Università degli Studi di Salerno
Corso di Ingegneria del Software

mobilFarm
System Design Document
Versione 0.0

Data: 13/12/2016

Partecipanti:

Nome	Matricola
D'Avanzo Giuseppina	0512103032
Fiorentino Giuseppa	0512103158
Menichino Alfonso	0512102852
Volpe Alberto	0512103311

Revision History

Data	Versione	Descrizione	Autore

Indice

1. INTRODUZIONE

1.1 Scopo del Sistema

Lo scopo è quello di realizzare un sistema di assistenza virtuale operante nel settore farmaceutico.
Permettendo all'utente di ottenere sostegno medico per la gestione di piani terapeutici e visite polispecialistiche.
Un sistema di supporto che gestisce la somministrazione di medicinali nei tempi e dosi corrette ed altre varie componenti sia che l'utente registrato è un medico si che l'utente registrato è un paziente.

1.2 Obiettivi di Design

L'interfaccia del sistema e il sistema stesso saranno realizzati a partire dai requisiti funzionali, non funzionali, dagli scenari e dai casi d'uso.
Successivamente grazie ad un intenso testing (anche con gli utenti) raffineremo le funzionalità e le relative interfacce applicando la tecnica del Cognitive Walktrough ed effettuando testing di usabilità attraverso il mago di Oz.

1.2.1 Performance

Il sistema deve garantire una risposta alle richieste dell'utente in tempo reale con una latenza massima di 10 secondi, ciò implica l'utilizzo di un sistema back-end basato su un sistema cloud scalabile che permetta di garantire un eccellente tempo di risposta in maniera indipendente dal throughput.
Anche la quantità di memoria deve essere scalabile per permettere di aumentare e diminuire le performance del sistema in maniera automatica in base alle richieste.
AWS EC2 risponde in maniera efficace a questi obiettivi dato che offre una gestione automatica del Load Balancing, di AutoScaling e Provisioning.

1.2.2 Affidabilità

Il sistema deve essere implementato in maniera tale da gestire tutti i tipi di errori generati dall'utente con la relativa gestione attraverso l'utilizzo delle eccezioni.
Inoltre si devono prevedere messaggi di errori relativi all'assenza di connessione ed a messaggi di errore provenienti dal back-end.
Nel back-end viene verificato la titolarità dell'utente in relazione alla chiamata.
Date tali premesse e considerando che il sistema deve essere disponibile sia al paziente che al dottore 7 giorni su 7, l'utilizzo di AWS EC2 è consigliato per garantire i criteri di affidabilità.

1.2.3 Costi

I costi necessari per realizzare mobilFarm sono relativi al pagamento degli stipendi degli sviluppatori al fine di sviluppare e manutenere il sistema e alle riunioni di training rivolte ai dottori per presentare il sistema.
I costi di amministrazione sono minimi grazie all'outsourcing della gestione

del sistema hardware ad Amazon*.*

1.2.4 Manutenzione

Il sistema deve essere flessibile per permettere l'estendibilità del sistema, la modificabilità delle funzioni già esistenti e deve consentire di riutilizzare l'implementazione esistente per altri domini applicativi che risultano paralleli *a quello in corso di studio.* Il sistema back-end garantisce un minor impegno risorse per effettuare il porting su client aventi piattaforme differenti.
Inoltre, grazie all'utilizzo di Apache e MySql nel back-end e di tecnologie opensource, questo potrà essere spostato facilmente su altri sistemi hardware.
Grazie alla documentazione sarà semplice mappare le funzionalità ai relativi requisiti.

1.2.5 Utente finale

L'usabilità viene garantita grazie all'utilizzo di una interfaccia omogenea e accessibile a tutte le categorie di utenti.

1.3 Definizioni, acronimi, e abbreviazioni

Acronimi e abbreviazioni:

RAD: Requirements Analysis Document;
SDD: System Design Document;
S.O.S. : Save Our Souls / Soccorso Occorre Subito / Richiesta di soccorso in codice Morse;
VMF: VirtualmobilFarm;
DP: dot per inch.

Definizioni:

VirtualmobilFarm: Armadietto farmaceutico virtuale;
Storia Clinica: insieme di tutti i piani terapeutici di un paziente;
Deuteranopia: l'insensibilità al verde;
Protanopia: insensibilità al rosso;
Tritanopia: insensibilità al blu.

