



# ERASMUS<sup>✈</sup>SMART

## Integration Testing

Riferimento	
Versione	1.0
Data	16/12/2018
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Andrea Valenti Ruggero Tammaro
Approvato da	



## Revision History

Data	Versione	Descrizione	Autori
10/12/2018	0.1	Primate stesura	Martin Cioffi Cristiano Binetti Andrea Massaro
13/12/2018	0.2	Aggiunti i punti 1,2, 3 e 4.	Martin Cioffi Cristiano Binetti Andrea Massaro
14/12/2018	0.3	Aggiunta sezione 3.1	Martin Cioffi Cristiano Binetti Andrea Massaro
16/12/2018	1.0	Modifiche e revisione	Andrea Valenti Ruggero Tammaro



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

## Project Team

Nome	Cognome
Andrea	Massaro
Cristiano	Binetti
Martin	Cioffi



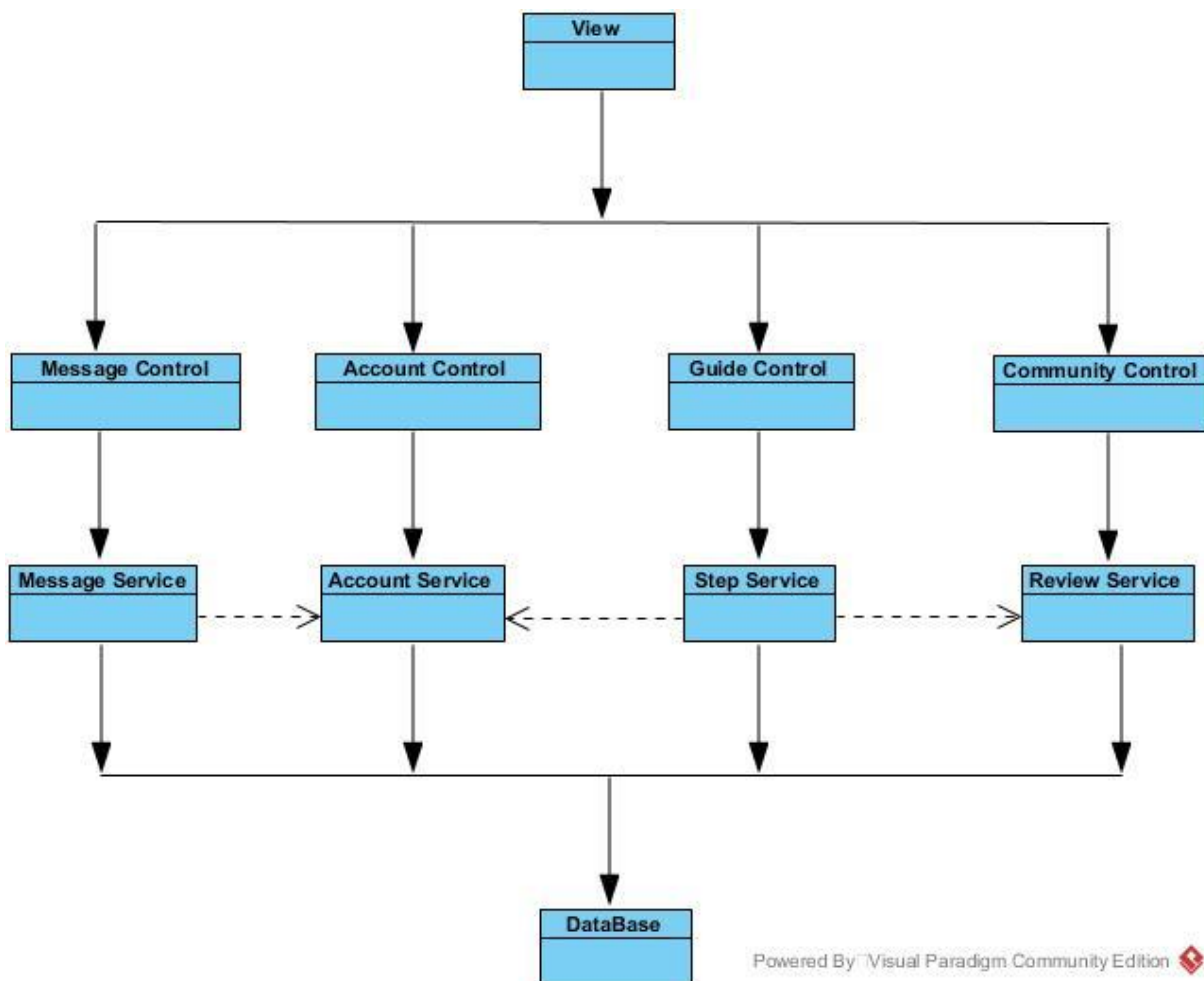
## Sommario

Revision History .....	2
1. Integration test strategy .....	5
2. Integration test plan .....	5
3. Test case specification.....	6
3.1 Introduzione .....	6
3.2 Test case .....	6
4. Pass/Fail Criteria .....	10

