



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



Statement of Work

Comun-ity

Riferimento	
Versione	1.0
Data	08/11/2022
Destinatario	Comuni Italiani
Presentato da	Testa Elio, Iannucci Michele
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
10/10/2022	0.1	Prima stesura	Testa Elio, Iannucci Michele
17/10/2022	0.2	Revisione dell'idea e creazione degli scenari con Criteri d'accettazione	Testa Elio, Iannucci Michele
19/10/2022	0.3	Revisione del documento	Testa Elio, Iannucci Michele
08/11/2022	0.4	Aggiunta Logo e Revisione	Testa Elio
24/01/2023	1.0	Revisione finale	Testa Elio, Iannucci Michele



Statement of Work (SOW) del Progetto

Comun-ity

1. Piano Strategico/Strategic Plan

I Comuni hanno come obiettivo nel prossimo triennio il corretto impiego dei fondi PNRR in materia di sostenibilità, transizione ecologica e transizione digitale. Il fine è quello di migliorare la qualità e lo stile di vita dei cittadini, rendendosi al contempo più attrattivi agli occhi di nuovi investitori e residenti. In relazione a ciò si cercherà di migliorare anche quelli che sono gli indicatori Ministeriali che riguardano principalmente la qualità dell'aria.

2. Obiettivi di Business/Business Needs

I Comuni intendono fornire sia ai propri cittadini che a futuri residenti un servizio di “Smart Community” che permetta loro di risolvere piccoli problemi di vita quotidiana in maniera semplice ed efficace, soddisfacendo le loro esigenze ed andando a sfruttare e rafforzare il senso di comunità digitale.

3. Ambito del Prodotto/Product Scope

L'obiettivo del progetto è quello di fornire uno strumento di supporto alle attività e alle necessità quotidiane dei cittadini, migliorandone la qualità della vita. La piattaforma dovrà permettere:

- a chiunque di poter chiedere e dare aiuto per commissioni o favori;
- ai professionisti accreditati di dare aiuto per lavoretti;
- di organizzare spostamenti utilizzando il minor numero possibile di auto.

4. Data di Inizio e di Fine

Inizio: Ottobre 2022

Fine: Gennaio-Febbraio 2023. E' possibile concordare la data di consegna che potrà essere una delle seguenti:

- I: dal 9 al 25 Gennaio 2023
- II: dal 26 Gennaio al 10 Febbraio 2023



- III: dal 13 al 24 Febbraio 2023

5. 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.

6. 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 $50 \cdot n$ 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 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.).

9. Situazione attuale

Alice si è da poco trasferita in un paese di provincia per motivi di lavoro. Se sfortunatamente si ammala di Covid o ha un guasto alla lavatrice, ha bisogno di qualcuno che possa darle una mano. Attualmente l'unica cosa che può fare è chiedere aiuto, magari bussando i vicini di casa se si trova in un condominio, o chiedendo in giro numeri di telefono. Un'alternativa potrebbe essere l'utilizzo di un gruppo Facebook cittadino.

Se non esiste, Alice è costretta a crearlo e deve sperare che gli abitanti lo notino e decidano di aderirvi in breve tempo, in modo da poterla aiutare.

Se invece il gruppo esiste già, allora Alice posta l'annuncio. Ma come fa a capire se qualcuno decide di aiutarla? Deve contattare tutte le persone che semplicemente mettono “mi piace” al post? E se invece decidono di mandarle un messaggio, deve controllare continuamente la sezione “richieste di messaggi”? E se diverse persone le scrivono, come fa a scegliere a chi affidarsi? In fondo non conosce nessuno, potrebbe scegliere per simpatia o in ordine temporale. Una volta trovata la persona, deve cancellare il post per evitare che le arrivino altri messaggi? Ed una volta scelta la persona, se questa poi si tira indietro, Alice deve riscrivere il post?



10. Scenario proposto

Primo Caso: richiesta di aiuto per favori o commissioni

Alice accede alla piattaforma Comun-ity ed inserisce una richiesta di aiuto: ha bisogno che qualcuno le faccia la spesa.

Bob che è un cittadino modello, entra nella piattaforma e visualizza l'annuncio di Alice. Egli decide quindi di aiutarla, prendendo in carico la richiesta.

Il sistema notifica ad Alice che Bob è pronto ad aiutarla.

Nel caso in cui Bob annulli l'azione, il sistema rende di nuovo visibile l'annuncio.

Dopo aver ricevuto aiuto, Alice lascia un feedback sull'affidabilità di Bob.

Criteri d'accettazione Primo Caso:

- Notifiche in tempo reale dello stato della richiesta.

Secondo Caso: richiesta di un lavoretto

Alice accede alla piattaforma Comun-ity ed inserisce una richiesta di aiuto: ha bisogno di un idraulico ma non conosce nessuno in città.

Bob che è un professionista accreditato, entra nella piattaforma e visualizza l'annuncio di Alice. Egli decide quindi di aiutarla, prendendo in carico la richiesta.

Il sistema notifica ad Alice che Bob è pronto ad aiutarla.

Nel caso in cui Bob annulli l'azione, il sistema rende di nuovo visibile l'annuncio.

Dopo aver ricevuto aiuto, Alice lascia un feedback sull'affidabilità di Bob.

Criteri d'accettazione Secondo Caso:

- Permettere solo ai professionisti accreditati di rispondere a richieste lavorative;
- Notifiche in tempo reale dello stato della richiesta.