1.4 Riferimenti

- B. Bruegge, A.H. Dutoit, Object Oriented Software Engineering – Using UML, Patterns and Java, Prentice Hall;
- Sommerville, Software Engineering, Addison Wesley.
- Lehrstuhl für Angewandte Softwaretechnik.

2. ARCHITETTURA DEL SISTEMA PROPOSTO

2.1 Panoramica

L'applicazione mobilFarm è possibile dividerla in **due sottosistemi principali**:
- il **front-end** relativo alla presentazione delle informazioni all'utente e alla gestione delle stesse tramite l'applicazione Android;
- il **back-end** che fornisce delle API di lettura/scrittura nel database.

Parallelamente il sistema verrà organizzato con un'architettura software **Three-Tier** mentre l'architettura dei sottosistemi sarà **chiusa** al fine di garantire alta manutenibilità e portabilità al costo dell'efficienza.
Quest'ultima verrà compensata dal sistema di AutoScaling di AWS EC2.

Il sottosistema **Presentation** si occupa della gestione del layout e della presentazione dei risultati all'utente, con le relative azioni applicabili.
Il sottosistema **Application** si occupa della logica di business sia lato app che lato server. Infine, il sottosistema **Logic** si occupa delle classi di dati utili al funzionamento dell'applicazione.

2.2 Decomposizione in sottosistemi

Abbiamo suddiviso il sistema in diversi sottosistemi seguendo il principio di **alta coesione e basso accoppiamento**.
Ogni sottosistema assume un ruolo preciso grazie alla specializzazione delle classi presenti al suo interno.

Il sistema mobilFarm è dotato di **3 sottosistemi**:

- **GestoreAccount**

  Questo sottosistema comprende le funzionalità relative sia agli account dei medici che dei pazienti, offrendo metodi utili alla registrazione, modifica, visualizzazione dei propri dati e visualizzazione dei dati relativi ai propri dottori o ai propri pazienti. Inoltre si occupa della verifica delle credenziali durante l'accesso all'applicazione e della modalità SOS.
- **GestoreFarmaco**

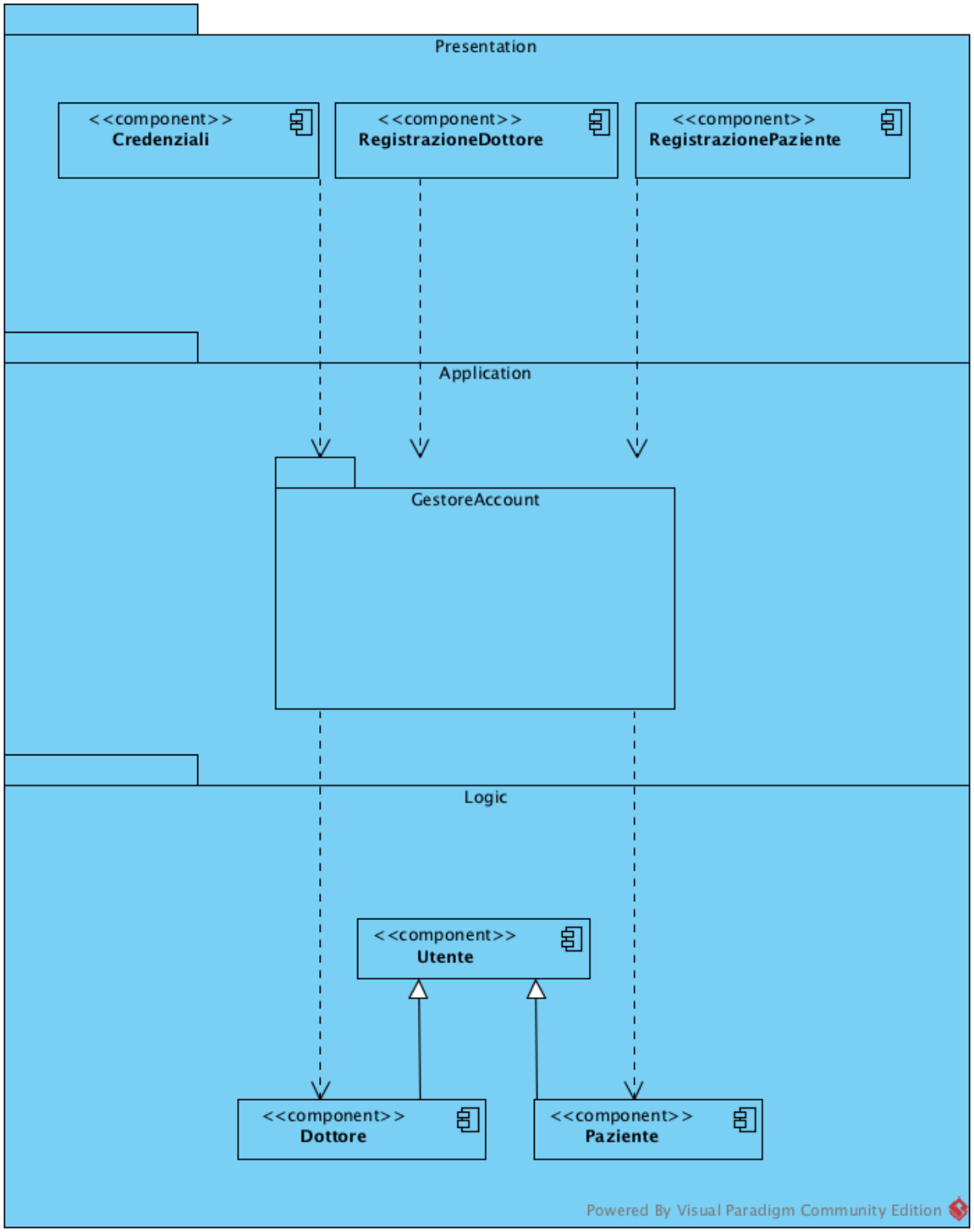
  Questo sottosistema comprende le funzionalità relative ai farmaci pubblici e quelli relativi al proprio virtualmobilFarm, offrendo metodi di inserimento, ricerca, notifica di scadenza e creazione (da parte dei soli dottori) dei farmaci.
- **GestorePiano**

