



UnfinItaly

PIANO DI PROGETTO

Autori:

- **Ismailaj Fatjona - 860833**
- **Rossetto Elena - 857182**
- **Vettori Enrico - 857763**
- **Zanatta Giacomo - 859156**

INDICE

1.INTRODUZIONE	3
1.1 OVERVIEW DEL PROGETTO	3
1.2 DELIVERABLES DEL PROGETTO	3
1.3 EVOLUZIONE	4
1.4 MATERIALE DI RIFERIMENTO	4
1.5 DEFINIZIONI E ABBREVIAZIONI	4
2. ORGANIZZAZIONE DEL PROGETTO	5
2.1 MODELLO DEL PROCESSO	5
2.2 STRUTTURA ORGANIZZATIVA	5
2.3 INTERFACCE ORGANIZZATIVE	5
2.4 RESPONSABILITÀ DI PROGETTO	5
3. PROCESSI GESTIONALI	6
3.1 OBIETTIVI E PRIORITÀ	6
3.2 ASSUNZIONI, DIPENDENZE, VINCOLI	6
3.3 GESTIONE DEI RISCHI	6
3.4 MECCANISMI DI MONITORAGGIO E DI CONTROLLO	9
3.5 PIANIFICAZIONE DELLO STAFF	10
4. PROCESSI TECNICI	11
4.1 METODI, STRUMENTI, TECNICHE	11
4.2 DOCUMENTAZIONE DEL SOFTWARE	11
4.3 FUNZIONALITÀ DI SUPPORTO AL PROGETTO	12
5. PIANIFICAZIONE DEL LAVORO, DELLE RISORSE UMANE E DEL BUDGET	13
5.1 WBS (WORK BREAKDOWN STRUCTURE)	13
5.2 DIPENDENZE	14
5.3 RISORSE NECESSARIE	16
5.4 ALLOCAZIONE DEL BUDGET E DELLE RISORSE	17
5.5 PIANIFICAZIONE	17

1.INTRODUZIONE

1.1 OVERVIEW DEL PROGETTO

Il progetto in questione consiste nel presentare all'utente un applicazione Android in cui l'utente ha a disposizione una mappa interattiva delle opere pubbliche Italiane rimaste incomplete (aggiornate al 31/12/2016). I dati riguardanti tali opere sono prelevati da un dataset messo a disposizione dal Ministero delle Infrastrutture e dei Trasporti in modo del tutto trasparente e gratuito. Oltre all'applicazione Android verrà creata una pagina web dove l'utente potrà contattare direttamente il team di sviluppo e troverà informazioni riguardanti l'applicazione.

L'applicazione permetterà di:

1. visualizzare su mappa le opere incomplete;
2. filtrare la visualizzazione delle opere incomplete in base alla categoria;
3. filtrare la visualizzazione in base alla regione;
4. visualizzare in una scheda apposta per ogni opera incompiuta eventuali informazioni: l'ente promotore, l'identificativo CUP, una descrizione, la causa del fallimento, gli importi per l'ultimo qe approvato e ultimo sal, lo stato di avanzamento in percentuale;
5. visualizzare sulla mappa i dati con diverse modalità:
 - a. tramite 4 diversi colori dei pins per osservare la percentuale dell'avanzamento dei lavori (tra 0% e 25% rosso, tra 25% e 50% arancione, tra 50% e 75% giallo, tra 75% e 100% verde);
 - b. visualizzare sulla mappa la distribuzione delle opere tramite una serie di colori che evidenziano il numero di opere nel territorio agendo direttamente sulla cartina dell'Italia.

Il progetto è stato pensato avendo a cuore la sensibilizzazione degli utenti rispetto all'ambiente pubblico e lo sviluppo di una maggiore consapevolezza sulla gestione dei fondi pubblici in ambito architettonico e infrastrutturale, volendo mettere in risalto il degrado che le opere incomplete nel nostro paese.

1.2 DELIVERABLES DEL PROGETTO

1 - Piano di Progetto	08/11/2017 presso Via Torino 155 Mestre, DAIS
2 - Documento di analisi e specifica	10/11/2017 presso Via Torino 155 Mestre, DAIS
3 - Piano di testing	23/11/2017 presso Via Torino 155 Mestre, DAIS
4 - Documento di Progettazione	23/12/2017 presso Via Torino 155 Mestre, DAIS
5 - Realizzazione e messa in linea	31/01/2018 presso Via Torino 155 Mestre, DAIS

1.3 EVOLUZIONE

Il progetto vedrà la seguente evoluzione:

1. Analisi delle richieste, ricerca dei dati e ideazione dell'applicazione;
2. Progettazione più dettagliata dell'app;
3. Stesura della documentazione richiesta;
4. Sviluppo del codice.

