



SDD – System Design Document

Versione	2.0
Data	01/12/2021
Destinatario	Prof.ssa F. Ferrucci, Prof. Fabio Palomba
Presentato da	Tutti i team member
Approvato da	Salvatore Amideo Alice Vidoni

Revision History

Data	Versione	Descrizione	Autori
19/11/2021	0.1	Prima stesura	[Tutti]
29/11/2021	1.0	Completamento del documento	[Tutti]
30/11/2021	1.1	Correzione formattazione del documento	Buono Claudio, Salzano Alessio
01/12/2021	2.0	Revisione Definitiva	Salvatore Amideo Alice Vidoni

Sommario

Revision History	2
1. INTRODUZIONE	4
1.1 Obiettivi del sistema	4
1.2 Design Goals	4
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.....	10
3.1 Panoramica	10
3.2 Decomposizione in sottosistemi	11
3.2.1 Decomposizione in layer	11
3.2.2 Decomposizione in sottosistemi	12
3.3 Mapping Hardware/Software.....	14
3.4 Gestione dei dati persistenti	15
3.4.1 Descrizioni delle entità persistenti	15
3.5 Controllo degli accessi e sicurezza	19
3.6 Controllo flusso globale del sistema	24
3.7 Condizioni Boundary	25
4. Servizi dei sottosistemi.....	26

1. INTRODUZIONE

1.1 Obiettivi del sistema

Il sistema che si vuole realizzare ha come scopo la semplificazione e la velocizzazione delle funzionalità del servizio mensa UNISA, rendendole più agevoli sia per chi ci lavora sia per chi ne usufruisce.

I nostri obiettivi riguardano la realizzazione di un sistema che permetta di avere una massima automazione per quanto riguarda l'intera gestione del tesserino mensa ed automatizzare le ordinazioni dei pasti. Si vuole velocizzare le comunicazioni tra Cliente e personale ADISU, ottimizzare la metodologia d'acquisto dei pasti, facilitare le mansioni dell'operatore Mensa nella distribuzione dei pasti e agevolare le funzioni gestionali dell'intero servizio.

1.2 Design Goals

Rank/ Priorità	ID Design Goal	Descrizione Design Goal	Categoria	Origine	Trade off
1	DG_1	Il software sviluppato deve avere i minimi tempi di risposta nell'assolvere alle funzionalità per cui è stato creato; l'utente dovrà percepire fluidità e prontezza nel ricevere i dati richiesti	Performance - Response Time	RNF_3	Response Time vs Memory: Se il software non adempie ai requisiti di tempo di risposta è imprescindibile l'utilizzo di più spazio di memoria per velocizzare il sistema.
2	DG_2	Il software deve essere accessibile 24 ore su 24, tranne durante i periodi di manutenzione. La memoria necessaria per il funzionamento del sistema è in diretta correlazione con quella utilizzata per mantenere il Database.	Performance - Memory	RNF_3	Performance vs Maintenance: La manutenibilità sarà preferita alla performance per poter favorire i processi di aggiornamento software agli sviluppatori.
3	DG_3	Il sistema avvertirà l'utente nel caso riceva input non validi con l'aiuto di messaggi.	Dependability - Robustness	RNF_2	

4	DC_4	Il sistema deve garantire l'affidabilità dei servizi presentati. La funzionalità di login dell'utente sarà gestita in modo affidabile.	Dependability - Dependability	RNF_2	
5	DG_5	Tutti gli utenti per accedere alla propria area personale necessitano di username e password. Ogni utente può, in totale sicurezza, può compiere le varie azioni nel sistema.	Dependability - Security	RNF_2	
6	DG_6	Il sistema è accessibile a tutti gli utenti autorizzati, quindi iscritti/dipendenti nell'Università degli Studi di Salerno, e dipendenti ADISU.	Dependability - Availability	RNF_2	Disponibilità vs Fault Tolerance: Il sistema deve essere sempre disponibile all'utente in caso di errori di una funzionalità, anche al costo di non renderla disponibile per un lasso di tempo
7	DG_7	Il sistema permette di salvare i dati automaticamente, così da poterli recuperare in caso di fallimenti	Dependability - Fault Tolerance	RNF_2	
8	DG_8	Si stima un costo di 50 ore per ogni project member per la progettazione e sviluppo del sistema. In totale 350 ore	Cost - Development Cost		Development Cost vs Security: Il sistema non permette lo sviluppo di un software simultaneamente sicuro e costi poco
9	DG_9	Il sistema presenterà il supporto multilingua, particolarmente per la lingua inglese. Supporterà adeguatamente l'utente per quanto riguarda l'usabilità del sistema. Inoltre, il sistema sarà	Maintenance - Extensibility	RNF_4	Memory vs Extensibility: Il sistema deve permettere l'estensibilità a discapito della memoria utilizzata, in modo tale da permettere al cliente di richiedere agli sviluppatori nuove

		preposto a adattarsi a nuovi cambiamenti			funzionalità.
10	DG_10	Il sistema non permette agli utenti di poter modificare le sue funzionalità.	Maintenance - Changeability	RNF_4	
11	DG_11	L'interazione con il sistema avviene tramite browser essendo sviluppato come una web application. È quindi accessibile da un qualunque dispositivo dotato di qualsiasi browser.	Maintenance - Portability	RNF_4	
12	DG_12	I requisiti sono tracciati attraverso l'utilizzo di una matrice di tracciabilità.	Maintenance - Requirements Traceability	RNF_4	
13	DG_13	Il sistema prevede un'interfaccia ordinata, user- friendly e di facile utilizzo.	End User - Usability	RNF_1	
14	DG_14	Il sistema si rende utile in quanto velocizza la gestione del servizio mensa offerto, gestendo i dati in formato digitale e non in maniera cartacea.	End User - Utility	RNF_1	

1.3 Definizioni, Acronimi e abbreviazioni

UE: UnisaEAT.

