



Progetto di Ingegneria del Software 2022/23

Università Ca' Foscari Venezia

Piano di progetto

1.0

0-Budget

17/10/2022



Document Informations

NomeProgetto	Bookito!	Acronimo	BKT
Derivable	Piano di progetto		
Data di consegna	17/10/2022		
Team Leader	Giulia Cogotti		884383@stud.unive.it
Team Members	Diego Passarella		886711@stud.unive.it
	Marco Chinellato		886217@stud.unive.it
	Marta Cazzin		883396@stud.unive.it
	Jinpeng Zhang		886854@stud.unive.it
	Tudor Croitor		886005@stud.unive.it

Document History

Version	Issue Date	Stage	Changes	Contributors
1.0	17/10/2022	Draft	Prima stesura del documento	Giulia Cogotti Diego Passarella Marco Chinellato
2.0		Final		



Indice

1. Introduzione	3
1.1. Overview del progetto	3
1.2. Evoluzione del progetto	5
1.3. Materiali di riferimento	6
1.4. Glossario	6
2. Organizzazione del progetto	7
2.1. Modello del processo	7
2.2. Struttura organizzativa	7
2.3. Interfacce organizzative	7
2.4. Responsabilità di progetto	8
3. Processi gestionali	9
3.1. Obiettivi e priorità	9
3.2. Assunzioni, Dipendenze, Vincoli	9
3.3. Gestione dei rischi	10
3.4. Meccanismi di monitoraggio e di controllo	12
3.5. Pianificazione dello staff	12
4. Processi tecnici	13
4.1. Metodi, strumenti e tecniche	13
4.2. Documentazione del software	14
4.3. Funzionalità di supporto al progetto	14
5. Pianificazione del lavoro, delle risorse umane e del budget	15
5.1. WBS	15
5.2. Dipendenze	17
5.3. Risorse necessarie	19
5.4. Allocazione del budget e delle risorse	20
5.5. Pianificazione	21



1. Introduzione

1.1. Overview del progetto

Il documento in questione illustra il piano di progetto, ovvero la pianificazione del lavoro che il gruppo dovrà svolgere per portare a termine la creazione dell'applicazione Android.

Il progetto ha come obiettivo la progettazione e la seguente creazione di un'applicazione android rivolta agli utenti più fragili appartenenti a un certo quartiere, in particolare viene scelta la provincia di Venezia.

L'idea proposta è quella di creare un'applicazione che chiameremo **Bookito!** la quale ha lo scopo di favorire lo scambio e il prestito di libri usati. Il tutto con uno sguardo alle interazioni sociali e al coinvolgimento di persone economicamente fragili. Ovviamente utilizzabile anche da altri utenti non fragili i quali vorrebbero fornire un aiuto concreto verso il prossimo.

Una situazione problematica è dettata dal prestito di un libro tra due utenti. Quando questo accade essi devono necessariamente concordare una data di restituzione, ovviamente salvata su database, tramite un'apposita chat.

Nel giorno prestabilito, i due utenti devono confermare l'avvenuta restituzione del libro, dove il proprietario di quest'ultimo dovrà segnalare le condizioni in cui esso è stato restituito. Il proprietario del libro può inoltre segnalare la mancata restituzione (o il suo ritardo).

In Bookito varrà la regola del "*silenzio assenso*", quindi se un utente non dovesse confermare (o rifiutare) l'avvenuta restituzione, dopo 7 giorni dalla data prestabilita, verrà in automatico contrassegnato come restituito. Tuttavia al fine di evitare che gli utenti si dimentichino di contrassegnare l'avvenuta riconsegna, Bookito manderà in modo persistente alcune notifiche per ricordare di compiere questa azione.

Inoltre, per evitare utenti "malevoli" (che non restituiscono libri o che li restituiscono in pessime condizioni) è necessario inserire delle "red flag" nei profili degli utenti dopo un certo numero di recensioni negative subite. In questo modo si spingerà gli utenti ad utilizzare l'applicazione in modo onesto creando un clima di fiducia e sicurezza.



