11/10/2021

Università
Ca'Foscari
Venezia

Corso:

Ingegneria del Software [CT0090] 2021/2022

Team:

Unità dei mandrilli

Membri:

Andrea Gentilini 880141
Runjie Xia 879779
Marco Di Fresco 881286
Pierluigi Marchioro 881929

PIANO DI PROGETTO

VERSIONE 1.1

INDICE

1. INTRODUZIONE

1.1. OVERVIEW DEL PROGETTO

Il progetto consiste nello sviluppo di una estensione per la già esistente applicazione "Families Share". Questa offre una soluzione dal basso verso l'alto per l'equilibrio tra lavoro e vita privata supportando le famiglie con assistenza all'infanzia, consulenza per i genitori e attività di doposcuola.

L'estensione da noi proposta mira a fornire le principali funzionalità per la gestione e l'utilizzo di un marketplace dove ogni utente ha la possibilità di effettuare prestiti, scambi o donazioni di oggetti.

I principali utilizzatori dell'estensione proposta saranno tutte quelle famiglie che si trovano in una particolare situazione economica che non consente loro di effettuare spese fuori budget. Loro saranno la "domanda". L' "offerta" invece è composta da tutte quelle famiglie che vogliono liberarsi di oggetti che non utilizzano più. Così facendo possiamo definire i servizi che il progetto offre come il perfetto punto di incontro tra domanda e offerta.

1.2. DELIVERABLES DEL PROGETTO

Le scadenze per questo progetto sono state programmate nel seguente modo:

- Definizione dei gruppi di progetto (22/09/2021)
- Proposta iniziale (01/10/2021)
- 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 e allineamento documentazione (15/01/2022)

1.3. EVOLUZIONE DEL PROGETTO

Il progetto sarà composto da 3 parti fondamentali:

- Front-End, la parte grafica del progetto verrà implementata tramite “React”, cioè una libreria open-source in JavaScript per la creazione di interfacce utente.

- Back-End, l’implementazione del back-end del progetto invece avverrà in Node.js, cioè un runtime system multipiattaforma orientato agli eventi per l'esecuzione di codice JavaScript.

- Database, la logica della base di dati invece verrà implementata tramite MongoDB, cioè un DBMS non relazionale classificato come un database di tipo NoSQL.

1.4. MATERIALE DI RIFERIMENTO

Durante l’interezza del progetto il team sarà dipendente dai materiali didattici presenti sulla piattaforma Moodle del corso.

Per l’implementazione del progetto il nostro team partirà dal codice sorgente di Families Share già presente in GitHub al seguente link:
https://github.com/vilabs/Families_Share-platform

Il sito developer.android.com sarà, per il team, una guida per il design visivo, dinamico e interattivo di Android.

Infine le principali informazioni riguardanti l’applicazione originale sono disponibili al seguente link:
https://www.families-share-toolkit.eu/it/

1.5. DEFINIZIONI ED ABBREVIAZIONI

- **Estensione**: componente di un software che aggiunge delle funzionalità.
- **Marketplace**: sono in generale il luogo in cui avvengono degli scambi commerciali
- **GitHub**: GitHub è un servizio di hosting per progetti software.
- **Open-Source**: è un software rilasciato con una licenza in cui il detentore del copyright concede agli utenti i diritti di utilizzare, studiare, modificare e distribuire il software e il suo codice sorgente.
- **Runtime System**: è un software che fornisce i servizi necessari all'esecuzione di un programma.
- **DBMS**: (Database Management System) è un sistema software progettato per consentire creazione, manipolazione e interrogazione efficiente di database.
- **IDE**: un ambiente di sviluppo integrato (in inglese Integrated development environment), è un software che supporta i programmatori nello sviluppo del codice di un

programma
- **Android**: sistema operativo per dispositivi mobili.
- **JavaScript**: linguaggio di programmazione.
- **JetBrains**: azienda di sviluppo software.
- **React.js**: una libreria JavaScript per la creazione di interfacce utente.
- **Node.js**: un framework usato per scrivere applicazioni in JavaScript.
- **Framework**: piattaforma che funge da strato intermedio tra un sistema operativo e il software che lo utilizza.
- **Jest**: framework di testing per il linguaggio JavaScript
- **JSON**: un formato testuale per la strutturazione di dati.
- **WebStorm**: un IDE prodotto da JetBrains.
- **Front-end**: la parte visibile all'utente di un programma con cui può egli interagire.
- **Back-end**: la parte che permette l'effettivo funzionamento delle interazioni fra utente e programma.
- **Windows** : sistema operativo prodotto da Microsoft Corporation.
- **macOS**: sistema operativo sviluppato da Apple Inc.
- **RAM**: un componente hardware di un pc.
- **Main Branch**: ramo principale di lavoro all'interno di GitHub
- **Branch**: ramo di lavoro all'interno di GitHub
- **Pull Request**: è una nostra richiesta, fatta all'autore originale di un software o di un documento, di includere le nostre personali modifiche al suo progetto.
- **Backup**: creazione di una copia di file
- **ES (earliest start time)**: indica il giorno minimo di inizio dell'attività a partire dal minimo tempo necessario per le attività che precedono.
- **EF (earliest start time)**: dato ES e la durata dell'attività indica il minimo giorno in cui l'attività può terminare.
- **LF (latest finish time)**: il giorno massimo in cui l'attività deve finire senza creare ritardo alle attività che dipendono da lui.
- **LS (latest star time)**: dato LF e la durata dell'attività indica il giorno massimo in cui l'attività deve iniziare senza provocare ritardi alle attività che dipendono da lui.
- **ClickUp**: strumento molto potente e versatile di project management e gestione delle attività, che può integrarsi con vari altri applicativi tra cui GitHub.
- **TDD - Test Driven Development**: modello di sviluppo del software che prevede che la stesura dei test automatici avvenga prima di quella del software che deve essere sottoposto a test, e che lo sviluppo del software applicativo sia orientato esclusivamente all'obiettivo di passare i test automatici precedentemente predisposti
- **API**: Application Programming Interface, indica un insieme di procedure (in genere raggruppate per strumenti specifici) o interfacce atte all'espletamento di un dato compito
- **Stack tecnologico**: insieme di tecnologie che permettono la realizzazione di un dato progetto

2. ORGANIZZAZIONE DEL PROGETTO

2.1 MODELLO DEL PROCESSO

Lunedì	Mercoledì	Giovedì	Venerdì
Mattino	Pomeriggio	Giornata intera	Pomeriggio
Incontro per organizzare al meglio le attività e gli obiettivi settimanali	Lavoro al progetto in presenza così da poter essere più efficienti nel caso di problematiche nello sviluppo	Lavoro al progetto in presenza così da poter essere più efficienti nel caso di problematiche nello sviluppo	Controllare il lavoro svolto durante la settimana e se necessario correggerlo. Upload delle attività svolte ed aggiornamento della documentazione

Per organizzare al meglio le attività il team si avvarrà del sito web **ClickUp** (ClickUp™ | One app to replace them all) nel quale sarà disponibile consultare lo stato di avanzamento del progetto in tempo reale. Ciò sarà possibile grazie alle tabelle, diagrammi e altre funzioni di collaborazione e project management che l'applicazione mette a disposizione dell'utente.