RAD: Requirements Analysis Document.

SDD: System Design Document.

User-Friendly: letteralmente “amichevole per l’utente”, di facile utilizzo anche per chi non è esperto.

DB: Data Base.

MongoDB: è un DBMS non relazionale orientato ai documenti. Sfrutta il formato JSON per la memorizzazione e rappresentazione dei dati

DBMS: Database Management System.

JSON: JavaScript Object Notation, è un formato adatto all’interscambio di dati tra applicazioni client/server.

MVC: Model View Controller.

Event-driven: meccanismo per il flusso di controllo globale. Il flusso globale del sistema è guidato da una serie di eventi.

1.4 Riferimenti

- EA_RAD_V_1.0
- Slide del corso, presenti sulla piattaforma e-learning.
- Libro: Object-Oriented Software Engineering (Using UML, Patterns, and Java) Third Edition.
Autori: Bernd Bruegge & Allen H. Dutoit.

1.5 Panoramica

Capitolo 1: contiene l’introduzione con gli obiettivi del sistema, i design goals e un elenco di definizioni, acronimi e abbreviazioni utili alla comprensione dell’intera documentazione.

Capitolo 2: descrive le funzionalità offerte dal sistema corrente.

Capitolo 3: viene presentata l’architettura del sistema proposto, nella quale sarà gestita la decomposizione in sottosistemi, il mapping hardware/software, i dati persistenti, il controllo degli accessi e sicurezza, il controllo del flusso globale del sistema e le condizioni limite.

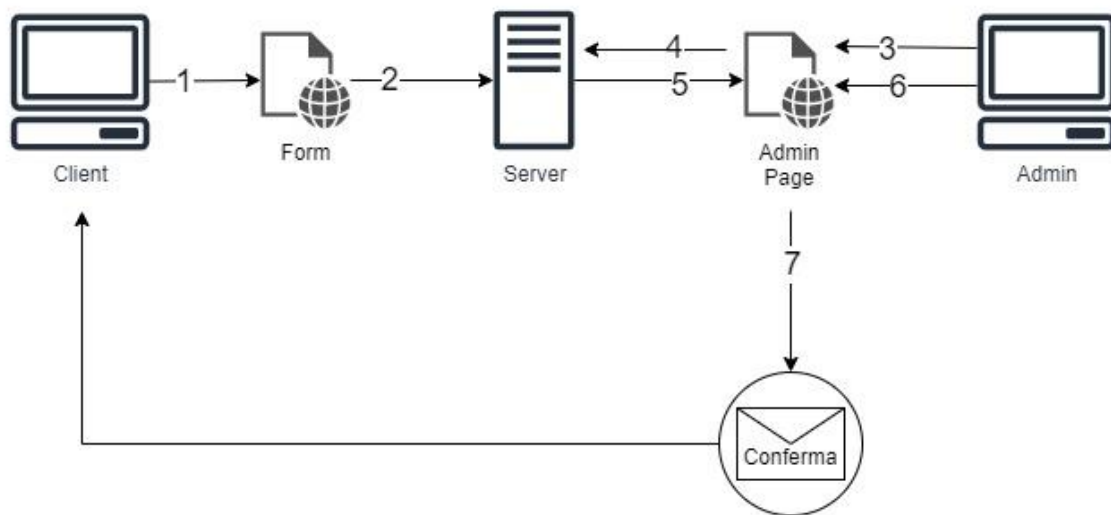
Capitolo 4: vengono presentati i servizi offerti dai sottosistemi.

2. Architettura del Sistema corrente

I servizi mensa di tutte le università campane vengono gestiti dall'ADISU attraverso un apposito portale. Non esiste, attualmente, un prodotto software finalizzato alla gestione dei servizi mensa dedicato esclusivamente all'Università degli Studi di Salerno.

Per usufruire del servizio di ristorazione bisogna disporre di un tesserino magnetico. Questo può essere attivato e rinnovato recandosi fisicamente al Punto Informazione, oppure compilando un form attraverso il portale dedicato e attendere l'e-mail di conferma da parte dell'admin. Una volta che il tesserino magnetico è stato attivato, bisogna recarsi fisicamente al Punto Informazione per il ritiro.

Lo schema sottostante descrive il funzionamento corrente per la richiesta del tesserino:



1. Il client visualizza sul suo browser il form per la richiesta del tesserino e lo compila;
2. Il client conferma i dati del form. Il form viene inviato, tramite protocollo HTTP, al web server;
3. L'admin fa richiesta di visualizzare l'Admin Page sul suo browser;
4. Avviene un HTTP request dei dati da visualizzare al server;
5. Il server restituisce i dati;
6. L'admin visualizza il form compilato dal client e ne conferma la correttezza;
7. Un'email di conferma viene autogenerata e inviata al client.

La ricarica del tesserino avviene o recandosi ai dispositivi di ricarica fisici situati all'interno della mensa, o utilizzando la piattaforma online pagoPA/myPay.

Il ritiro dei pasti avviene attendendo l'identificazione ai tornelli, seguendo una fila indiana, e completando la procedura uno per volta. Stesso discorso vale per il pagamento.

La richiesta del tesserino magnetico può risultare sveniente a molti clienti, in quanto richiede il ritiro del tesserino recandosi fisicamente al Punto Informazione. I possessori di tesserino potrebbero, non volutamente, smarrirlo, costringendoli a dover ripetere la procedura di richiesta.