## 1. Integration test strategy

Utilizzeremo una strategia Top-Down in quanto la complessità del nostro sistema è incentrata nella parte alta della struttura, in questo modo oltre ad utilizzare un approccio più naturale otterremo rapidamente uno “scheletro” del prodotto semi funzionante. In base a queste considerazioni, anche se i stub da produrre sono tipicamente complessi, i vantaggi ottenuti dall'utilizzo della strategia Top-Down si adattano alle esigenze dello sviluppo del testing del nostro sistema.

## 2. Integration test plan





## 3. Test case specification

### 3.1 Introduzione

Poiché adottiamo la strategia di testing Top-Down, gli input utilizzati per effettuare i seguenti test case sono forniti dagli stub, i quali sostituiscono la parte chiamata restituendo solamente dei valori di ritorno fittizi.

### 3.2 Test case

<b>Test Case Name</b>	Test_StepService_ReviewService
<b>Test Case ID</b>	TC_3.1
<b>Condizione di Entrata</b>	Lo studente ha effettuato l'accesso e non ha ancora avviato lo step-by-step.
<b>Flusso degli Eventi</b>	<ol style="list-style-type: none"><li>1. Lo studente, attraverso il menu, accede alla sezione "Guida".</li><li>2. Il sistema mostra la lista delle università con il numero di recensioni annesse.</li><li>3. StepService richiede a ReviewService la lista delle università con il numero di recensioni annesse.</li></ol>
<b>Condizione di uscita</b>	Il sistema mostra la lista delle università con il numero di recensioni annesse.

<b>Test Case Name</b>	Test_StepService_ReviewService
<b>Test Case ID</b>	TC_3.1
<b>Percorso Test</b>	-
<b>Input</b>	La lista delle recensioni che viene fornita dal ReviewService
<b>Oracolo</b>	Lo StepService mostra la lista delle università con il numero di recensioni annesse.
<b>Log</b>	-



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



<b>Test Case Name</b>	Test_StepService_AccountService
<b>Test Case ID</b>	TC_3.2
<b>Condizione di Entrata</b>	Lo studente ha effettuato l'accesso e è arrivato allo step4.
<b>Flusso degli Eventi</b>	<ol style="list-style-type: none"><li>1. Lo studente clicca sullo Spinner per visualizzare la lista dei tutor disponibili.</li><li>2. Lo stepService richiede ad AccountService la lista di tutti i tutor</li></ol>
<b>Condizione di uscita</b>	Lo studente visualizza la lista di tutti i tutor disponibili.

<b>Test Case Name</b>	Test_StepService_AccountService
<b>Test Case ID</b>	TC_3.2
<b>Percorso Test</b>	-
<b>Input</b>	StepService riceve la lista di tutti i tutor da AccountService
<b>Oracolo</b>	Lo studente visualizza la lista di tutti i tutor disponibili.
<b>Log</b>	-





<b>Test Case Name</b>	Test_MesageService_AccountService
<b>Test Case ID</b>	TC_3.3
<b>Condizione di Entrata</b>	Il tutor ha effettuato l'accesso.
<b>Flusso degli Eventi</b>	<ol style="list-style-type: none"><li>1. Il tutor accede alla pagina relativa ai messaggi</li><li>2. Il MessageService richiede all' AccountService la lista dei nomi e cognomi degli studenti collegati all'account del tutor</li></ol>
<b>Condizione di uscita</b>	Il tutor visualizza tutta la lista degli studenti collegati ad esso.

<b>Test Case Name</b>	Test_MesageService_AccountService
<b>Test Case ID</b>	TC_3.3
<b>Percorso Test</b>	-
<b>Input</b>	MessageService riceve la lista di tutti gli studenti da AccountService
<b>Oracolo</b>	Il tutor visualizza tutta la lista degli studenti collegati ad esso.
<b>Log</b>	-



## 4. Pass/Fail Criteria

---

Il testing proposto ha successo se l'output osservato è diverso dall'output atteso, quindi un test ha successo se rileva uno o più malfunzionamenti del programma. In tal caso il codice testato che ha generato una failure, verrà analizzato e si procederà alla correzione del fault. Sarà infine ripetuta la fase di testing per verificare che la modifica non abbia impattato su altri componenti del sistema. Viceversa, il testing fallisce se non riesce ad individuare un errore.