Nel corso dello sviluppo del progetto è possibile vengano identificate nuove funzionalità (aggiuntive e non richieste) che potrebbero portare a un arricchimento del servizio fornito dall'applicazione. Queste potrebbero quindi essere introdotte in corso d'opera senza però andare a modificare la struttura generale e le funzionalità già previste dell'applicazione: un esempio è quello dell'inserimento di una componente social in cui gli utenti possano condividere immagini riguardanti le varie opere incompiute. Inoltre, grazie allo studio delle varie categorie di opere incompiute, alle date riguardanti l'ideazione dei lavori e l'effettivo inizio, e alle percentuali di completamento dei lavori, riteniamo possibile un'implementazione che riesca a scovare dati al di fuori dei valori medi per risaltare eventuali appalti truccati. Inoltre riteniamo possibile un'evoluzione dove dei processi interni all'applicazione controllino periodicamente la presenza di dati aggiornati e di opere nelle vicinanze.

1.4 MATERIALE DI RIFERIMENTO

- Slides del corso di Ingegneria del Software (Materiali ISA)
- Documentazione toolkit sviluppato per il progetto CEVID - <https://cevid.unive.it/open-data/toolkit/>
- Sito web StackOverflow per eventuali problematiche simili riscontrate da altri utenti
- Android 5 Programming by Example - Kyle Mew
- Android Programming for Beginners - John Orton
- Documentazione Google - <https://developer.android.com/reference/>
- Android Studio - <https://developer.android.com/studio>
- Linee guida Material Design - <https://material.io/>

1.5 DEFINIZIONI E ABBREVIAZIONI

<u>Android</u>	Sistema operativo per smartphone e altri dispositivi
<u>App.</u>	Applicazione che viene eseguita su uno smartphone
<u>Bug</u>	Problema riscontrato nell'applicazione
<u>Cloud</u>	Servizio di archiviazione online offerto da aziende
<u>CUP</u>	Codice Unico Progetto
<u>DAIS</u>	Dipartimento di Scienze Ambientali, Informatica e Statistica - Ca Foscari
<u>IDE</u>	Ambiente di sviluppo che agevola il programmatore nello sviluppo software
<u>SDK</u>	Software Development Kit
<u>Pin</u>	Marcatore nella mappa
<u>Pins</u>	Plurale di 'Pin'

<u>Processo</u>	Entità software che svolge una o più operazioni
<u>Responsive</u>	Interfaccia in grado di adattarsi a diversi tipi di risoluzione di schermo
<u>Tool</u>	Strumento di lavoro
<u>Thread</u>	Parte di un processo con scopo ben definito

2. ORGANIZZAZIONE DEL PROGETTO

2.1 MODELLO DEL PROCESSO

Il modello scelto per realizzare il progetto è il modello a cascata. Il team di sviluppo ha scelto questo modello per la sua semplicità e chiarezza nel definire le varie fasi di specifica e di sviluppo. Inoltre, data la dimensione non troppo elevata del progetto, il limite di questa scelta, ossia la difficoltà di effettuare cambiamenti nel corso del processo, pensiamo possa non incidere in maniera molto rilevante nel caso in cui si debba modificare qualche piccolo parametro o decisione in corso d'opera.

2.2 STRUTTURA ORGANIZZATIVA

Il team ha deciso di utilizzare una struttura di tipo democratico decentralizzato. Vista l'assenza di un leader permanente le decisioni del gruppo verranno prese in modo equo. La comunicazione orizzontale permetterà uno scambio di idee in modo semplice e veloce che pensiamo possa rendere più efficiente la gestione di eventuali problemi o errori.

2.3 INTERFACCE ORGANIZZATIVE

Nel corso del processo il gruppo si rivolgerà a entità esterne per:

- richiedere consulenze riguardanti l'andamento dello sviluppo del progetto (Professore Cortesi);
- richiedere eventuali spiegazioni aggiuntive riguardanti la programmazione Android (Dottor Spanò);
- coinvolgere un maggior numero di persone nel testing dell'applicazione.

2.4 RESPONSABILITÀ DI PROGETTO

Il gruppo di lavoro non è stato suddiviso per categorie di lavoro, di conseguenza tutte le mansioni verranno suddivise in modo equo tra tutti i membri sia per quanto riguarda la scrittura della documentazione sia per quanto riguarda la scrittura del codice sorgente.

