allocare su un server web delle pagine web di un sito web o di un'applicazione web, rendendolo così accessibile dalla rete Internet e ai suoi utenti.
- **Branch:** un branch è una diramazione, ovvero un percorso diverso e staccato da quello principale (master) del progetto. Questo è molto utile per avere dei flussi paralleli indipendenti in cui possiamo sperimentare nuove funzioni o correggere bug senza intaccare il progetto principale e poi, eventualmente, unirli (merge) nel ramo principale o in altri rami. Questo termine viene utilizzato in GitHub.

Fonte: <https://devdev.it/guida-git-versioning/i-branch-ramificazioni/>

- **Master:** anche questo termine viene utilizzato in GitHub. Con “master branch” si intende il branch principale del progetto che viene creato per la prima volta quando si invia il primo file nel proprio repository.
- **Node.JS:** Node.JS è un software che fornisce i servizi necessari all'esecuzione di codice Javascript. Quest'ultimo è un linguaggio di programmazione orientato agli oggetti e agli eventi comunemente utilizzato nella programmazione Web lato client per la creazione, in siti web e applicazioni web, di effetti dinamici interattivi tramite funzioni di script.

Fonte: <https://it.wikipedia.org/wiki/Node.js>, <https://it.wikipedia.org/wiki/JavaScript>

- **Java:** Java è un linguaggio di programmazione ad alto livello, orientato agli oggetti che si appoggia sull'omonima piattaforma software di esecuzione, specificamente progettato per essere il più possibile indipendente dalla piattaforma hardware di esecuzione.

Fonte: [https://it.wikipedia.org/wiki/Java_\(linguaggio_di_programmazione\)](https://it.wikipedia.org/wiki/Java_(linguaggio_di_programmazione))

- **SQL:** SQL è un linguaggio per database progettato per le seguenti operazioni:
 - creare e modificare schemi di database;
 - inserire, modificare e gestire dati memorizzati;
 - interrogare i dati memorizzati;
 - creare e gestire strumenti di controllo e accesso ai dati.

Fonte: https://it.wikipedia.org/wiki/Structured_Query_Language

2. ORGANIZZAZIONE DEL PROGETTO

2.1. Modello del processo

Il modello del processo che il team ha deciso di adottare è di tipo **Agile**. Il metodo agile permette di ridurre il rischio di fallimento sviluppando il software in finestre di tempo limitate chiamate iterazioni. Ogni iterazione è un piccolo progetto a sè stante e deve contenere tutto ciò che è necessario per rilasciare un piccolo incremento nelle funzionalità del software.

Per garantire un'ottimizzazione del lavoro e per far sì che tutti i membri partecipino attivamente allo sviluppo del progetto, il team si organizza durante la settimana con meeting (stabiliti dal project manager definito nel paragrafo 2.4) per identificare i compiti da svolgere in quella settimana ed effettuare un recap della situazione attuale del progetto. Gli incontri settimanali avverranno secondo la seguente programmazione.

Lunedì	Giovedì	Giorno a Scelta
Pomeriggio/Sera	Pomeriggio/Sera	Avanzamento del progetto individualmente.
Incontro per stabilire ed identificare i compiti da svolgere durante la settimana	Recap della situazione attuale del progetto ed eventuali task aggiuntivi svolti dall'intero team.	

Per tenere traccia degli avanzamenti e dello sviluppo del progetto il team utilizza una tabella che identifica l'avanzamento delle attività:

Elenco Attività	
• Attività 1	DONE
• Attività 2	DOING
• Attività 3	TO DO
• Attività 4	TO DO

DONE : Verranno inserite le attività completate fino a quel momento dall'inizio della pianificazione del progetto.

DOING : Indica che l'attività cominciate che devono essere ancora terminate (non è detto che l'attività verrà completata in quella determinata settimana).

TO DO : Verranno riportate le attività ancora da svolgere programmate per quella settimana.

Le attività saranno identificate e assegnate direttamente ad un componente del team, oppure svolte dall'intero gruppo nei giorni della settimana indicati in precedenza con le attività presenti nella sezione 5.1.

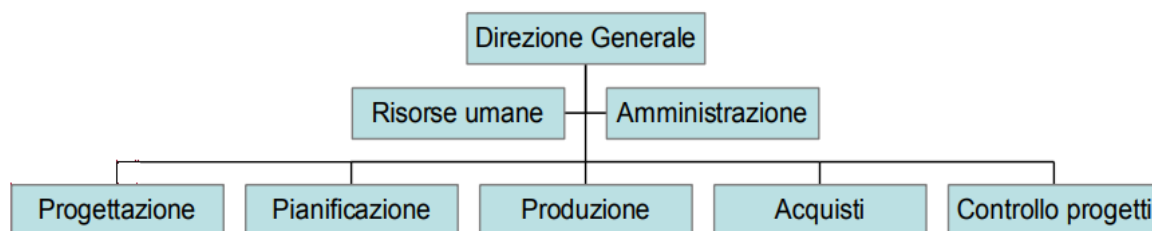
Le fasi principali del progetto sono:

- Analisi e definizione del progetto
- Definizione della struttura del team
- Pianificazione delle attività
- Costruzione WBS, Gantt e Pert
- Implementazione e test delle singole componenti
- Progettazione software
 - Sviluppo prima versione dell'app
 - Implementazioni delle funzioni base prestabilite
- Integrazione e test del sistema
- Sviluppo della versione completa dell'app
- Sviluppo interfaccia grafica e migliorie
- Controllo e consegna

Ogni fase del progetto avrà una durata con un certo margine di errore e verrà relazionata alle altre come spiegato nella sezione 5.1.

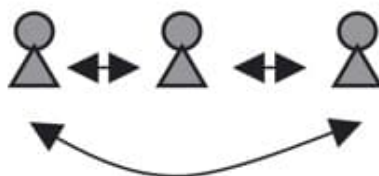
2.2. Struttura organizzativa

Un gruppo presenta al suo interno una **struttura organizzativa**, in questo caso il team si ha deciso di adottare per questo progetto un'**Organizzazione per Funzione**. Tale organizzazione non prevede un personale assegnato ad uno specifico progetto e non è presente un coordinatore delle attività.



Per quanto riguarda la **tipologia del team**, il gruppo ha scelto di adoperare un tipo:

- **Democratico Decentralizzato:** la seguente tipologia presenta alcune caratteristiche come:
 - L'assenza di un leader permanente all'interno del team
 - La partecipazione attiva di tutti i membri
 - Il consenso di gruppo sulle soluzioni e sulla organizzazione del lavoro
- **Comunicazione orizzontale:** comunicazione che prevede tutti i membri del gruppo sullo stesso livello, le informazioni fluiscono tra persone che ricoprono la stessa posizione nell'organizzazione.