2.2 STRUTTURA ORGANIZZATIVA

Ci sono 3 tipologie di strutture organizzative:

- **Funzionale**
  - Le risorse vengono raggruppate in aree di specializzazione in funzione delle competenze, supervisionate da un manager funzionale. Il project manager ha un potere molto limitato perché deve confrontarsi con i responsabili di funzione per tutte le scelte riguardanti il budget e l'allocazione delle risorse.
- **Per progetto**
  - Il project manager ha un potere ed un'autorità assoluta su tutti, ha completo accesso al budget così da poterlo gestire arbitrariamente. Il suo ruolo principale nella gestione di un progetto è la coordinazione del personale. Le risorse vengono assegnate ad un progetto con dei ruoli chiari e ben definiti per poi, alla conclusione, cambiare attività così da poter rendere il lavoro più dinamico
- **A matrice**
  - Consente di mediare gli aspetti positivi dell'organizzazione funzionale e a progetti diminuendo gli aspetti negativi. A capo dei progetti c'è un manager funzionale che gestisce i project manager. Ciò ha come vantaggio la condivisione di conoscenze ed informazioni utili fra i vari progetti.

Come gruppo abbiamo deciso di optare per una **struttura organizzativa per progetto** per perseguire i seguenti obiettivi:

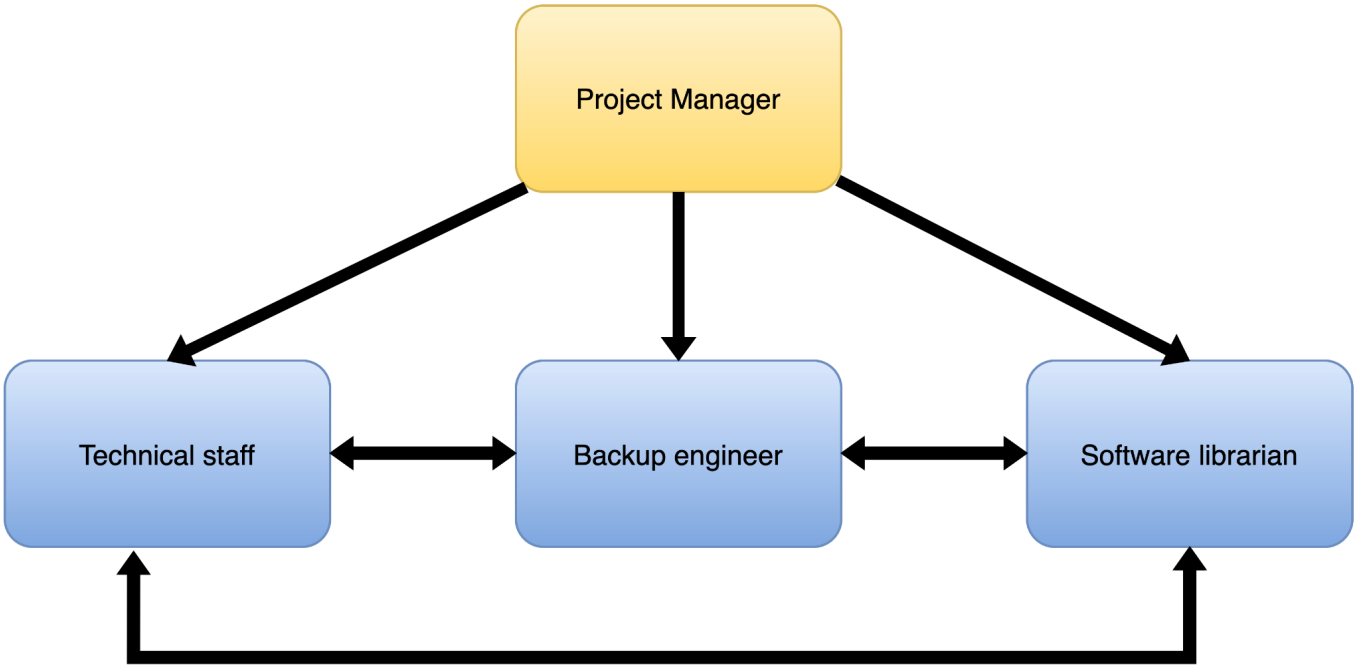
- **Flessibilità e rapidità decisionale**: non avendo un manager funzionale c'è un contatto diretto tra project manager e risorse.
- **Chiarezza ruoli**: nel caso di problematiche specifiche, la risorsa designata saprà prontamente come gestire la situazione nel modo più rapido ed efficace.

Come **tipologia di team** si è optato per un **controllo decentralizzato**. Tale scelta è stata effettuata perché:

- si necessita la presenza di un leader permanente che coordini il lavoro ma che non prenda scelte importanti da solo;
- risoluzione dei problemi svolta in gruppo ma lo sviluppo designato a chi più specializzato

Tale scelta permette di avere una **comunicazione orizzontale** tra tutti i membri del team.

Qui viene riportato il grafico con le specifiche dei livelli e dei soggetti che fanno parte del team di sviluppo del progetto.



2.3 INTERFACCE ORGANIZZATIVE

Durante lo svolgimento del plugin per Families Share, il team necessita di interfacciarsi con altre entità con funzione di:

- **Supervisione e feedback**
  - il team si interfaccerà con il Professor A. Cortesi così da avere un secondo parere sullo svolgimento dell'elaborato.
- **Consultazione codice**
  - il team si interfaccerà con il Professor A. Spanò.

2.4 RESPONSABILITÀ DEL PROGETTO

Qui sono elencati i vari ruoli assegnati ai membri del team con relativi compiti e responsabilità. Questi sono compiti specifici che caratterizzano maggiormente ciascun membro del team, ma ognuno sarà in grado di sostituire chiunque altro membro nel caso egli sia impossibilitato ad adempiere il proprio ruolo, per sviluppare l'applicativo e per stilare i documenti relativi al progetto

- **Project manager**:
    - Membro: di Fresco Marco;
    - Responsabilità: definisce gli obiettivi e coordina il team. Al compimento delle attività ha il compito di consegnare i deliverables entro le date previste.
- **Technical staff**:
    - Membro: Marchioro Pierluigi;
    - Responsabilità: conduce l'analisi e lo sviluppo del progetto, avrà il compito di confrontarsi con il software librarian così da poter integrare le possibili soluzioni da esso fornite. Dovrà anche monitorare il codice degli altri sviluppatori.
- **Backup engineer**:
    - Membro: Gentilini Andrea;
    - Responsabilità: supporta il project manager ed è responsabile della validazione del codice ovvero gestirà tutte le pull request su github.
- **Software librarian**:
    - Membro: Xia Runjie;
    - Responsabilità: mantiene la documentazione aggiornata ed espone i possibili problemi del software. Ha anche il compito di ideare e sviluppare delle soluzioni ai problemi lavorando fianco a fianco al technical staff.

