



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof.ssa F. Ferrucci



SDD System Design Document

Riferimento	
Versione	1.2
Data	13/12/2018
Destinatario	Prof.ssa F. Ferrucci
Presentato da	Daniele De Vinco, Donatella Cioffi, Federica Ungherese, Luigi Di Palma
Approvato da	



Revision History

Data	Versione	Descrizione	Autori
04/12/2018	1.0	Prima stesura	Daniele De Vinco, Donatella Cioffi, Federica Ungherese, Luigi Di Palma
13/12/2018	1.1	Terza stesura. Aggiornamento dei paragrafi 3.3, 3.4, 3.6, 4	Daniele De Vinco, Donatella Cioffi, Federica Ungherese, Luigi Di Palma
12/01/2019	1.2	Revisione finale	Daniele De Vinco, Donatella Cioffi, Federica Ungherese, Luigi Di Palma



Sommario

Revision History	2Error! Bookmark not defined.
1. Introduzione	4
1.1 Obbiettivi del Sistema	4
1.2 Design Goals.....	5
1.3 Definizioni,Acronimi e Abbreviazioni	7
1.4 Riferimenti	7
1.5 Panoramica.....	7
2. Architettura del Sistema Corrente.....	8
3 Architettura del Sistema proposto	9
3.1 Panoramica	9
3.2 Decomposizione in sottosistemi	9
3.3 Mapping Hardware/Software	11
3.4 Gestione dati persistenti	12
3.5 Controllo degli accessi e sicurezza.....	13
3.6 Controllo globale del Software	15
3.7 Condizioni limite	15
4 Servizi dei sottosistemi.....	18



1. Introduzione

1.1 Obbiettivi del Sistema

Il sistema che si vuole realizzare ha come scopo quello di informare al meglio i diversi studenti che hanno intenzione di partecipare al progetto Erasmus+. Il nostro obiettivo è di realizzare un sistema che permetta agli utenti tramite la pubblicazione di post e commenti, di effettuare diverse domande al fine di ridurre al minimo i dubbi relativi alla loro partecipazione al progetto. Il manager del sistema inserirà diverse bacheche, dove avverranno discussioni tra gli utenti, a seconda delle mete offerte. Una maggiore informazione permetterà agli studenti che si vogliono affacciare al mondo Erasmus di avere un quadro generale più chiaro mediante le risposte fornite da utenti che hanno già partecipato al progetto o che si trovano in Erasmus. Il sistema progettato è un vero e proprio Forum. Per quanto riguarda le bacheche il sistema dovrà fornire all'utente la possibilità di visualizzare una bacheca che mostra informazioni (post e commenti) relativi alla meta desiderata, dovrà fornire all'utente la possibilità di selezionare la meta desiderata, mediante una specifica bacheca dove si visualizzeranno i diversi post e commenti relativi a tale destinazione.

Per i post il sistema dovrà fornire all'utente la possibilità di:

- creare un nuovo post, tale operazione può essere effettuata dai soli utenti loggati;
- ricercare di uno specifico post, mediante la barra di ricerca apposita, può essere fatta da tutti gli utenti, anche quelli non registrati;
- modificare un post, può essere effettuata dal creatore del post o dal manager del sistema;
- eliminare un post, effettuata analogamente alla modifica;

Per i commenti del sistema dovrà fornire all'utente la possibilità di:

- creare di un nuovo commento, tale operazione può essere effettuata dai soli utenti loggati;
- ricercare uno specifico commento all'interno di un post, mediante la barra di ricerca apposita, può essere fatta da tutti gli utenti, anche quelli non registrati;
- modificare un commento, può essere effettuata dal creatore del post o dal manager del sistema;
- eliminare un commento, essa viene effettuata analogamente alla modifica;

Relativamente al Forum invece il sistema dovrà fornire all'utente la possibilità di autenticarsi all'interno del sistema e dovrà fornirgli la possibilità di scegliere quale operazione effettuare cliccando sugli appositi pulsanti.