Il tipo **democratico decentralizzato** presenta a sua volta:

- **VANTAGGI**
 - Ricerca degli errori in modo molto efficace (intero team assieme può risultare più veloce nell'individuare tali "problematiche")
- **SVANTAGGI**
 - E' una tipologia difficile da imporre, in quanto non sempre tutti i membri all'interno del team sono attivi e partecipi.

2.3. Interfacce organizzative

Durante la realizzazione del progetto, il team avrà bisogno di interfacciarsi con persone estranee del team. Il team attraverso questo strumento potranno osservare l'andamento del progetto ed effettuare eventuali piccoli cambiamenti grazie al parere delle entità esterne. In particolare il team avrà bisogno di:

- **Controllo del progetto:** il team ottiene dei task riguardanti il progetto, periodicamente, dal Professor A.Cortesi che ne effettua una valutazione e notifica il team sull'andamento dello svolgimento del progetto.
- **Consultazione:** per quanto riguarda la consultazione il team durante lo sviluppo del progetto avrà la possibilità di interfacciarsi con il Professor A.Spanò, per ottenere maggiori informazioni o argomenti non approfonditi a lezione, utili per lo sviluppo dell'applicazione.

2.4. Responsabilità di progetto

All'interno di questa sezione verranno elencati i vari ruoli assegnati ai membri del gruppo. Nel team i membri del gruppo svolgono tutti quanti le stesse mansioni, ovvero, quando si parlerà di sviluppo dell'applicazione, il team verrà riunito ed attraverso le conoscenze di tutti i membri verrà svolto ciascun task assieme. Tuttavia all'interno del team è stato scelto un:

- **Project manager**
 - Membro: Marco Pistollato
 - Responsabilità: si classifica come tutti gli altri membri del gruppo, però ha lo scopo di rispondere e mantenere un minimo di organizzazione all'interno del team. Le mansioni sono esattamente quelle degli altri membri del team e partecipa anch'esso allo sviluppo del progetto insieme al team.
 - Nota: all'interno del team non esiste un Project manager vero e proprio, ma tutti i membri sono allo stesso livello e ciascuno offre le sue competenze che possono aiutare il gruppo a sviluppare l'applicazione.

3. DESCRIZIONE DEI PROCESSI GESTIONALI

3.1. Obiettivi e priorità

- **OBIETTIVO**

L'obiettivo di questo **progetto** è la realizzazione di un'**applicazione Android Nativa** che vada ad estendere il sistema **"Families Share"** aggiungendo delle **Feature** non già presenti in esso. In particolare si vuole implementare la possibilità all'interno di un gruppo di effettuare la **CONDIVISIONE DI OGGETTI** (*E.g.* Giochi da Tavolo, Videogiochi, Devices, Libri, Carte, Skateboard, Monopattini, Palloni, Vestiti, Attrezzi per la casa... etc) attraverso dei **prestiti a tempo determinato o indeterminato**.

L'applicazione dovrà essere in grado di interfacciarsi al server **Node.JS** che si occuperà del lato **backend** e dell'interazione con il **database**.

Un **utente** che si trova all'interno di un gruppo e che **intende prestare** un oggetto potrà pubblicare un **annuncio di disponibilità** in un'apposita sezione. Durante la creazione dell'annuncio l'utente dovrà selezionare una **categoria** di appartenenza dell'oggetto, inserire una **descrizione** e caricare delle **foto** che lo rappresentino. Quando un altro utente appartenente al gruppo è interessato a un annuncio può richiedere il prestito proponendo un periodo utile e iniziando una **conversazione** con l'utente che intende prestare al fine di **accordarsi su quando e dove incontrarsi per il prestito**. Una volta trovato un accordo tra i due utenti **viene stabilita una data per la restituzione** e all'avvicinarsi di questa **il sistema notificherà i due utenti** che potranno proseguire la conversazione accordandosi su dove e quando trovarsi per la **restituzione**.

Un **utente** che è in cerca di un oggetto e desidera un prestito può inserire un **annuncio di richiesta** in un gruppo di cui fa parte e se uno degli altri utenti intende prestare un oggetto di quel tipo può rispondere all'annuncio con le stesse modalità di quanto scritto prima.

È possibile accordarsi per un **prestito a tempo indeterminato**. Questa funzionalità consente agli utenti che hanno oggetti che non usano più di sbarazzarsene senza buttarli via, **aiutando un'altra famiglia** che ne ha bisogno e favorendo la **"Second Hand Economy"** che

punta a **ridurre i consumi** attraverso il riutilizzo di oggetti non nuovi ma che sono ancora nelle condizioni di svolgere le funzionalità adibite ad esso.

La **Piattaforma non si assume nessuna responsabilità** per eventuali **smarrimenti, danni** agli oggetti o **mancate restituzioni**. Lo scopo dei **gruppi** è la creazione di una **cerchia di persone fidate** alle quali affidare i propri oggetti.

● **PRIORITÀ**

Le Priorità per lo sviluppo di una piattaforma efficace per lo svolgimento degli obiettivi di progetto sono:

- Un sistema di notifica in tempo reale per la pubblicazione di annunci e per le notifiche di restituzione
- Un'interfaccia intuitiva e semplice da usare che porti l'utente a utilizzare la piattaforma invece di una semplice app di messaggistica istantanea.
- La creazione di un annuncio deve essere semplice e veloce da creare.
- La gestione di immagini da salvare nel database.
- La gestione di una conversazione tra due utenti che si stanno accordando per un prestito tramite messaggi in tempo reale e uno storico delle conversazioni.
- Avvisare l'utente che la piattaforma si esonera dalle responsabilità sul trattamento degli oggetti e assicurarsi che tra i due utenti ci sia un rapporto di fiducia in modo da ridurre questi casi.
- Un sistema di notifica che ricordi per tempo la scadenza del prestito (*E.g.* 30 giorni prima, 7 giorni prima, 3 giorni prima, 1 giorno prima) e che inviti gli utenti a iniziare una conversazione per stabilire la data di restituzione

3.2. Assunzioni, dipendenze, vincoli

● **ASSUNZIONI**

Si assume che ogni membro del team sia responsabile e in grado di organizzare il proprio tempo da dedicare a questo progetto e agli altri corsi di studio al fine di completare gli

obiettivi di progetto entro le deadline stabilite. Si assume inoltre che gli studenti si organizzino autonomamente per l'acquisizione delle competenze necessarie, dei materiali e della strumentazione per portare a termine la varie parti del progetto.

