



Ingegneria del Software

Piano di progetto

Nome progetto: “Rent-Sciò”

Nome gruppo: Ramma

Michele Lotto: 875922

Andrea Chinellato: 875422

Alessandro Furlan: 854723

Michele Gatto: 875446

Riccardo Zuliani: 875532

Committente: Agostino Cortesi

INDICE

1. Introduzione	3
1.1 Overview del Progetto	3
1.2 Deliverables del Progetto	3
1.3 Evoluzione del Progetto	3
1.4 Materiali di riferimento	4
1.5 Definizioni e Abbreviazioni	4
2. Organizzazione del progetto	6
2.1 Modello del Processo	6
2.2 Struttura Organizzativa	6
2.3 Interfacce Organizzative	6
2.4 Responsabilità di Progetto	7
3. Processi gestionali	8
3.1 Obiettivi e Priorità	8
3.2 Assunzioni, Dipendenze, Vincoli	8
3.3 Gestione dei rischi	9
3.4 Meccanismi di monitoraggio e controllo	10
3.5 Pianificazione dello staff	10
4. Processi tecnici	11
4.1 Metodi, Strumenti e Tecniche	11
4.2 Documentazione del Software	11
4.3 Funzionalità di supporto al progetto	11
5. Pianificazione del lavoro, delle risorse umane e del budget	12
5.1 WBS (Work breakdown structure)	12
5.2 Dipendenze	14
5.2.1 Schedulazione del lavoro	14
5.2.2 Diagramma di PERT	16
5.2.3 Diagramma di GANTT	17
5.3 Risorse Necessarie	18
5.4 Allocazione del Budget e delle Risorse	18
5.5 Pianificazione – Deadlines e Milestones	18

1. Introduzione

1.1 Overview del Progetto

Il progetto, commissionato dal professor Agostino Cortesi per l'esame di Ingegneria del Software presso l'Università Ca' Foscari di Venezia, prevede la realizzazione di un applicativo Android che sfrutti la tecnologia peer-to-peer e alcuni sensori intrinseci allo smartphone.

La nostra idea consiste nell'implementazione di un sistema per la gestione ed il noleggio di veicoli a pedali (es. Risciò) e veicoli affini, molto comuni in località balneari e turistiche (es. Jesolo).

Negoziante e cliente avranno a disposizione un'interfaccia personalizzata per ognuno, in grado di tener conto di diverse specifiche relative al viaggio.

Il **negoziante** avrà a disposizione le seguenti features:

1. Configurare l'area dove i suoi veicoli potranno circolare;
2. Controllo in tempo reale della posizione e dello stato dei suoi veicoli.

Il **cliente** avrà a disposizione le seguenti features:

1. Un catalogo per scegliere il percorso desiderato;
2. Notifiche in caso di velocità troppo elevata, posizione non consentita, tempo sufficiente per ritornare al negozio.

1.2 Deliverables del Progetto

- Piano di Progetto (18/10/2020)
- Documento di analisi e specifica (27/10/2020)
- Piano di testing (14/11/2020)
- Documento di Progettazione (09/12/2020)
- Codice e messa in linea del sistema (15/01/2021)

1.3 Evoluzione del Progetto

Il progetto nasce per realtà piccole e limitate, ove non vi è possibilità di avere la situazione sotto controllo tramite app e web server, bensì con carta e penna. Pensando più in grande, sarebbe possibile implementare la stessa idea per il noleggio di automobili e veicoli affini, servendosi della tecnologia fornita da Android Auto, per integrare al meglio l'applicazione al mezzo di trasporto noleggiato. Si potrebbe inoltre introdurre un metodo di pagamento dinamico (in base alle ore trascorse del noleggio) tramite app, attraverso collegamento ad un conto bancario (carta di debito/credito oppure Paypal).

1.4 Materiali di riferimento

Google:

- <https://developer.android.com/design>
- <https://material.io/design>
- <https://developer.android.com/guide>

Moodle:

- Slides del professor Agostino Cortesi;
- Libri su Android e il suo sviluppo, disponibili su moodle.

1.5 Definizioni e abbreviazioni

Android: Sistema operativo per dispositivi mobili sviluppato da Google Inc., basato su kernel Linux.

Android Studio: Ambiente di sviluppo integrato per lo sviluppo su piattaforme Android.

API (Application Programming Interface): Insieme di librerie software di un linguaggio di programmazione.

Applicazione: Il termine applicazione individua un software installato o una serie di software in fase di esecuzione su un computer o un dispositivo mobile con lo scopo e il risultato di rendere possibile una o più funzionalità o servizi.

App: abbreviazione di “applicazione”, in questo caso un programma per dispositivi mobili.

GitHub: È un servizio di hosting per progetti software. Vi si può caricare codice sorgente rendendolo scaricabile agli utenti.

Android Auto: È uno standard basato su smartphone, sviluppato da Google, per consentire ai dispositivi mobili con il sistema operativo Android di essere utilizzati in automobili mediante la head unit del cruscotto.

Paypal: Metodo di pagamento e di trasferimento di denaro mediante l'utilizzo di Internet.

Discord: Mezzo di comunicazione vocale nel quale si può parlare, discutere e lavorare in gruppo a distanza.

GPS: Sistema di posizionamento di navigazione satellitare che attraverso una rete dedicata di satelliti artificiali in orbita, fornisce ad un terminale mobile o ad un ricevitore GPS informazioni sulle sue coordinate geografiche.