Tutti i membri del team hanno un ruolo secondario da **Technical staff** essendo comunque un gruppo di 4 persone tutti devono partecipare attivamente allo sviluppo del codice affinché le tempistiche vengano rispettate

3. DESCRIZIONE DEI PROCESSI GESTIONALI

3.1. OBIETTIVI E PRIORITÀ

- **Obiettivo:** l'obiettivo principale consiste nel creare un'estensione per l' applicazione Families Share già presente nel Play Store.

  L' idea del nostro team è quella di offrire un servizio aggiuntivo (di scambio, prestito o donazione di oggetti) che riesca a risultare utile per molti utenti della piattaforma.

  È necessario stabilire quali utenti saranno disposti ad utilizzare questo "marketplace" e dare loro la possibilità di mettere in vetrina i propri oggetti. Inoltre si rende necessario creare una piattaforma tale da favorire la comunicazione tra il soggetto offerente e il soggetto richiedente in modo che la transazione possa avvenire senza problemi o difficoltà.

  Il primo obbiettivo sarà quello di creare una prima versione dell'estensione (demo o 1.0) contenente le principali funzionalità, una volta ottenuta questa prima versione il team potrà proseguire con lo sviluppo di una versione più avanzata (1.1) che andrà ad aggiungere funzionalità e dettagli grafici migliorati.

  L'obiettivo si può considerare raggiunto nel momento in cui gli utenti usufruiranno della piattaforma in maniera continuativa e con un adeguato livello di soddisfazione.

- **Priorità:** a livello software, il team dovrà necessariamente occuparsi della gestione delle eccezioni in modo da evitare malfunzionamenti che porterebbero ad un fallimento dell'applicazione e ad una cattiva esperienza per l'utente. L'applicazione dovrà supportare tutte le funzionalità in maniera efficiente e fluida.

  A livello di interfaccia e user experience, il nostro team prenderà in considerazione le principali euristiche sulla usabilità di un sistema informatico per lo sviluppo dell'interfaccia utente. Questo in modo da garantire un buon livello di usabilità e un'ottima esperienza per l'utente.

3.2. ASSUNZIONI, DIPENDENZE, VINCOLI

- **Assunzioni:** è necessario assumere che ogni membro del team sia responsabile, diligente e che operi nel bene comune del team che sia autonomo nella gestione del suo tempo per il completamento degli obiettivi settimanali. Assumiamo inoltre che ogni soggetto del gruppo partecipi in maniera equilibrata al completamento del progetto e che copra prontamente le lacune riguardanti le nuove tecnologie che ci è stato chiesto di utilizzare.

- **Dipendenze:** per la riuscita del progetto è necessaria la conoscenza del linguaggio JavaScript, della struttura dei database NoSQL, dell'utilizzo di framework come Node.js e React e le conoscenze base per poter utilizzare GitHub. Nel caso di Node.js, esso verrà parzialmente appreso attraverso le lezioni di approfondimento frontali, per React e MongoDB invece bisognerà effettuare una preparazione da autodidatta.

- **Vincoli:** i vincoli principali che ci vengono imposti saranno i giorni di consegna dei lavori, in modo da evitare ritardi a cascata su consegne successive. Sarà critico non consegnare documenti non controllati, superficiali, e non intestati in maniera corretta. Cercare (possibilmente tutti gli elementi del team o almeno una persona) di essere sempre a lezione in modo da rimanere sempre aggiornati sulle ultime comunicazioni e per poter chiarire eventuali dubbi riguardo lo sviluppo.

3.3. GESTIONE DEI RISCHI

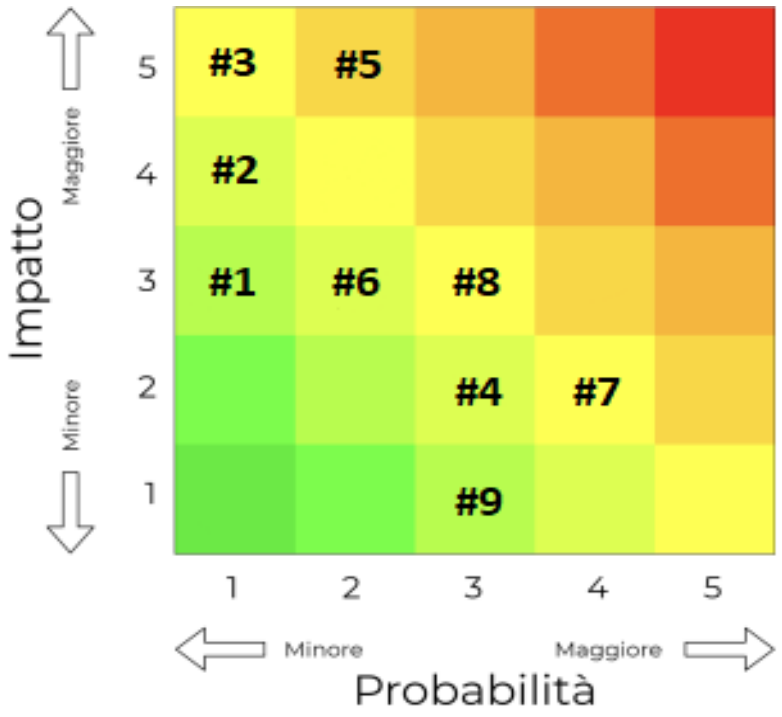
Identificazione: per evitare ritardi o addirittura la cancellazione del progetto è necessario identificare i rischi, ecco un elenco di eventuali rischi da dover evitare:

- **Mancato adempimento dei task assegnati:** rischio causato dal fatto che un membro del team non riesca (per colpa o per dolo) ad adempiere alle sue mansioni.
- **Incapacità a collaborare:** nel caso in cui tra i membri del team si crei una situazione di ostilità tale da impedire una efficiente collaborazione.
- **Disfacimento del team o dei suoi elementi:** se le controversie all'interno del team si accentuano in maniera troppo grave. Per questo motivo, è importante chiarire sin da subito qualsiasi diverbio tra i membri in modo da lavorare in maniera più coesa possibile.
- **Impossibilità a lavorare:** impossibilità di un soggetto di lavorare e portare a termine i propri task. Situazione causata da un qualsiasi fattore esterno tipo salute, famiglia, ecc…
- **Perdita dati:** rischio causato da malfunzionamenti/problemi tecnici che causano la perdita di informazioni vitali per il progetto intero. Per superare questo problema vengono inserite determinate regole per la creazione di backup.
- **Malfunzionamento macchine:** situazione creata da uno o più guasti alle macchine utilizzate dai membri del gruppo per lo sviluppo del progetto.
- **Scarsa conoscenza tecnica:** rischio molto probabile dato il fatto che nessun membro del team conosce le tecnologie che verranno utilizzate. E' un rischio poco impattante siccome con materiali online di vario genere si possono sopperire tutte le lacune.
- **Previsione sbagliata:** nel caso in cui lo sviluppo del progetto stia seguendo una strada sbagliata/diversa da quella precedentemente approvata.
- **Chiusura luoghi fisici di lavoro:** dovuta a nuovi regolamenti/decreti messi in atto per la salvaguardia della popolazione dall'emergenza COVID-19. La soluzione a questo è lo spostamento degli spazi di lavoro da fisici a online.