La ricarica del tesserino tramite dispositivo fisico presenta i seguenti problemi: impossibilità di ricevere il resto; ricarica solo con banconote; presenza di lunghe file negli orari di punta; lentezza della procedura; continui malfunzionamenti dei dispositivi di ricarica.

La ricarica del tesserino tramite la piattaforma pagoPA/myPay richiede una lunga attesa prima dell'effettivo accredito e di e-mail di conferma, in quanto l'admin dovrà confermare la correttezza dei dati inseriti dal cliente.

Il ritiro dei pasti risulta piuttosto lento, in quanto, prima del ritiro, durante il ritiro e per effettuare il pagamento, bisogna attendere il proprio turno in fila indiana.

La piattaforma online UnisaEat, per risolvere le problematiche sopra riportate, consentirà di:

- richiedere un tesserino digitale e sostituirlo a quello magnetico;
- ricaricare il tesserino online, attraverso la piattaforma, senza dover attendere lunghi lassi di tempo per l'esito positivo dell'operazione;
- permettere la prenotazione dei pasti ed il pagamento con procedura online, in modo da eliminare le lunghe file per queste operazioni e velocizzare i tempi;
- altre funzioni per rendere più efficiente il servizio mensa...

3. Architettura del Sistema proposto

3.1 Panoramica

Il sistema proposto consiste in una piattaforma Web che si rivolge a tutti i clienti (studenti e non) che usufruiscono del servizio di ristorazione dell'Università degli Studi di Salerno, agli operatori della mensa e al personale ADISU. Gli utenti che avranno accesso alla piattaforma saranno quindi: cliente, operatore mensa, personale ADISU e l'admin. Tutti gli utenti potranno effettuare il log-in e il log-out, nessun utente può effettuare la registrazione, in quanto i clienti effettuano il log-in tramite le credenziali di esse3, l'operatore mensa sarà inserito dal personale ADISU e il personale ADISU sarà inserito dall'. Una volta effettuato il log-in, la piattaforma metterà a disposizione diverse funzionalità a seconda del tipo di utente ad aver effettuato l'accesso.

Il cliente potrà richiedere, rinnovare e ricaricare il tesserino digitale, visualizzare la sezione delle FAQ, visualizzare il menu giornaliero e potrà effettuare la prenotazione di un pasto, il sistema a questo punto verificherà l'effettivo acquisto del pasto e genererà un QR che verrà inviato al cliente, che verrà usato per il ritiro del pasto nella mensa.

L'operatore mensa potrà gestire il menù giornaliero e visualizzare le statistiche settimanali.

Il personale ADISU potrà gestire le FAQ e gestire gli operatori mensa.

Il cliente e il personale ADISU avranno accesso ad un sistema di messaggistica istantanea, dove il cliente può inviare dei messaggi al personale ADISU, per richiedere ulteriori informazioni e risolvere problematiche, e inoltre il cliente, il personale ADISU e l'operatore mensa potranno inviare dei ticket che conterranno la descrizione del problema riscontrato con la piattaforma, quest'ultimi arriveranno all'admin che si occuperà di risolverli.

L'architettura utilizzata è di tipo repository, in quanto adatta quando il sistema è basato sull'archivio dei dati e la loro modifica è frequente e coinvolge più parti. In questo tipo di architettura i sottosistemi che compongono il software accedono e modificano una singola struttura dati, chiamata repository; questo fa sì che i vari sottosistemi siano fra di loro relativamente indipendenti, in quanto interagiscono solo mediante il repository. Come pattern è utilizzato l'MVC (Model-View-Controller), un pattern architetturale molto diffuso nello sviluppo di sistemi software, in particolare nell'ambito object-oriented, in grado di separare la logica di presentazione dei dati, dalla logica di business.

3.2 Decomposizione in sottosistemi

3.2.1 Decomposizione in layer

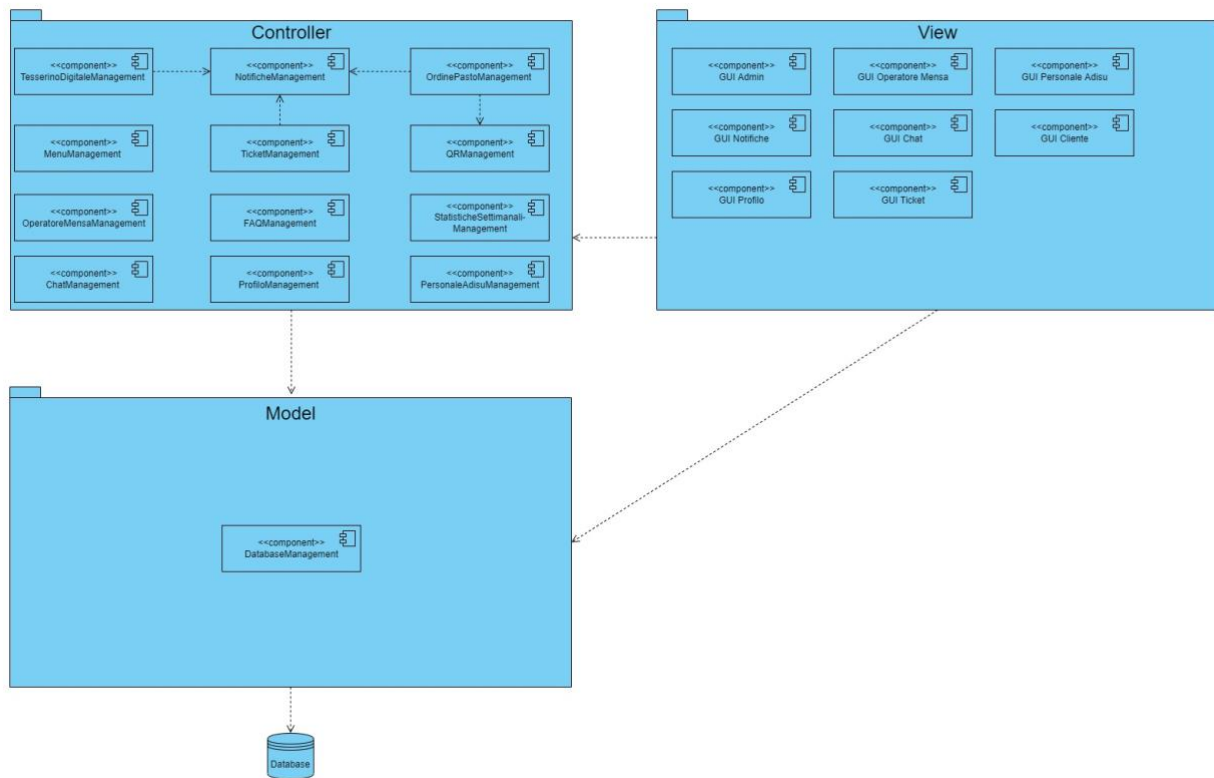
Il sistema prevede una decomposizione in tre layer. Quest'ultimi gestiscono diversi aspetti e funzionalità del sistema:

- **Model:** si occupa della gestione dei dati e contiene i metodi d'accesso ad essi;
- **View:** si occupa di visualizzare i dati all'utente e gestisce l'interazione con quest'ultimo;
- **Controller:** si occupa della logica del sistema. Riceve i comandi dell'utente attraverso la view e reagisce eseguendo delle operazioni che possono interessare il model e che portano ad un cambiamento di stato del view.

Si è quindi deciso di utilizzare un'architettura MVC, in quanto UnisaEat sarà una piattaforma interattiva che fornirà interfacce multiple dello stesso Model.

3.2.2 Decomposizione in sottosistemi

Il sistema è suddiviso nei seguenti sottosistemi:



Il livello View prevede la gestione dei seguenti sottosistemi:

- GUI Admin
- GUI Cliente
- GUI Operatore Mensa
- GUI Personale Adisu
- GUI Notifiche
- GUI Chat
- GUI Profilo
- GUI Ticket

Il livello Controller prevede la gestione dei seguenti sottosistemi:

- TesserinoDigitaleManagement

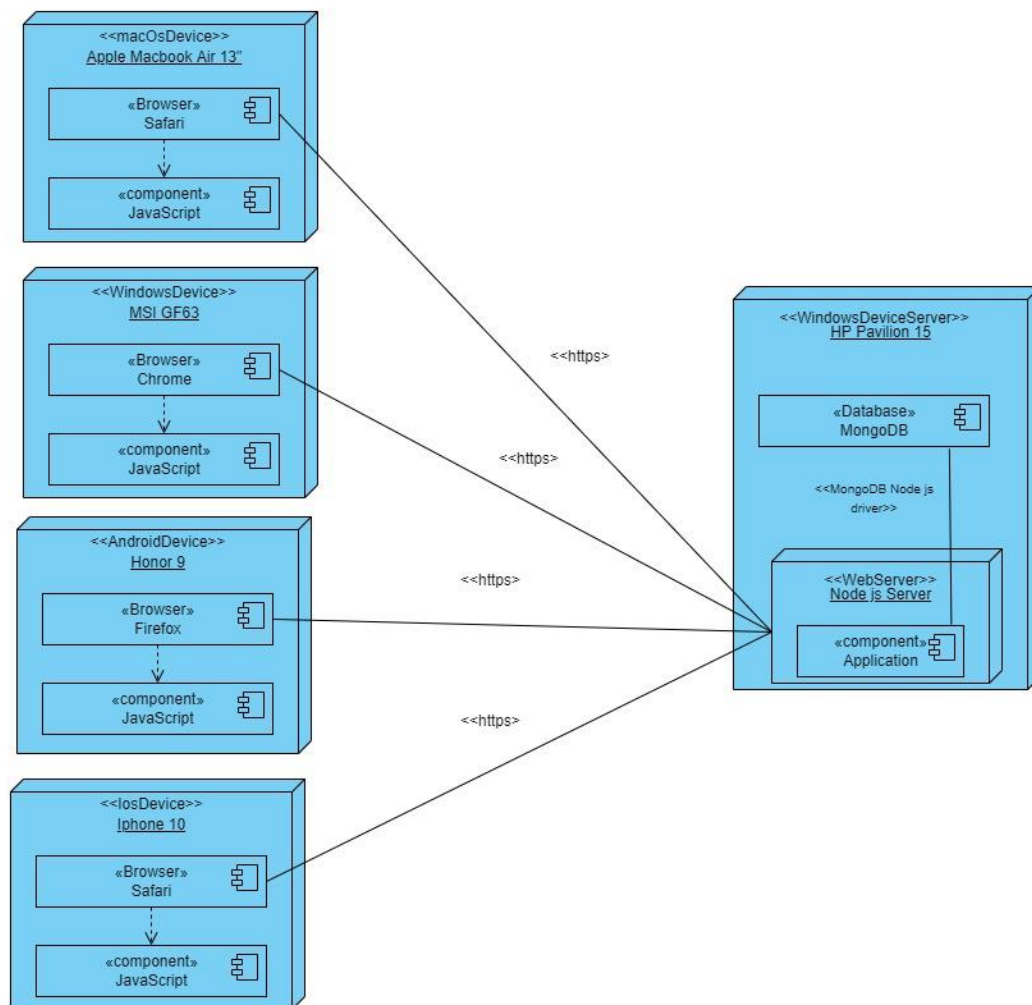
- NotificheManagement
- OrdinePastoManagement
- MenuManagement
- TicketManagement
- QRManagement
- OperatoreMensaManagement
- FAQManagement
- StatisticheSettimanaliManagement
- ChatManagement
- ProfiloManagement
- PersonaleAdisuManagement