Dato che il sistema ha accesso a dati sensibili degli studenti il sistema deve fornire un metodo di autenticazione sicuro in modo che i dati siano protetti da accessi fraudolenti. Per una migliore usabilità, inoltre, il sistema:

- dovrà essere facile da apprendere ed intuitivo da utilizzare,
- dovrà consentire la navigazione agevole per la fruizione delle funzionalità da esso offerte.



1.2 Design Goals

Design	Rank	Goals	Descrizione	Categoria	Origine
DG_1	15	Tempo di risposta	Il tempo di risposta di una qualsivoglia operazione è di al più 5 secondi.	Criteri di performance	3.4.3 (RAD)
DG_2	14	Memoria	La dimensione complessiva del sistema dipende dalla memoria utilizzata per il mantenimento del database.	Criteri di performance	3.4.3 (RAD)
DG_3	2	Robustezza	Il sistema, nel momento in cui è soggetto ad un accesso concorrente sia da parte dell'amministratore che da parte dell'utente, per quanto concerne la modifica/eliminazione di un post/commento, permetterà l'accesso alla modifica/eliminazione alla prima persona che ha effettuato la richiesta.	Criteri di affidabilità	3.4.2 (RAD)
DG_4	3	Affidabilità	I risultati visualizzati quali mete del progetto Erasmus+ e informazioni riguardanti le varie università d'interesse, sono attendibili. Essi infatti sono coerenti con le informazioni presenti sul sito Unisa.it.	Criteri di affidabilità	3.4.2 (RAD)
DG_5	4	Disponibilità	Il sistema è accessibile sempre eccetto i giorni in cui viene effettuata la manutenzione.	Criteri di affidabilità	3.4.2 (RAD)
DG_6	5	Tolleranza ai guasti	Il sistema non ammette tolleranze ai guasti.	Criteri di affidabilità	3.4.2 (RAD)
DG_7	6	Security	L'accesso al sistema è garantito mediante e-mail e password. La sicurezza del	Criteri di affidabilità	3.4.2 (RAD)



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof.ssa F. Ferrucci

			database è garantita dal fatto che lo stesso è in locale, quindi accessibile solo all'addetto ai lavori.		
DG_8	7	Safety	Integrità dei dati - stato del sistema in caso di crash	Criteri di affidabilità	3.4.2 (RAD)
DG_9	16	Costi di sviluppo	È stimato un costo complessivo di 600 ore per la progettazione e lo sviluppo del sistema (150 ore per ogni team member).	Criteri di costo	3.4.6 (RAD)
DG_10	8	Estendibilità	Il sistema può essere esteso ad altri dipartimenti o ad altre università cambiando i dati su cui lavora	Criteri di manutenzione	3.4.4 (RAD)
DG_11	10	Modificabilità	Il sistema è soggetto a modifiche mensili.	Criteri di Manutenzione	3.4.4 (RAD)
DG_12	9	Tracciabilità	Nel RAD è possibile visualizzare tutti i requisiti. Per ogni paragrafo sarà specificato, se necessario, l'identificativo del requisito a cui si fa riferimento.	Criteri di Manutenzione	3.4.4 (RAD)
DG_13	11	Affidabilità	La manutenzione del sistema viene effettuata mensilmente.	Criteri di Manutenzione	3.4.4 (RAD)
DG_14	13	Portabilità	L'app è utilizzabile solo su dispositivi Android.	Criteri di Manutenzione	3.4.4 (RAD)
DG_15	1	Usabilità	L'usabilità di un sistema può essere analizzata considerando diverse caratteristiche. Questo sistema sarà molto semplice da apprendere senza la consultazione di documentazione associata. L'intuitività è garantita in quanto il sistema avrà una buona prevedibilità, cioè la risposta del sistema ad un'azione utente sarà corrispondente alle aspettative.	Criteri finali	3.4.1 (RAD)
DG_16	12	Utilità	Il sistema risulterà essere utile agli studenti che vorranno	Criteri finali	3.4.1 (RAD)



TRADE-OFF

Functionality vs. Usability:

Il nostro sistema mira ad una maggiore usabilità rispetto alla funzionalità perché è stato ideato per semplificare al meglio il suo utilizzo da parte degli utenti.