Tabella dei rischi:

ID	Rischio	Probabilità	Impatto	Azione
1	Mancato adempimento dei task assegnati	1	3	Richiamo da parte del team ed eventuale monitoraggio del lavoro
2	Incapacità a collaborare	1	4	Richiamo da parte del team con possibile espulsione del membro non collaborativo
3	Disfacimento del team o dei suoi elementi	1	5	Riedificazione del team con elementi rimasti
4	Impossibilità a lavorare (problemi di salute e familiari)	3	2	Ripartizione temporanea dell'attività del membro mancante
5	Perdita dati	2	5	Eseguire il rollback all'ultimo backup valido
6	Malfunzionamento macchine	2	3	Manutenzione e eventuale sostituzione temporanea
7	Scarsa conoscenza tecnica	4	2	Studio approfondito e ricerca della documentazione adeguata
8	Previsione sbagliata	3	3	Ricalcolo delle tempistiche e adattamento del codice
9	Chiusura luoghi fisici di lavoro	3	1	Sostituzione modalità di incontro

Classificazione:



3.4. MECCANISMI DI MONITORAGGIO E CONTROLLO

I nostri canali di comunicazione principali saranno WhatsApp e Discord, mentre per la condivisione della documentazione e del codice verranno utilizzati GitHub, per la condivisione del codice sorgente, e Google Drive, per la condivisione dei documenti. Inoltre saranno previsti una serie di incontri settimanali in modo da confrontarsi sul lavoro svolto, quello da svolgere ed eventuali problemi da risolvere. La divisione dei compiti e i relativi controlli fanno riferimento al punto 2.4 del presente documento.

3.5. PIANIFICAZIONE DELLO STAFF

Per poter sviluppare adeguatamente questo progetto, sono necessarie diverse conoscenze esterne al corso, la prima tra tutte è l'utilizzo di Node.js come strumento per il back-end. Inoltre è importante conoscere la struttura dei database NoSQL come MongoDB e saper utilizzare in maniera efficace ed efficiente l'ambiente di sviluppo con il quale abbiamo deciso di sviluppare l'applicazione (WebStorm - https://www.jetbrains.com/webstorm/).

Saranno infine necessarie le conoscenze base di Github come software per il versionamento di una produzione.

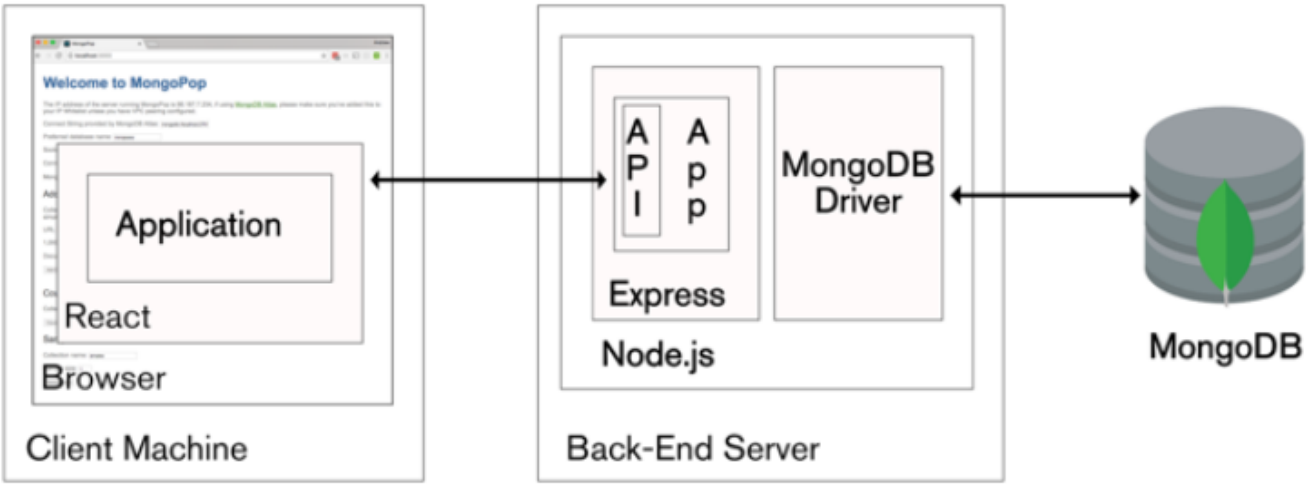
4. DESCRIZIONE DEI PROCESSI TECNICI

4.1 METODI, STRUMENTI E TECNICHE

In questa sezione verranno mostrati i processi tecnici e la strumentazione utilizzata nei vari processi, è necessario definirli accuratamente per facilitare il team ad adottare le misure fisse all'interno del progetto.

La maggior parte delle decisioni tecniche è stata presa a seguito di un'approfondita analisi della base di codice e della documentazione ufficiale di Families Share. Lo stack tecnologico utilizzato all'interno di tale applicazione è composto da React.js per il front-end, Node.js per il back-end, MongoDB per il database e Jest come framework di testing. Per questo motivo, dato che il nostro progetto è un'estensione dell'applicazione Families Share, il nostro team adotterà il suo stesso stack tecnologico.

L'immagine seguente mostra com'è fatta, dal punto di vista architetturale e tecnologico, l'applicazione Families Share.



- **Ambiente di sviluppo:** il progetto verrà sviluppato nell'IDE WebStorm sviluppato da JetBrains.
- **Linguaggi**: lo sviluppo avverrà tramite il framework Node.js per quanto riguarda la programmazione del back-end, mentre per lo sviluppo del front-end si lavorerà con il framework React.js, per il salvataggio dei dati dell'utente in locale verrà usato il formato JSON per mantenere la persistenza dei dati stessi.
- **Strumentazione di lavoro:** per lo sviluppo del progetto sarà necessario fare uso di macchine con i seguenti requisiti tecnici minimi: 2 GB di RAM, Windows 8 o macOS 10.13, 3.5 GB di spazio su disco, e una connessione dati.
- **Strumentazione di test:** per le operazioni di testing si seguirà un modello di sviluppo chiamato TDD (Test Driven Development), in cui il codice dei test viene scritto prima dell'implementazione, con il vantaggio di definire sin da subito il

comportamento di quest'ultima. Per questo motivo, l'attività di testing sarà intrinseca allo sviluppo del codice sorgente.
Saranno poi necessari ulteriori test d'integrazione nel momento in cui diversi componenti dell'applicazione saranno uniti e si userà Jest come framework di testing.

- **Standards di sviluppo**: è necessario organizzare e strutturare il codice affinché sia modulare e scalabile. Questo modo di realizzare il codice permette una visualizzazione più pulita e una lettura più facile, semplificando il lavoro all'interno del team per una gestione più agevole del codice scritte da altri membri del team, permettendo in futuro di testare le singole funzioni ed apportare eventuali modifiche in modo rapido. Inoltre, è opportuno affiancare il codice con dei commenti per spiegare ad altri membri del team, i quali si occuperanno di controllare la qualità ed il corretto funzionamento del codice. Questo permetterà ai membri del progetto di facilitare l'evoluzione del progetto (descritta nella sezione 1.3). In aggiunta, per omogeneizzare il lavoro e l'innesto di termini per le variabili/campi/metodi con le keyword di base, l'applicazione sarà creata utilizzando la lingua inglese in tutto ciò che la riguarda.