WBS: Work Breakdown Structure, struttura di scomposizione del lavoro. Si intende con questo termine l'elenco di tutte le attività di un progetto.

Linguaggio Java: Linguaggio di programmazione ad alto livello orientato agli oggetti, avente tipizzazione statica.

Modello di tipo evolutivo: Il modello evolutivo è uno dei modelli del ciclo di vita del software che cerca di superare i limiti principali del modello a cascata. Si basa sulla prototipazione che consiste nell'uso di specifici strumenti software per la realizzazione rapida di una versione semplificata del sistema informativo, con la quale sperimentare le sue funzionalità. La verifica del prototipo può portare ad una modifica dei requisiti e ad un'eventuale revisione del progetto.

Deliverables (“consegnabile”): In un progetto indica un risultato significativo, quindi un output che rappresenta una fase importante per il committente che deve essergli consegnata. Può essere un oggetto fisico o un documento.

Diagramma di Pert: è un diagramma reticolare (grafo), nel quale i rettangoli rappresentano le attività, mentre linee e frecce (archi orientati) rappresentano i rapporti di dipendenza e le relazioni tra attività.

Diagramma di Gantt: è un diagramma a barre che rappresenta le attività in un piano cartesiano, con il tempo in ascissa e la successione delle attività in ordinata, mediante barre la cui lunghezza è proporzionale alla durata. Dal diagramma si ricava quanto tempo dura ogni singola attività, quando è allocata nel tempo e quanto tempo è necessario per completare tutto il progetto.

Modello a cascata: Questo modello chiamato anche ciclo di vita a cascata è il più tradizionale modello di ciclo di vita del software. Secondo questo modello il processo di realizzazione del software è strutturato in una sequenza lineare di fasi o passi, che comprende: analisi dei requisiti, progettazione, sviluppo, collaudo e manutenzione.

Fake GPS: Applicazione utilizzata in ambito informatico per permettere ad un dispositivo Android di ricevere coordinate “errate” impostate dall'utente in modo tale da simulare la funzionalità di un vero e proprio sensore GPS.

Funzionalità di Back-end: Funzionalità che permette l'effettivo funzionamento di un programma su un qualsiasi dispositivo. Spesso utilizzata in coppia con la funzionalità front-end, che riguarda invece l'interfaccia utente.

Prototipo: Un prototipo è, dunque, un modello approssimato o parziale del sistema che vogliamo sviluppare, realizzato allo scopo di valutare determinate caratteristiche

Prototipazione di tipo evolutivo: L'obiettivo è lavorare con il cliente ed evolvere verso il sistema finale a partire da una specifica di massima. Lo sviluppo inizia con le parti del sistema che sono già ben specificate, aggiungendo via via nuove caratteristiche.

Prototipazione di tipo usa e getta: L'obiettivo è capire i requisiti del sistema e quindi sviluppare una definizione migliore dei requisiti, infatti il prototipo sperimenta le parti del sistema che non sono ancora ben comprese

2. Organizzazione del progetto

2.1 Modello del Processo

Per questo progetto abbiamo pensato di adottare un modello di tipo evolutivo, perché, a nostro parere, è quello che si attiene di più alle nostre esigenze: il modello a cascata ci è sembrato troppo limitante, in particolare per la difficoltà di effettuare cambiamenti nel corso del processo. Volevamo infatti avere la libertà e la sicurezza di poter apportare modifiche più o meno rilevanti, durante l'intera evoluzione del progetto.

Pensavamo inoltre di adottare una prototipazione di tipo evolutivo per quanto concerne gli aspetti tecnici dell'applicazione: la costruzione di vari prototipi partendo da una descrizione di massima, fino ad evolvere al sistema finale. Inoltre la prototipazione di tipo "usa e getta" richiederebbe troppo tempo per funzionalità back-end.

Per quanto riguarda l'interfaccia grafica, applicheremo una prototipazione di tipo usa e getta, in quanto è molto facile proporre diverse soluzioni per poi scegliere la migliore.

Possibilmente cercheremo di adottare la tecnica del "Pair Programming" da remoto: un membro del team condividerà lo schermo mentre scrive il codice, mentre il collega potrà suggerirgli cambiamenti, nuove idee e segnalare eventuali errori. Siamo infatti convinti che lavorare in coppia sia il modo più semplice per sviscerare un problema e venirne a capo.

2.2 Struttura Organizzativa

La struttura organizzativa del team sarà di tipo democratico decentralizzato: l'assenza di un leader permanente giova al gruppo portando un'attitudine positiva, garantendo a tutti i membri del team di poter esprimere le proprie idee agli altri, senza un'imposizione superiore: garantisce una comunicazione orizzontale necessaria a sostenere un lavoro che si basa sulla continua creazione di prototipi.

I nostri canali di comunicazione saranno essenzialmente 3:

1. **WhatsApp**: utilizzato per comunicazioni veloci, quali organizzazione di meeting e scambi di opinioni e idee;
2. **Discord**: utilizzato come principale mezzo di comunicazione durante lo sviluppo (Pair programming) e per riunioni;
3. **GitHub**: utilizzato per sincronizzare modifiche apportate al codice di ogni singolo membro del team.

2.3 Interfacce Organizzative

Per chiarimenti riguardo a management e scheduling del progetto, ci rivolgeremo al professor Agostino Cortesi, mentre per questioni relative allo sviluppo in ambiente Android faremo riferimento al professor Alvis Spanò e al tutor Giulio Zausa.

