



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

# Statement of Work

## Progetto UniSeats

Riferimento	
Versione	1.1
Data	12/10/2020
Destinatario	Dipartimento di Informatica dell'Università degli Studi di Salerno
Presentato da	Vincenzo Russo
Approvato da	



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

## Revision History

---

Data	Versione	Descrizione	Autori
07/10/2020	1.0	Prima stesura	V. Russo
12/10/2020	1.1	Revisione e aggiunta scenari	V. Russo



# Statement of Work (SOW) del Progetto

## UniSeats

### 1. Piano Strategico/Strategic Plan

Il Dipartimento di Informatica dell'Università di Salerno sta costruendo una didattica migliore e più sostenibile coinvolgendosi attivamente nella risoluzione dei problemi. La ricerca multidisciplinare, l'insegnamento e l'apprendimento di alto livello generano nuove conoscenze e know-how a beneficio della comunità. Un pensiero innovativo, un atteggiamento attento e proattivo e l'interazione creativa aiuteranno il Dipartimento a consolidare la sua posizione tra i più importanti in Italia.

Per realizzare la sua visione, il Dipartimento di Informatica ha selezionato diversi obiettivi strategici per il periodo 2020-2021, tra cui il miglioramento della gestione delle infrastrutture, nella fattispecie le aule adibite alla didattica.

### 2. Obiettivi di Business/Business Needs

Il Dipartimento di Informatica dell'Università degli Studi di Salerno intende incrementare la sicurezza degli studenti ed allo stesso tempo permettere uno svolgimento più sicuro delle lezioni in presenza.

Precedentemente non è mai stato implementato un sistema di gestione degli spazi comuni, in tal caso le aule. Data la singolare situazione che gli studenti e i docenti stanno affrontando nell'anno 2020-2021 è nata la necessità di implementare un sistema che permetta agli iscritti di studiare e socializzare in tutta sicurezza.

### 3. Ambito del Prodotto/Product Scope

L'obiettivo del progetto è fornire uno strumento di supporto per selezionare un posto in aula. Il sistema che proponiamo supporterà:

- Prenotare i posti nelle aule studio;
- Modificare le prenotazioni;
- Eliminare le prenotazioni;
- Ottimizzare i posti disponibili per rispettare le misure anti-Covid;
- Consigli all'utente le aule dove è più probabile che incontri studenti con le stesse caratteristiche.



#### Scenario 1: *Prenotazione Posto*

Marco è uno studente dello studente del Dipartimento di Informatica iscritto al terzo anno. Alla fine delle lezioni decide di andare a studiare in aula studio, e desidera trovare altri studenti del Dipartimento con cui studiare e confrontarsi. Marco dopo essersi autenticato sulla piattaforma UniSeats, chiede al sistema di trovare un posto libero in un'aula che contenga almeno un altro studente del Dipartimento.

Il sistema riceve la richiesta di Marco, controlla i posti disponibili e gli propone 2 soluzioni:

- Aula F9 della struttura F contenente altri 3 studenti.
- Aula F15 della struttura F2 contenente uno studente.

Marco riceve la risposta dal sistema e sceglie l'aula F15. Invia, la prenotazione al sistema.

Il sistema elabora la richiesta e invia a Marco il QR Code per accedere all'aula. Marco riceve il QR code.

#### Scenario 2: *Cancellazione posto*

Andrea è uno studente fuorisede del primo anno, ha notato che la corsa dei pullman è stata cancellata, informa di ciò i suoi compagni, quindi desidera cancellare la sua prenotazione del posto in aula.

Andrea dopo essersi autenticato sulla piattaforma UniSeats, seleziona la prenotazione precedentemente effettuata. Il sistema mostra ad Andrea i dettagli della prenotazione, Andrea collegato da smartphone tocca sull'icona del cestino per eliminare la prenotazione. Il sistema chiede ad Andrea se è sicuro con un dialogo. Andrea Tocca Sì. Il sistema elabora la richiesta ed elimina la prenotazione.

#### Scenario 3: *Modifica posto*

Simone è uno studente del primo secondo anno, frettolosamente e con poco giudizio ha prenotato un'altra aula. Simone si accorge che non è l'aula corretta e desidera modificare la sua scelta.

Simone si autentica sulla piattaforma UniSeats, seleziona la prenotazione precedentemente effettuata. Il sistema mostra a Simone i dettagli della prenotazione, Simone collegato da PC clicca sull'icona di modifica per modificare la prenotazione. Il sistema chiede ad Simone se è sicuro con un dialogo. Simone Tocca Sì. Il sistema elabora la richiesta ed elimina la prenotazione. Il sistema riceve la richiesta di Simone, controlla i posti disponibili e gli propone 2 soluzioni:

- Aula F3 della struttura F2 contenente altri 3 studenti.
- Aula F8 della struttura F2 contenente altri 2 studenti.

Simone riceve la risposta dal sistema e sceglie l'aula F8. Invia, la prenotazione al sistema. Il sistema elabora la richiesta e apporta tutte le modifiche necessarie.



## 4. Data di Inizio e di Fine

---

Inizio: Ottobre 2020

Fine: Gennaio 2021

## 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

---

- Rispetto scadenze;
- Budget/Effort non superiore a  $50 \cdot n$  ore dove  $n$  sono i membri del team (compresi PM);
- Applicazione in Java o derivati;
- Uso di tre Design Pattern;
- Uso di UML;
- Utilizzo di un sistema di versioning, dove tutti i membri del team forniscono il loro contributo;
- Utilizzo di tool di management (Trello, Asana,...) per divisione compiti;
- Utilizzo di Slack per comunicazione;
- Utilizzo di quality tool come Checkstyle;
- Parte di progetto con approccio Agile (Scrum).



## 7. Criteri di Accettazione/Acceptance Criteria

---

- Branch coverage dei casi di test: almeno 75%;
- Buona manutenibilità;
- Il numero di warning dati in output da Checkstyle inferiore ad una soglia da definire (molto bassa).

## 8. Criteri di premialità

---

- Utilizzo di sistemi di build, come Maven o Gradle;
- Utilizzo del pull-based development tramite l'applicazione di code review;
- Utilizzo di un processo di Continuous Integration, tramite l'utilizzo di Travis.