3. PROCESSI GESTIONALI

3.1 OBIETTIVI E PRIORITÀ

L'obiettivo consiste nel creare un'applicazione per dispositivi Android che permetta di visualizzare nella mappa le opere incompiute e applicare a questi dati dei filtri (vedere introduzione 1.1).

Il gruppo si pone come priorità e obiettivi:

- Produrre un prodotto di qualità,
- Rispettare le scadenze stabilite,
- Collaborare,
- Rispettare i vincoli del progetto.

3.2 ASSUNZIONI, DIPENDENZE, VINCOLI

Assunzioni:

Gli utenti sono persone di età compresa tra i 16 e 70 anni, interessati a comprendere meglio eventuali sprechi di denaro nella realizzazione di opere incompiute.

Gli utenti usano dispositivi Android con accesso a Internet.

Dipendenze:

Il software verrà sviluppato quasi interamente con il linguaggio Java.

Utilizzo di smartphone e tablet con sistema operativo Android (versione minima richiesta 5.0).

Vincoli:

Il primo vincolo che si incontra è quello temporale, in quanto vogliamo rispettare le consegne del progetto stando però anche al passo con gli altri corsi che seguiamo: questo comporta la collaborazione da parte di tutti i membri del gruppo.

3.3 GESTIONE DEI RISCHI

Essendo un progetto che si sviluppa in un certo intervallo di tempo, bisogna prendere in considerazione gli eventuali rischi che possono sorgere e che è necessario vengano gestiti lavorando in gruppo.

Alcuni dei rischi possono essere:

1. **Problemi di salute:** in questo periodo c'è un maggior rischio di ammalarsi e di conseguenza rallentare lo sviluppo del progetto. In questo caso bisogna essere pronti ad affrontare questo rischio aiutandosi a vicenda senza lasciare che il proseguimento dei lavori ne risenta;
2. **Difficoltà con il linguaggio di programmazione:** ci si deve impegnare di più per raggiungere la meta e i membri del gruppo devono aiutarsi a vicenda per avere dei risultati migliori;

3. **Perdita del lavoro fatto:** questo rischio potrebbe andare a danneggiare irrimediabilmente il progetto, è necessario quindi effettuare un backup del lavoro fatto per eliminare la possibilità di riscontrare dei problemi di questo genere. Inoltre grazie alle soluzioni cloud usate il rischio non è frequente ma da tenere in considerazione;
4. **Guasto software ai computer:** questo rischio di per sé non è un gran problema, perché con le nostre conoscenze siamo in grado di riparare il danno. Se dovesse avvenire un guasto software, il problema più grande sarebbe la perdita di tempo per rimediare al guasto. Il caso peggiore prevede una formattazione del disco e una reinstallazione del software;
5. **Guasto hardware ai computer:** se dovesse avvenire un guasto hardware ad una macchina da noi utilizzata per sviluppare il progetto, la soluzione è quella di cercare di cambiare l'hardware guasto, oppure prevedere l'utilizzo di una macchina di riserva, quali ad esempio i computer del laboratorio di informatica del campus;
6. **Bugs:** durante lo sviluppo è molto probabile che si trovino bachi sul codice. È necessario, per evitare situazioni spiacevoli, definire un appropriato piano di testing per consentire di trovare eventuali bug e di conseguenza correggerli. Inoltre, per evitare di scrivere codice malfunzionante, si potrebbe attuare il peer programming. Questo tipo di rischio ha un impatto variabile, in quanto un bug potrebbe essere scovato subito e corretto facilmente e in modo efficace, oppure potrebbe risultare complesso da trovare e gestire.