Inoltre coinvolgeremo terze parti per testare ed ottenere feedback riguardo l'usabilità dell'interfaccia ed il corretto funzionamento dell'applicativo.

2.4 Responsabilità di Progetto

Il team sarà diviso in sottogruppi. Ognuno di essi lavorerà ad un ambito specifico (es. Creazione di Web server, creazione dell'interfaccia, gestione delle connessioni Admin/Cliente) e avrà la propria responsabilità. Ci aspettiamo comunque una certa flessibilità da parte di ogni componente del team a sostituire e/o integrare il lavoro non assegnatogli.

2.4.1 Relazioni pubbliche

Responsabile: Lotto Michele.

Funzioni: Relazioni tra Team e committente ed utenti.

Attività: Chiedere chiarimenti al committente, stipulare accordi e organizzare incontri tra il Team e le varie parti.

2.4.2 Interfaccia grafica

Responsabile: Chinellato Andrea.

Funzioni: Gestione dell'intera parte visuale dell'applicazione.

Attività: Scelta del layout in tutte le sue parti e gestione della scalabilità di tutta l'interfaccia.

2.4.3 Strutturazione del codice

Responsabili: Zuliani Riccardo.

Funzioni: Controllo sulla struttura e la leggibilità del codice.

Attività: Scelta dei patterns, suddivisione dei package, controllo sull'interazione e sulla comprensione del codice.

2.4.4 Stesura della documentazione

Responsabili: Furlan Alessandro.

Funzioni: Redazione e revisione dei documenti.

Attività: Redazione e controllo sulla correttezza, comprensibilità e standardizzazione dei documenti affiancati all'applicazione.

2.4.5 Configurazione Web Server

Responsabili: Gatto Michele.

Funzioni: Gestire tutte le funzioni che concernono con il server di supporto.

Attività: creazione di web server, gestione e sviluppo dello stesso.

3. Processi gestionali

3.1 Obiettivi e Priorità

L'obiettivo di questo progetto consiste nella realizzazione di un sistema di gestione dei noleggi di veicoli a pedali tramite un applicativo Android. Ciò aiuterebbe i negozianti ad informatizzare la loro attività, garantendo un livello di sicurezza e di affidabilità maggiore rispetto all'utilizzo di carta e penna.

Le nostre priorità sono:

- Rispettare le specifiche e le scadenze assegnate dal professore;
- Garantire agli utenti un sistema di qualità;
- Fornire i prototipi in maniera regolare per la validazione agli altri membri del team;
- Mantenere un gruppo coeso e propenso all'obiettivo.

3.2 Assunzioni, Dipendenze, Vincoli

Assumiamo che ogni membro del gruppo partecipi e si impegni attivamente affinché il progetto sia portato a termine con successo e nei tempi previsti. Inoltre si assume che tutti i membri abbiano una conoscenza almeno sufficiente per quanto riguarda lo sviluppo dell'applicazione e della documentazione, in caso contrario sarà loro priorità colmare le eventuali lacune. In aggiunta assumiamo che tutti i futuri utenti dell'applicazione abbiano uno smartphone con una versione di Android maggiore o uguale alla 7.0 (API 24 - Nougat).

Per quanto riguarda i vincoli abbiamo identificato il rispetto delle scadenze nei tempi previsti stabiliti dal professore e la capacità di conciliare il lavoro del progetto con gli impegni universitari.

Infine l'unica dipendenza individuata riguarda la velocità dell'apprendimento dell'ambiente di sviluppo in relazione alla realizzazione dell'intero progetto.

3.3 Gestione dei rischi

#	Identificazione rischio	Probabilità	Impatto	Strategia
1	Problemi di salute	Medio - bassa	Basso	Distribuzione: il carico di lavoro viene redistribuito equamente.
2	Complessità di sviluppo dell'app maggiore delle previsioni	Media	Alto	Valutazione di alternative: cambio di strategia nella realizzazione del prototipo.
3	Poca coordinazione e collaborazione tra le componenti del gruppo	Bassa	Alto	Riunione: si discute su come migliorare la situazione e l'ambiente di lavoro.
4	Abbandono di un componente del gruppo	Bassa	Alto	Accettarlo: il carico di lavoro del componente mancante viene redistribuito.
5	Non rispettare i termini della scadenza	Bassa	Alto	Disponibilità: i membri del gruppo si impegnano a garantire una maggiore disponibilità all'avvicinarsi della scadenza.
6	Perdita o danneggiamento del codice (durante un'operazione di merge)	Medio - Bassa	Alto	Supporto: Si cerca di aiutare e riparare il danno effettuato.
7	Sovrapposizione di impegni universitari	Alto	Medio	Organizzazione: ogni membro del gruppo ha il dovere di non trascurare i suoi compiti.
8	Scioglimento del gruppo	Bassa	Alta	Mitigazione: i membri del gruppo si impegnano a mantenere toni consoni all'ambiente di lavoro.
9	Utenti insoddisfatti	Medio	Medio	Raccolta feedback: Raccogliamo il maggior numero di feedback possibile al fine di migliorare l'esperienza utente.
10	Nuovo Lockdown	Medio-Alto	Medio	Attesa: Dobbiamo rimanere in attesa che la situazione venga sbloccata. Nel frattempo si può testare l'applicazione mediante un fake GPS.

