



Laurea Magistrale in informatica-Università di Salerno  
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

# Statement of Work

## Progetto

# Chemioterapia

Versione	1.0
Data	19/10/2022
Destinatario	Istituzioni medico-sanitarie
Presentato da	Alessandro Bergamo, Francesco Pio Ianuzziello
Approvato da	Filomena Ferrucci

---



Laurea Magistrale in informatica-Università di Salerno  
Corso di Gestione dei Progetti Software- Prof.ssa F. Ferrucci

## Revision History

---

Data	Versione	Descrizione	Autori
17/10/2022	0.1	Prima stesura del documento	A. Bergamo, F. P. Ianuzziello
18/10/2022	0.2	Revisione degli scenari	A. Bergamo, F. P. Ianuzziello
19/10/2022	1.0	Conclusione del documento e revisione	A. Bergamo, F.P. Ianuzziello



# Statement of Work (SOW) - Progetto "Chemioterapia"

---

## 1. Piano Strategico/Strategic Plan

---

Le istituzioni mediche e sanitarie intendono possedere una gestione sempre più di qualità, efficace e vicina alle esigenze dei pazienti migliorando la gestione del tempo, degli spazi sanitari (sale d'attesa, poltrone, reparti, ecc.) e del personale.

## 2. Obiettivi di Business/Business Needs

---

Le istituzioni mediche e sanitarie intendono possedere uno strumento che possa semplificare la schedulazione e organizzazione dell'uso dei farmaci chemioterapici e del calendario delle terapie, in modo tale da diminuire i tempi necessari e organizzare meglio il numero del personale che si occupa di tali mansioni.

## 3. Ambito del Prodotto/Product Scope

---

L'obiettivo del progetto è fornire uno strumento di ottimizzazione per l'uso di farmaci chemioterapici e la schedulazione delle terapie. Deve supportare:

- uno schedatore intelligente dei calendari delle terapie, dell'uso delle stanze e delle poltrone,
- registrazione su app di pazienti, farmaci, terapie, personale, parametri e schedulazione dei calendari,
- l'utilizzo di algoritmi genetici e modelli di deep learning per l'ottimizzazione della schedulazione.

## 4. Data di Inizio e di Fine

---

Data di Inizio: Ottobre 2022

Data di Fine: Gennaio 2023



## 5. Scenari

### Scenario as-is

Marco è un paziente che deve ricevere una cura a base di farmaci chemioterapeutici in seguito alla diagnosi di una grave malattia. Marco dovrà sostenere una terapia all'ospedale. Viene convocato dall'ospedale alle ore 10:00 del mattino, ma recatosi in sede si accorge che non è il solo ad esser stato convocato per quell'orario. Banalmente, l'ospedale convoca tutti i pazienti che devono effettuare la terapia allo stesso orario, intasando quelli che sono gli spazi sanitari e distribuendo i pazienti nel corso della giornata facendo attendere i pazienti molte ore inutilmente. Marco accorgendosi di questa situazione si ritrova ad esser più frustrato del dovuto, abbassando anche quella che è la sua fiducia e considerazione nei confronti dell'ospedale.

### Scenario Visionario

Marco è un paziente che deve ricevere una cura a base di farmaci chemioterapeutici in seguito alla diagnosi di una grave malattia. Marco dovrà sostenere una terapia all'ospedale, pertanto viene inserito dall'ospedale sulla piattaforma di gestione dei pazienti e delle terapie ad essi associate. Tale piattaforma schedula per Marco le date e gli orari in cui dovrà effettuare le varie fasi della terapia. Grazie all'utilizzo di tale piattaforma, Marco non dovrà più attendere inutilmente all'ospedale, ma potrà recarsi direttamente all'orario definito dalla piattaforma, aiutandolo a risparmiare tempo della sua giornata ed a rendere tutto il processo di terapia meno tedioso.

## 6. Deliverables

- Project Management: business case, charter, team contract, scope statement, WBS, schedule, PM Plan, cost baseline, status reports, final project presentation, final project report, lessons-learned report, e ogni altro documento richiesto per gestire il progetto.
- Di Prodotto: RAD, SDD, ODD, Matrice di Tracciabilità, Test Plan, Test Case Specification, Test Incident Report, Test Summary Report, Manuale D'Uso, Manuale Installazione e ogni altro documento richiesto per lo sviluppo del sistema.



## 7. Vincoli/Constraints

### Vincoli collaborativi e comunicativi.

- Rispetto scadenze delle scadenze intermedie/di fine progetto (\*/**definite dai project manager**, per i progetti di tipo A, e **definite nello statement of work**, per i progetti di tipo B/\*)
- Budget/Effort non superiore a  $50 * n$  ore dove  $n$  sono i membri del team (compresi PM)
- Uso di sistemi di versioning - GitHub in particolare
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo
- Utilizzo di tool di per la suddivisione dei task e attività (Trello)
- Utilizzo di tool di comunicazione tracciabile (Slack)

### Vincoli tecnici

#### *Analisi e specifica dei requisiti*

- Specifica di **minimo 2** e **massimo 4** scenari per ogni membro del team;
- Specifica di **minimo 2** e **massimo 4** requisiti funzionali e non funzionali per ogni membro del team;
- **Esattamente** uno use case per ogni membro del team - i casi d'uso aggiuntivi **non** saranno valutati;
- **Esattamente** un sequence diagram ogni due membri del team - i sequence diagram aggiuntivi **non** saranno valutati;
- **Esattamente** un diagramma a scelta tra statechart e activity diagram ogni due membri del team - ulteriori diagrammi **non** verranno valutati;
- Specifica di un class diagram per team - eventuali object diagram **non** verranno valutati.

#### *System Design*

- Specifica di **minimo 2** e **massimo 4** design goal per ogni membro del team.
- Definizione di **un diagramma** di decomposizione dei sottosistemi per team, con annessa descrizione e motivazione all'uso.
- Definizione di **un deployment diagram** per team, con annessa descrizione e motivazione all'uso.



### **Object Design**

- Uso di **minimo** uno e **massimo** due design pattern per team (devono essere selezionati tra quelli presentati a lezione);
- Uso di UML.

### **Testing**

- Ogni studente dovrà effettuare il testing di unità, tramite category partition, di **esattamente** un metodo di una classe sviluppata.
- Ogni studente dovrà effettuare il testing di sistema, tramite category partition, di **esattamente** una funzionalità del sistema sviluppato.

## 8. Criteri di Accettazione/Acceptance Criteria (Criteri che, se non rispettati, portano al fallimento del progetto)

---

- Utilizzo appropriato di GitHub, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Adeguato utilizzo del pull-based development, che preveda il **rispetto delle linee guida definite nel contesto del primo lab**.
- Adeguato utilizzo di Slack, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Adeguato utilizzo di Trello, che preveda il **rispetto delle linee guida definite nel contesto del secondo lab**.
- Documentazione adeguata. Verranno usati tool di **plagiarism detection** per identificare casi in cui gli studenti hanno copiato da progetti di anni precedenti e/o da altre fonti.
- Appropriato test di unità di un metodo sviluppato, che preveda il **rispetto dei vincoli**.
- Appropriato test di sistema di una funzionalità del sistema sviluppato, che preveda il **rispetto dei vincoli**.

## 9. Criteri di premialità

---

- Uso adeguato di sistemi di **build**;



- Uso adeguato di un processo di **continuous integration** tramite Travis;
- Uso adeguato di tool di controllo della qualità (ad esempio, **CheckStyle**);
- Adozione di processi di **code review**;
- Uso adeguato di tool avanzati di testing (e.g., **Mockito**, **Cobertura**, etc.).