**PROJECT PLAN**

***1. INTRODUZIONE:***

Lo scopo del progetto è creare un software per la gestione di un ospedale, in particolare per tenere traccia dei pazienti operati e degli interventi a cui devono sottoporsi. L'obbiettivo principale è raggiungere una forma stabile e funzionante del programma, anche se non ancora commercializzabile, soprattutto dal punto di vita grafico.

*Spiegazione Problema*

Il paziente va in visita dal medico e si decide che deve essere operato; il medico inserisce il paziente nel sistema, il quale contiene una pagina anagrafica per ogni paziente (all’interno sono contenute le informazioni anagrafiche e mediche del paziente ed è importante che in ogni momento si possa accedere a queste informazioni).

Il paziente entra in una lista di graduatoria dell’intervento specifico a cui si deve sottoporre. La cabina di regia contatta il paziente per fornirgli le date degli esami che dovrà eseguire prima dell’intervento. Il paziente svolge gli esami e, una volta completati, viene chiamato per fare l’intervento; a questo punto, la cabina di regia lo inserisce nella lista delle persone pronte per fare l’intervento, in modo che il medico, una settimana prima, può preparare la lista operatoria, che schedula giorno per giorno chi verrà operato (le liste operatorie possono essere modificate in qualsiasi momento dal medico).

Nel giorno dell’intervento si tiene traccia di tutti i movimenti, i rischi, l'anestesia adoperata, la diagnosi e la procedura relative all'operazione eseguita sul paziente in un verbale; nel verbale sono inoltre contenuti gli orari di inizio e fine dell'intervento, le tecniche effettuate durante l'operazione e i nomi dell'equipe medica cha ha effettuato l'operazione (i ruoli e le competenze rispetto al verbale possono variare per ogni componente).

Il software, quindi, deve tenere traccia di tutte le liste a cui il paziente appartiene, nonché monitorare come vengono gestiti i vari pazienti e medici per ogni intervento effettuato.

***2. MODELLO DEL PROCESSO***

Il Modello del progetto in questione è un modello *A Cascata*:

* ogni fase del progetto deve essere definita e la successiva fase fa riferimento alla precedente.
* per ogni fase si richiede Verifica e Validazione (V&V), rispettivamente per soddisfare i requisiti del progetto stesso (è necessario valutare la correttezza del passaggio alla fase successiva) e per capire se il sistema risponde alle esigenze dell'utente (dobbiamo capire se stiamo costruendo il progetto in modo corretto).

Prima della stesura del progetto stesso, è necessaria un'attenta analisi per capire le varie problematiche legate ad esso.

***3. ORGANIZZAZIONE DEL PROGETTO***

Il gruppo di lavoro è composto da tre persone:

* Nicola Cattaneo
* Matteo Mangili
* Marco Saini

I tre collaboratori si dividono i vari compiti:

- **Stesura del Project Plan** / **Presentazione Progetto** (Mangili)

- **Creazione dei Diagrammi** (Saini)

- **Stesura codice sorgente** (Cattaneo).

Spesso è necessario lavorare in coppia, al fine di ottimizzare la collaborazione tra i singoli individui e minimizzare il margine di errore.

I tre collaboratori lavorano principalmente da soli per le varie parti, tenendosi aggiornati quotidianamente, e in settimana si incontrano nell'Aula Studio dell'Edificio C dell'Università degli Studi di Bergamo per discutere sull'andamento del progetto.

GitHub è lo strumento prescelto per la gestione della documentazione relativa al progetto sia per la condivisione di codice, diagrammi e richieste di modifica.

***4. STANDARD, LINEE GUIDA, PROCEDURE***

Il software implementato è un’architettura *Client-Server*:

- **Client** (Paziente che deve essere operato)

- **Server** (Sistema dell'Ospedale)

***5. ATTIVITA' DI GESTIONE***

Per gestire tutte le parti prodotte nello sviluppo del progetto è necessario fissare delle attività di gestione:

* Ogni membro del gruppo lavora sui suoi compiti e in caso di difficoltà, si rivolge agli altri membri, al fine di risolvere problematiche.
* Ogni settimana i membri del gruppo si incontrano per definire eventuali dubbi, sistemare gli errori e per fare un resoconto di quanto fatto nei precedenti giorni.

Tali attività permettono un controllo continuo sullo sviluppo del progetto. Prima di apportare una modifica, ogni membro deve confrontarsi con gli altri collaboratori, al fine di capire se tale modifica è necessaria e, in caso affermativo, ottenere la loro approvazione a procedere.