In caso di utilizzo dell'applicazione da parte di un minorenni si dovrà pensare allo sviluppo di un sistema di *"controllo parentale"* in modo che gli incontri avvengano tramite o insieme all'adulto responsabile del minore.

1.2. Deliverables del progetto

La seguente sezione si occupa di definire i documenti che verranno consegnati con conseguente data.

ID	Titolo derivabile	Data di consegna
–	Definizione dei gruppi di lavoro	26/09/2022
D0	Proposta iniziale	03/10/2022
D1	Piano di progetto	17/10/2022
D2	Documento dei requisiti	31/10/2022
D3	Piano di testing	14/11/2022
D4	Documento di progettazione	28/11/2022
D5	Versione 1.0 del codice sorgente	15/12/2022
D6	Versione 2.0 del codice e allineamento documentazione	15/01/2023



1.3. Evoluzione del progetto

Il progetto ha il fine di conseguire un esame universitario. Se, al termine di esso, il team continuasse a collaborare ampliando ulteriormente l'applicazione creata, le funzionalità che essa fornirebbe aumenterebbero. In particolare di seguito vengono proposte alcune idee:

- In caso di molta affluenza da parte degli utenti, sarebbe utile una *“collaborazione”* con librerie o biblioteche della provincia di Venezia, in modo da aumentare i libri a disposizione e il controllo.
- Per incentivare l'utilizzo corretto dell'applicazione e aumentare la fiducia degli utenti, si potrebbe introdurre un sistema di *“punteggi/coins”*, nel quale gli utenti guadagnano un certo punteggio effettuando le operazioni supportate (prestito, scambio, regalo). In particolare assumendo un corretto comportamento l'utente avrà un punteggio maggiore, in caso contrario sarà minore. Un parametro di valutazione di esempio potrebbe essere quello della restituzione del libro in buone condizioni e secondo la data prestabilita.
- In caso di interesse da parte dell'utenza si potrebbe introdurre una sezione *“Booklist”*, in cui l'utente può avere una lista di tutti i suoi libri letti ed assegnar loro un voto con anche (possibilmente) una piccola recensione. La booklist potrà essere sia pubblica, che privata. In caso fosse pubblica potrebbe fornire un aiuto ad altri utenti nella scelta dei libri.
- Sarebbe interessante introdurre una funzionalità in cui l'utente richiede un libro di suo interesse da ricevere in prestito o in regalo che non è disponibile nella libreria virtuale di nessun utente. In questo modo si potrebbe trovare qualcuno disposto a inserirlo.
- Infine, si potrebbe pensare ad un eventuale ampliamento del range dell'area geografica in cui viene utilizzata l'applicazione.



1.4. Materiali di riferimento

I materiali utilizzati per la stesura del piano del progetto sono indicati di seguito:

- Slides fornite dal professore durante lo svolgimento del corso, allo scopo di avere indicazioni sul contenuto di ciascun paragrafo;
- Progetti degli anni passati forniti sulla piattaforma moodle;
- Documenti e dati online per calcolare medie di budget e risorse;
- [Project Libre](#), software gratuito per la progettazione del diagramma di [Gantt](#) e del [grafo delle attività](#);
- [Edrawmax](#), software con alcune funzionalità gratuite per la creazione di diagrammi, utilizzato per l' [WBS](#);

1.5. Glossario

API: Application Program Interface, sono il metodo con cui le librerie vengono usate per sopperire ad un problema specifico di scambio di informazioni

IDE: Integrated Development Environment (Ambiente di sviluppo integrato) è un software che in fase di programmazione, supporta i programmatori nello sviluppo e debugging del codice sorgente di un programma.

LaTeX: Linguaggio di markup per la scrittura di documenti.