Il livello Model prevede la gestione del seguente sottosistema:

- DatabaseManagement

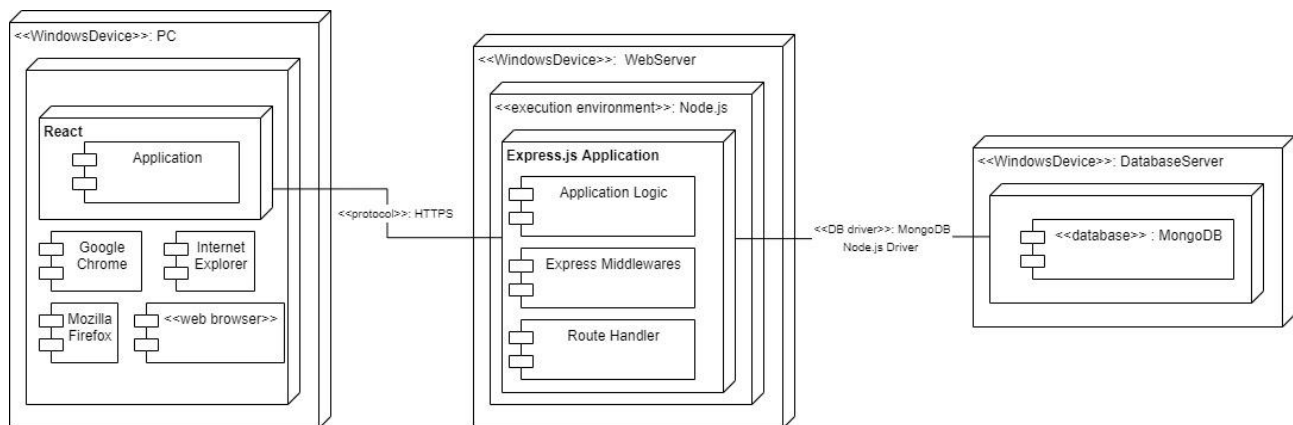
Deployment Diagram

Gli utenti che vogliono interagire con l'interfaccia del sistema hanno bisogno di un browser che sappia interpretare Javascript. I client richiedono le risorse al Web Server, creato con l'ausilio del run-time system Node js, tramite protocollo HTTPS. L'applicazione si interfaccia con il database MongoDB utilizzando il driver MongoDB Node js.



3.3 Mapping Hardware/Software

Il sistema, per garantire i servizi, utilizza un'architettura di tipo Client-Server. Lato Server è presente un ambiente run-time, open-source e multiplatforma chiamato **Node.js**, grazie a esso abbiamo la possibilità di eseguire il codice JavaScript lato server e produrre pagine web dinamiche prima che le stesse siano inviate al web browser dell'utente. Il Client è rappresentato dal Browser utilizzato dall'utente. Abbiamo scelto Node.js poiché possiede un'architettura ad eventi capace di realizzare operazioni input/output (I/O) asincrone, questa scelta di design punta a ottimizzare la velocità di esecuzione e la scalabilità nella nostra applicazione web. Dal punto di vista di memorizzazione dei dati, poiché utilizziamo il formato JSON, la soluzione migliore è l'utilizzo di **MongoDB**, un database non relazionale (NoSQL).



3.4 Gestione dei dati persistenti

Per la memorizzazione dei dati è stato scelto un Database non relazionale, utilizzando il DBMS orientato ai documenti MongoDB. Il vantaggio della scelta di un Database orientato ai documenti consiste nella possibilità di conservare le informazioni in documenti JSON, ciò consente di memorizzare molti dati, anche annidati uno nell'altro; i documenti sono raggruppati in collezioni che possono essere anche eterogenee, ovvero non c'è uno schema fisso per i documenti che ne fanno parte. Un'organizzazione di questo tipo consente di apportare modifiche agli attributi di un'entità anche in corso d'opera, inoltre le tecniche di indicizzazione garantiscono alte performance di accesso ai dati nonostante la possibile assenza di relazioni tra gli oggetti.

3.4.1 Descrizioni delle entità persistenti

Cliente

- ClienteID: string
- Nome: string
- Cognome: string
- Password: string
- Città: string
- E-mail: string
- Indirizzo: string
- Tesserino: tesserino

La collection Client contiene le informazioni anagrafiche del cliente, la sua e-mail, il numero del suo tesserino e la sua password. Il cliente è identificato univocamente dalla sua matricola in quanto essa è unica all'interno dell'università.

Admin

- E-mail: string
- Password: string
- Cognome: string
- Nome: string
- AdminID: string

La collection Admin contiene le informazioni anagrafiche, un identificativo univoco, la sua e-mail e la sua password.

Personale

- Nome: string
- Cognome: string
- Password: string
- E-mail: string
- Numerotelefono: string
- Datadinascita: date
- Indirizzo: string
- Ruolo: string
- PersonaleID: string
- Disponibilita: boolean

La collection Personale contiene le informazioni relative all'anagrafica, la sua e-mail, la sua password, un identificativo per riconoscere univocamente una persona e il ruolo che può essere: personale ADISU o operatore mensa. Nel caso del personale ADISU, il campo Disponibilita permette al cliente di conoscere la disponibilità del personale nell'iniziare o meno una chat.

Messaggio

- SenderID: string

- ReceiverID: string
- MessaggioID: string
- Testo: string
- Data: date

La collection Messaggio contiene l'id del singolo messaggio, il testo del messaggio con il relativo id del mittente e del destinatario e la data di quando è stato inviato il messaggio.