● DIPENDENZE

Per il corretto svolgimento del progetto è necessario che i membri del gruppo abbiano o acquisiscono le seguenti competenze:

- Ambiente di sviluppo: **Android Studio** (introdotto nelle lezioni frontali dal Prof. Alvis Spanò ma che va comunque approfondito autonomamente)
- Linguaggio: **Java** (studiato nel corso di Programmazione a Oggetti del secondo anno)
- Database Relazionali: **SQL** (studiato nel corso di Basi di Dati)
- Backend: **Node.JS** (Introdotto nella lezione frontale con la Tutor Alessia Michela di Campi ma che va appreso autonomamente)
- Linguaggio: **Javascript** (che stiamo apprendendo nel corso libero di Linguaggi per la Rete)
- Sistema di Versioning: **Github** (visto col prof. Alvis Spanò e già usato per progetti precedenti)

● VINCOLI

Il team si impegna a Rispettare le Scadenze delle Deadline e a coordinarsi per finire in tempo ogni passo del progetto, a impegnarsi individualmente con lo studio e l'apprendimento delle competenze necessarie a soddisfare le dipendenze e a essere presente durante le lezioni del corso per aver chiari gli obiettivi del progetto.

3.3. Gestione dei rischi

- **Strategia per la prevenzione e la riduzione dei rischi**

Per prevenire e ridurre i rischi il team adotta tecniche di gestione dei rischi preventive e reattive di fronte ai problemi come spiegato dettagliatamente nella tabella dei rischi. Si prediligono tuttavia tecniche di prevenzione finalizzate a ridurre la probabilità che un certo rischio si verifichi invece di cercare di correggere i problemi dopo che sono già avvenuti.

- **Identificazione dei rischi**

- **Mancanza di competenze Tecniche**

Può verificarsi che uno o più membri del gruppo non abbiano molta confidenza con gli ambienti di sviluppo, i linguaggi usati e i paradigmi di programmazione necessari a creare la piattaforma.

- **Irresponsabilità dei singoli**

Può accadere che uno o più membri del gruppo per motivi personali o organizzativi non siano in grado di completare per tempo il lavoro assegnatogli o che non lo cominci nemmeno.

- **Scissione del gruppo o abbandono di singoli**

Può accadere che di fronte a problematiche scolastiche, personali o per conflitti con i colleghi uno o più membri del gruppo decidano di non proseguire con lo sviluppo del progetto.

- **Difficoltà nel Team-Working**

Vista l'inesperienza di noi studenti a lavorare in gruppo possono verificarsi problemi nella coordinazione e nella suddivisione del lavoro.

- **Problemi di salute**

Può accadere che uno o più membri si trovino impossibilitati a contribuire attivamente al progetto per problemi di salute (vista anche la stagione invernale e la situazione Covid-19).

- **Malfunzionamenti Tecnici**

Possono verificarsi problemi tecnici di malfunzionamento hardware delle macchine di lavoro o di problemi software degli ambienti di sviluppo (E.g. Compatibilità OS, Virtualizzazione Android, Server Node...)

- **Perdita Dati**

Possono, in seguito a problemi hardware o software, verificarsi perdite di dati (E.g. Documentazione, Codice Sorgente...etc) che aumenterebbero notevolmente i tempi di sviluppo.

- **Previsioni Sbagliate**

I tempi previsti per lo sviluppo potrebbero essere sottostimati causando ritardi nelle consegne comunicate al cliente.

- **Non conformità rispetto alle specifiche**

Una comunicazione scarsa col cliente potrebbe generare delle specifiche non chiare e l'ambiguità dell'italiano potrebbe portare allo sviluppo non conforme rispetto alle esigenze del cliente

- **Chiusura Spazi Universitari**

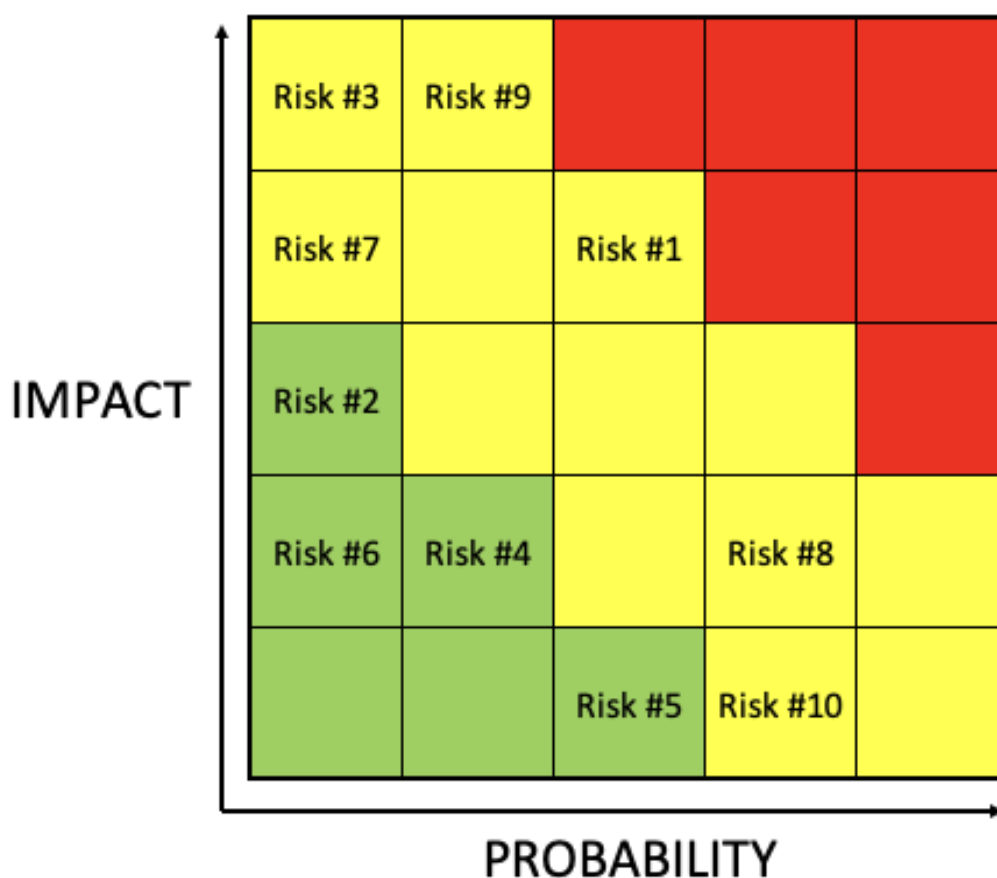
Vista la situazione della Pandemia Covid-19 potrebbe verificarsi la chiusura degli spazi universitari per spostarsi in modalità telematica.