Tabella dei rischi

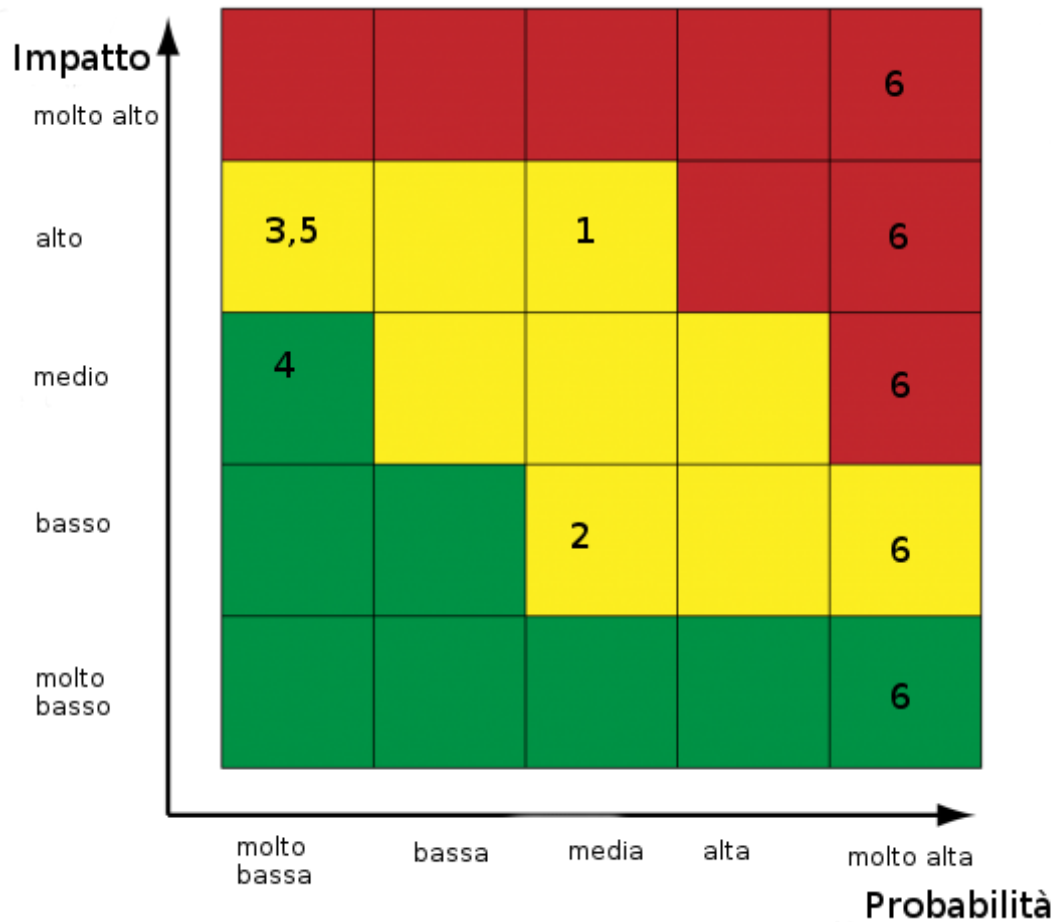
ID	RISCHI	CATEGORIA	PROBABILITÀ	IMPATTO	AZIONE
1.	Problemi di salute	Personale	Medio	Alto	Collaborare con gli altri membri.
2.	Difficoltà con il linguaggio di programmazione	Sviluppo	Medio	Basso	Rivedere le regole di programmazione di Android, documentarsi su Internet.
3.	Perdita del lavoro fatto	Sviluppo	Molto basso	Alto	Ripristinare una copia del lavoro fatto dal backup salvato in cloud. Il codice sorgente su GitHub è al sicuro da eventuali perdite.
4.	Guasto software ai computer	Materiali	Molto basso	Medio	Cercare di sistemare il programma in errore. Nel caso non si riesca a farlo, formattare il computer.

5	Guasto hardware ai computer	Materiali	Molto basso	Alto	Procurarsi componenti di ricambio o, nel caso peggiore, una macchina che sostituisca quella guasta.
6	Bugs	Sviluppo	Molto alto	Variabile	Rivedere il codice e apportare modifiche.

Prevenzione e monitoraggio dei rischi

ID	RISCHI	CATEGORIA	PREVENZIONE	MONITORAGGIO
1.	Problemi di salute	Personale	Non esporre il fisico a situazioni di malattia	Controllo sintomi comuni della febbre, controllo temperatura corporea
2.	Difficoltà con il linguaggio di programmazione	Sviluppo	Studiare il linguaggio	Controllo del codice scritto
3.	Perdita del lavoro fatto	Sviluppo	Utilizzare il Cloud e Git	Controllare che il lavoro sia stato sincronizzato online
4.	Guasto software ai computer	Materiali	Non installare programmi pericolosi	Controllare eventuali segnali di errore software
5	Guasto hardware ai computer	Materiali	Non esporre il computer a situazioni di pericolo (quali calore)	Controllare l'hardware e tenere aggiornati i driver
6	Bugs	Sviluppo	Prestare attenzione alla scrittura del codice	Controllo di errori nel codice scritto

Matrice dei rischi



3.4 MECCANISMI DI MONITORAGGIO E DI CONTROLLO

Il progetto verrà sviluppato in gruppo e quindi ci sarà il bisogno di condividere il lavoro fatto con gli altri membri affinché ognuno lo possono guardare, controllare ed eventualmente criticare.

Inoltre ogni componente deve monitorare in modo periodico il lavoro fatto per non perdere la continuità ed essere sempre al passo con lo sviluppo per rientrare nei tempi delle scadenze.