4.2 DOCUMENTAZIONE DEL SOFTWARE

La documentazione dell'estensione del software sarà realizzata dal team durante lo sviluppo del progetto, sarà disponibile entro le scadenze indicate nella sezione 1.2.
Inoltre se sarà necessario, verranno apportate modifiche al documento in caso di rischedulazione dei compiti secondo la programmazione citata nella sezione 2.1.
Link alla repository GitHub sulla quale viene raccolta tutta la documentazione relativa allo sviluppo:
https://github.com/Saygong/Progetto-Ingegneria-Del-Software

4.3 FUNZIONALITÀ DI SUPPORTO AL PROGETTO

Per garantire un corretto svolgimento del progetto e mantenere aggiornato il resto del team sul progresso, è necessario adottare delle misure di comunicazione, le quali agiranno come strumenti di supporto per lo stesso progetto.
Qui sotto verranno elencate le principali funzionalità di supporto adottate del team:

- **Pianificazione della qualità:** è necessario controllare lo sviluppo del codice man mano che verranno implementate nuove funzioni, affidando tale pratica agli altri membri del team, o almeno al soggetto identificato come backup engineer. Tale controllo avverrà rispettando determinati pattern di programmazione e la modularità del codice stesso, inoltre è necessario effettuare test frequentemente implementando dei test di integrazione nella repository del progetto.

- **Pianificazione della gestione delle configurazioni:** il lavoro del singolo avverrà su un nuovo branch dove ci si impone di sviluppare/correggere una feature, se tale implementazione verrà accettata dal backup engineer, si potrà effettuare una pull request sul main branch.

- **Funzionalità di supporto:** il team utilizzerà **GitHub** per mantenere il codice sorgente con i relativi backup, dove, sia il team che il responsabile di tutorato potranno controllare l'avanzamento del progetto, è molto utile inoltre per tornare a versioni precedenti in caso di malfunzionamenti importanti sul software che si sta sviluppando attualmente.

  Parallelamente, si utilizzerà la suite di Google, in particolare **Google drive** e **documents** per mantenere e redigere la documentazione relativa al progetto.

  La logica della base di dati invece verrà implementata tramite MongoDB, un DBMS non relazionale classificato come un database di tipo NoSQL.

5. PIANIFICAZIONE DEL LAVORO, DELLE RISORSE UMANE E DEL BUDGET

5.1 WBS (WORK BREAKDOWN STRUCTURE)

Si esplicita qui la struttura del progetto nelle diverse funzioni, attività e milestones, ovvero la work breakdown structure del progetto.

Il team ha proseguito con l'individuazione di una serie di funzioni per avere un costante riscontro, monitoraggio e sviluppo settimanale del progetto nella sua interezza, nonché per l'assegnamento e controllo delle diverse attività svolte e da svolgere.

Funzioni

Project Management

Funzione che sarà presente durante lo svolgimento di tutto il progetto. Sarà necessaria al team per organizzare il lavoro di ogni settimana, mostrando poi i risultati prodotti o problematiche da risolvere durante l'ultimo giorno lavorativo della settimana e se necessario apportare modifiche alla documentazione e riorganizzare il lavoro tra le componenti del team.

Documentazione

La redazione dei documenti avverrà prevalentemente durante lo sviluppo del codice sorgente (ad eccezione del piano di progetto che sarà redatto prima dello sviluppo del codice sorgente), sia per essere più organizzati nel lavoro, sia per favorire una conoscenza comune a tutti i membri del team, e, se necessario, anche per apportare modifiche ai vari documenti che potrebbero riportare dati non aggiornati, man mano che il progetto evolve.
La documentazione prodotta sarà principalmente quella specificata al punto 1.2.

Controllo qualità

Controllo della qualità dell'applicazione e della documentazione: ci si deve assicurare che esse siano in linea con gli obiettivi descritti nel piano di progetto e con i requisiti che caratterizzano ogni deliverable.

Testing

La fase di testing è la fase dove verrà eseguita una verifica del codice sviluppato. Il nostro team, come specificato al punto 4.1, seguirà un modello di sviluppo chiamato TDD (Test Driven Development), in cui il codice dei test viene scritto prima dell'implementazione, con il vantaggio di definire sin da subito il comportamento di quest'ultima. Per questo motivo, l'attività di testing sarà intrinseca allo sviluppo del codice sorgente.
Saranno poi necessari ulteriori test d'integrazione nel momento in cui diversi componenti dell'applicazione saranno uniti.

Attività e Task

A Ideazione del progetto
Comprende le attività di brainstorming tra i membri del team per ideare un possibile progetto da sviluppare per il corso di Ingegneria del software.

A.1 Brainstorming
Ogni membro del team ha la possibilità di fare una qualsiasi proposta di progetto, che verrà annotata e valutata nella fase successiva.

A.2 Analisi Fattibilità
In questa fase si analizzano obiettivi, requisiti e vincoli per la realizzazione delle proposte ideate nella fase di Brainstorming

A.3 Definizione dell'idea di progetto
Si vuole mostrare al team quale progetto verrà sviluppato e a quale scopo.

B Pianificazione
Discussione, definizione e pianificazione di attività, funzioni, ruoli, rischi e delle relative tempistiche.

B.1 Definizione struttura organizzativa
Vengono definiti i ruoli e le relative mansioni per ogni membro del gruppo in modo da spartire il lavoro in modo equo e definire una struttura lavorativa.

B.2 Analisi e gestione dei rischi
Vengono presi in considerazione le eventuali problematiche che potranno manifestarsi durante lo sviluppo dell'applicativo e la loro gestione.

B.3 WBS, Gantt, Pert, Allocazione Risorse
Definire l'ordine cronologico delle varie attività e la loro durata, tenendo conto dei vincoli temporali tra un'attività e l'altra.

C Sviluppo Versione Alpha
Sviluppo della versione iniziale del progetto. Questa dovrà avere le funzionalità strettamente necessarie ad adempiere agli obiettivi prefissati nella proposta di progetto.

C.1 Definizione dei Requisiti
Si definiscono formalmente le caratteristiche che il prodotto finale deve avere.

C.2 Progettazione Architetturale
Progettazione dell'architettura del software che il team dovrà realizzare.

C.3 Progettazione Modello ed API Database
Definizione del modello, ovvero degli oggetti che sono necessari all'implementazione della logica di business dell'applicazione.

C.4 Sviluppo API Database
Sviluppo del database MongoDB e del codice, lato server, riguardante tale database e l'API che ne espone l'accesso.

C.5 Sviluppo Back-end lato client
Sviluppo della logica di collegamento tra i vari componenti dell'interfaccia grafica ed il server.