● **Tabella dei Rischi**

ID	RISCHIO	CATEGORIA	PROBABILITÀ	IMPATTO	GESTIONE
1	Mancanza di competenze Tecniche	Risorse Umane	3	4	REATTIVA: Presenza alle lezioni tecniche e approfondimenti attraverso lo studio individuale
2	Irresponsabilità dei singoli	Risorse Umane	1	3	PREVENTIVA: Suddivisione puntuale del lavoro e orario di ritrovo settimanale fisso REATTIVA: Aiuto da parte di altri membri del gruppo
3	Scissione del gruppo o abbandono di singoli	Risorse Umane	1	5	PREVENTIVA: Comunicazione e Organizzazione strutturata REATTIVA: Riorganizzazione del lavoro in modo da portare a termine comunque il progetto nella sua interezza
4	Difficoltà nel Team-Working	Risorse Umane	2	2	PREVENTIVA: Suddivisione del lavoro ben strutturata
5	Problemi di salute	Risorse Umane	3	1	REATTIVA: Aumento di lavoro degli altri membri del team per sostituire la persona malata
6	Malfunzionamenti Macchine di lavoro	Tecnologico	1	2	PREVENTIVA: Manutenzione macchina di lavoro REATTIVA: Riparazione o Sostituzione macchina malfunzionante
7	Perdita Dati	Tecnologico	1	4	PREVENTIVA: Salvataggio di dati in Cloud (gitHub, Google Drive...) per minimizzare il rischio

8	Previsioni sbagliate	Requisiti	4	2	REATTIVA: Slittamento dei tempi di consegna o aumento di lavoro per concludere entro la deadline
9	Non conformità rispetto alle specifiche	Requisiti	2	5	PREVENTIVA: Analisi accurata del documento di specifica
10	Chiusura spazi universitari	Covid-19	4	1	REATTIVA: Incontri online (Discord)

- **Classificazione**



3.4. Meccanismi di monitoraggio e controllo

Per quanto riguarda la comunicazione interna del team utilizzeremo un gruppo Whatsapp per aggiornarci sull'avanzamento del progetto e per stabilire con più precisione la presenza ad un incontro, inoltre durante i ritrovi in presenza per le lezioni di università ci aggiorniamo sulla situazione del progetto. Per quanto riguarda la condivisione di documenti e codice sorgente si adotta nel primo caso un archivio Google Drive Condiviso (<https://drive.google.com/drive/folders/1J78RNh169IJpKmPhDe4LpvKCgHfTWoPw?usp=sharing>) e nel secondo un repository GitHub (https://github.com/giova239/Families_share-extension). La suddivisione dei ruoli fa fede a quanto indicato nel punto 2.4 del Piano di Progetto.

3.5. Pianificazione dello staff

Per un corretto sviluppo del progetto sono necessarie diverse Skills e Competenze, in particolare la conoscenza del Linguaggio Java e dei principi fondamentali della Programmazione a Oggetti, un buon livello di confidenza con l'IDE di Android Studio, una buona conoscenza del linguaggio Javascript, del meccanismo delle Callback e della programmazione asincrona al fine di usare correttamente NodeJS per creare un BackEnd che gestisca le richieste dei Client, Inoltre serve una buona conoscenza dei Database Relazionali e del linguaggio SQL e infine è necessario avere un buon livello di confidenza con Github per una condivisione efficace del sorgente e per la conservazione dei dati.

4. DESCRIZIONE DEI PROCESSI TECNICI

4.1. Metodi, strumenti e tecniche

In questo paragrafo vengono esposti i vari strumenti che il gruppo di lavoro ha in mente di utilizzare per realizzare il progetto.

- **Ambiente di sviluppo:** lo sviluppo dell'applicazione Android avverrà attraverso l'IDE Android Studio, il quale mette anche a disposizione un emulatore su cui poter testare l'applicazione prodotta. La versione 4.2 di Android Studio, rilasciata nel maggio 2021, sarà quella utilizzata dal gruppo di lavoro.
- **Linguaggi:** lo sviluppo dell'applicazione Android avverrà facendo uso del linguaggio di programmazione Java. Lo sviluppo delle interfacce, invece, vedrà l'utilizzo del linguaggio di modellazione e di specifica UML.
- **Strumentazione di lavoro:** come scritto precedentemente, l'applicazione Android verrà sviluppata all'interno dell'IDE Android Studio; quest'ultimo ha bisogno che le macchine sulle quali viene eseguito tale ambiente di sviluppo soddisfino i seguenti requisiti minimi: 8GB di RAM, 8GB di spazio su disco e risoluzione minima dello schermo pari a 1280*800. Le macchine di cui il gruppo si servirà saranno dotate dei sistemi operativi Windows 10 con architettura a 64-bit, Linux e macOS.
- **Strumentazione di test:** per le operazioni di testing verranno utilizzati smartphones dotati di sistema operativo Android 6.0 o versioni successive. Dato che l'obiettivo del progetto è quello di permettere a due o più utenti appartenenti ad un gruppo di condividere oggetti dove ci sarà una persona che rappresenterà il prestatore e un'altra che figurerà come acquirente, gli smartphones utilizzati per testare l'applicazione saranno almeno 2.
- **Standards di sviluppo:** si cercherà di rendere il codice modulare e scalabile in modo tale che la lettura del codice sia chiara e velocemente intuibile. Questo approccio permetterà anche una maggiore facilità nell'apportare modifiche in futuro. In parti del programma che potrebbero risultare di difficile comprensione oppure che si ritengono importanti, verranno inseriti opportuni commenti atti soprattutto a far capire il significato del codice a coloro che dovessero mantenerlo in futuro.



- **Linee guida:** per capire bene come muoversi all'interno dell'IDE Android Studio verrà consultata la documentazione ufficiale, disponibile al seguente link:
<https://developer.android.com/studio/intro>.

Per quanto riguarda lo sviluppo di codice all'interno del medesimo IDE, verrà esaminata un'altra documentazione ufficiale reperibile al seguente link:
<https://developer.android.com/reference/packages>.