Classificazione dei rischi

IMPATTO	PROBABILITÀ' DI VERIFICARSI				
	Bassa	Medio-Bassa	Media	Medio-Alta	Alta
Alto	3 - 4 - 5 - 8	6	2		
Medio-Alto					
Medio			9	10	7
Medio-Basso					
Basso		1			

3.4 Meccanismi di monitoraggio e controllo

Il progetto viene controllato e monitorato costantemente durante tutta la sua realizzazione. Ogni membro del team avrà un preciso compito e utilizzeremo GitHub per controllare le modifiche apportate da ogni componente. Alla fine della creazione di ogni prototipo, ci riuniremo tramite Discord per discutere e validare la versione proposta. Inoltre cercheremo di essere presenti (tutti e 5) ad almeno due incontri settimanali per poter discutere sull'andamento graduale di ogni prototipo.

3.5 Pianificazione dello staff

I prerequisiti utili per poter portare a termine il progetto sono:

- La conoscenza del linguaggio di programmazione Java;
- La conoscenza e la dimestichezza con il pacchetto di Google Docs;
- La conoscenza di GitHub per poter caricare e modificare codice utile allo sviluppo del progetto;

Se queste conoscenze non sono ancora state assimilate completamente, i membri in difficoltà dovranno informarsi e apprendere il necessario attraverso:

- Le lezioni del professor Alvise Spanò per quanto riguarda l'ambiente di sviluppo Android Studio e la programmazione in Java. Inoltre seguiranno anche sue spiegazioni riguardo l'utilizzo concreto di GitHub e le modifiche apportabili al codice Master presente online;
- Video tutorial per approfondire ed integrare ciò che risulta più ostico da assimilare;
- Spiegazione da parte dei membri del team più esperti;
- Lettura di testi tecnici e specifici relativi all'argomento interessato.

4. Processi tecnici

4.1 Metodi, Strumenti e Tecniche

Gli strumenti utilizzati per la realizzazione del progetto sono:

- IDE Android Studio per l'effettivo sviluppo dell'applicazione;
- GitHub per apportare modifiche al codice e caricarle sul sito;
- Google Docs per la stesura della documentazione;
- Discord per la comunicazione tra i membri del team;
- Un computer per ognuno dei membri per la programmazione e la consultazione;
- Smartphone Android (almeno 2: uno per la figura di negoziante e uno per la figura di cliente) per testare l'applicazione;
- Whatsapp per scambi di idee rapide.

Il metodo di sviluppo che adotteremo sarà basato sulla prototipazione: partiremo da una specifica generale per poi andare a sviluppare tre prototipi principali. Durante la fase dello sviluppo utilizzeremo la tecnica del pair programming per mantenere un tasso di concentrazione ed attitudine adeguato.

4.2 Documentazione del Software

La documentazione del Software sarà redatta in gruppo e consegnata entro le scadenze citate nel punto 1.2 - Deliverables del progetto. Si tiene a specificare che durante la stesura di questo documento alcune caratteristiche del progetto potrebbero variare e/o non sono del tutto definite.

I documenti necessari saranno consegnati al committente utilizzando la pagina Moodle del corso di Ingegneria del Software, in un'apposita sezione relativa al documento. Il gruppo inoltre conserverà una copia del documento per confrontarlo con i futuri, così da ottenere un'autovalutazione relativa ai cambiamenti effettuati o meno.

4.3 Funzionalità di supporto al progetto

Per quanto riguarda la pianificazione delle attività, bisogna rispettare alcuni punti chiave per assicurare un prodotto di qualità al cliente:

- Mantenere una documentazione aggiornata e ben dettagliata;
- Essere coerenti con il modello scelto;
- Fornire i prototipi e documentazione annessa a tempo debito per la validazione di essi;
- Mantenere un buon grado di coordinazione e di attitudine nel gruppo.

Sarà compito di ogni responsabile accertarsi che lo sviluppo relativo al suo campo proceda senza intoppi. Durante le due riunioni settimanali si discuterà sull'attuale andamento di ogni singola attività così da riferire al responsabile eventuali sviluppi.

Per la pianificazione della gestione delle configurazioni, occorrerà mantenere le varie versioni del codice in locale, onde evitare spiacevoli perdite di dati durante la fase di upload nella piattaforma GitHub.

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

5.1 WBS (Work breakdown structure)

1. Specifiche del progetto
 - 1.1. Definizione obiettivi generali e ambito del progetto
 - 1.1.1. Definizione concept dell'applicazione
 - 1.1.2. Identificazione degli utenti
 - 1.1.3. Definizione obiettivi e priorità
 - 1.2. Definizione risorse necessarie
 - 1.3. Analisi di fattibilità
 - 1.3.1. Proposta di fattibilità dell'intero progetto
 - 1.3.2. Progettazione di un eventuale piano alternativo
 - 1.4. Preparazione delle specifiche
 - 1.4.1. Definizione del ciclo di sviluppo
 - 1.4.2. Suddivisione del progetto in macroaree
 - 1.5. Definizione struttura del team
 - 1.5.1. Definizione del modello da adottare
 - 1.5.2. Definizione metodi di comunicazione
 - 1.5.3. Definizione responsabilità e responsabili
 - 1.5.4. Definizione dei sottogruppi
 - 1.6. Gestione dei rischi
 - 1.7. Stima dei costi
 - 1.8. Pianificazione del lavoro
 - 1.8.1. Definizione attività e preparazione WBS
 - 1.8.2. Definizione milestones
 - 1.8.3. Stime dei tempi necessari
2. Progettazione concettuale
 - 2.1. Analisi dei Requisiti
 - 2.1.1. Individuazione requisiti utente e di sistema
 - 2.1.2. Individuazione tipo di requisito
 - 2.2. Analisi e studio interfacce grafiche
3. Progettazione tecnica
 - 3.1. Scelta versione Android
 - 3.2. Scelta DBMS
 - 3.3. Scelta libreria di geolocalizzazione