C.6 Prototipazione UI
Verrà sviluppato un mockup semplice per avere una interfaccia grafica che mostri le funzionalità dell'applicazione.

C.7 Sviluppo Front-end lato client
Sviluppo dell'interfaccia grafica e della logica annessa.

C.8 Integrazione Server-Client
Integrazione delle varie componenti dell'architettura del progetto. Prevede anche una fase di testing per verificare che tale procedura avvenga correttamente.

C.9 Consegna Versione Alpha
Consegna della versione iniziale del progetto. Ci si deve assicurare che il formato ed il contenuto previsti dalla consegna del progetto siano rispettati.

D Sviluppo Versione completa
Sviluppo della versione finale del progetto. Questa dovrà essere completamente rifinita a livello di interfaccia grafica ed esperienza dell'utente, introducendo delle migliorie e potenzialmente espandendo la versione Alpha con delle funzionalità extra.

D.1 Definizione Rifiniture e Funzionalità Extra
Definizione formale degli interventi di rifinitura ed espansione che andranno svolti sulla versione Alpha.

D.2 Lato Server - Rifinitura e Funzionalità Extra
Implementazione di rifiniture e funzionalità extra volte a migliorare l'esperienza dell'utente, lato server.

D.3 Lato Client - Rifinitura e Funzionalità Extra
Implementazione di rifiniture e funzionalità extra volte a migliorare l'esperienza dell'utente, lato client.

D.4 Integrazione Server-Client
Integrazione delle varie componenti dell'architettura del progetto. Prevede anche una fase di testing per verificare che tale procedura avvenga correttamente.

D.5 Consegna Versione Completa
Consegna della versione finale del progetto. Ci si deve assicurare che il formato ed il contenuto previsti dalla consegna del progetto siano rispettati.

Milestones

Rilascio Versione Alpha

Verrà rilasciata la versione dimostrativa dell'applicazione per mostrare il suo effettivo funzionamento nelle procedure di base, quali, la presenza di un marketplace in cui esporre etc.

Rilascio Versione Completa

Rilascio della versione completa dell'applicazione, rifinita e migliorata nei punti in cui l'esperienza dell'utente nella versione Alpha era carente. Verranno implementate delle funzionalità extra qualora si dovessero rivelare importanti per migliorare in maniera significativa l'esperienza dell'utente.

Diagramma WBS

Dopo aver analizzato la WBS nelle sue varie parti, viene riportata la sua struttura graficamente in uno schema. Qui sono inserite tutte le funzioni, le attività, i task e le milestones secondo un ordine logico, in più si identifica anche un sequenzialità o parallelizzazione delle stesse attività (o task).

5.2 TEMPISTICHE E DIPENDENZE

Per dipendenze si intendono quelle attività che sono strettamente legate al termine di una attività precedente. Questo perché alcune attività dipendenti hanno bisogno di alcune parti del progetto terminate, senza contare che la forza lavoro limitata può saturarsi facilmente, ed è quindi necessario distribuire il lavoro in maniera equa e dilatata nel tempo. Le attività che prevedono una dipendenza sono forzate ad essere sequenziali, mentre, per attività che non hanno obblighi l'una verso l'altra, è possibile utilizzare una parallelizzazione delle stesse attività per ottimizzare i tempi di lavoro.

Qui di seguito è stilata una tabella delle dipendenze riguardante i task che verranno svolti dal team con relativa durata, data di inizio e di fine, da cui è possibile ricavare il cammino critico delle attività che devono essere assolutamente assolte nel periodo indicato (vedi Diagramma di Pert).

Oltre alla data di inizio e di fine prestabilite, sono anche segnate, per ogni attività (o task), le date di inizio e di fine ultime a cui si può fare riferimento per non sforare con i tempi previsti per la durata totale del progetto.

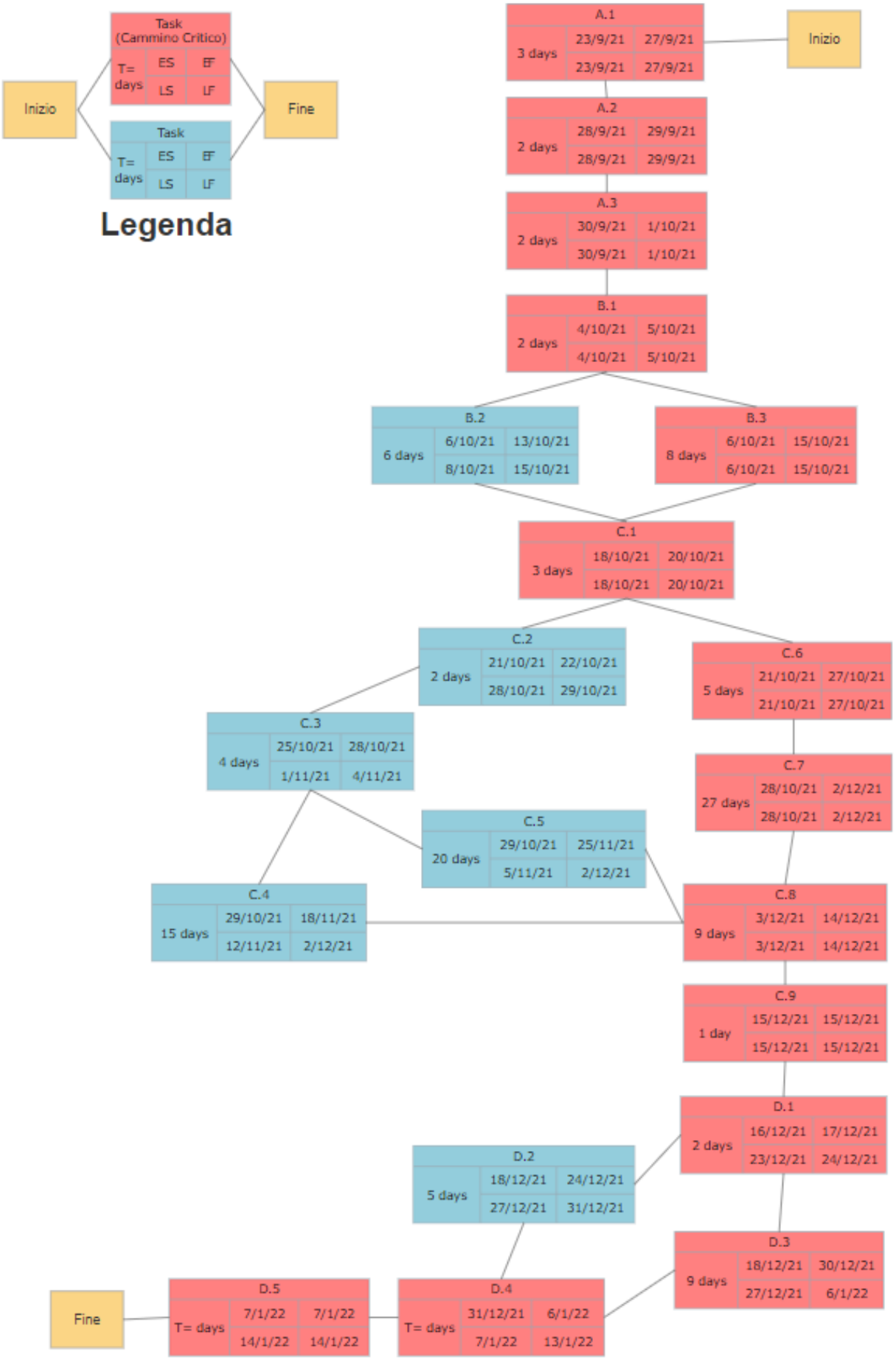
È stato tenuto conto che il team lavorerà durante la settimana dal Lunedì fino al Venerdì di ogni settimana, tutti i giorni Sabato e Domenica sono stati esclusi. Esclusi anche i giorni dove cadranno alcune festività importanti, ovvero Natale, Santo Stefano ed il 1° dell'anno 2022, che sono tuttavia giorni del weekend.
Per questioni di tempo, altre festività verranno considerate come giorni lavorativi, pertanto, lo svolgimento delle attività proseguirà regolarmente per quelle giornate.

