



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

Statement of Work

Progetto

SmartGym

Riferimento	thISTeam_SOW
Versione	1.0
Data	12/11/2022
Destinatario	Docente di Ingegneria del Software 2022/23 (Carmin Gravino)
Presentato da	thISTeam (FABIANO Daniele, MASELLI Mariantonietta, SPINELLI Michele)
Approvato da	



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

Revision History

Data	Versione	Descrizione	Autori
18/10/2022	0.1	Prima Stesura	thlSTeam
29/10/2022	0.2	Revisione	thlSTeam
12/11/2022	1.0	Revisione	thlSTeam



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

Statement of Work (SOW) del Progetto SmartGym

1. Piano Strategico/Strategic Plan

La Palestra Nettuno intende ampliare la sua clientela attraverso un programma di allenamento “Bodyweight Challenge” per avvicinare dei novizi e recuperare vecchi atleti, definendo un modo innovativo di accedere alla scheda degli esercizi.

2. Obiettivi di Business/Business Needs

La Palestra Nettuno pianifica di creare un sistema digitale per distribuire la scheda degli esercizi, in modo da diminuire l'uso del cartaceo e ottimizzare i tempi e i mezzi di rilascio. Si offre la possibilità di scegliere tra diversi modalità di creazione della scheda, adatti sia agli atleti che vogliono lavorare in autonomia che agli atleti assistiti dai trainer.

3. Ambito del Prodotto/Product Scope

L'obiettivo del prodotto è fornire uno strumento di supporto alle attività di allenamento assicurando che tutti gli atleti coinvolti possano gestire la scheda degli esercizi.

Deve supportare le seguenti caratteristiche base:

1. La registrazione degli utenti all'applicazione (atleti e trainer).
2. L'inserimento da parte dell'atleta delle sue attuali caratteristiche fisiche e di esperienza.
3. La selezione da parte dell'atleta, del trainer che lo assisterà.
4. La creazione della scheda degli esercizi nelle seguenti modalità:
 - 4.1. Manuale
 - 4.2. Generata
 - 4.3. Trainer
5. L'operazione di modifica e cancellazione della scheda degli esercizi.
6. La valutazione del trainer da parte dell'atleta, a conclusione del periodo di allenamento.

Inoltre, può supportare le seguenti caratteristiche aggiuntive:

7. La visualizzazione di schede degli esercizi condivise dai trainer.
8. Uno storico complessivo degli allenamenti svolti dall'atleta.



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

Scenario relativo alla Funzionalità 1:

Daniele è un ragazzo sportivo ma non si è mai approcciato al fitness, decide quindi di informarsi sulle possibilità presenti nelle vicinanze. Scopre la Palestra Nettuno e installa la loro applicazione sul suo dispositivo. All'avvio dell'app viene pregato di registrarsi, per farlo inserisce all'interno degli appositi campi i suoi dati anagrafici personali (Nome, Cognome, Data di Nascita, Email...).

Scenario relativo alla Funzionalità 2:

Daniele vuole accedere alla sua prima scheda degli esercizi ma viene notificato dall'app con un avviso che gli impedisce di continuare, ricordandogli di completare il profilo. Accede alla sezione dedicata al profilo e inserisce le caratteristiche mancanti (Sesso, Peso, Altezza, Esperienza, Obiettivo settimanale...) sbloccando la funzionalità per ottenere una scheda degli esercizi.

Scenario relativo alla Funzionalità 3:

Mariantonietta si è recata per la prima volta in palestra e ha conosciuto di persona un trainer. Decide di iniziare un periodo di allenamento, assistita dal trainer in questione. Mariantonietta accede all'applicazione e ricerca il trainer, lo seleziona e attende l'accettazione per poi dare inizio al periodo di allenamento.

Scenario relativo alla Funzionalità 4 (4.1-4.2-4.3)

Mariantonietta accede all'applicazione per scegliere la scheda degli esercizi. Si trova davanti a tre possibilità:

1. Manuale: Mariantonietta crea la scheda in autonomia, scegliendo dall'elenco dei possibili esercizi presenti in app.
2. Generata: Mariantonietta non deve preoccuparsi della creazione e in base alle sue caratteristiche fisiche e di esperienza, l'applicazione genererà in maniera intelligente la scheda degli esercizi.
3. Trainer: Mariantonietta si è precedentemente accordata con un trainer al quale lascia il compito di creare la scheda a lei dedicata.

Mariantonietta decide di scegliere la terza opzione, e resta in attesa della ricezione della propria scheda.



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

Scenario relativo alla Funzionalità 5:

Nel corso della sua permanenza in palestra, Mariantonietta ha l'esigenza di effettuare dei cambiamenti alla scheda degli esercizi. Decide di modificare la scheda degli esercizi che avverrà in modalità diverse a seconda della scheda scelta:

1. Manuale: Mariantonietta modifica la scheda in autonomia, scegliendo dall'elenco dei possibili esercizi presenti in app.
2. Generata: Mariantonietta non deve preoccuparsi delle modifiche, le basterà aggiornare le sue caratteristiche fisiche e di esperienza. L'applicazione la notifica che, visti i cambiamenti alle caratteristiche, verrà generata una nuova scheda.
3. Trainer: Mariantonietta si è precedentemente accordata con un trainer al quale lascia il compito di modificare la scheda a lei dedicata.

Scenario relativo alla Funzionalità 6:

Michele ha completato un periodo di allenamento con il suo trainer. Nel momento in cui conferma il completamento del periodo, dall'app ha la possibilità di lasciare una valutazione al trainer attraverso una scala da 1 a 5 su determinati aspetti (Puntualità, Disponibilità, Cordialità, Professionalità...).

Scenario relativo alla Funzionalità 7:

Michele ha cominciato a lavorare come trainer per la Palestra Nettuno ed è in cerca di potenziali nuovi atleti. Per far conoscere il suo metodo di allenamento e il suo stile, decide di creare alcune schede degli esercizi e di pubblicarle sull'app nell'apposita sezione, in modo da renderle visibili a tutti gli atleti.

Scenario relativo alla Funzionalità 8:

Mariantonietta ha svolto vari mesi di allenamento e vuole visualizzare il progresso di questi mesi. Accedendo all'applicazione nella sezione storico, può visualizzare e controllare per ogni settimana passata, in quali giorni si è allenata e quali esercizi ha svolto.



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

4. Data di Inizio e Fine

Inizio: Ottobre 2022.

Fine: Gennaio-Febbraio 2023.

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 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 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.



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

- 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.).