4. Sviluppo del prodotto

4.1. Prototipo iniziale

4.1.1. Specifica

4.1.2. Sviluppo

4.1.3. Testing

4.1.4. Validazione

4.2. Prototipo intermedio

4.2.1. Specifica

4.2.2. Sviluppo

4.2.3. Testing

4.2.4. Validazione

4.3 Prototipo definitivo

4.3.1. Specifica

4.3.2. Sviluppo

4.3.3. Testing

4.3.4. Validazione

5. Rilascio Versione Definitiva

6. Documentazione

6.1. Piano di Progetto

6.2. Documento di analisi e specifica

6.3. Piano di testing

6.4. Documento di Progettazione

6.5. Manuale Utente

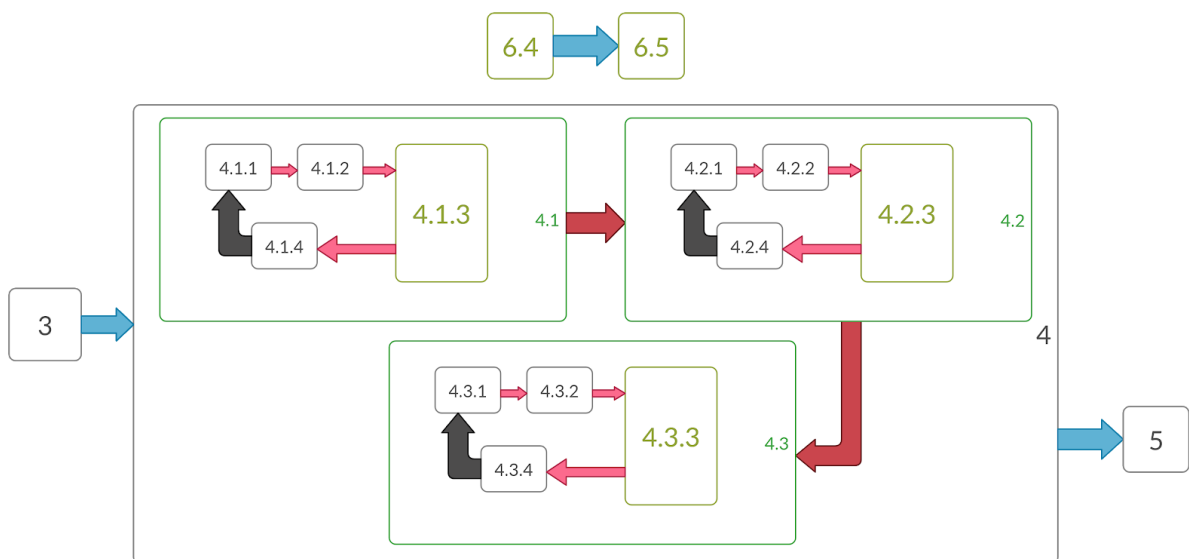
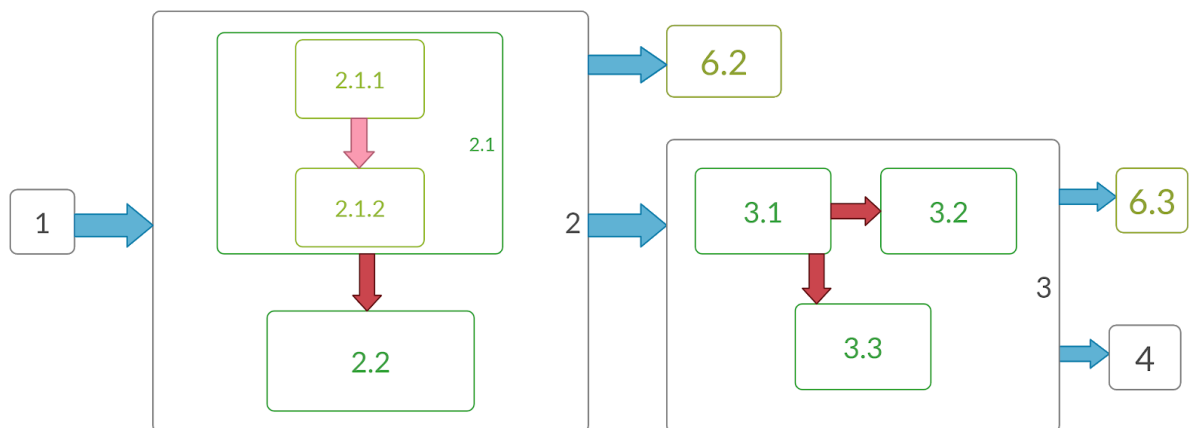
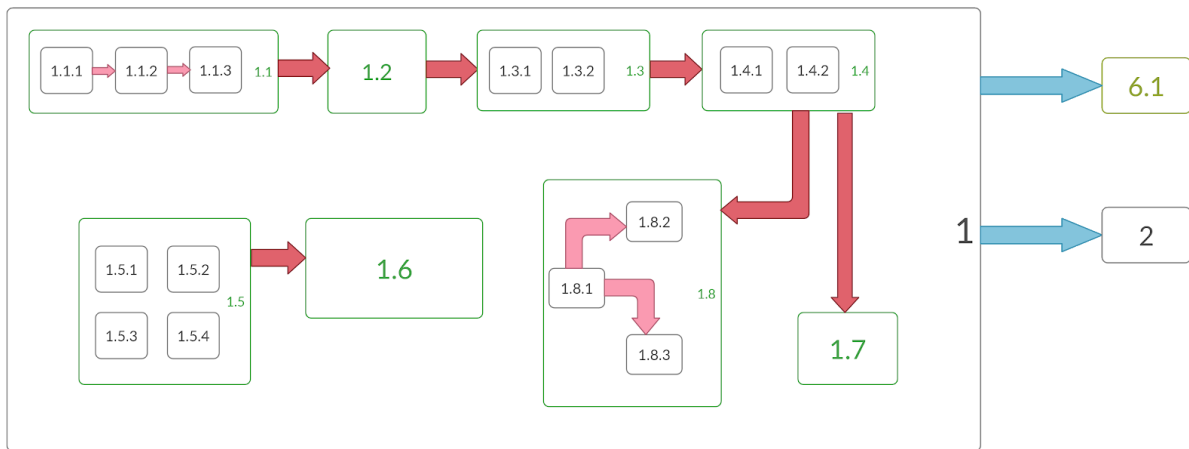
5.2 Dipendenze