Cost vs. Robustness

Il nostro sistema predilige una maggiore robustezza rispetto ai costi perché sarà in grado comportarsi in modo ragionevole qual ora si presentino alcune situazioni impreviste.

Efficiency vs. Portability

Il sistema ideato predilige efficienza alla portabilità perché è stato ideato al fine di utilizzare un numero minimo di risorse del dispositivo da cui è utilizzato.

Cost vs. Reusability

Il sistema predilige la riusabilità rispetto ai costi perché sarà possibile in futuro riutilizzare tale progetto, o parte di esso, per inserirlo all'interno di uno nuovo.

1.3 Definizioni, Acronimi e abbreviazioni

Erasmus+: progetto gestito da UNISA per studiare all'estero

UNISA: Università degli studi di Salerno

DG_N: Acronimo che indica il codice del Design Goal

1.4 Riferimenti

- B.BRUGGE, A.H.DUTOIT, *OBJECT ORIENTED SOFTWARE ENGINEERING – USING UML, PATTERNS AND JAVA*, PRENTICE HALL, 3D EDITION, 2009
- C. GHEZZI, D. MANDRIOLI, M. JAZAYERI, *INGEGNERIA DEL SOFTWARE – FONDAMENTI E PRINCIPI*, PRENTICE HALL, 2004
- SOMMERVILLE, *SOFTWARE ENGINEERING*, ADDISON WESLEY
- Materiale didattico reperibile su <http://elearning.informatica.unisa.it/el-platform/> nella sezione del corso di *Ingegneria del Software* (professoressa Filomena Ferrucci)

1.5 Panoramica

Al secondo punto del documento verrà presentato il sistema corrente.



Laurea Triennale in informatica-Università di Salerno
Corso di Ingegneria del Software- Prof.ssa F. Ferrucci





Al terzo punto verrà presentata l'architettura del sistema proposto in cui gestiremo la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema, le condizioni limite.

Al quarto punto verranno presentati i servizi dei sottosistemi



2. Architettura del Sistema corrente

Il sistema sviluppato non si rifà ad alcun sistema preesistente, tuttavia è stato individuato un sistema (un Forum di ProBoards) che ha funzionalità affini a quelle presenti.

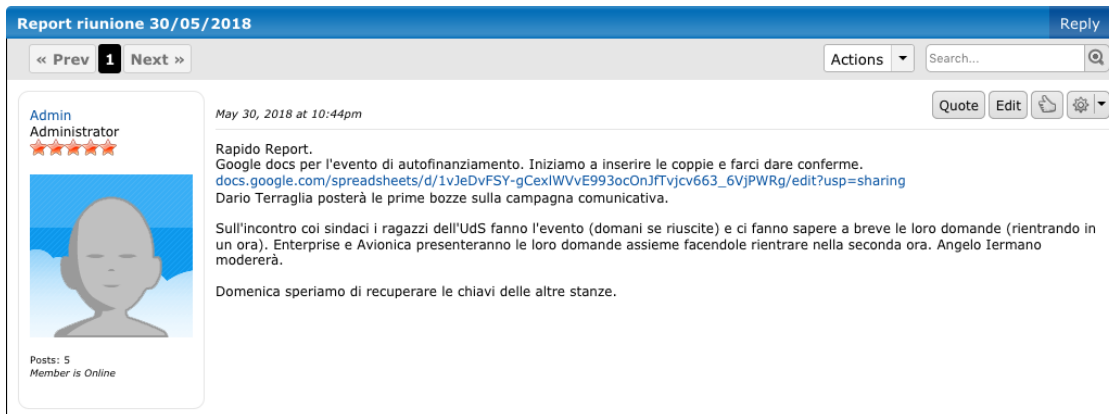
Nella home è possibile vedere l'elenco delle bacheche principali, con le relative informazioni riguardo ognuna di esse.