4.2. Documentazione del software

La documentazione del software verrà realizzata dal gruppo di lavoro nel corso dello sviluppo del progetto, rispettando le consegne definite nella sezione 1.2.

Eventuali modifiche a tale documentazione verranno apportate seguendo la programmazione inserita all'interno della sezione 2.1.

4.3. Funzionalità di supporto al progetto

Di seguito vengono specificati gli strumenti di supporto al progetto che verranno utilizzati dal gruppo, utili ai membri del team per consentire loro di comunicare informazioni in modo rapido.

- **Pianificazione della qualità:** è importante controllare il codice man mano che viene sviluppato e tale operazione verrà compiuta da tutti i membri del team, in quanto si assume che nella scrittura dell'applicazione partecipino tutti attivamente fornendo un feedback costante ed in tempo reale; in fase di progettazione questa metodologia permetterà anche di avere diversi punti di vista su eventuali dubbi di implementazione che potrebbero nascere. Nel corso della scrittura del codice dell'applicazione sarà necessario effettuare una certa mole di test per verificare la bontà del lavoro prodotto e non trovarsi negli ultimi istanti a dover correggere parti consistenti del programma. La strumentazione per i test è stata indicata nella sezione 4.1.
- **Pianificazione della gestione delle configurazioni:** il codice sorgente verrà conservato all'interno del servizio di hosting GitHub che permetterà al Professor Cortesi Agostino, al Professor Spanò Alvise e alla responsabile di tutorato Di Campi Alessia Michela di controllare l'avanzamento del progetto. Il link al repository GitHub è presente nella sezione 3.4. All'interno del repository Git, il team lavorerà sul solo branch master senza andare creare altri branch eseguendo successivamente un merge, in quanto si è deciso che alla scrittura del codice parteciperanno tutti i membri del gruppo e quindi ci si concentrerà sul solo branch master. Google Drive verrà utilizzato dai componenti del gruppo per condividere la documentazione relativa al progetto redatta facendo uso del programma Google Documents; in questo modo ciascun utente potrà apportare le modifiche che ritiene necessarie (e che sono state prima concordate con il resto del gruppo) al documento quando vorrà, mostrando in tempo reale all'intero team le effettive correzioni apportate.

5. PIANIFICAZIONE

5.1. WBS (Work Breakdown Structure)

La WBS definisce la struttura del progetto e lo suddivide nelle diverse attività, scadenze e funzioni. Il gruppo ha individuato alcune funzioni con lo scopo di avere un riscontro costante per la durata del progetto, e per questo sono comprese dei confronti settimanali.

FUNZIONI:

- Project Management (**A**): funzione presente durante lo sviluppo di tutto il progetto, il carico di lavoro sarà organizzato in base ai risultati e alle scadenze adattandosi alle necessità (come descritto nel paragrafo 2.1 il gruppo utilizzerà una metodologia agile). Si prevede anche una possibile riorganizzazione del lavoro in base a questi ultimi fattori.
- Apprendimento delle tecnologie (**B**): comporta lo studio e lo sviluppo delle capacità per utilizzare linguaggi e strumenti atti alla realizzazione del progetto (in particolare lo sviluppo in Android e la gestione del progetto)
- Documentazione (**C**): deve rispettare le scadenze prefissate e per questo motivo sarà un'attività parallela allo sviluppo delle funzionalità aggiuntive dell'applicazione (fatta eccezione per il piano di progetto che verrà consegnato prima), tutto il gruppo parteciperà alla stesura della documentazione per motivi organizzativi e di conoscenza comune dell'andamento del progetto
 - Piano di progetto (**C.1**): descrizione degli obiettivi di progetto e gli elementi necessari per il loro raggiungimento.
 - Documento di analisi e specifica (**C.2**): descrive i servizi richieste ed i vincoli dati dal committente.
 - Piano di testing (**C.3**): contiene gli obiettivi, le procedure e le tempistiche per effettuare le verifiche.
 - Documento di progettazione (**C.4**): definizione dell'architettura di sistema
- Controllo qualità (**D**): l'applicazione deve essere in grado di compiere le funzioni descritte nel piano di progetto
- Testing (**E**): è la fase in cui verranno effettuate le prove necessarie per assicurare il corretto funzionamento dell'applicazione

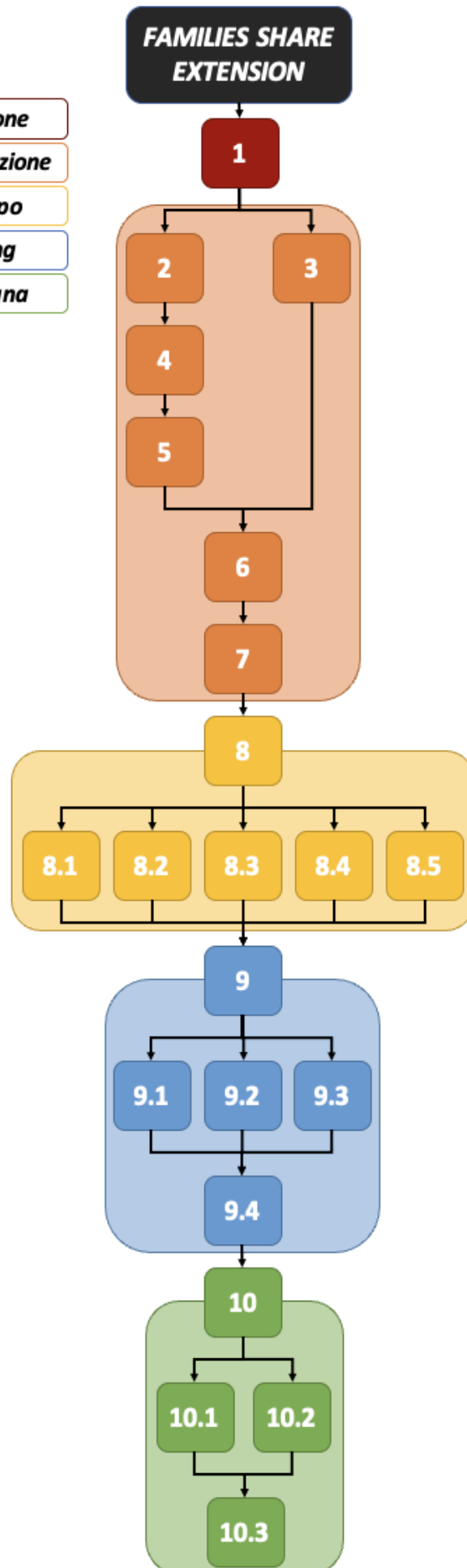
- Unit (**E.1**): test delle singole parti
- Integration (**E.2**): test dell'integrazione della singola parte con il resto dell'applicazione (compresa la parte già esistente).