  Questo sottosistema comprende tutte le funzionalità relative alla gestione dei piani terapeutici, offrendo classi e relativi metodi inerenti alla creazione di un nuovo piano, alla modifica e alla predisposizione di notifiche negli orari di assunzione.

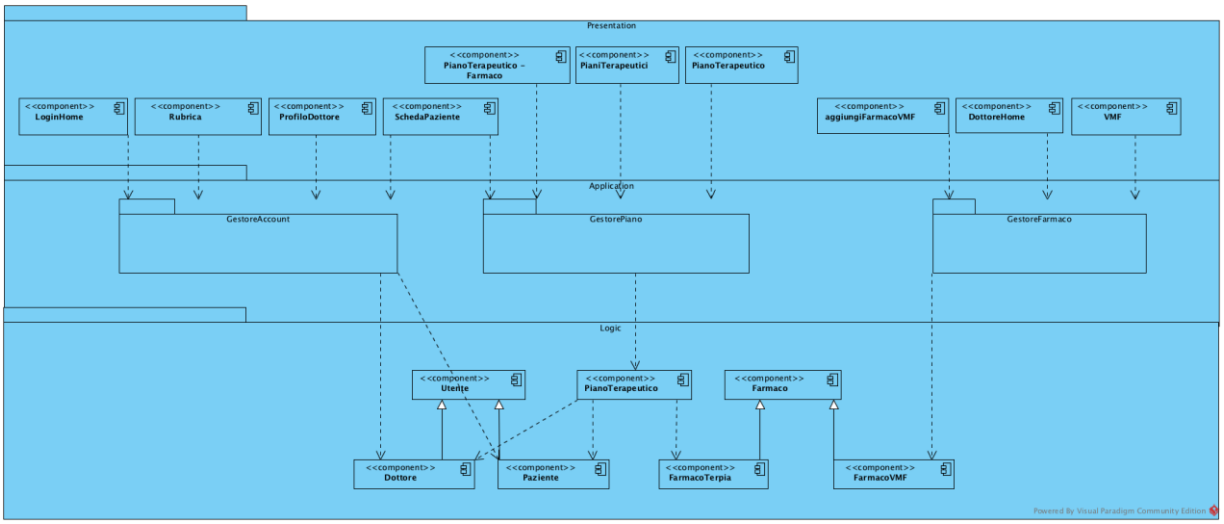
Di seguito vengono riportate le decomposizioni dei vari sottosistemi aventi un'architettura chiusa e strutturati nella tipologia Three-Tier inerenti ad ogni tipo di

utente (Utente non registrato, Dottore, Paziente).