3.5 PIANIFICAZIONE DELLO STAFF



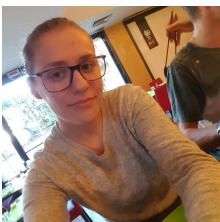
Giacomo Zanatta: Conoscenza base di GitHub, ottima familiarità con l'IDE Android Studio, buona conoscenza del linguaggio di programmazione Java, conoscenza di base di sviluppo applicazioni Android. Ottima capacità di problem solving. Attualmente si sta impegnando per migliorare le proprie capacità di programmazione e di project management.



Enrico Vettori: Conoscenza basilare di GitHub, conoscenza discreta dell'IDE Android Studio, conoscenza discreta del linguaggio di programmazione Java e dei principali pattern, nessuna conoscenza di sviluppo applicazioni Android, nessuna conoscenza di linguaggio XML. Buona capacità di problem solving.



Elena Rossetto: Conoscenza base di GitHub, bassa conoscenza nello sviluppo di applicazioni Android e nell'uso di Android Studio. Buona capacità di problem solving. Si sta impegnando nel miglioramento delle sue capacità di programmazione.



Fatjona Ismailaj: Conoscenza base di GitHub, bassa conoscenza dell'IDE Android Studio e nello sviluppo di applicazioni Android. Buona capacità di problem solving.

4. PROCESSI TECNICI

4.1 METODI, STRUMENTI, TECNICHE

Sistemi di calcolo

Per sviluppare questo progetto useremo il software Android Studio disponibile per Linux, Mac OSX e Microsoft Windows. Una delle caratteristiche dello sviluppo di software per dispositivi Android è la sua indipendenza dal Sistema Operativo della macchina con cui si andrà a sviluppare. Ogni membro del gruppo sceglierà quindi il Sistema Operativo che più gli aggrada.

L'SDK di Android (fornito in bundle con Android Studio) fornisce utili strumenti di testing per l'applicazione, permettendo di creare dispositivi virtuali in cui eseguire il debug del progetto.

Inoltre, useremo i nostri dispositivi fisici (due OnePlus 3, dotati di Android 7.1 e 8.0, un Galaxy J6 dotato di Android 5.1, un Huawei P9 Lite dotato di Android 7.0) per verificare il corretto funzionamento dell'applicazione su dispositivi eterogenei (per la versione del Sistema Operativo, e per l'hardware) ed emulatori virtuali per testare il software.

Metodi di sviluppo

Ogni membro del gruppo deciderà in base alle sue preferenze e capacità su cosa concentrarsi dello sviluppo. Sarà necessaria la collaborazione tra i membri e l'aiuto reciproco per portare a termine il progetto nel migliore dei modi. Lo sviluppo dell'applicazione verrà inoltre accompagnata da prototipi interattivi per simulare l'applicazione prima di sviluppare la parte del codice sorgente.

Tecniche

Useremo le linee guida sulla stesura di applicazioni Android fornite da Google e dal Dott. Spanò Alvisè. In particolare, ci affideremo alle linee guida del Material Design per lo sviluppo dell'interfaccia grafica e ad eventuali Design Pattern. Questo permetterà di abbracciare la filosofia grafica di Google e di realizzare un'interfaccia grafica funzionale, semplice da usare e responsive.

4.2 DOCUMENTAZIONE DEL SOFTWARE

Il software verrà documentato durante lo sviluppo del progetto, nello specifico verranno redatti i seguenti documenti:

- Piano di Progetto
- Documento di analisi e specifica
- Piano di testing
- Documento di Progettazione

I documenti nella loro versione finale verranno consegnati entro il 31/01/2017

4.3 FUNZIONALITÀ DI SUPPORTO AL PROGETTO

Sviluppare un progetto con codice di qualità è il principale obiettivo del gruppo. In particolare si desidera:

- Mantenere aggiornata la documentazione;
- Rispettare le linee guida di sviluppo Android;
- Utilizzare Programming Pattern quando richiesto per uno sviluppo il più possibile riusabile, privo di errori e collaudato;
- Rispettare la struttura organizzativa;
- Scrivere il codice più performante possibile vista la categoria di dispositivi sui quali verrà eseguita l'applicazione;
- Eseguire periodicamente testing sul codice sorgente scritto, in modo da provare tutti i casi particolari possibili e rilevare eventuali bugs.