Dalla tabella delle tempistiche e delle dipendenze, è stato ricavato il Diagramma di Pert che mostra graficamente il cammino del progetto con il relativo cammino critico. Per ogni task sono stati indicati l'ID, la durata, l'ES, l'EF, il LS ed il LF.

Tabella delle tempistiche e delle dipendenze

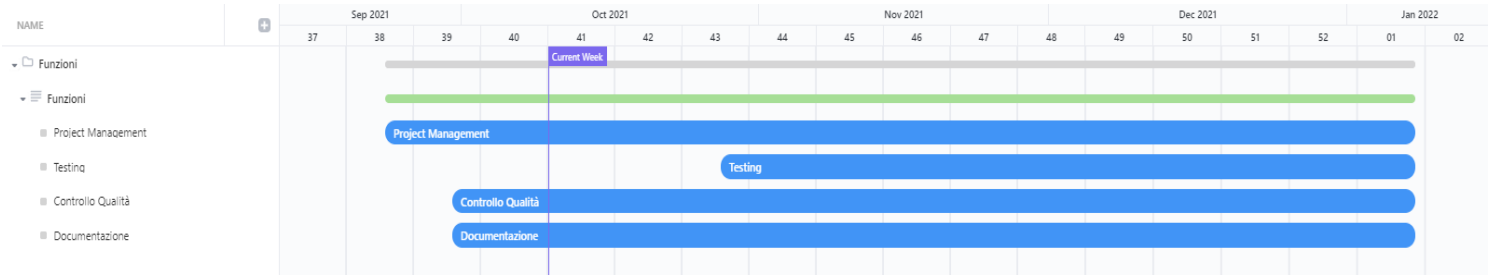
TO DO	ACTIVITY ID	ID	DEPENDENCIES	DURATION	ES	EF	LS	LF
✓ Brainstorming	A	A.1	–	3d	23/09/21	27/9/21	23/9/21	27/9/21
✓ Analisi Fattibilità	A	A.2	A.1	2d	28/09/21	29/9/21	28/9/21	29/9/21
✓ Definizione dell'idea di progetto	A	A.3	A.2	2d	30/09/21	1/10/21	30/9/21	1/10/21
✓ Definizione struttura organizzativa	B	B.1	A	2d	4/10/21	5/10/21	4/10/21	5/10/21
✓ Analisi e gestione dei rischi	B	B.2	B.1	6d	6/10/21	13/10/21	8/10/21	15/10/21
✓ WBS, Gantt, Pert, Allocazione Risorse	B	B.3	B.1	8d	6/10/21	15/10/21	6/10/21	15/10/21
✓ Definizione dei Requisiti	C	C.1	B	3d	18/10/21	20/10/21	18/10/21	20/10/21
✓ Progettazione Architetturale	C	C.2	C.1	2d	21/10/21	22/10/21	28/10/21	29/10/21
✓ Progettazione Modello ed API Database	C	C.3	C.2	4d	25/10/21	28/10/21	1/11/21	4/11/21
✓ Sviluppo API Database	C	C.4	C.3	15d	29/10/21	18/11/21	12/11/21	2/12/21
✓ Sviluppo Back-end lato client	C	C.5	C.3	20d	29/10/21	25/11/21	5/11/21	2/12/21
✓ Prototipazione UI	C	C.6	C.1	5d	21/10/21	27/10/21	21/10/21	27/10/21
✓ Sviluppo Front-end lato client	C	C.7	C.6	27d	28/10/21	2/12/21	28/10/21	2/12/21
✓ Integrazione Server-Client	C	C.8	C.4, C.5, C.7	9d	3/12/21	14/12/21	3/12/21	14/12/21
✓ Consegna Versione Alpha	C	C.9	C.8	1d	15/12/21	15/12/21	15/12/21	15/12/21
✓ Definizione Rifiniture e Funzionalità Extra	D	D.1	C	2d	16/12/21	17/12/21	23/12/21	24/12/21
✓ Lato Server - Rifinitura e Funzionalità Extra	D	D.2	D.1	5d	18/12/21	24/12/21	27/12/21	31/12/21
✓ Lato Client - Rifinitura e Funzionalità Extra	D	D.3	D.1	9d	18/12/21	30/12/22	27/12/21	6/1/22
✓ Integrazione Server-Client	D	D.4	D.2, D.3	5d	31/12/22	6/1/22	7/1/22	13/1/22
✓ Consegna Versione Completa	D	D.5	D.4	1d	7/1/22	7/1/22	14/1/22	14/1/22