5.2.1 Schedulazione del lavoro

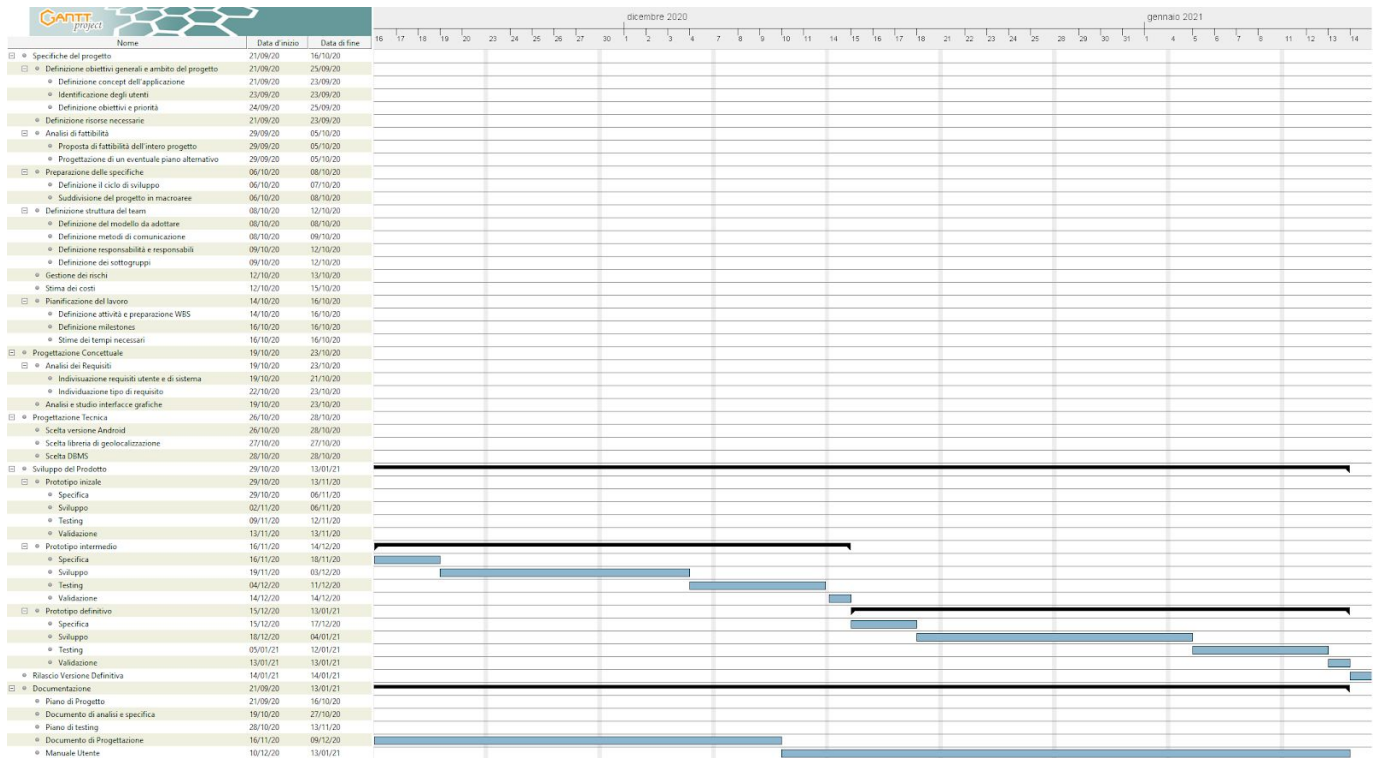
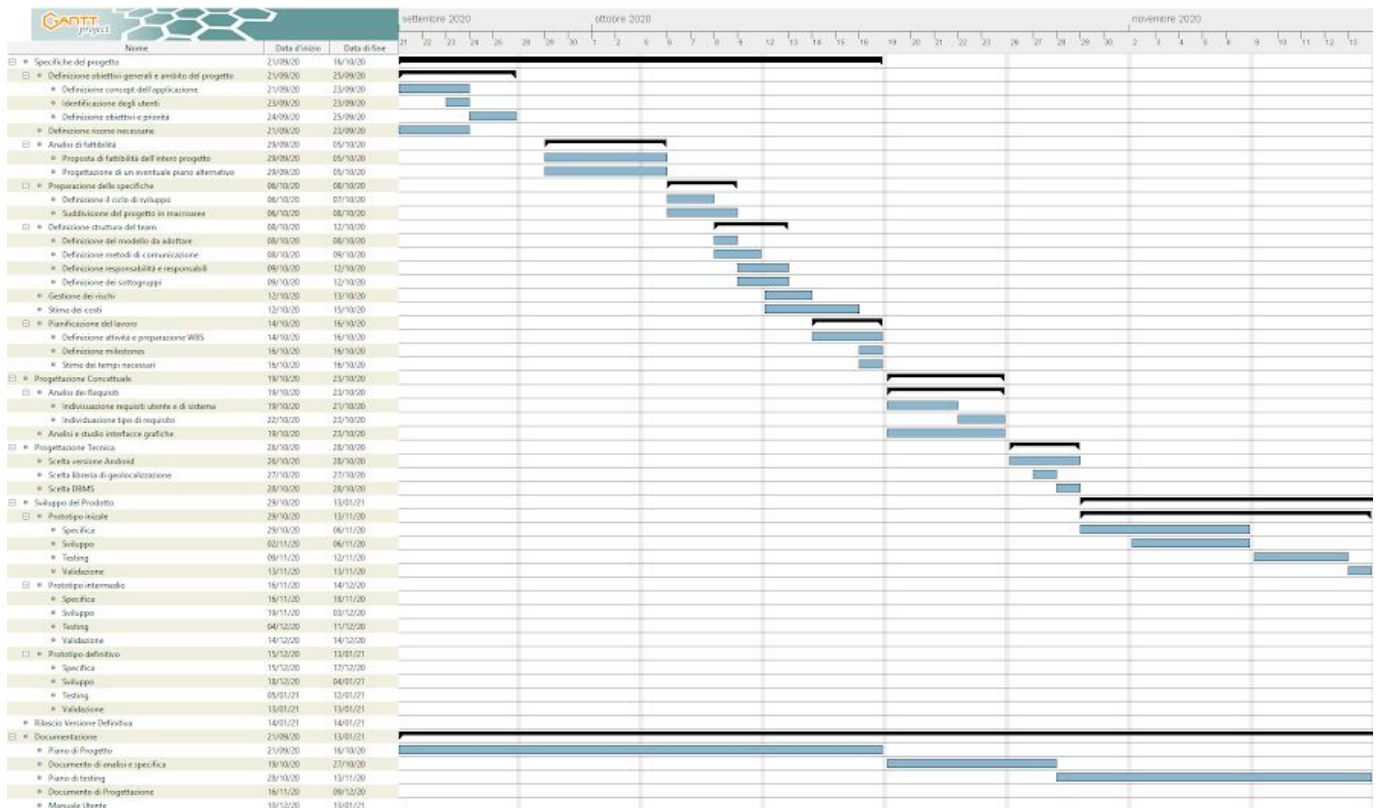
Attività		ES	EF	LS	LF	Giorni
1. Specifiche del progetto		18/09/20	14/10/20	21/09/20	16/10/20	26
	1.1 Definizione obiettivi generali e ambito del progetto	18/09/20	20/09/20	21/09/20	23/09/20	3
	1.1.1 Definizione concept dell'applicazione	18/09/20	20/09/20	21/09/20	23/09/20	3
	1.1.2 Identificazione degli utenti	20/09/20	20/09/20	23/09/20	23/09/20	1
	1.1.3 Definizione obiettivi e priorità	21/09/20	22/09/20	24/09/20	25/09/20	2
	1.2 Definizione risorse necessarie	18/09/20	20/09/20	21/09/20	23/09/20	3
	1.3 Analisi di fattibilità	27/09/20	03/10/20	29/09/20	05/10/20	7
	1.3.1 Proposta di fattibilità dell'intero progetto	27/09/20	03/10/20	29/09/20	05/10/20	7
	1.3.2 Progettazione di un eventuale piano alternativo	27/09/20	03/10/20	29/09/20	05/10/20	7
	1.4 Preparazione delle specifiche	03/10/20	05/10/20	06/10/20	08/10/20	3
	1.4.1 Definizione il ciclo di sviluppo	03/10/20	04/10/20	06/10/20	07/10/20	2