General				
	Board	Threads	Posts	Last Post
	Calendarizzazione Gruppo programmazione attività Moderator: Admin	0	0	No posts have been made on this board.
	Arredamento Gruppo Arredamento Moderator: Admin	0	0	No posts have been made on this board.
	Comunicazione Gruppo Comunicazione Moderator: Admin	0	0	No posts have been made on this board.
	Generale Argomenti generali Moderator: Admin	2	2	Bug e Segnalazioni by Admin May 30, 2018 at 10:49pm

Se si seleziona una bacheca, si accede a tutti i post contenuti in essa, visualizzati come elenco con le relative informazioni.

Generale						Create Thread
« Prev 1 Next »		Select	Actions	Search...		
	Subject	Created By	Replies	Views	Last Post	
	Bug e Segnalazioni	Admin	0	2	by Admin May 30, 2018 at 10:49pm	
	Report riunione 30/05/2018	Admin	0	4	by Admin May 30, 2018 at 10:44pm	

Se si seleziona un post, viene mostrata una pagina con le informazioni di quel post e gli eventuali commenti generati da altri utenti.



3. Architettura del Sistema proposto

3.1 Panoramica

Il sistema proposto è un'applicazione mobile, con un server online che gestisce il database condiviso. L'obiettivo è quello di proporre un sistema che consenta la condivisione di informazioni in modo rapido ed efficiente, gestendo più utenti nello stesso momento.

Ci sarà quindi una divisione tra client e server: il primo gestirà l'interfaccia e la logica, il secondo si occuperà di fornire i dati richiesti ed effettuare diverse operazioni sul database.

I sistemi saranno divisi in 3 layer che consentiranno una più semplice gestione dei sottosistemi

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in Layer

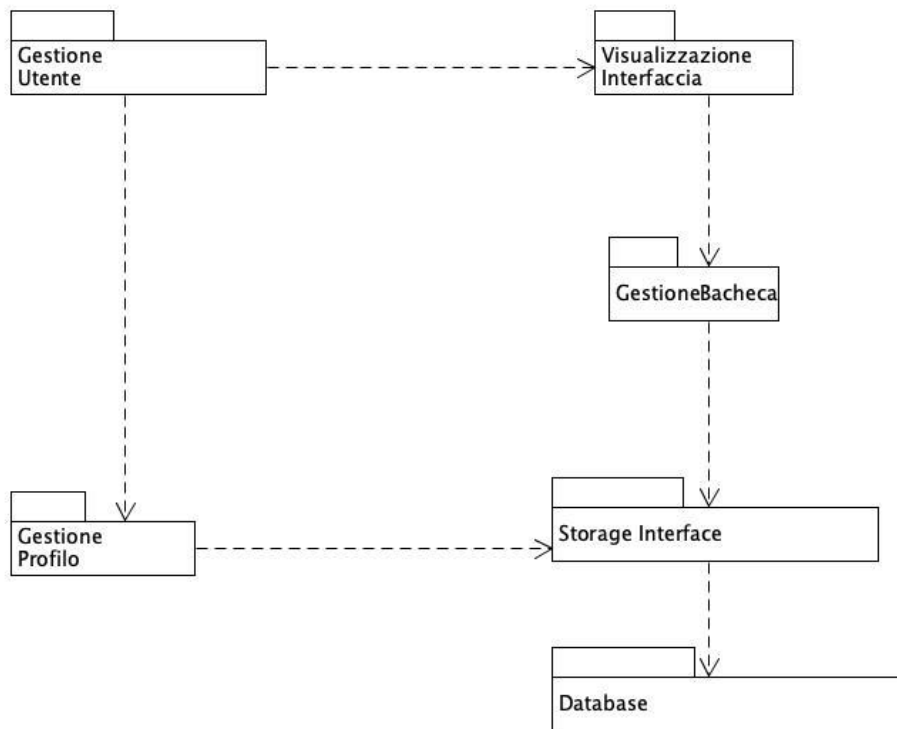
Il Sistema è decomposto in tre layer che si occupano di gestire aspetti e funzionalità differenti:

- Presentazione: raccoglie e gestisce l'interfaccia e gli eventi generati dall'utente;
- System Logic: si occupa della gestione della logica del sistema;
- Storage: si occupa della gestione e dello scambio dei dati tra i sottosistemi.