***6. RISCHI***

I rischi maggiori sono:

- mancare la consegna finale del progetto causa impegni universitari e non di ciascun membro del team.

- trovare giorni della settimana in cui tutti i membri del team potevano incontrarsi, per potersi confrontare insieme sul progetto, causa impegni extra-scolastici

di ognuno.

- difficoltà a sviluppare un software che rispondesse a tutte le esigenze/requisiti del problema proposto.

***7. PERSONALE E TECNICHE***

Il team è composto da tre persone:

* Nicola Cattaneo (project manager, progettista database, product owner, frontend)
* Matteo Mangili (backend, staticals models developper, tester)
* Marco Saini (frontend, reviewer, staticals models developper)

***8. METODI E TECNICHE***

Per modellare il programma con un'ampia visione dell'insieme di funzioni del progetto, si sfruttano vari tipi di Diagrammi in UML.

Per prima cosa, vanno identificati i casi d'uso e gli attori del problema proposto, al fine di costruire lo *USE CASE DIAGRAM*.

Vengono, in seguito, definite le varie classi, le relazioni tra di esse e tutti gli attributi e metodi di ciascuna classe, che verranno implementate nel *CLASS DIAGRAM*.

I legami che uniscono le singole classi possono essere visti come messaggi, ciò permette di definire lo *SEQUENCE DIAGRAM*, con cui è possibile identificare l'ordine corretto di istruzioni da eseguire per raggiungere lo scopo del problema.

Se si vuole vedere nel dettaglio lo stato di un oggetto del problema, si sfrutta lo *STATE DIAGRAM.*

Per visualizzare/comprendere le azioni che specificano il comportamento definito dall'utente rispetto al sistema generale, si sfrutta l'*ACTIVITY DIAGRAM*.

***9. GARANZIA DI QUALITA'***

Per la progettazione del software applicativo, si fa riferimento alla norma ISO 9001 (specifica i requisiti per un sistema di qualità per l'organizzazione, che deve dimostrare la propria capacità di fornire prodotti che soddisfino i requisiti dei clienti) e alle sue varie voci (in particolar modo alla sezione Pianificazione), per garantire il successo della creazione e utilizzo del progetto.

***10. PACCHETTI DI LAVORO***

Il Progetto è stato suddiviso in diverse parti distinte:

1. Intervista ad un'infermiera dell'Ospedale Papa Giovanni XXIII di Bergamo (BG), per raccogliere le informazioni generali alla stesura dell'intero progetto.

2. *Project Plan,* per definire i vari passaggi della stesura del progetto.

3. *Diagrammi,* per avere una rappresentazione schematica del progetto.

4. *Codice*, per elaborare il programma attraverso un linguaggio di alto livello.

Ogni membro del team collabora in maniera attiva allo sviluppo dei vari componenti.

***11. RISORSE***

Per la realizzazione finale del progetto il team ha usufruito di varie risorse:

* Ogni membro ha sfruttato il proprio computer portatile (PC) o il proprio tablet per lavorare sulle singole componenti dell'intero progetto.
* Utilizzo di varie applicazioni, quali *Google Drive, Microsoft Teams, Microsoft OneDrive* per raccogliere le informazioni necessarie al fine di realizzare un buon progetto.
* *Eclipse IDE*, per la stesura del codice.
* *StarUML,* per la creazione dei vari diagrammi.
* *GitHub,* per condividere il progetto e notificare/gestire eventuali modifiche.

***12. BUDGET***

Il tempo è molto importante per la stesura dell'intero progetto: stimando che ogni membro del team lavorerà 60 h per conto proprio, per un totale complessivo di 180 h, la maggior parte di esse verrà impegnata nella progettazione, mentre le restanti saranno di studio (al fine di raccogliere nozioni utili) e revisione (per controllare/risolvere eventuali errori).

***13. CAMBIAMENTI***

Ogni cambiamento apportato al progetto viene discusso tra i membri del team: qualora un membro volesse modificare una parte del progetto, è tenuto ad avvisare i restanti membri della modifica effettuata, attraverso l'uso di app di messaggistica.

L'uso di GitHub permette ai membri del team di poter essere sempre aggiornati sullo sviluppo del progetto, qualora la comunicazione diretta non fosse possibile.

***14. CONSEGNA***

La consegna finale del Progetto avverrà entro il 05/02/2024, tenendo aperta la possibilità di eventuali manutenzioni e modifiche entro il 10/02/2024