	1.4.2 Suddividere il progetto in macroaree	03/10/20	05/10/20	06/10/20	08/10/20	3
	1.5 Definizione struttura del team	05/10/20	09/10/20	08/10/20	12/10/20	5
	1.5.1 Definizione del modello da adottare	05/10/20	05/10/20	08/10/20	08/10/20	1
	1.5.2 Definizione metodi di comunicazione	05/10/20	06/10/20	08/10/20	09/10/20	2
	1.5.3 Definizione responsabilità responsabili	06/10/20	09/10/20	09/10/20	12/10/20	4
	1.5.4 Definizione dei sottogruppi	06/10/20	09/10/20	09/10/20	12/10/20	4
	1.6 Gestione dei rischi	9/10/20	10/10/20	12/10/20	13/10/20	2
	1.7 Stima dei costi	9/10/20	12/10/20	12/10/20	15/10/20	4
	1.8 Pianificazione del lavoro	11/10/20	13/10/20	14/10/20	16/10/20	3
	1.8.1 Definizione attività e preparazione WBS	11/10/20	13/10/20	14/10/20	16/10/20	3
	1.8.2 Definizione milestones	13/10/20	13/10/20	16/10/20	16/10/20	1
	1.8.3 Stime dei tempi necessari	13/10/20	13/10/20	16/10/20	16/10/20	1
2. Progettazione Concettuale		17/10/20	21/10/20	19/10/20	23/10/20	5
	2.1 Analisi dei Requisiti	17/10/20	21/10/20	19/10/20	23/10/20	5