Per lo sviluppo del progetto il team utilizzerà programmi di supporto quali:

- **GitHub**: servizio che permette di tenere nel cloud il codice sorgente dell'applicazione, di tenere traccia di tutte le modifiche che vengono apportate con relativa cronologia e di permette una sincronizzazione di lavoro tra tutti i membri del progetto;
- **Google Drive**: servizio cloud offerto da Google, in cui i componenti del gruppo condividono una cartella dove sono inseriti tutti i vari documenti scritti a computer e scaricati da internet;
- **Fogli e Documenti Google**: il gruppo ha optato per scrivere i documenti utilizzando i fogli e i documenti che Google permette di condividere in modo da scrivere tutti nello stesso file anche contemporaneamente;
- **InvisionApp**: tool online per creare prototipi interattivi e Storyboard;
- **Adobe Experience Design**: tool di Adobe che permette di creare interfacce grafiche Android in modo rapido e accattivante;
- **Telegram**: piattaforma di comunicazione open source.

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

5.1 WBS (WORK BREAKDOWN STRUCTURE)

- A. Definizione linee generali del progetto (2 giorni), vengono definite le idee base del progetto.
- B. Ricerca dati (2 giorni): vengono cercati i dati aperti necessari per sviluppare l'applicazione.
- C. Definizione e creazione ambiente di lavoro (1 giorno): in questa fase viene configurato l'ambiente di lavoro (verrà installato Android Studio, preparata la repository, creato il progetto e configurato l'accesso a gitHub da Android Studio).
- D. Decisione ruoli interni (1 giorno): in questa fase viene effettuata una riunione tra i membri del gruppo per stabilire i ruoli interni e l'organizzazione.
- E. Stesura calendario (1 giorno): viene stabilito un calendario di progetto, in base alle scadenze da rispettare.
- F. Identificazione delle fasi di sviluppo (1 giorno): vengono identificate e analizzate le varie fasi che il progetto dovrà superare per poter essere portato a termine.
- G. Definizione delle risorse (1 giorno): definiamo le risorse che abbiamo a disposizione, a partire da noi stessi.
- H. Definizione del budget (1 giorno): viene definito il budget a disposizione e l'eventuale prezzo del progetto.
- I. Definizione struttura organizzativa del progetto (1 giorno): definiamo l'organizzazione del progetto.
- J. Analisi dei rischi (1 giorno): da ciò che è emerso dai punti precedenti, analizziamo i possibili rischi che potrebbero accadere durante il progetto.
- K. Stesura del piano di progetto (4 giorni): dopo aver completato i punti precedenti, stiliamo il piano di progetto.
- L. Analisi documentazione CEVID (2 giorni): viene analizzata attentamente la documentazione messa a disposizione dal Professor Cortesi Agostino.
- M. Definizione interfacce utente (2 giorni): viene redatta una bozza e un eventuale prototipo dell'interfaccia utente.
- N. Stesura documento di analisi e specifica (9 giorni): in questa fase viene definito il documento di analisi e specifica, richiesto come secondo assignment. Il termine di questa fase coincide con la data di consegna.
- O. Stesura piano di testing (12 giorni): viene elaborato il piano di testing.
- P. Stesura documento di progettazione (24 giorni): viene elaborato e scritto il documento di progettazione, quarto assignment del corso.
- Q. Sviluppo interfaccia utente (3 giorni): dopo aver steso il documento vero e proprio, inizierà lo sviluppo vero e proprio dell'applicazione. In questa fase, viene sviluppata l'interfaccia utente, sulla base della bozza definita precedentemente.
- R. Scrittura del codice dell'applicazione e della landing page (32 giorni): questa lunga fase consiste nella stesura del codice dell'applicazione e della landing page. In

seguito alla stesura del piano di progettazione e del piano di testing, potrà venire suddivisa in sottofasi.

- S. Test applicazione e debugging (10 giorni): questi 10 giorni consistono nel testare l'applicazione e nella correzione di eventuali bug riscontrati.
- T. Revisione applicazione (2 giorni): viene eseguito un check completo dell'applicazione, al termine del quale sarà pronta per la pubblicazione.
- U. Test finale (2 giorni): viene effettuato un ultimo grande test per essere sicuri del corretto funzionamento.
- V. Pubblicazione sul Play Store (1 giorno): l'ultima fase consiste nella pubblicazione del progetto sul Play Store di Google.