Diagramma di Pert

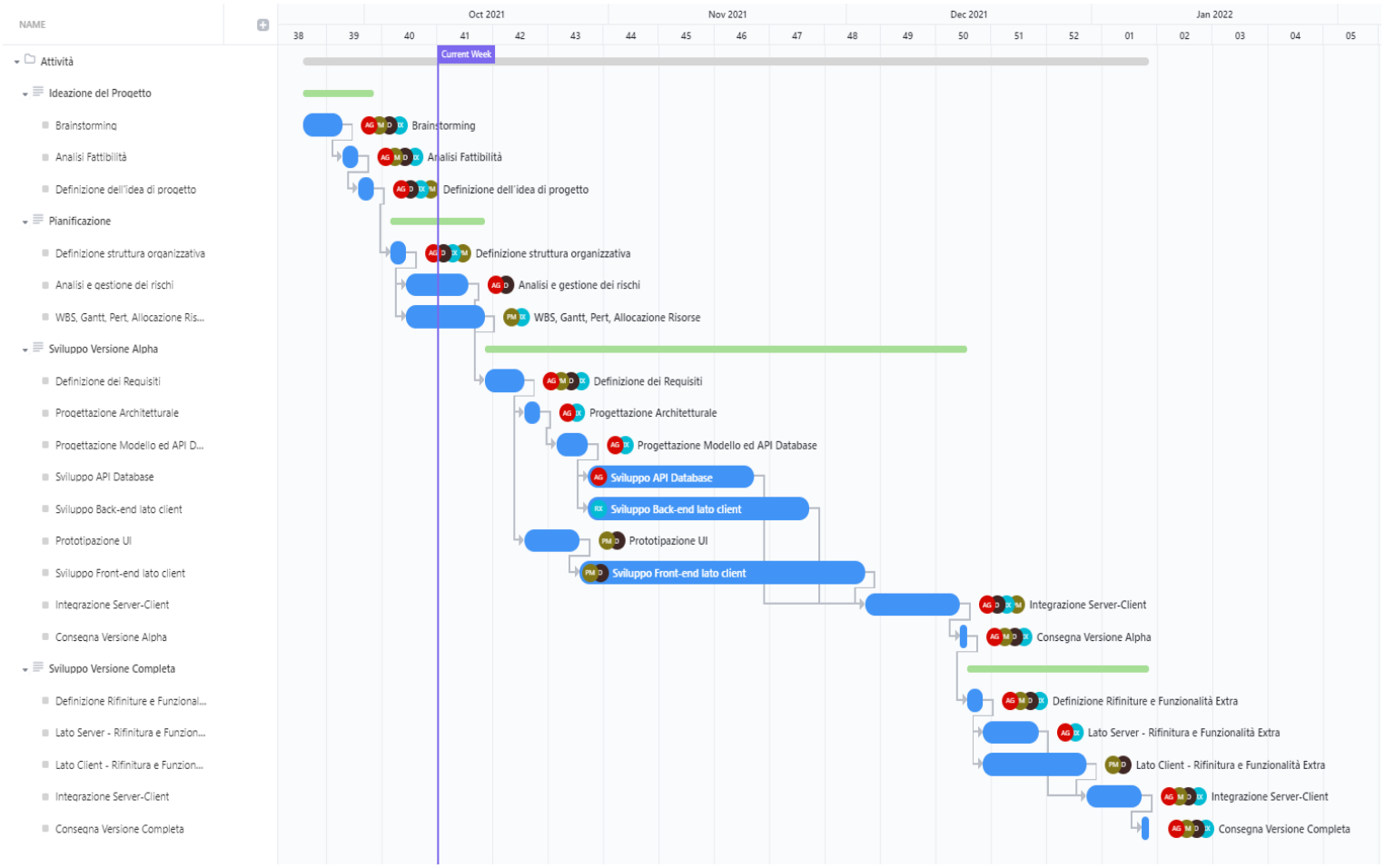


Diagrammi di Gantt

Funzioni



Attività e Task



Distribuzione delle attività dello staff

Di seguito viene mostrata l'organizzazione dello staff in base alle attività da svolgere per singolo membro del team. Le attività fanno riferimento alla tabella delle tempistiche e delle dipendenze indicata nella sezione 5.1, a cui fanno anche riferimento i diagrammi di Gantt: in questa tabella si vede l'associazione tra i task ed il personale assegnato per il loro svolgimento.

Di seguito specificati a chi appartengono le etichette che rappresentano il personale a cui un determinato task viene assegnato:

RX - Azzurro = Runjie Xia
PM - Marrone = Pierluigi Marchioro
D - Grigio Scuro = Marco Di Fresco
AG - Rosso = Andrea Gentilini

TO DO	ASSIGNEE	ACTIVITY ID	ID
Brainstorming	RX D PM AG	A	A.1
Analisi Fattibilità	RX D PM AG	A	A.2
Definizione dell'idea di progetto	PM RX D AG	A	A.3
Definizione struttura organizzativa	PM RX D AG	B	B.1
Analisi e gestione dei rischi	D AG	B	B.2
WBS, Gantt, Pert, Allocazione Risorse	RX PM	B	B.3
Definizione dei Requisiti	RX D PM AG	C	C.1
Progettazione Architetturale	RX AG	C	C.2
Progettazione Modello ed API Database	RX AG	C	C.3
Sviluppo API Database	AG	C	C.4
Sviluppo Back-end lato client	RX	C	C.5
Prototipazione UI	D PM	C	C.6
Sviluppo Front-end lato client	D PM	C	C.7
Integrazione Server-Client	PM RX D AG	C	C.8
Consegna Versione Alpha	RX D PM AG	C	C.9
Definizione Rifiniture e Funzionalità Extra	RX D PM AG	D	D.1
Lato Server - Rifinitura e Funzionalità Extra	RX AG	D	D.2
Lato Client - Rifinitura e Funzionalità Extra	D PM	D	D.3
Integrazione Server-Client	RX D PM AG	D	D.4
Consegna Versione Completa	RX D PM AG	D	D.5

5.3 RISORSE NECESSARIE

Si riconoscono molteplici categorie di risorse necessarie per la realizzazione di tale progetto:

- **Risorse umane:** identificato dai membri del team e le loro competenze tecniche (come lo sviluppo di software e la stesura della documentazione).
- **Risorse hardware:** le macchine da lavoro personali dei membri del team.
- **Risorse software:** sarà necessario munirsi di software come WebStorm per lo sviluppo generale dell'estensione, Google Documenti per la condivisione e la stesura della documentazione del progetto, Git per tenere traccia e facilitare lo sviluppo del software ed una piattaforma di comunicazione istantanea tra i membri del gruppo (WhatsApp e Discord).
- **Risorse temporali:** comprende le ore lavorative necessarie alla realizzazione del progetto, è necessaria una discreta gestione del tempo in quanto i membri del team non si dedicheranno solamente a questo progetto, ma anche alla frequenza e allo studio di altri corsi.
- **Risorse documentative:** include tutta la documentazione di riferimento per sviluppare l'applicazione, quali la documentazione e le guide sulle API, un piccolo manuale per usare GitHub e i requisiti di progetto.

5.4 ALLOCAZIONE DEL BUDGET E DELLE RISORSE

Non sono presenti spese economiche per lavorare su questo progetto, in quanto la strumentazione di lavoro e la strumentazione di test sono effettivamente i dispositivi già in possesso da ogni membro del team, inoltre i software che verranno utilizzati sono gratuiti.

Si potrebbe assumere di quantificare economicamente il lavoro di ogni singolo membro del team in Euro per ora di lavoro. A tal proposito, sapendo che i meeting di lunedì e venerdì saranno complessivamente di 2 ore, lavorando dal lunedì al venerdì per 3 ore al giorno, arrivando ad un monte di 15 ore settimanali. Sapendo che dal 23 settembre al 7 gennaio trascorrono poco più di 17 settimane e definendo 15 Euro per ogni ora lavorativa del singolo membro (il team è composto da 4 persone), si può stimare che solamente per le ore lavorative di tutti i membri, il progetto avrà un costo base di €15.300.

Gruppo	Settore	Prezzo	Totale parziale
Ore contate	Uomo-ora * 4	€15/ora	€15.300
Documenti	Piano di progetto	€250	€2.150
	Documento di analisi e specifica	€800	
	Piano di testing	€600	
	Documento di progettazione	€500	
Features	Business logic	€1.600	€3.100
	Database dinamico	€400	
	Partite simultanee	€400	
	Chat di gruppo	€400	
	Extra	€300	
Grafica	Grafica applicazione	€500	€500
		Totale finale (IVA esclusa)	€21.050

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.