	2.1.1 Individuazione requisiti utente e di sistema	17/10/20	19/10/20	19/10/20	21/10/20	3
	2.1.2 Individuazione tipo di requisito	20/10/20	21/10/20	22/10/20	23/10/20	2
	2.2 Analisi e studio interfacce grafiche	17/10/20	21/10/20	19/10/20	23/10/20	5
	<u>3. Progettazione Tecnica</u>	24/10/20	26/10/20	26/10/20	28/10/20	3
	3.1 Scelta versione Android	24/10/20	26/10/20	26/10/20	28/10/20	3
	3.2 Scelta libreria di geolocalizzazione	25/10/20	25/10/20	27/10/20	27/10/20	1
	3.3 Scelta DBMS	26/10/20	26/10/20	28/10/20	28/10/20	1
	<u>4. Sviluppo del Prodotto</u>	27/10/20	11/1/21	29/10/20	13/01/21	77
	4.1 Prototipo iniziale	27/10/20	11/11/20	29/10/20	13/11/20	16
	4.1.1 Specifica	27/10/20	4/11/20	29/10/20	06/11/20	9
	4.1.2 Sviluppo	31/10/20	04/11/20	02/11/20	06/11/20	5
	4.1.3 Testing	07/11/20	10/11/20	09/11/20	12/11/20	4
	4.1.4 Validazione	11/11/20	11/11/20	13/11/20	13/11/20	1
	4.2 Prototipo intermedio	14/11/20	12/12/20	16/11/20	14/12/20	29
	4.2.1 Specifica	14/11/20	16/11/20	16/11/20	18/11/20	3
	4.2.2 Sviluppo	17/11/20	1/12/20	19/11/20	03/12/20	15
	4.2.3 Testing	02/12/20	09/12/20	04/12/20	11/12/20	8
	4.2.4 Validazione	12/12/20	12/12/20	14/12/20	14/12/20	1
	4.3 Prototipo definitivo	13/12/20	11/1/21	15/12/20	13/01/21	30
	4.3.1 Specifica	13/12/20	15/12/20	15/12/20	17/12/20	3
	4.3.2 Sviluppo	16/12/20	2/1/21	18/12/20	04/01/21	18
	4.3.3 Testing	03/01/21	10/01/21	05/01/21	12/01/21	8
	4.3.4 Validazione	11/01/21	11/01/21	13/01/21	13/01/21	1
	<u>5. Rilascio Versione Definitiva</u>	14/01/21	14/01/21	14/01/21	14/01/21	1
	<u>6. Documentazione</u>	18/09/20	10/01/21	21/09/20	13/01/21	115
	6.1 Piano di Progetto	18/09/20	13/10/20	21/09/20	16/10/20	26
	6.2 Documento di analisi e specifica	16/10/20	24/10/20	19/10/20	27/10/20	9
	6.3 Piano di testing	25/10/20	10/11/20	28/10/20	13/11/20	17
	6.4 Documento di Progettazione	13/11/20	6/12/20	16/11/20	09/12/20	24
	6.5 Manuale Utente	7/12/20	10/1/21	10/12/20	13/01/21	35

5.2.2 Diagramma di PERT



5.2.3 Diagramma di GANTT



5.3 Risorse Necessarie

Le risorse necessarie per l'evoluzione del progetto sono le seguenti:

- Umane: sono gli effettivi membri del team e eventuali terzi per le fasi di testing.
- Software: tutto il software utile per lo sviluppo. Nel nostro caso tutti i software per la comunicazione già citati precedentemente e il software per lo sviluppo effettivo del codice sorgente: Android Studio.
- Hardware: ogni membro del team utilizzerà il proprio PC con una relativa connessione internet e uno Smartphone Android con connessione ad Internet per la fase di testing.

5.4 Allocazione del Budget e delle Risorse

Attività	Durata in Giorni	Budget in gg/u
Specifiche del progetto	26	130
Progettazione Concettuale	5	25
Progettazione Tecnica	3	15
Sviluppo del progetto	77	385
Rilascio versione definitiva	1	1
Documentazione	115	115
Totale	116	576

Valore del progetto: Considerati i dati presenti nella tabella soprastante abbiamo concluso che il valore stimato dell'applicazione si dovrebbe aggirare attorno ai 11.520€ (da non confondere con il prezzo di vendita). Durante questo calcolo abbiamo considerato di lavorare in media 2 ore al giorno e di chiedere una retribuzione di 10 €/hr. Inoltre si è assunto che in media ogni componente dedicherà circa un quinto del proprio lavoro alla stesura della documentazione relativa alle proprie attività.

5.5 Pianificazione – Deadlines e Milestones

La pianificazione è dettata dall'ordine delle deadlines:

1. Piano di Progetto (18/10/2020)
2. Documento di analisi e specifica (27/10/2020)
3. Piano di testing (14/11/2020)
4. Documento di Progettazione (09/12/2020)
5. Codice e messa in linea del sistema (15/01/2021)

In più sono state inserite 3 Milestones significative, ognuna relativa alla fase di validazione di ogni prototipo di sviluppo dell'applicazione.