Kotlin: Linguaggio di programmazione atto allo sviluppo di applicazioni android.

Java: Linguaggio di programmazione ad oggetti.

Overleaf: Web app che supporta la scrittura di documenti in latex condivisibili con altri utenti.

Discord: Applicazione che permette la comunicazione orale e testuale fra molteplici utenti.

Smart-working: è una particolare modalità di esecuzione della prestazione di lavoro subordinato introdotta al fine di incrementare la competitività e di agevolare la conciliazione dei tempi di vita e lavoro.

Backup: In informatica si indica con la messa in sicurezza di informazioni di un sistema, da utilizzare come recupero (ripristino) in caso di danni accidentali.



2. Organizzazione del progetto

2.1. Modello del processo

Il modello scelto per gestire le relazioni tra le varie fasi del processo è il cosiddetto “*modello evolutivo*” il quale permette la suddivisione del lavoro partendo da una descrizione di massima di arrivare al risultato finale, attraversando le fasi intermedie.

Questo modello dona flessibilità al progetto con il rischio accettabile di non sapere a che punto si è arrivati di preciso.

Inoltre verrà in parte utilizzato un modello di processo basato sul *riutilizzo*, il quale permette di sfruttare librerie di codice già esistenti.

Un ulteriore modello che verrà usato è quello di sviluppo agile, in particolare la sua funzionalità di *pair programming* sfruttando i vantaggi della collaborazione lavorativa.

2.2. Struttura organizzativa

La comunicazione tra i membri del gruppo avverrà nei seguenti modi:

- Discussione di gruppo nelle ore libere tra una lezione e un'altra;
- Utilizzo di un server discord provvisto di canali vocali e testuali utili nella comunicazione remota e nell'organizzazione delle informazioni;
- Utilizzo di un gruppo whatsapp per comunicazioni rapide e urgenti.

Ogni membro deve essere ben a conoscenza delle proprie competenze, in modo tale da fornire la task più adatta al fine di evitare un rallentamento del lavoro.

2.3. Interfacce organizzative

Per interfacce organizzative si intende le relazioni con altre entità. In particolare per quanto riguarda il docente esso si interfacerà con il team leader che avrà l'ulteriore compito di effettuare le consegne finali. Rispetto ai potenziali utenti sarà compito di tutti i membri gestire un possibile dialogo per coinvolgerli nell'utilizzo dell'applicazione o per avere un punto di vista esterno.



2.4. Responsabilità di progetto

Potenzialmente ogni membro del gruppo può contribuire in egual modo nello sviluppo del progetto. Si è deciso tuttavia di adottare indicativamente la seguente suddivisione dei compiti, in modo tale da favorire la collaborazione e migliorare le interazioni tra i componenti.

Occupazione	Descrizione	Componenti
<i>Sviluppo applicazione</i>	Scrittura del codice sorgente.	Chinellato Marco Cogotti Giulia Passarella Diego Croitor Tudor
<i>Revisione codice</i>	Controllo della qualità del codice.	Chinellato Marco Cogotti Giulia Passarella Diego Croitor Tudor
<i>Testing e feedback</i>	Testing delle funzionalità dell'applicazione, invio di feedback e segnalazioni di problemi.	Cogotti Giulia Chinellato Marco Passarella Diego Cazzin Marta Zhang Jinpeng Tudor Croitor
<i>Stesura documentazione</i>	Stesura dei vari documenti da consegnare descritti nella sezione 1.2 del corrente documento.	Cogotti Giulia Passarella Diego Chinellato Marco Zhang Jinpeng
<i>Revisione documentazione</i>	Controllo dei deliverable da consegnare.	Cogotti Giulia Tudor Croitor
<i>Graphic UI Design</i>	Prototipazione dell'interfaccia grafica e dell'esperienza utente.	Cazzin Marta Zhang Jinpeng



3. Processi gestionali

3.1. Obiettivi e priorità

Per concludere con successo il progetto, ovvero il completamento dell'applicazione secondo le premesse prestabilite, sono rilevanti gli aspetti che riguardano:

- *rispetto delle scadenze*: altrimenti sarà difficile arrivare fino in fondo;
- *coinvolgimento utenti*: per garantire che l'applicazione abbia uno scopo;
- *cooperazione*: evitare che si creino meccanismi disfunzionali tra i membri del team;
- *rispetto delle funzionalità descritte*: implementare poche funzionalità ma fatte bene.