ATTIVITÀ:

- Ideazione del progetto (**1**): comprende le idee per implementare le funzionalità aggiuntive
- Analisi di fattibilità (**2**): vengono analizzati gli obiettivi e si verifica che sia effettivamente possibile implementare le idee
- Definizione della struttura del team (**3**): vengono definiti i ruoli di ogni membro e suddiviso il lavoro
- Analisi dei requisiti (**4**): analisi dei requisiti tecnici imposti dal committente e dalla struttura già esistente
- Analisi e gestione dei rischi(**5**): vengono presi in considerazione eventuali problemi che si potranno presentare durante lo sviluppo dell'app
- Pianificazione delle attività (**6**): organizzazione delle attività e previsione delle tempistiche, assegnazione di parti ad ogni individuo
- Costruzione WBS, Gantt e Pert (**7**): definizione dell'ordine cronologico dei vari task e la loro durata, si tiene conto delle scadenze prefissate
- Progettazione software (**8**): sviluppo della prima versione funzionante dell'app con le funzioni di base
 - gestione di immagini da salvare nel database (**8.1**)
 - creazione di un'interfaccia grafica (**8.2**)
 - gestione della creazione di un annuncio (**8.3**)
 - gestione delle conversazioni tra utenti (**8.4**)
 - implementazione sistema di notifiche (**8.5**)
- Sviluppo della versione definitiva (**9**)
 - miglioramento delle funzionalità (**9.1**)
 - semplificazione e miglioramento interfaccia (**9.2**)
 - semplificare la creazione di un annuncio (**9.3**)



- fase di test **(9.4)**
- Controllo e consegna **(10)**
 - fase di test aggiuntiva **(10.1)**
 - controllo della documentazione finale **(10.2)**
 - consegna **(10.3)**



5.2. Dipendenze

Le dipendenze si riferiscono alle attività strettamente legate tra loro e che quindi non possono essere svolte in qualsiasi momento ma dipendono dalla riuscita di altre, questo perché hanno bisogno di alcune parti del progetto terminate o definite in modo più chiaro.

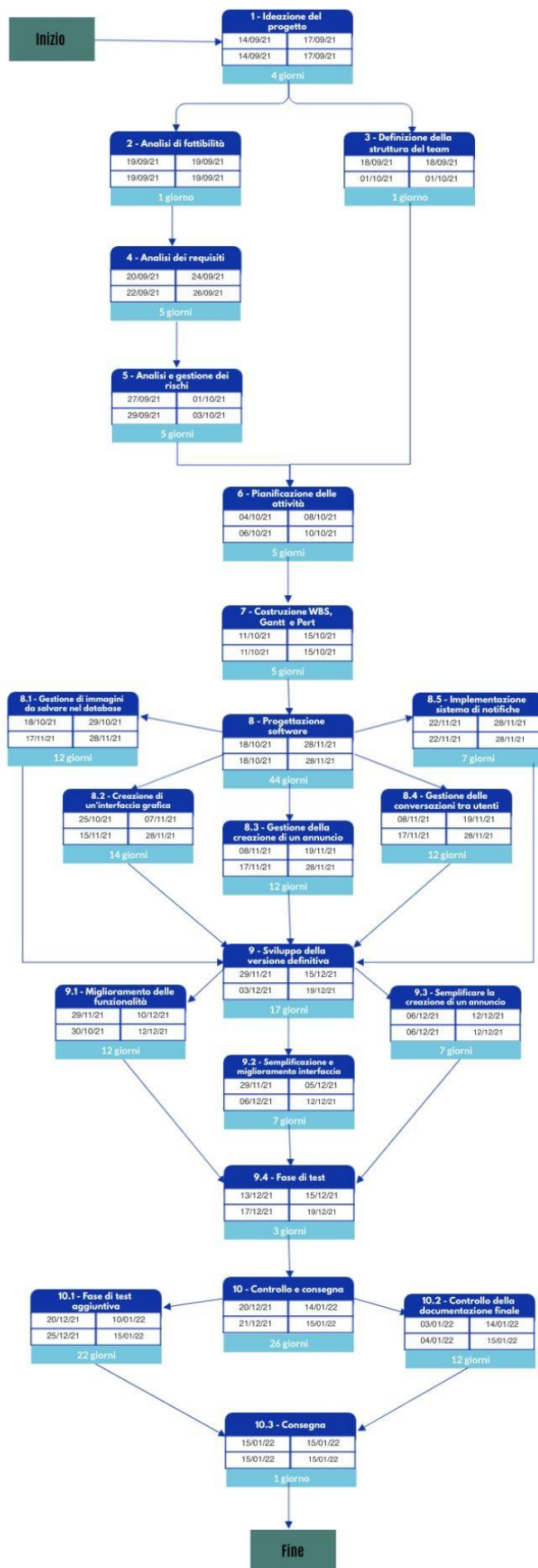
Le attività che prevedono una dipendenza sono obbligatoriamente svolte in sequenza, al contrario nel caso in cui non ci siano dipendenze possono essere svolte in parallelo (gestendo comunque la forza lavoro che non può essere infinita). Il parallelismo permette nonostante i limiti organizzativi di lavorare su diverse attività contemporaneamente per completare il lavoro con più efficienza.

ID	Dipendenze	Durata (gg)	ES	EF	LS	LF
1	-	4	14/09/21	17/09/21	17/09/21	14/09/21
2	1	1	19/09/21	19/09/21	19/09/21	19/09/21
3	1	1	18/09/21	18/09/21	01/10/21	01/10/21
4	2	5	20/09/21	24/09/21	26/09/21	22/09/21
5	4	5	27/09/21	01/10/21	03/10/21	29/09/21
6	5,3	5	04/10/21	08/10/21	10/10/21	06/10/21
7	6	5	11/10/21	15/10/21	15/10/21	11/10/21
8	7	44	18/10/21	28/11/21	28/11/21	18/10/21
8.1	7	12	18/10/21	29/10/21	28/11/21	17/11/21
8.2	7	14	25/10/21	07/11/21	28/11/21	15/11/21
8.3	7	12	08/11/21	19/11/21	28/11/21	17/11/21
8.4	7	12	08/11/21	19/11/21	28/11/21	17/11/21
8.5	7	7	22/11/21	28/11/21	28/11/21	22/11/21
9	8.1, 8.2, 8.3, 8.4, 8.5	17	29/11/21	15/12/21	19/12/21	03/12/21
9.1	8	12	29/11/21	10/12/21	12/12/21	30/12/21
9.2	8	7	29/11/21	05/12/21	12/12/21	06/12/21
9.3	8	7	06/12/12	12/12/21	12/12/21	06/12/21
9.4	9.1, 9.2, 9.3	3	13/12/21	15/12/21	19/12/21	17/12/21
10	9.4	26	20/12/21	14/01/22	15/01/22	21/12/21
10.1	9.4	22	20/12/21	10/01/22	15/01/22	25/01/22
10.2	9.4	12	03/01/22	14/01/22	15/01/22	04/01/22
10.3	10.1, 10.2	1	15/01/22	15/01/22	15/01/22	15/01/22

