



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

Statement of Work

Progetto

UniPass

Versione	1.1
Data	10/02/2023
Presentato da	Barretta Alessandro Di Lauro Antonio Malafronte Sabato Zampaglione Gerardo
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
18/10/2022	0.1	Prima stesura	[Gruppo]
04/11/2022	0.2	Rettifica di alcune macro-funzionalità e relativi scenari	[Gruppo]
07/11/2022	0.3	Aggiunta macro-funzionalità	[Gruppo]
12/12/2022	0.4	Aggiunta macro-funzionalità	[Gruppo]
16/12/2022	1.0	Revisione pre-consegna intermedia	[Gruppo]
10/02/2023	1.1	Cambiamento della data di consegna	[Gruppo]



Statement of Work (SOW) del Progetto

UniPass

1. Piano Strategico/Strategic Plan

Il cliente, viste le difficoltà organizzative presenti nel sistema di trasporto regionale, desidera realizzare un software che permetta agli studenti dell'Università di Salerno di organizzare il proprio viaggio di ritorno, alleggerendo il carico agli enti di trasporto pubblico ed evitando disagi agli studenti per eventuali difficoltà logistiche, assicurandogli il ritorno a casa.

2. Obiettivi di Business/Business Needs

Attualmente i continui ritardi e la carenza di mezzi pubblici necessari a garantire il ritorno a casa agli studenti stanno creando grossi disagi. Per questo motivo si è deciso di sviluppare un sistema che fornisca la possibilità di organizzare viaggi tra studenti tramite una piattaforma di car pooling, garantendo ad essi un'alternativa ai sistemi di trasporto pubblico tradizionali.

3. Ambito del Prodotto/Product Scope

Il sistema deve:

- Consentire agli utenti di registrarsi alla piattaforma fornendo le proprie informazioni.
- Consentire ai guidatori di inserire le informazioni riguardanti il proprio veicolo.
- Consentire ai guidatori di creare un viaggio inserendo destinazione, data e orario di partenza, posti disponibili e prezzo.
- Consentire ai passeggeri di cercare viaggi compatibili in base a dei filtri (destinazione, data e orario di partenza, prezzo).
- Consentire ai passeggeri di prenotare un viaggio tra quelli disponibili.
- Consentire ai guidatori di chiudere le prenotazioni dei loro viaggi (chiusura automatica a 30 minuti dall'orario di partenza).
- Consentire ai guidatori di escludere passeggeri prenotati ai suoi viaggi.
- Prevedere un meccanismo di valutazione dei guidatori da 1 a 5 stelle.



- Prevedere un meccanismo di valutazione dei passeggeri da 1 a 5 stelle.
- Consentire a tutti gli utenti registrati di modificare le informazioni del proprio profilo.
- Consentire a tutti gli utenti registrati di visualizzare lo storico dei viaggi.
- Consentire ai guidatori di eliminare un viaggio da loro creato in precedenza.
- Consentire ai passeggeri di eliminare una prenotazione precedentemente effettuata.
- Consentire ai guidatori di modificare le informazioni del loro veicolo.
- Consentire ai guidatori di rimuovere il veicolo associato al proprio profilo.

Scenari:

Scenario: Gennaro, studente dell'Università di Salerno, ha perso il pullman di ritorno dall'università verso Pompei delle 14:00. Il prossimo pullman verso Pompei è in arrivo alle 17:30. Non volendo attendere così a lungo, decide di iscriversi ad UniPass e si mette alla ricerca di un viaggio verso Pompei per la data odierna con partenza alle ore 15:30. Gennaro seleziona uno dei viaggi tra quelli ottenuti dalla ricerca e prenota un posto.

Scenario: Biagio, studente del Dipartimento di Informatica all'Università di Salerno, raggiunge abitualmente il campus con la propria auto. Di solito è da solo, quindi decide di iscriversi ad UniPass e registrare il proprio veicolo.

Scenario: Carmine, dottorando in Ingegneria Industriale all'Università di Salerno, è iscritto ad UniPass. Nella data odierna deve partire alle 16:00 per tornare a casa sua in località Sarno. Decide quindi di creare un viaggio sulla piattaforma inserendo le informazioni relative ad esso.



Scenario: Costantino ha appena completato il viaggio di ritorno verso Cava de' Tirreni ma la sua esperienza non è stata ottimale: il guidatore durante il viaggio ha compiuto diverse infrazioni al codice stradale. Arrivato a casa, Costantino decide quindi di lasciare una valutazione di 2 stelle al guidatore.

Scenario: Michele ha utilizzato per la prima volta UniPass come guidatore ed ha appena completato il suo primo viaggio con due passeggeri. Egli si è trovato molto bene con i due ragazzi e decide così di lasciare una valutazione di 5 stelle ad entrambi.

Scenario: Elmo, utente abituale di UniPass, effettua una ricerca per un passaggio verso Anghi con partenza alle ore 14:00. Questa genera tre risultati. A questo punto Elmo controlla il suo storico viaggi e nota che ha già viaggiato con uno dei tre guidatori a cui aveva attribuito una valutazione di 5 stelle. Decide quindi di selezionare tra i tre il viaggio che ha lo stesso guidatore.

Scenario: Adele, iscritta ad UniPass, ha cambiato il suo numero di telefono. Decide quindi di recarsi sul suo profilo ed effettuare una modifica delle informazioni personali, cambiando nello specifico il suo numero di telefono. Terminata l'operazione, Adele salva le modifiche.

4. Data di Inizio e di Fine

Inizio: ottobre 2022

Fine: febbraio 2023. La data di consegna è stata fissata al 17 febbraio 2023.



5. Deliverables

- RAD, SDD, ODD, 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.).