3.2.2 Decomposizione in sottosistemi

Il Sistema verrà suddiviso in tre sottosistemi, in cui si raggrupperanno le funzionalità in base all'area di gestione di appartenenza. Si è creata, inoltre, un'interfaccia intermedia tra i sistemi della logica di Sistema e il database, in modo da non dover fare grossi cambiamenti nel caso in cui sia necessario cambiare database.

La suddivisione logica del Sistema è rappresentata di seguito:



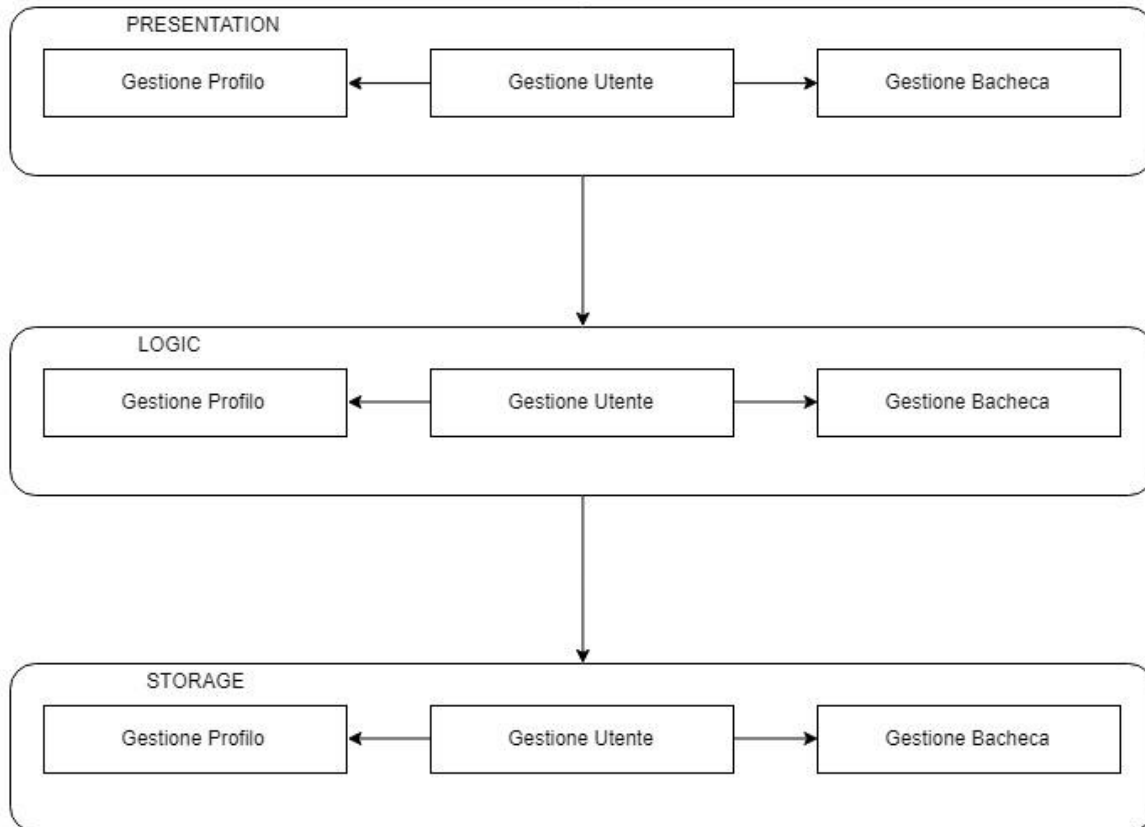
Il livello VisualizzazioneInterfaccia prevede un unico sottosistema.