5.3. Risorse necessarie

Per la realizzazione del progetto si possono individuare diverse categorie di risorse necessarie:

- Risorse Umane: i membri del gruppo, sono incluse le competenze tecniche (come l'utilizzo degli strumenti necessari alla documentazione e lo sviluppo di software).
- Risorse Temporal: le ore di lavoro necessarie al completamento del progetto, comprese quelle dedicate alla gestione del tempo in modo da poter svolgere il lavoro rispettando le scadenze.
- Risorse Documentative: fanno riferimento a tutta la documentazione di riferimento (riguardo al progetto e al software/librerie che verranno utilizzate) e alla documentazione che verrà prodotta durante lo svolgimento del progetto, ne fanno parte anche i requisiti di progetto
- Risorse Software: comprende tutto il software che verrà utilizzato, dallo sviluppo (es. con Android Studio) alla stesura della documentazione (Google Drive o Office), la condivisione e la gestione del codice (git/github), la gestione del tempo e del lavoro (GanttProject, Google Drive), per finire software per la comunicazione tra i membri del gruppo (Whatsapp/Telegram)
- Risorse Hardware: tutta la strumentazione hardware necessaria, come ad esempio laptop, smartphone e dispositivi android dedicati ai test dell'applicazione



5.4. Allocazione del budget e delle risorse

Non sono previste spese economiche per la realizzazione del progetto in quanto gli strumenti che verranno utilizzati sono già in possesso di ogni membro del team, il software utilizzato invece sarà gratuito (Google Drive, Whatsapp, Telegram, Github, GanttProject, Android Studio).

Si potrebbe assumere di quantificare economicamente il lavoro di ogni singolo membro del team in Euro per ora di lavoro. Sapendo che i meeting di lunedì e giovedì saranno complessivamente di 2 ore, più l'avanzamento individuale del progetto dal singolo membro del team con ore arbitrarie. Sapendo che dal 21 settembre al 28 dicembre trascorrono all'incirca 13 settimane e definendo all'incirca 342 euro per ogni settimana di lavoro si può stimare che il progetto avrà un costo base di €4.446.

Prendendo in considerazione la scelta della metodologia Agile vengono contate solamente le ore di lavoro in gruppo, mentre per le ore individuali la gestione del tempo varia in base ad ogni membro del gruppo in base alle sue possibilità.

Gruppo	Settore	Prezzo	Totale parziale
Ore contate	Uomo-ora*4	€342/settimana	€4.446
Documenti	Piano Progetto	€250	€1.900
	Documento di analisi e specifica	€200	
	Piano di testing	€350	
	Documento di progettazione	€400	
	Versione 1.0	€300	
	Versione 1.1	€400	
Features	Gestione di immagini da salvare nel database	€1.150	€3.300
	Creazione di un'interfaccia grafica	€550	
	Gestione della creazione di un annuncio	€400	
	Gestione delle conversazioni tra utenti	€750	
	Implementazione sistema di notifiche	€450	
Grafica	Grafica applicazione	€450	€400
Totale finale (IVA esclusa)			€10.046