3.2. Assunzioni, Dipendenze, Vincoli

Per portare a termine del progetto si assume che tutti i membri del team siano responsabili e autonomi nel proprio lavoro per poter eseguire correttamente i propri compiti senza intralciare il lavoro complessivo. Inoltre si assume che ognuno abbia le capacità di lavorare in gruppo senza creare situazioni disfunzionali.

Riguardo le conoscenze tecniche, si assume che ogni membro sappia utilizzare gli strumenti, elencati al punto [4.1](#), necessari a svolgere i compiti assegnati al punto [2.4](#).

Altre assunzioni riguardano le ore di lavoro, considerando che la consegna finale andrà effettuata a gennaio, è molto probabile che nel periodo natalizio il lavoro eseguito sarà minore. Di conseguenza sarà responsabilità di ognuno anticipare il lavoro per evitare un sovraccarico nel periodo indicato.



3.3. Gestione dei rischi

La tabella seguente visualizza la probabilità e i rischi relativi allo svolgimento del progetto. I rischi vengono identificati tramite una scala di colori che varia dal verde (basso impatto e bassa probabilità) al rosso (alto impatto e alta probabilità). Al di sotto vengono definiti i rischi con possibili misure per risolvere il problema.

Impatto	Molto alto					
	Alto		#2			#7
	Medio	#3		#6		
	Basso			#4	#1	
	Molto basso	#5				
		0 - 20%	20 - 40%	40 - 60%	60 - 80%	80 - 100%
Probabilità						



La seguente tabella illustra i rischi con la relativa probabilità e impatto, inoltre viene fornita una possibile soluzione ai problemi nel caso in cui essi si dovessero verificare:

N°	Rischio	Probabilità	Impatto	Soluzione
#1	Malattia di uno o più componenti.	75%	Basso	Lavoro in <i>smart-working</i> . In caso di gravi condizioni, suddivisione del lavoro tra i restanti membri del team.
#2	Un membro non porta a termine i suoi incarichi.	40%	Alto	Richiamo ed eventuale esclusione dal gruppo con conseguente ridistribuzione del lavoro.
#3	Un componente abbandona spontaneamente il gruppo.	10%	Medio	Ripartizione del lavoro fra i restanti membri del team.
#4	Competenze richieste troppo elevate con una complessità del progetto troppo elevata.	55%	Basso	Studio approfondito degli strumenti e ridimensionamento delle funzionalità.
#5	Perdita dei dati.	8%	Molto basso	Ripristino da backup precedenti.
#6	Impegni personali di un membro per un certo periodo prestabilito.	60%	Medio	Gestione del lavoro in data precedente, in modo da alleggerire il carico durante il periodo di assenza.
#7	Concorrenza con altri corsi.	95%	Alto	Provare a distribuire il tempo, in caso di fallimento dare minore priorità ad altri corsi.



3.4. Meccanismi di monitoraggio e di controllo

Ogni documento seguirà un format specifico fornito dal docente, in particolare dovrà essere presente il logo dell'università Ca' Foscari nell'intestazione e titolo del corso con nome del docente nel piè di pagina.

I documenti e il codice stesso partono dalla versione 1.0, in seguito quando verranno apportate importanti modifiche essi passeranno alla versione 1.X, fino al raggiungimento della versione finale 2.0.

Per quanto riguarda il monitoraggio dei rischi elencati nella [gestione dei rischi](#), essi si possono prevedere e controllare tramite una comunicazione attiva tra i membri del team.