Il livello di System Logic prevede a sua volta una suddivisione in 3 sottosistemi:

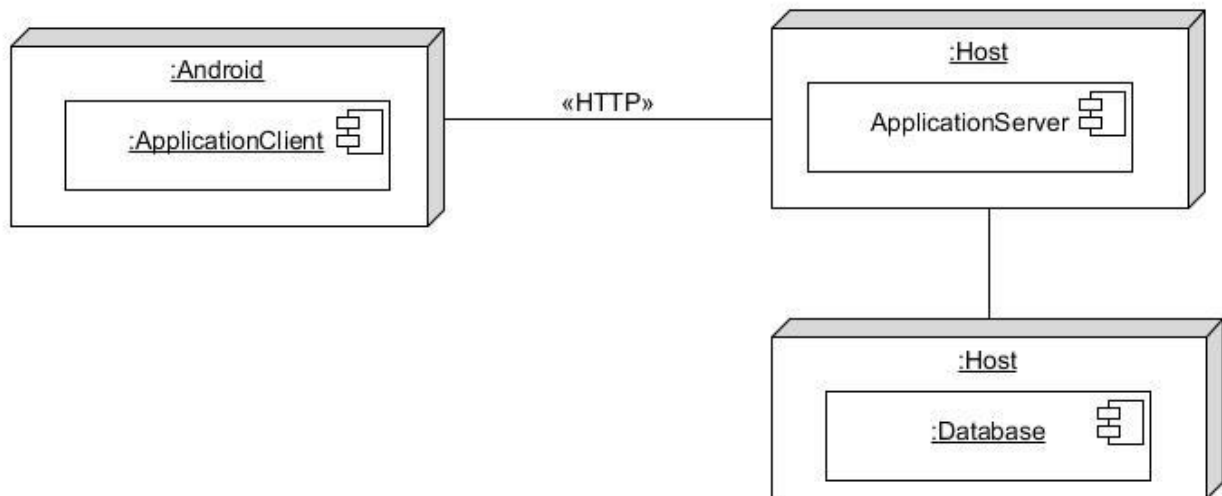
- GestioneUtente: il sottosistema si occupa della gestione delle funzionalità messe a disposizione dell'utente;
- GestioneProfilo: il sottosistema si occupa della gestione dei dati di un profilo utente o manager;
- GestioneBacheca: il sottosistema si occupa gestione delle funzionalità proprie della bacheca.

Il livello Storage Interface prevede un unico sottosistema già spiegato in precedenza.

3.2.3 Diagramma di Deployment



3.3 Mapping Hardware/Software



Di seguito la lista delle componenti utilizzate:

- Firebase: piattaforma per gestire il database ed i servizi di autenticazione utente in maniera sicura;
- Application server: ruolo svolto da Firebase.
- Android: sistema operativo al quale è rivolta l'applicazione.

Il sistema verrà implementato, testato e distribuito su diversi tipi di smartphone. Non è richiesta una grande potenza di calcolo o più processori, in quanto il carico di lavoro non è eccessivo.

3.4 Gestione dei dati persistenti

Gli oggetti persistenti provengono dagli oggetti entity individuati nella fase di analisi dei requisiti e sono i seguenti:

- Bacheca
- Post
- Commento
- Profilo utente

Gli oggetti persistenti vengono gestiti mediante l'utilizzo di un database NoSQL. Tutti i dati hanno priorità elevata.

Il modello E-R rispecchia il class diagram presente al paragrafo 3.5.4 del RAD.

3.5 Controllo degli accessi e sicurezza

Il sistema supporta una gestione multi-utente, dove ogni user accede a specifiche funzionalità a seconda dei permessi che gli vengono assegnati.

Ci sono tre tipi di user:

- Ospite
- Utente



- Manager

L'ospite è un user che non ha effettuato il login o non ha completato il processo di registrazione, e può effettuare solamente operazioni di ricerca o visualizzazioni delle bacheche, dei post, dei commenti e visualizzazione dei profili di utenti registrati.

L'utente è un user che ha effettuato il login e può creare un nuovo post o un nuovo commento, modificare un post o commento creato da lui precedentemente e modificare le informazioni contenute nel proprio profilo.

Il Manager ha accesso a tutti i dati presenti nel sistema.