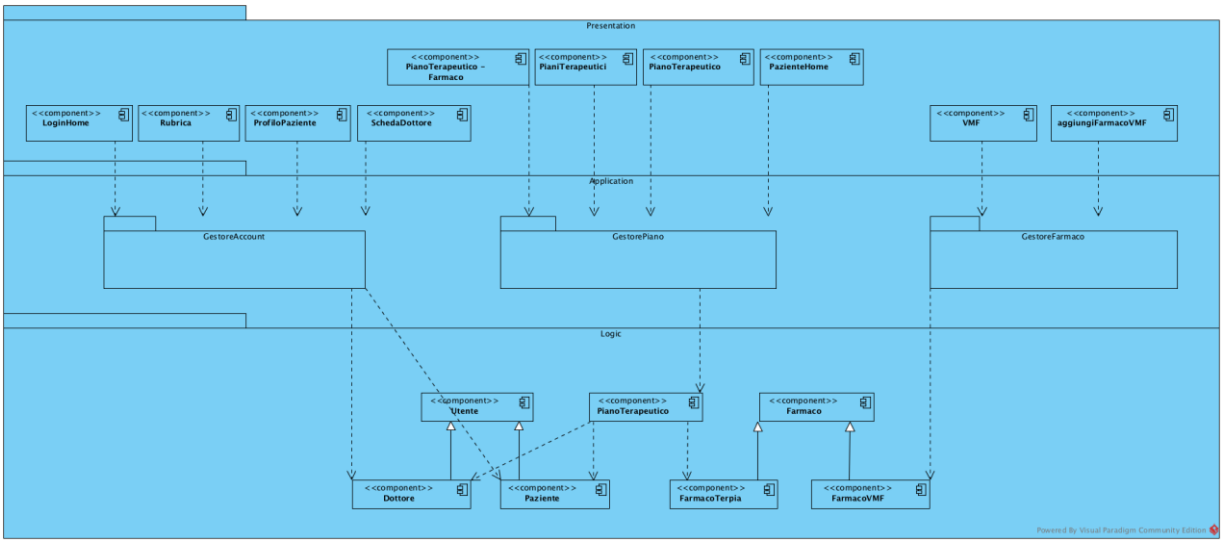
Utente non registrato



Dottore



Paziente



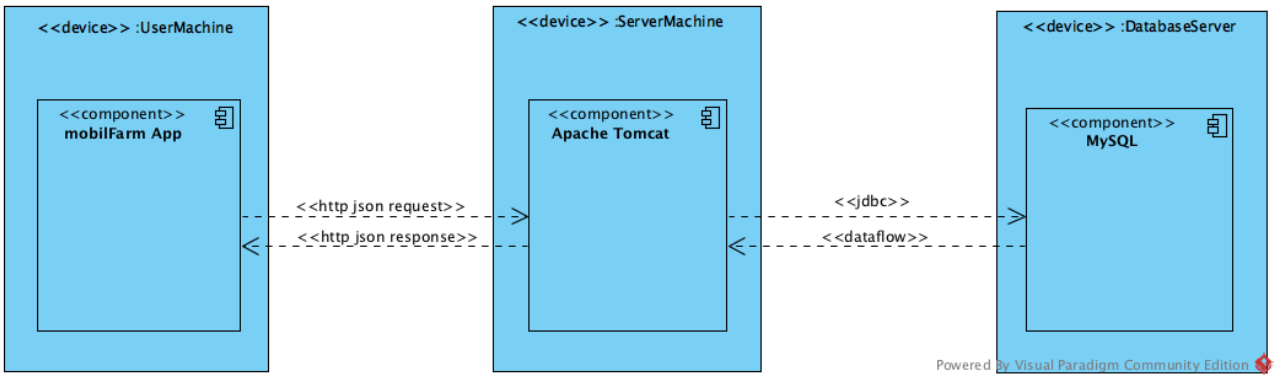
2.3 Hardware/Software Mapping

Di seguito è presente il deployment diagram che mostra il funzionamento e l'interazione dei vari componenti del sistema.