3.5. Pianificazione dello staff

Tutti i componenti del team devono collaborare e comunicare tra di loro, esponendo problemi e/o idee relativi al progetto stesso.

Il progetto richiede competenze nello sviluppo di applicazioni android, al momento nessun componente del gruppo ha queste capacità. Per risolvere il problema sarà necessario seguire attentamente le lezioni presentate in aula e documentarsi personalmente tramite internet.

Inoltre è fortemente richiesta una solida base di programmazione e capacità di problem-solving. Qualora ci fosse qualche componente con lacune in questo, esso dovrà approfondire la conoscenza tramite risorse a sua disposizione evitando di rallentare lo svolgimento del progetto.



4. Processi tecnici

4.1. Metodi, strumenti e tecniche

Nello sviluppo del progetto verranno utilizzati i seguenti strumenti:

- **Android Studio** come ambiente di sviluppo per Java
- **Java/XML** per il FrontEnd, **Firestore & Javascript** per il Back-end
- **API e Framework**: Firestore per gestire autenticazione degli utenti, [Google Books API](#) per avere una lista di tutti i libri indicizzati da google.
- **Git/GitHub** come strumento di controllo di versione del codice e piattaforma di backup decentralizzato.
- [Figma](#) come strumento di prototipazione dell'interfaccia ed esperienza utente.

La documentazione relativa al progetto viene fornita utilizzando i seguenti strumenti, ognuno di essi permette la scrittura simultanea di più utenti rendendone più semplice la stesura:

- LaTeX
- OverLeaf
- GoogleDocs

Di seguito sono indicate le linee guida a cui ci si farà riferimento durante il progetto:

- Documentazione Java
(<https://docs.oracle.com/javase/7/docs/api/overview-summary.html>)
- Documentazione Android Studio
(<https://developer.android.com/docs>)
- Linee guida material design per Android
(<https://material.io/>)
- Documentazione Firestore:
(<https://firebase.google.com/docs>)
- Documentazione Javascript:
(<https://developer.mozilla.org/en-US/docs/Web/JavaScript>)



4.2. Documentazione del software

In linea teorica è prevista la stesura di un manuale d'utilizzo dell'applicazione da fornire agli utenti. Questo documento verrà creato soltanto nel caso in cui tutte le attività del progetto finissero nei tempi previsti dall'EF (*earliest finish time*), rappresentato al punto [5.2](#).

4.3. Funzionalità di supporto al progetto

Per le funzionalità di supporto viene fatto riferimento allo standard *ISO/IEC 25010* che descrive la qualità del software sulla base di otto attributi: idoneità funzionale, efficienza/performance, compatibilità, usabilità, affidabilità, sicurezza, manutenibilità e portabilità. In seguito questa sezione verrà aggiornata con la qualità relativa al progetto svolto.



5. Pianificazione del lavoro, delle risorse umane e del budget

5.1. WBS

La seguente sezione mostra il WBS, ovvero il *Work Breakdown Structure*, il quale è un diagramma che suddivide il progetto principale in diverse attività, suddivise a loro volta in task. Inizialmente è presente una descrizione testuale del WBS e in seguito il vero e proprio diagramma.

Per convenzione si è deciso che le lettere greche indicano le funzioni, le lettere maiuscole le attività e le lettere minuscole i task. I numeri che seguono le lettere aiutano ad individuare la gerarchia tra attività, sotto attività e task.

Le dipendenze tra le varie componenti vengono mostrate nella sezione successiva in modo da non sovraccaricare graficamente il WBS.

Funzioni:

Le funzioni qua descritte prevedono un affiancamento *costante* allo sviluppo del progetto.