Il controllo di accesso non cambia dinamicamente, ma è sempre uguale in tutti i sottosistemi.

Sottosistema Attore	Gestione		
	Gestione Bacheche	Gestione Profilo	Gestione Utente
Ospite	<ul style="list-style-type: none"> • Visualizzazione bacheca • Visualizzazione post • Visualizzazione commenti • Ricerca post • Ricerca commento 	<ul style="list-style-type: none"> • Visualizzazione profilo (solo di altri utenti) • Ricerca profilo utente 	<ul style="list-style-type: none"> • Registrazione • Login
Utente	<ul style="list-style-type: none"> • Visualizzazione bacheca • Visualizzazione post • Visualizzazione commenti • Ricerca post • Ricerca commento 	<ul style="list-style-type: none"> • Modifica profilo di propria proprietà • Elimina profilo di propria proprietà • Visualizzazione profilo 	<ul style="list-style-type: none"> • Logout



	<ul style="list-style-type: none"> • Pubblicazione post • Pubblicazione commento 	<ul style="list-style-type: none"> • Ricerca profilo utente 	
Manager	<ul style="list-style-type: none"> • Visualizzazione bacheca • Visualizzazione post • Visualizzazione commenti • Ricerca post <ul style="list-style-type: none"> • Ricerca commento • Pubblicazione post • Pubblicazione commento <ul style="list-style-type: none"> • Aggiungi bacheca • Modifica bacheca • Modifica post di un utente <ul style="list-style-type: none"> • Modifica commento di un utente • Elimina post di un utente <ul style="list-style-type: none"> • Elimina commento di un utente 	<ul style="list-style-type: none"> • Modifica profilo di propria proprietà • Elimina profilo di propria proprietà • Visualizzazione profilo • Ricerca profilo utente 	<ul style="list-style-type: none"> • Logout • Elimina profilo di un utente • Pannello manager (nomina utente come manager)



3.6 Controllo globale del Software

Il sistema è event-driven: presenta un'interfaccia che resta in attesa di una richiesta utente; ogni interazione dell'utente (evento) avvia l'elaborazione da parte del sottosistema appropriato e l'invio della relativa risposta.

3.7 Condizioni limite

3.7.1 Start-up

Per il primo start-up del sistema "UniShare" è necessario l'avvio di un web server (Firebase) che fornisca il servizio di un Database NoSQL per la gestione dei dati persistenti e l'interpretazione ed esecuzione del codice lato server. Il sistema recupera le informazioni per mostrare a schermo la Home. In seguito, tramite l'interfaccia di Login, sarà possibile autenticarsi con le opportune credenziali (e-mail e password) come utente così da avere accesso a tutte le funzionalità dedicate del sistema.

3.7.2 Start-up (a seguito di un fallimento)

Il sistema può subire guasti dovuti al sovraccarico del database con successivo fallimento. Il sistema non permette di salvare dati incorretti o incompleti automaticamente per avere una migliore consistenza del database in caso di emergenza.

3.7.3 Terminazione

Al momento della chiusura dell'applicativo si ha la terminazione del sistema con un regolare Logout dal sistema. Viene assicurata la consistenza dei dati, annullando eventuali operazione che erano in esecuzione.

3.7.4 Fallimento

Possono verificarsi diversi casi di fallimento del sistema:

- Nel caso in cui si verifichi un'interruzione inaspettata dell'alimentazione, non sono previsti metodi che riportino lo stato del sistema a prima dello spegnimento inaspettato.
- Un altro caso di fallimento potrebbe derivare dal software stesso che causa una chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. Non sono previste politiche correttive, l'unico processo che potrà essere eseguito è la chiusura del sistema e il suo successivo riavvio.
- Un altro caso di fallimento potrebbe essere dovuto ad un errore critico nell'hardware, non è prevista alcuna misura correttiva.