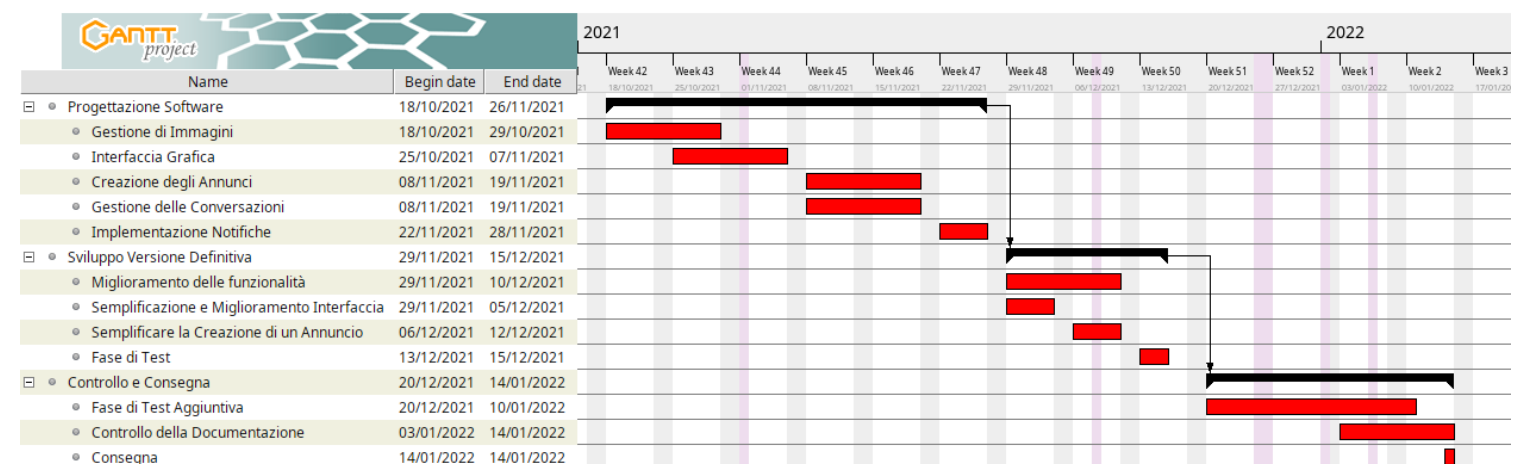
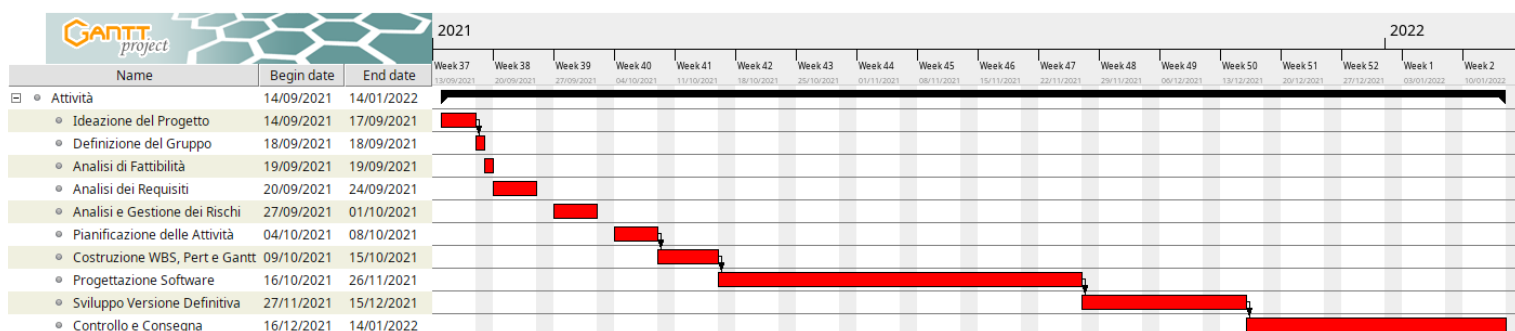
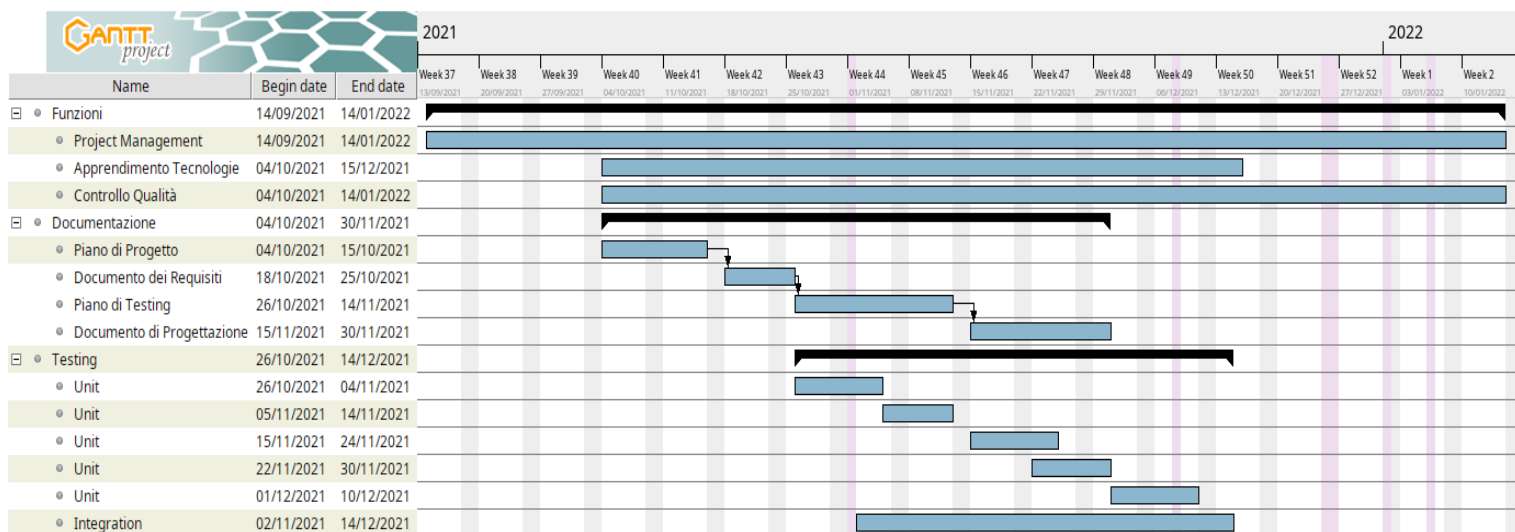
5.5. Pianificazione

La pianificazione del progetto prevede di fissare le scadenze che dovranno essere rispettate, per quelle riguardanti la documentazione ci sono date ben definite, al contrario sarà compito nostro gestire eventuali deadlines aggiuntive. Il compito di queste scadenze è quello di migliorare la possibilità di riuscita del progetto, suddividendolo in parti più piccole; di seguito sono riportate le deadline assegnate:

- Piano di Progetto (15/10/2021)
- Documento dei Requisiti (25/10/2021)
- Piano di Testing (14/11/2021)
- Documento di Progettazione (30/11/2021)
- Versione 1.0 del codice sorgente (15/12/2021)
- Versione 1.1. del codice e allineamento documentazione (15/01/2022)

Diagramma di Gantt:

Il diagramma di Gantt permette di avere più chiara la distribuzione del lavoro nel tempo, in modo da poter gestire con più precisione le attività necessarie al compimento dell'obiettivo.



Distribuzione attività:

Le attività indicate nelle sezioni 5.1 e 5.2 sono state puntualmente suddivise tra i membri del team per favorire una migliore schedulazione dei processi necessari al raggiungimento del risultato finale di progetto.

La tabella che segue indica le mansioni che i singoli membri del team dovranno svolgere su base settimanale:

SETTIMANA DEL PROGETTO: primo lunedì di ogni mese:	SETTEMBRE				OTTOBRE				NOVEMBRE					DICEMBRE				GENNAIO				
	6	13	20	27	4	11	18	25	1	8	15	22	29	6	13	20	27	3	10	15	24	31
MARCO PISTOLLATO	1	4	5	6	7	8.1				8.3	8.5	9.1	9.4			10.1						
	2						8.2			8.4		9.2					10.2			10.3		
	3							8				9				10						
GIOVANNI STEVANATO	1	4	5	6	7	8.1				8.3	8.5	9.1	9.4			10.1						
	2						8.2			8.4		9.3					10.2			10.3		
	3							8				9				10						
MARCO QUARTA	1	4	5	6	7	8.1				8.3	8.5	9.1	9.4			10.1						
	2						8.2			8.4		9.3					10.2			10.3		
	3							8				9				10						
ALEXANDRU PETRONEL CONDURACHE	1	4	5	6	7	8.1				8.3	8.5	9.1	9.4			10.1						
	2						8.2			8.4		9.2					10.2			10.3		
	3							8				9				10						

6. RIFERIMENTI

Per creare questo documento sono stati utilizzati come riferimento:

- “piano di progetto” di alcuni gruppi degli anni passati;
- materiale messo a disposizione dal professore.