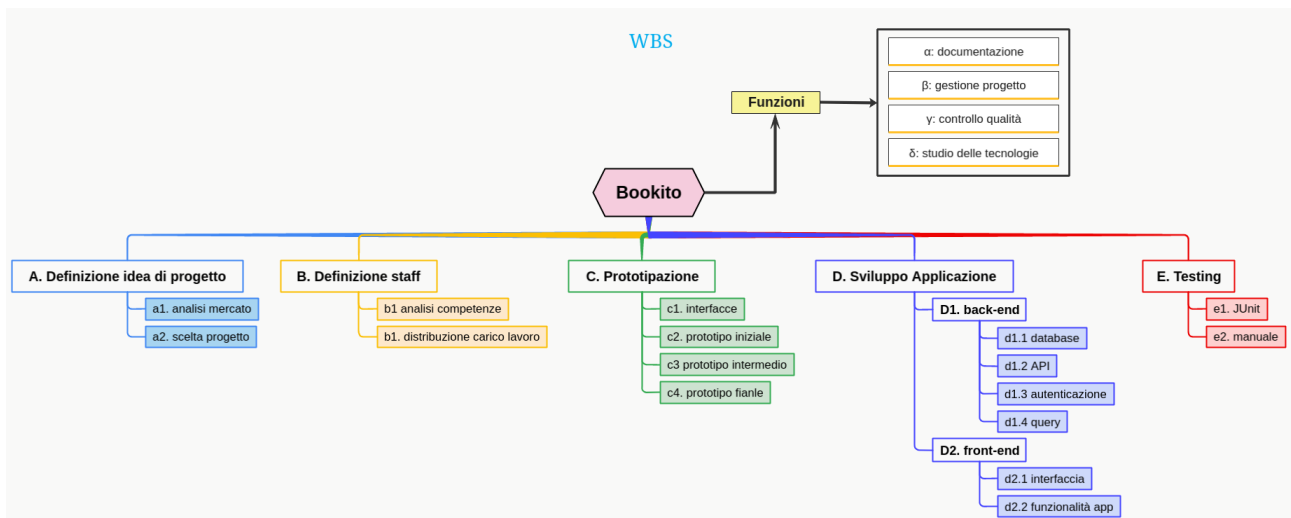
- **α Documentazione**, composta dagli elaborati descritti nei [deliverables di progetto](#).
- **β Gestione progetto**.
- **γ Controllo qualità** del codice e della documentazione.
- **δ Studio delle tecnologie** in modo da adeguarsi agli standard esistenti

Attività:

- **A Definizione idea di progetto:**
 - **a1** *Analisi del mercato* riguardante le applicazioni già esistenti
 - **a2** *Scelta del progetto*, ovvero una chiara definizione riguardo alle funzionalità e lo scopo dell'applicazione scelta
- **B Definizione dello staff:**
 - **b1** *Analisi competenze dei membri* tramite un attento sguardo al curriculum presentato in fase di creazione team
 - **b2** *Distribuzione del carico di lavoro* come mostrato nelle [responsabilità di progetto](#)
- **C Prototipazione interfaccia grafica:**
 - **c1** *Definizione interfacce*
 - **c2** *Disegno del primo prototipo*
 - **c3** *Analisi e miglioramento del prototipo*
 - **c4** *Implementazione del prototipo finale*



- **D Sviluppo applicazione:**
 - **D1 Configurazione backend:**
 - *d1.1 Configurazione database*
 - *d1.2 Configurazione API*
 - *d1.3 Configurazione autorizzazione e autenticazione utenti*
 - *d1.4 Generazione delle query per operazioni sui dati dell'applicazione*
 - **D2 Configurazione frontend:**
 - *d2.1 Implementazione dell'interfaccia*
 - *d2.2 Sviluppo delle funzionalità dell'applicazione*
- **E Testing:**
 - *e1 Testing automatici con JUnit*
 - *e2 Testing manuali installando l'applicazione in un dispositivo Android.*