Identificativo <i>UC_CL_1</i>	<i>Inizializzare il sistema</i>	<i>Data</i>	<i>29/11/2018</i>
		<i>Vers.</i>	<i>0.00.001</i>



Laurea Triennale in informatica-Università di Salerno
Corso di Ingegneria del Software- Prof.ssa F. Ferrucci

				Autore	De Vinco Daniele
Descrizione			Lo UC fornisce la funzionalità di portare da uno stato non inizializzato ad uno stabile		
Attore Principale			Utente		
Attori secondari			NA		
Entry Condition			Applicazione installata sul dispositivo		
Exit condition On success			Il dispositivo riesce ad aprire l'applicazione		
Exit condition On failure			L'applicazione non si apre		
Rilevanza/User Priority			Elevata		
Frequenza stimata			60 usi/giorno		
Extension point			NA		
Generalization of			NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO					
1	Utente:	Clicca sull'icona dell'applicazione			
2	Sistema:	Preleva i dati necessari per l'avvio			
3	Sistema:	Mostra la pagina iniziale			
Scenario/Flusso di eventi Alternativo: i dati non sono disponibili					
2.a1	Sistema:	Visualizza un messaggio di errore che segnala all'utente che non è stato possibile accedere al sistema			
2.a2	Sistema	Invita l'utente a riprovare più tardi			
Scenario/Flusso di eventi di ERRORE: sistema non riesce ad avviare l'applicazione					
3.a1	Sistema:	I dati visualizzati saranno incompleti o assenti.			
3.a2	Sistema	Termina con un insuccesso			



Laurea Triennale in informatica-Università di Salerno
Corso di *Ingegneria del Software*- Prof.ssa F. Ferrucci

Identificativo <i>UC_CL_2</i>	<i>Terminare il sistema</i>	<i>Data</i>	<i>29/11/2018</i>
		<i>Vers.</i>	<i>0.00.001</i>
		<i>Autore</i>	<i>De Vinco Daniele</i>
Descrizione	<i>Lo UC fornisce la funzionalità di terminare un sistema</i>		
Attore Principale	Utente		
Attori secondari	NA		
Entry Condition	Applicazione installata sul dispositivo		
Exit condition On success	Il dispositivo riesce a terminare l'applicazione		
Exit condition On failure			
Rilevanza/User Priority	Elevata		
Frequenza stimata	60 usi/giorno		
Extension point	NA		
Generalization of	NA		
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO			
1	Utente:	Termina l'applicazione	
2	Sistema:	Salva eventuali dati persistenti nel sistema	
3	Sistema:	Termina la connessione con il dispositivo	
Scenario/Flusso di eventi Alternativo: i dati sono incompleti o incorretti			
2.a1	Sistema:	Avvisa l'utente che ci sono dei dati incompleti che non verranno salvati	
2.a2	Sistema	Attende una risposta e poi termina	
Scenario/Flusso di eventi di ERRORE: sistema non riesce a salvare i dati			
2.b1	Sistema:	I dati visualizzati saranno incompleti o assenti.	
2.b2	Sistema	Termina con un insuccesso	



4. Servizi dei sottosistemi

Gestione Bacheca	Gestione Profilo	Gestione Utente
<p>Offre servizi per:</p> <ul style="list-style-type: none">• Visualizzare<ul style="list-style-type: none">◦ Bacheca◦ Post◦ Commento• Ricercare<ul style="list-style-type: none">◦ Post◦ Commento• Pubblicare<ul style="list-style-type: none">◦ Post◦ Commento• Aggiungere bacheca<ul style="list-style-type: none">• Modificare<ul style="list-style-type: none">◦ Post◦ Commento• Eliminare<ul style="list-style-type: none">◦ Post◦ Commento	<p>Gestisce le informazioni associate ad un utente registrato nel sistema. Ad ogni utente registrato offre servizi per:</p> <ul style="list-style-type: none">• Visualizzare profilo• Modificare proprio profilo• Eliminare proprio profilo <p>Sia l'utente che l'ospite invece possono effettuare la ricerca di un profilo utente registrato al sistema.</p>	<p>Offre servizi per:</p> <ul style="list-style-type: none">• Registrazione• Login• Logout• Eliminare profilo di un utente (solo Manager)• Nomina manager (solo Manager)