5.2 DIPENDENZE

ID	Attività	Durata (giorni)	Dipendenza
A	Definizione linee generali del progetto	2	-
B	Ricerca dei dati	2	A
C	Definizione e creazione ambiente di lavoro	1	G
D	Decisione dei ruoli interni al gruppo	1	A
E	Stesura calendario	1	A
F	Identificazione delle fasi di sviluppo	1	A
G	Identificazione delle risorse	1	A
H	Definizione del budget	1	D, G
I	Definizione struttura organizzativa del progetto	1	A
J	Analisi dei rischi	1	A, B, C, D, E, F, G, H
K	Stesura del piano di progetto	4	J
L	Analisi documentazione CEVID	2	-
M	Definizione interfacce utente	2	K
N	Stesura documento di analisi e specifica	9	M, L
O	Stesura piano di testing	12	N
P	Stesura documento di progettazione	24	O
Q	Sviluppo interfaccia utente	3	P
R	Scrittura del codice dell'applicazione	32	Q

S	Test applicazione e debugging	10	R
T	Revisione applicazione	2	S
U	Test finale	2	T
V	Pubblicazione sul PlayStore	1	U

Diagramma di Pert

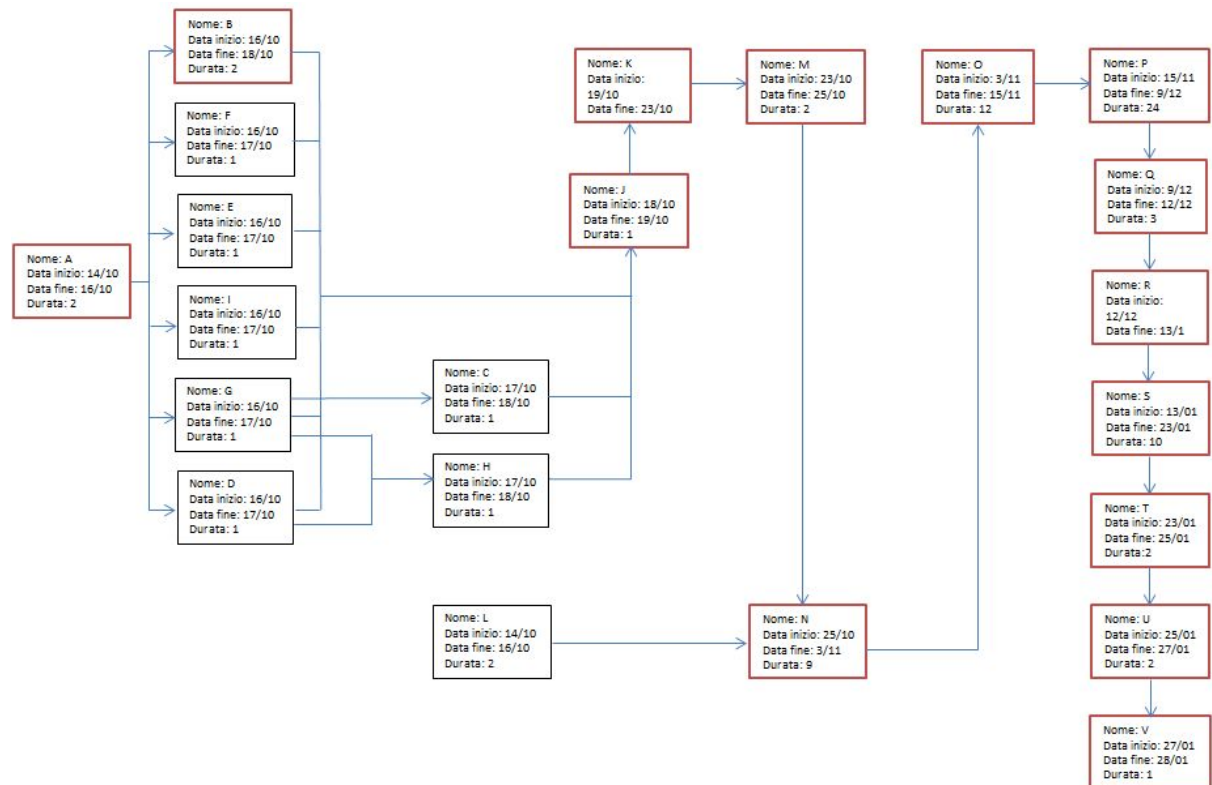
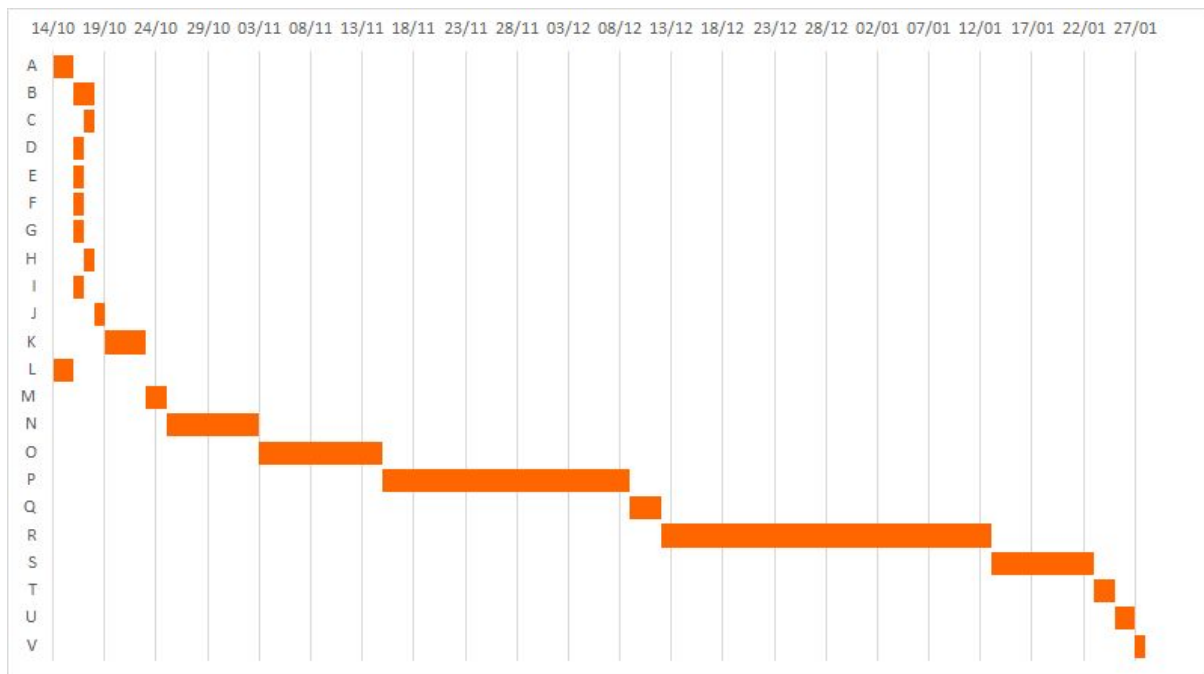