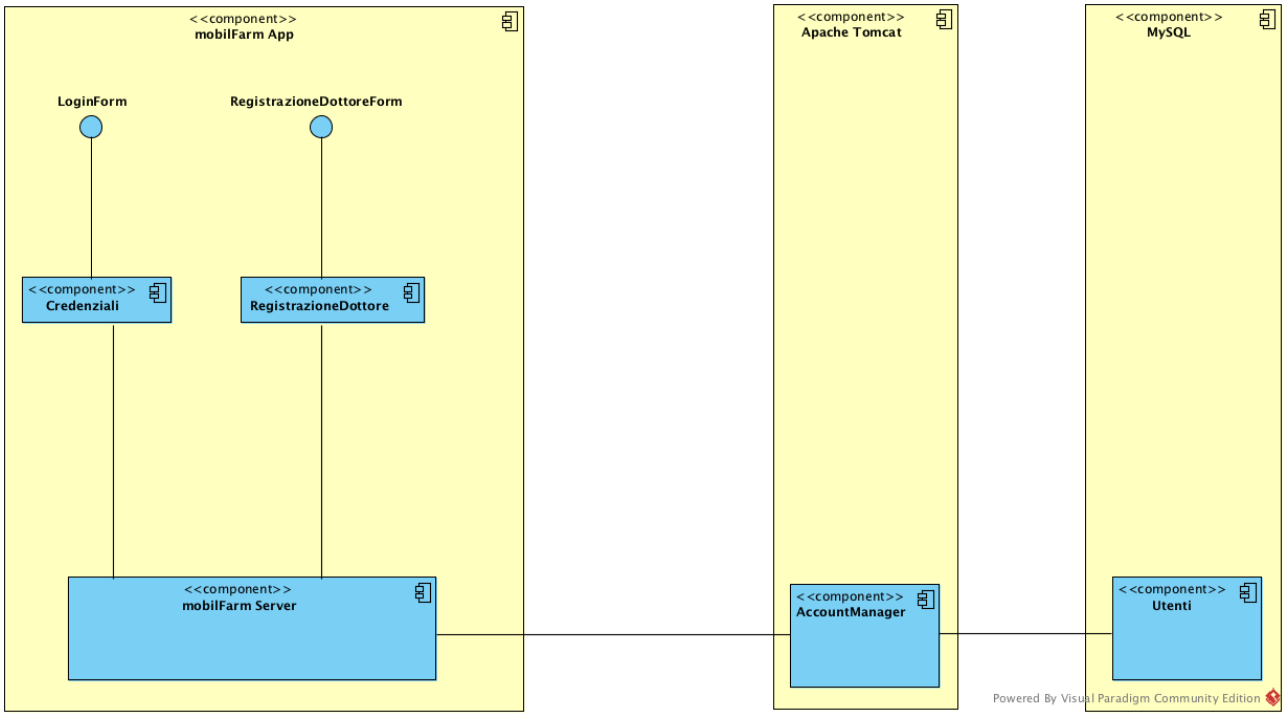
mobilFarm App: rappresenta l'app eseguibili sui device mobili.
Presenterà le informazioni all'utente grazie ad una **Json Response** ricevuta in seguito alla **Json Request** effettuata tramite **HTTP** al server Apache.

Apache: rappresenta il server attivo su AWS EC2 e si occuperà principalmente di processare le richieste ricevute dall'app e elaborare delle richieste al database MYSQL.