Notifica

- ReceiverID: string
- NotificationID: string
- Testo: string
- Tipo: string
- Titolo: string
- Visualizzazione: Boolean

La collection Notifica contiene il testo della notifica, l'id di chi riceve la notifica, un identificativo univoco, il tipo di notifica, il titolo che contiene la notifica e un valore che specifica se la notifica è stata visualizzata oppure no. Tutte le notifiche di un utente risultano visualizzate quando l'utente preme sull'icona per visualizzare tutte le notifiche.

Tesserino

- TesserinoID: string
- Saldo: float
- DataScadenza: date

La collection tesserino contiene un identificativo univoco, il saldo e la data di scadenza.

Ordine

- OrdineID: string
- Prezzo: string
- DataOrdine: date
- BoolPranzo: boolean
- Stato: string
- ListaPasti: pasto[]

- QR: byte[]
- AcquirenteID: string

La collection Ordine contiene un identificativo univoco, il prezzo dell'ordine, la data di quando è stato effettuato l'ordine, un valore booleano per identificare se l'ordine effettuato viene consumato a pranzo o a cena, lo stato dell'ordine, la lista dei pasti che contiene l'ordine, il QR code associato all'ordine dopo il pagamento e chi ha effettuato l'acquisto.

Menu

- Nome: string
- MenuID: string
- Pasti: Pasto[]

La collection Menu contiene un identificativo univoco, il nome del menu, il giorno in cui viene esposto, e i pasti che contiene.

Pasto

- PastoID: string
- Nome: string
- Descrizione: string
- Disponibilità: int
- Immagine: byte[]
- Categoria: string
- Ingredienti: string
- NumeroOrdinazioni: int

La collection Pasto contiene un id univoco per identificare il pasto, il nome del pasto, la descrizione, la categoria (primo, secondo, contorno, dolce), la disponibilità, l'immagine, gli ingredienti ed un numero di ordinazioni che si resetta settimanalmente.

FAQ

- FaqID: string
- Domanda: string
- Risposta: string

La collection FAQ contiene un identificativo univoco, la domanda e la risposta.

Ticket

- TicketID: string
- Titolo: string
- Problema: string
- Soluzione: string
- Data: date
- SenderID: string

La collection Ticket contiene un identificativo univoco, il titolo del ticket, il problema, la soluzione e la data di quando è stato presentato il ticket e l'identificativo di chi lo ha presentato.

StatisticheSettimanali

- StatisticheID: string
- Titolo: string
- Descrizione: string
- DataInizio: date
- DataFine: date
- PastiPiuOrdinati: array
- PastiMenoOrdinati: array

La collection Statistiche settimanali contiene un identificativo univoco, una descrizione, la data di inizio della settimana e la data di fine settimana. Contiene due array, contenenti id e numero di prenotazioni in una determinata settimana dei pasti più e meno ordinati.

3.5 Controllo degli accessi e sicurezza

All'interno del sistema "UnisaEAT", gli attori hanno il permesso di eseguire diverse operazioni. Grazie a questo sistema ci è possibile gestire gli accessi dei vari utenti alle funzionalità del software. Per approfondire al meglio il controllo degli accessi, viene utilizzata una matrice degli accessi, all'interno della quale le righe rappresentano gli attori e le colonne le classi. Ogni entry (attore, classe) della matrice contiene le operazioni consentite da quell'attore sulle istanze di quella classe.

Sottosistema Attore	Messaggio	Notifica	Ordine	Menu
Cliente	<ul style="list-style-type: none"> • Invio Messaggio • Rimozione messaggio • Ricezione messaggio • Modifica messaggio 	<ul style="list-style-type: none"> • Ricezione Notifica • Visualizza elenco Notifiche • Rimozione Notifiche 	<ul style="list-style-type: none"> • Scelta Pasti • Pagamento Pasto • Visualizza Riepilogo ordine • Ricezione QR • Visualizzazione QR 	<ul style="list-style-type: none"> • Visualizzazione Menu
Admin		<ul style="list-style-type: none"> • Generazione notifica 		
Personale ADISU	<ul style="list-style-type: none"> • Invio Messaggio • Rimozione messaggio • Ricezione messaggio • Modifica messaggio 	<ul style="list-style-type: none"> • Ricezione Notifica • Visualizzazione elenco Notifiche • Rimozione Notifica 		
Operatore Mensa		<ul style="list-style-type: none"> • Ricezione Notifica • Visualizzazione elenco Notifiche 		<ul style="list-style-type: none"> • Visualizzazione Menu • Inserimento Menu • Modifica Menu

		<ul style="list-style-type: none"> Rimozione Notifica 		
--	--	--	--	--

Sottosistema Attore	Tesserino	Pasto	FAQ	Ticket
Cliente	<ul style="list-style-type: none"> Richiesta Tesserino Rinnovo Tesserino Ricarica Tesserino Visualizzazione saldo 	<ul style="list-style-type: none"> Scelta Pasti Pagamento Pasto Visualizza Riepilogo Ordine 	<ul style="list-style-type: none"> Visualizzazione FAQ 	<ul style="list-style-type: none"> Compilazione Ticket
Admin				<ul style="list-style-type: none"> Visualizzazione lista ticket Risoluzione Ticket

Personale ADISU			<ul style="list-style-type: none"> • Visualizzazione FAQ • Inserimento domanda in FAQ • Modifica domanda in FAQ • Rimozione domanda in FAQ 	<ul style="list-style-type: none"> • Compilazione Ticket
Operatore Mensa				<ul style="list-style-type: none"> • Compilazione Ticket