5.2. Dipendenze

La tabella mostra i dati relativi all'[WBS](#) descritto nel punto precedente, aggiungendo le informazioni riguardanti le attività e i task. In particolare vengono mostrate le dipendenze, la durata espressa in giorni e i dati del diagramma di PERT (ES, EF, LS, LF). In particolare le sigle hanno i seguenti significati:

- ES (*earliest start time*): il minimo giorno di inizio dell'attività, a partire dal minimo tempo necessario per le attività che precedono.
- EF (*earliest finish time*): dato ES e la durata dell'attività, esso rappresenta il minimo giorno in cui l'attività può terminare.
- LF (*latest finish time*): il massimo giorno in cui quel task deve finire senza che si crei ritardo per i task che dipendono da lui.
- LS (*latest start time*): dato LF e la durata del task, esso rappresenta il massimo giorno in cui quel task deve iniziare senza provocare ritardo per i job che dipendono da lui.

ID	Dipendenze	Durata	ES	EF	LS	LF
A	-	3gg	19/09/2022	22/09/2022	23/09/2022	26/09/2022
a1	-	1gg	19/09/2022	20/09/2022	23/09/2022	24/09/2022
a2	a1	2gg	20/09/2022	22/09/2022	24/09/2022	26/09/2022
B	A	2gg	26/09/2022	28/09/2022	01/10/2022	03/10/2022
b1	A	1gg	28/09/2022	29/09/2022	01/10/2022	02/10/2022
b2	b1	1gg	29/09/2022	30/09/2022	02/10/2022	03/10/2022
C	B	15gg	03/10/2022	18/10/2022	08/12/2022	23/11/2022
c1	B	3gg	03/10/2022	06/10/2022	08/12/2022	11/12/2022
c2	c1	4gg	06/10/2022	10/10/2022	11/12/2022	15/11/2022
c3	c2	3gg	10/10/2022	13/10/2022	15/11/2022	18/11/2022
c4	c3	5gg	13/10/2022	18/10/2022	18/11/2022	23/11/2022
D	C	47gg	18/10/2022	4/12/2022	23/11/2022	15/01/2023
D1	C	14gg	18/10/2022	29/10/2022	23/11/2022	06/12/2022



d1.1	C	1gg	18/10/2022	19/10/2022	20/11/2022	21/11/2022
d1.2	d1.1	2gg	19/10/2022	20/10/2022	21/11/2022	23/11/2022
d1.3	d1.2	2gg	20/10/2022	22/10/2022	23/11/2022	25/11/2022
d1.4	d1.3	10gg	01/11/2022	11/11/2022	25/11/2022	05/12/2022
D2	C, D1	47gg	18/10/2022	05/12/2022	23/11/2022	15/01/2023
d2.1	C	7gg	18/10/2022	25/10/2022	23/11/2022	30/11/2022
d2.2	D1	40gg	29/10/2022	05/12/2022	06/12/2022	15/01/2023
E	d2.1	40gg	25/10/2022	05/12/2022	06/12/2022	15/01/2023
e1	d2.1	40gg	25/10/2022	05/12/2022	06/12/2022	15/01/2023
e2	d2.1	40gg	25/10/2022	05/12/2022	06/12/2022	15/01/2023



5.3. Risorse necessarie

Lo svolgimento del progetto richiede un certo quantitativo di risorse, in particolare risorse temporali. Ogni membro del team dovrà prendersi la responsabilità di garantire il suo contributo.

Al momento della stesura del presente documento, il team svolge almeno due incontri settimanali in presenza della durata di 5 ore ciascuno, per fare il punto sulla situazione e collaborare. In misura non quantificabile c'è una frequente comunicazione tramite discord e whatsapp per condividere idee e pareri sullo svolgimento del progetto.

Quando inizierà la scrittura del codice si presuppone che le ore dedicate al progetto aumentino.



5.4. Allocazione del budget e delle risorse

Tutte le API utilizzate prevedono un piano gratuito, quindi non sono stati spesi soldi da nessun componente del gruppo per il loro utilizzo.