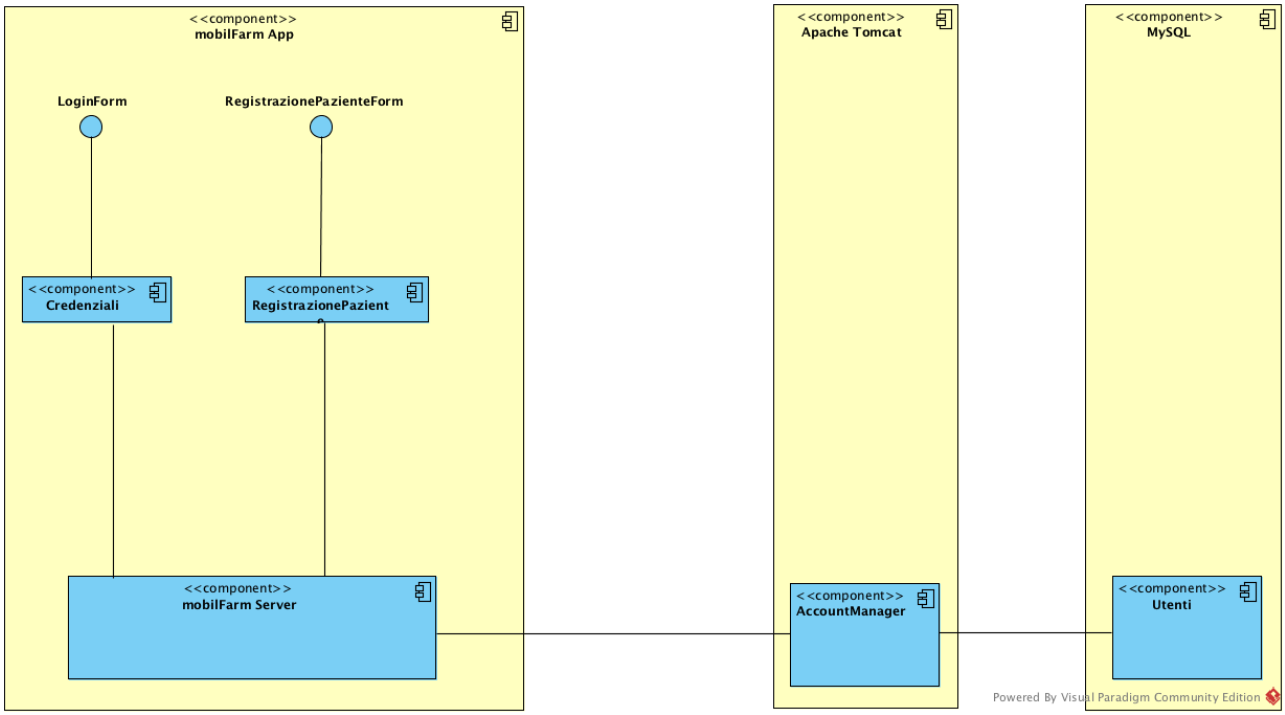
MySQL: rappresenta il database server **relazionale** che conterrà i dati di tutto il sistema e si occuperà di rispondere alle richieste ricevute da Apache fornendo, modificando, creando e cancellando record corrispondenti a determinate qualità.