Sottosistema Attore	Statistiche settimanali	Gestione Operatore Mensa	Gestione Personale ADISU
Admin			<ul style="list-style-type: none"> • Inserimento Personale ADISU • Rimozione Personale ADISU • Visualizzazione lista Personale ADISU

Personal e ADISU		<ul style="list-style-type: none"> • Inserimento Operatore Mensa • Rimozione Operatore Mensa • Visualizzazione lista operatori Mensa 	
Operat ore Mensa	<ul style="list-style-type: none"> • Visualizzazi one Statistiche settimanali 		

Sottosistema Attore	Profilo
Cliente	<ul style="list-style-type: none"> • Visualizzazione Area Personale

Admin	<ul style="list-style-type: none">• Visualizzazione Area Personale• Modifica Password
Personale ADISU	<ul style="list-style-type: none">• Visualizzazione Area Personale• Modifica Password
Operatore Mensa	<ul style="list-style-type: none">• Visualizzazione Area Personale• Modifica Password

3.6 Controllo flusso globale del sistema

Il flusso del sistema “UnisaEAT” fornisce una funzionalità che richiede una continua interazione da parte dell’utente, per cui il controllo del flusso globale del sistema è di tipo “Event-driven” control, ovvero guidato dagli eventi. Il ciclo principale attende un evento esterno, quando un evento si verifica è spedito all’oggetto appropriato (sulla base dell’info associata all’evento). Il controllo risiede in un dispatcher che chiama le funzioni di sottosistema. Tale meccanismo è da considerarsi flessibile e ideale per le interfacce utenti.

3.7 Condizioni Boundary

START-UP

Per il primo start-up del sistema “UnisaEat” è necessario l’avvio della runtime di Javascript “Node.js” che permette di eseguire il codice Javascript per la gestione dei dati persistenti.

Successivamente ci sarà:

- Un’interfaccia HOMEPAGE che contiene le informazioni riguardante i servizi offerti dal sistema accompagnate da uno slideshow, un footer, e una navbar con un bottone che permette di accedere all’interfaccia LOG-IN;
- Un’interfaccia per il LOG-IN che permette l’autenticazione attraverso opportune credenziali (e-mail e password). Una volta effettuato l’accesso il sistema “UnisaEat” riporta l’utente alla propria area personale dalla quale è possibile usufruire di tutte le operazioni che il sistema mette a disposizione (Es. Il cliente può effettuare la prenotazione di un pasto, l’operatore mensa può aggiungere il menù giornaliero, ecc.)

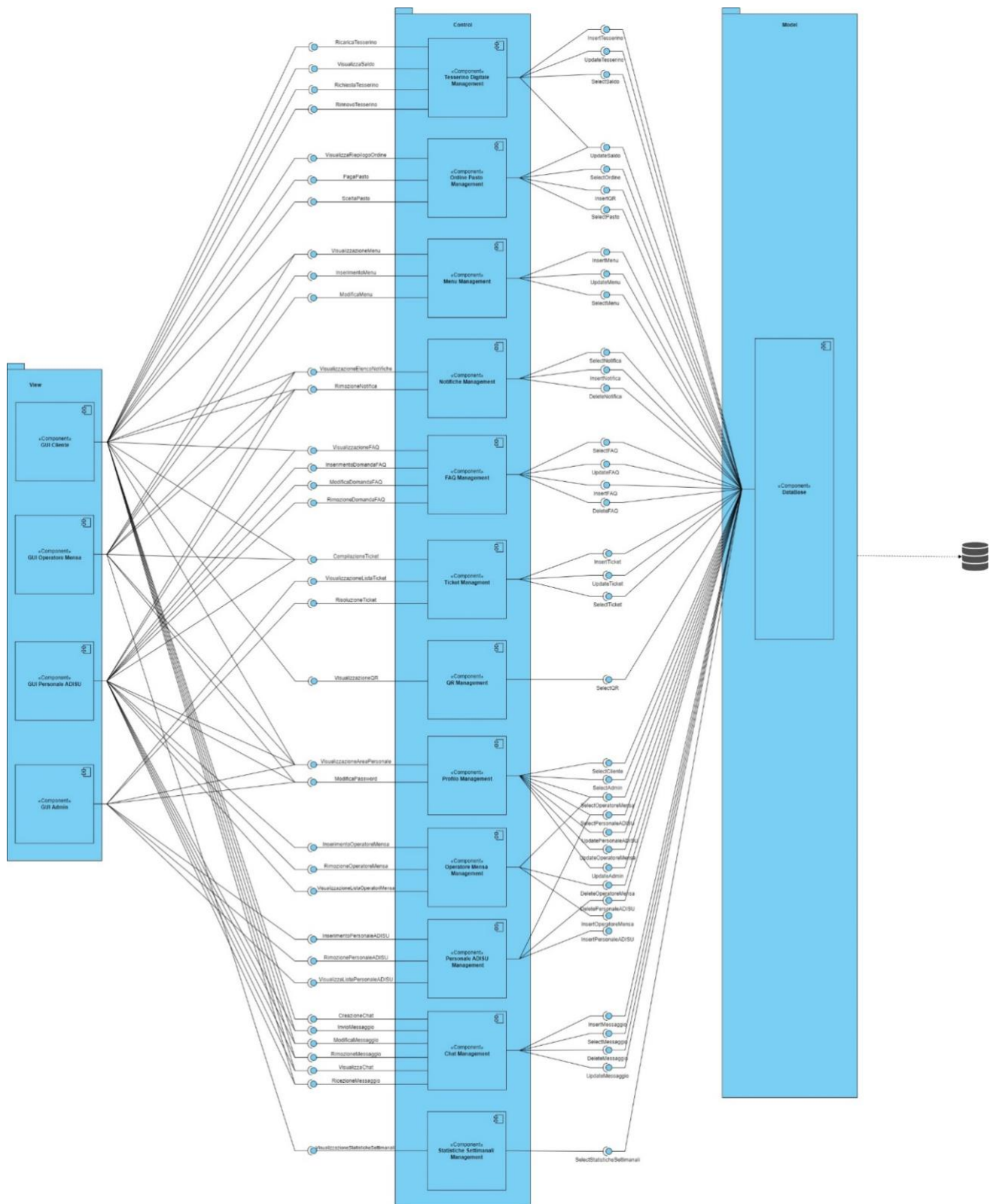
TERMINAZIONE

Al momento della corretta terminazione dell’applicazione, si ha una regolare chiusura del sistema accompagnata da un LOG-OUT che permette all’utente di effettuare l’uscita dalla sessione.

