



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof C. Gravino

Statement of Work

Progetto

MediCare

| | |
|---------------|---|
| Riferimento | |
| Versione | 0.1 |
| Data | 07/10/2023 |
| Destinatario | Prof C. Gravino |
| Presentato da | Andrea Gisolfi, Giacomo Favale, Giovanni Nigro, Antonio Merola |
| Approvato da | |



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof C. Gravino

Revision History

| Data | Versione | Descrizione | Autori |
|------------|----------|---------------|---|
| 07/10/2023 | 0.1 | Prima stesura | Andrea Gisolfi, Giacomo Favale, Giovanni Nigro, Antonio Merola |



Statement of Work (SOW) del Progetto

MediCare

1. Piano Strategico/Strategic Plan

Il team di MediCare intende fornire un accesso rapido ed efficace all'assistenza medica nazionale. La missione del team è quella di incoraggiare giovani e non ad interagire con il sistema medico al fine di diffondere consapevolezza e conoscenza, e a migliorare l'esperienza del paziente. Inoltre, promuovendo l'utilizzo tramite applicazione, punta ad avvicinare le nuove generazioni alla prevenzione medica.

2. Obiettivi di Business/Business Needs

MediCare ha come obiettivo ultimo quello di fornire un servizio per il sistema medico nazionale, che abbatti l'affluenza nelle sale d'attesa e che metta il paziente a suo agio nel fare domande. Inoltre, si impegna ad aiutare il paziente nella ricerca degli ospedali vicini per evitare situazioni indesiderabili.

3. Ambito del Prodotto/Product Scope

L'obiettivo del progetto è fornire un'applicazione di supporto ai cittadini fornendo una rapida geolocalizzazione degli ospedali, la possibilità di prenotare una visita medica, oltre ad un primo riscontro medico tramite chat-bot. L'applicazione deve supportare:

- fase di autenticazione,
- localizzazione ospedali vicini tramite mappa,
- ricerca ospedali tramite nome e/o zona,
- gestione di un sistema di prenotazione,
- presenza di un chat-bot a cui fare domande su possibili malattie basandosi su sintomi.

4. Data di Inizio e di Fine

Inizio: Ottobre 2023

Fine: Gennaio-Febbraio 2024



5. Deliverables

- 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.

6. Vincoli/Constraints

Vincoli collaborativi e comunicativi.

- Rispetto scadenze delle scadenze intermedie/di fine progetto **definite nello statement of work**
- Budget/Effort non superiore a $50 \cdot n$ ore dove n sono i membri del team
- 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 o similare)
- 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.

7. 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**.



8. 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.).