Diagramma di Gantt



5.3 RISORSE NECESSARIE

Le risorse necessarie alla realizzazione del progetto includono:

- Risorse umane: i 4 componenti del gruppo e eventuali persone che prestano del loro tempo per eseguire dei test;
- Risorse software: in ogni computer dei componenti del gruppo dovrà essere installato Android Studio obbligatoriamente, le restanti risorse software nominate possono essere sfruttate con un Browser e Adobe Experience Design;
- Risorse hardware: ogni componente del gruppo necessita di un calcolatore abbastanza performante da poter eseguire Android Studio. L'applicazione non richiede particolare Hardware aggiuntivo;

5.4 ALLOCAZIONE DEL BUDGET E DELLE RISORSE

Ogni membro del gruppo ha già a disposizione un calcolatore in grado di eseguire Android Studio, inoltre tutti gli strumenti che verranno utilizzati sono forniti in modo gratuito. I dati sulla quale si baserà l'applicazione sono open data forniti dal governo Italiano e quindi accessibili gratuitamente.

Essendo lo sviluppo del progetto un assignment del corso di Ingegneria del Software non verrà richiesto alcun costo per sviluppare l'applicazione.

Tuttavia, per questioni di trasparenza e rispetto verso il cliente riteniamo sia utile far notare i costi di sviluppo dell'applicazione, descritti qui di seguito:

- Account sviluppatore Google: 20€
- Ipotizzando 3 mesi di lavoro, 10 giorni di lavoro a mese, 4 ore al giorno (causa tempo dedicato allo studio) si ha un totale di $3 \times 10 \times 4 = 120$ ore a componente, calcolando un costo di 13€/h e considerando che il gruppo è composto da 4 membri il totale è:
 $120 \times 13 \times 4 = 6240$ €.
- Costo server Web: 40€
- Preventivo complessivo: 6300€

5.5 PIANIFICAZIONE

Il progetto seguirà la pianificazione stabilita dal professor Cortesi nel corso di Ingegneria del Software (Anno Accademico 2017/2018):

- | | |
|--|------------|
| - 1 - Piano di Progetto | 08/11/2017 |
| - 2 - Documento di analisi e specifica | 10/11/2017 |
| - 3 - Piano di testing | 23/11/2017 |
| - 4 - Documento di Progettazione | 23/12/2017 |
| - 5 - Realizzazione e messa in linea | 31/01/2018 |