Secondo varie fonti esaminate, lo stipendio di un Junior Software Engineer corrisponde a circa €13-14 l'ora, per semplicità questo valore verrà arrotondato a €13.

Mediamente vengono svolti 2 incontri settimanali tra i componenti del gruppo in presenza, di circa 10 ore in tutto, per un totale di €780 a settimana.

La consegna del progetto verrà effettuata in 13 settimane, dunque in totale la forza lavoro avrà un costo complessivo di €10.140.

Qui di seguito viene riportata una tabella rappresentante una stima approssimativa del costo totale di progetto comprendente il costo di implementazione delle varie funzionalità e dei servizi offerti.

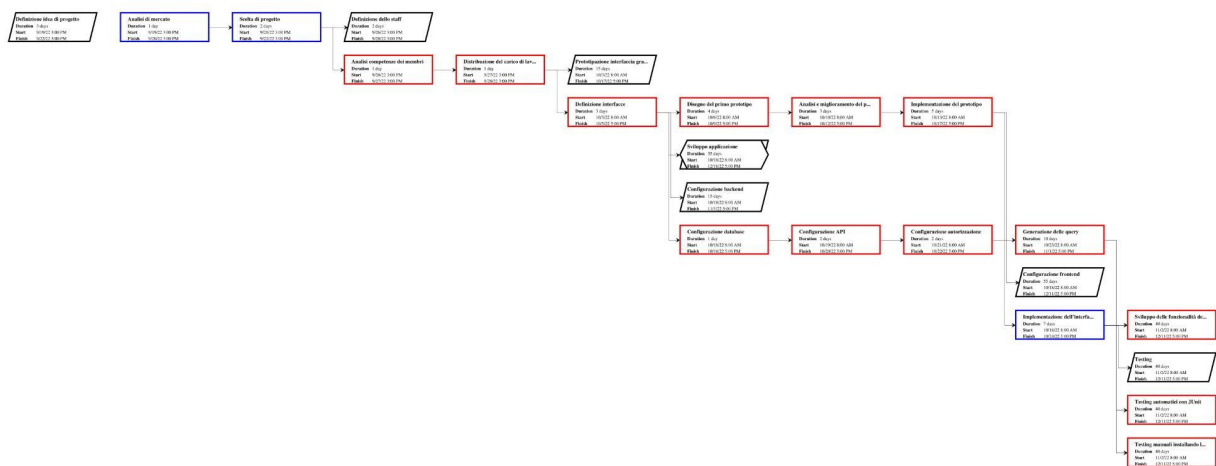
Gruppo	Tipologia	Prezzo	Totale parziale
<i>Tempo di lavoro</i>	Ore lavorative	€780/settimana	€10.140
<i>Documenti</i>	Piano di Progetto	€350	€350+
	Documento di analisi e Specifica	TBD	
	Piano di Testing	TBD	
	Documento di Progettazione	TBD	
<i>Funzionalità</i>	Chat fra utenti	€500	€4.700
	Creazione database	€1000	
	Autenticazione utente	€700	
	Funzionalità esclusive	€2500	
		Totale (IVA ESCLUSA)	€15.190+



5.5. Pianificazione

In questa sezione verrà mostrato il diagramma di PERT e il diagramma di Gantt strettamente correlati alle attività e task descritti nell' [WBS](#) con le relative informazioni specificate nella tabella delle [dipendenze](#).

Inizialmente viene mostrato il grafo delle attività, anche detto **diagramma di PERT**. Esso esprime le dipendenze e il cammino critico, cioè la sequenza di mansioni che porta ad arrivare alla consegna del progetto senza tollerare ritardi.



Infine viene mostrato il cosiddetto **diagramma di Gantt** che mostra i task del WBS esprimendo la temporizzazione. In questo modo si nota quali compiti possono essere sovrapposti e quali devono attendere la fine di un task precedente.