FALLIMENTO

1. Nel caso di guasti dovuti al sovraccarico del database e quindi con un fallimento di quest’ultimo, è possibile l’uso della funzionalità di sincronizzazione dei dati, chiamatasi “Replica”, di MongoDB per avere a disposizione più copie dei dati utili, e poterne effettuare così la rigenerazione;
2. Nel caso in cui si verifichi un’interruzione inaspettata dell’alimentazione, non sono previsti metodi che ripristino lo stato del sistema precedente allo spegnimento non volontario;
3. Altro caso di fallimento potrebbe essere dovuto ad un errore critico nell’hardware, per il quale non è prevista alcuna contromisura;
4. Ancora, un altro caso di fallimento potrebbe essere causato dal software stesso che provoca una ~~dian~~ non prevista per errori durante la fase d’implementazione. Non essendo stipulate politiche correttive, le uniche operazioni disponibili e consentite sono quelle di chiusura e riavvio successivo del sistema.

4. Servizi dei sottosistemi



GUI Cliente offre 20 servizi all'interfaccia Control:

- RicaricaTesserino
- RinnovoTesserino
- VisualizzazioneSaldo
- RichiestaTesserino
- VisualizzazioneAreaPersonale
- SceltaPasto
- PagaPasto
- VisualizzaRiepilogoOrdine
- VisualizzazioneMenu
- CreazioneChat
- VisualizzaChat
- RimuoviMessaggio
- InviaMessaggio
- ModificaMessaggio
- RicezioneMessaggio
- VisualizzazioneFAQ
- CompilazioneTicket
- VisualizzazioneElencoNotifiche
- RimozioneNotifica
- VisualizzazioneQR

GUI Operatore Mensa offre 9 servizi all'interfaccia Control:

- VisualizzazioneMenu
- InserimentoMenu
- ModificaMenu
- VisualizzazioneElencoNotifiche
- RimozioneNotifica
- CompilazioneTicket
- VisualizzazioneAreaPersonale

- ModificaPassword
- VisualizzazioneStatisticheSettimanali

GUI Personale ADISU offre 18 servizi all'interfaccia Control:

- VisualizzazioneElencoNotifiche
- RimozioneNotifica
- VisualizzaFAQ
- InserimentoDomandaFAQ
- ModificaDomandaFAQ
- RimozioneDomandaFAQ
- CompilazioneTicket
- VisualizzazioneAreaPersonale
- ModificaPassword
- VisualizzazioneStatisticheSettimanali
- InserimentoOperatoreMensa
- RimozioneOperatoreMensa
- VisualizzazioneListaOperatoreMensa
- InvioMessaggio
- ModificaMessaggio
- RimozioneMessaggio
- VisualizzaChat
- RicezioneMessaggio

GUI Admin offre 7 servizi all'interfaccia Control:

- VisualizzazioneListaTicket
- RisoluzioneTicket
- VisualizzazioneAreaPersonale
- ModificaPassword
- InserimentoPersonaleADISU
- RimozionePersonaleADISU
- VisualizzaListaPersonaleADISU

Tesserino Digitale Management offre 4 servizi all'interfaccia Model:

- InsertTesserino
- UpdateTesserino
- SelectSaldo
- UpdateSaldo

Ordine Pasto Management offre 4 servizi all'interfaccia Model:

- UpdateSaldo
- SelectOrdine
- InsertQR
- SelectPasto

Menu Management offre 3 servizi all'interfaccia Model:

- InsertMenu
- UpdateMenu
- SelectMenu

Notifiche Management offre 3 servizi offre 4 servizi all'interfaccia Model:

- SelectNotifica
- InsertNotifica
- DeletNotifica

FAQ Management offre 4 servizi all'interfaccia Model:

- SelectFAQ
- UpdateFAQ
- InsertFAQ
- DeleteFAQ

Ticket Management offre 3 servizi all'interfaccia Model:

- InsertTicket
- UpdateTicket
- SelectTicket

QR Management offre 1 servizio all'interfaccia Model:

- SelectQR

Profilo Management offre 7 servizi all'interfaccia Model:

- SelectCliente

- SelectAdmin
- SelectOperatoreMensa
- SelectPersonaleADISU
- UpdateAdmin
- UpdatePersonaleADISU
- UpdateOperatoreMensa

Operatore Mensa Management offre 3 servizi all'interfaccia Model:

- InsertOperatoreMensa
- DeleteOperatoreMensa
- SelectOperatoreMensa

Operatore Personale ADISU Management offre 3 servizi all'interfaccia Model:

- InsertPersonaleADISU
- DeletePersonaleADISU
- SelectPersonaleADISU

Chat Management offre 4 servizi all'interfaccia Model:

- InsertMessaggio
- SelectMessaggio
- DeleteMessaggio
- UpdateMessaggio

Statistiche Settimanali Management offre 1 servizio all'interfaccia Model:

- SelectStatisticheSettimanali