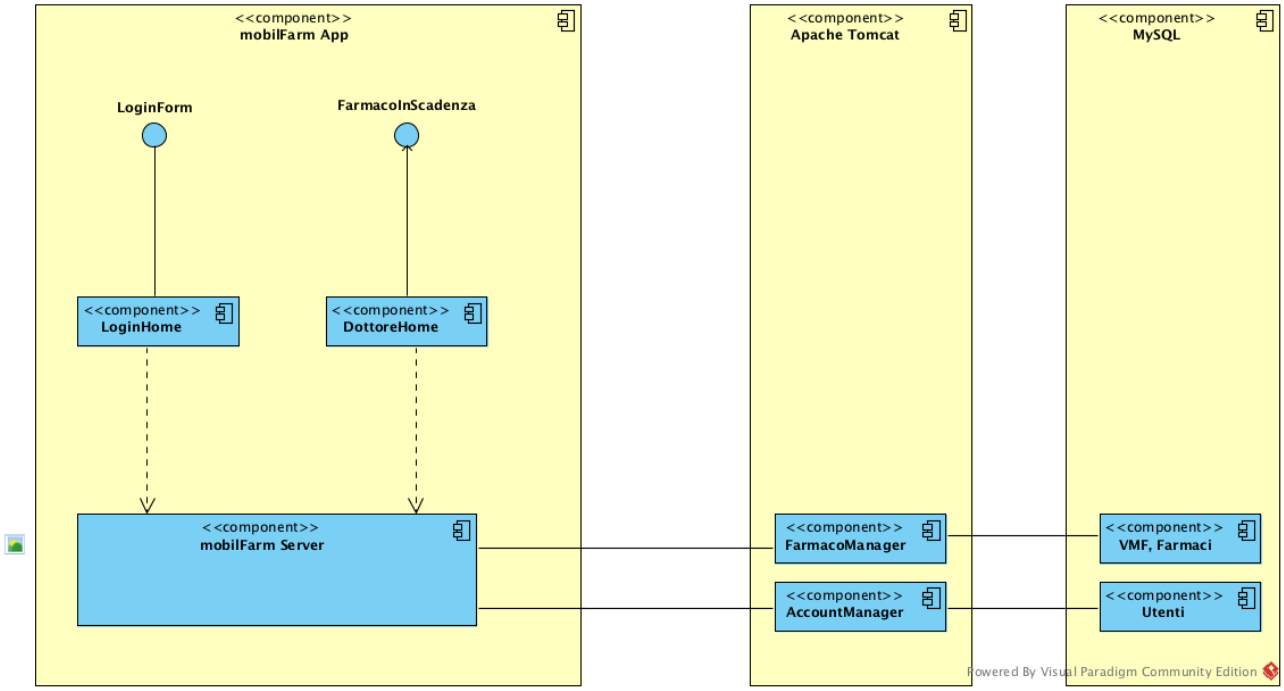
cd_0.1


<<component>>
mobilFarm App
LoginForm
RegistrazioneDottoreForm
<<component>>
Credenziali
<<component>>
RegistrazioneDottore
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
Utenti
Powered By Visual Paradigm Community Edition

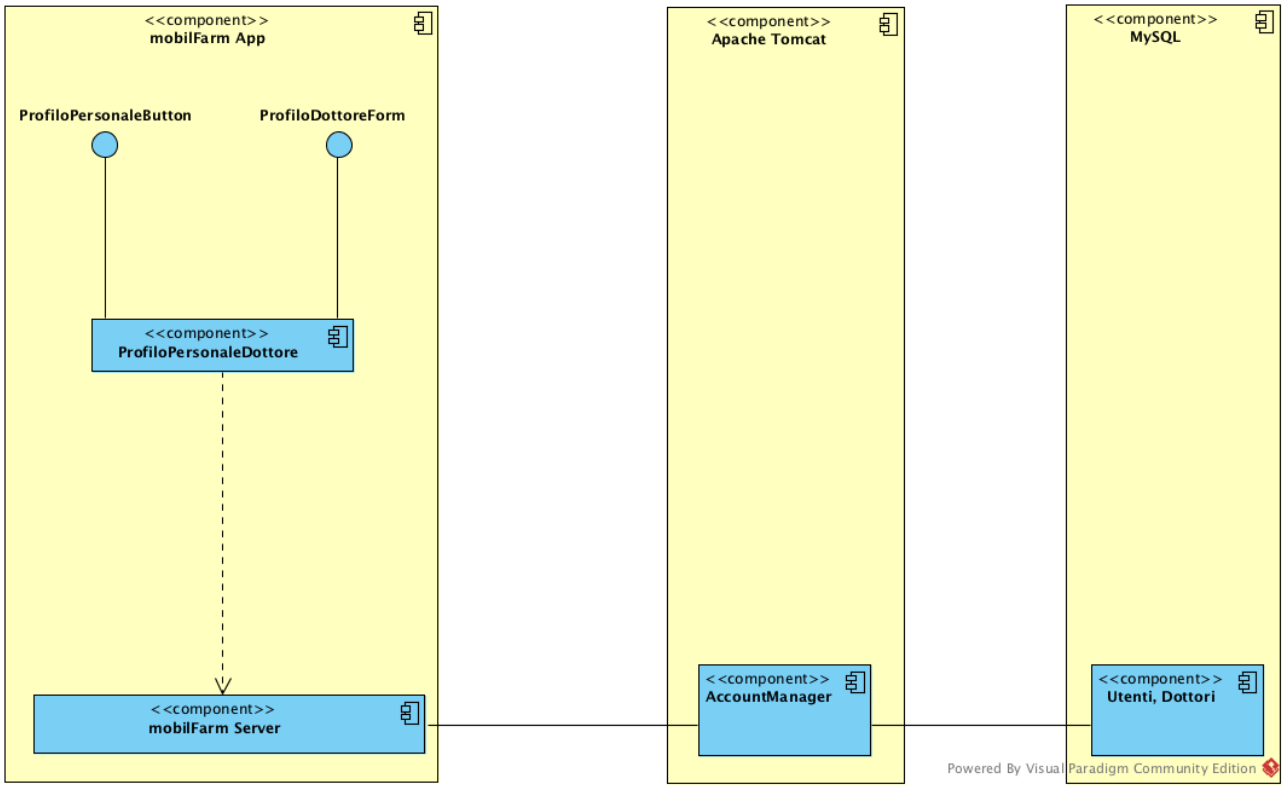
cd_0.2


<<component>>
mobilFarm App
LoginForm
RegistrazionePazienteForm
<<component>>
Credenziali
<<component>>
RegistrazionePazient
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
Utenti
Powered By Visual Paradigm Community Edition

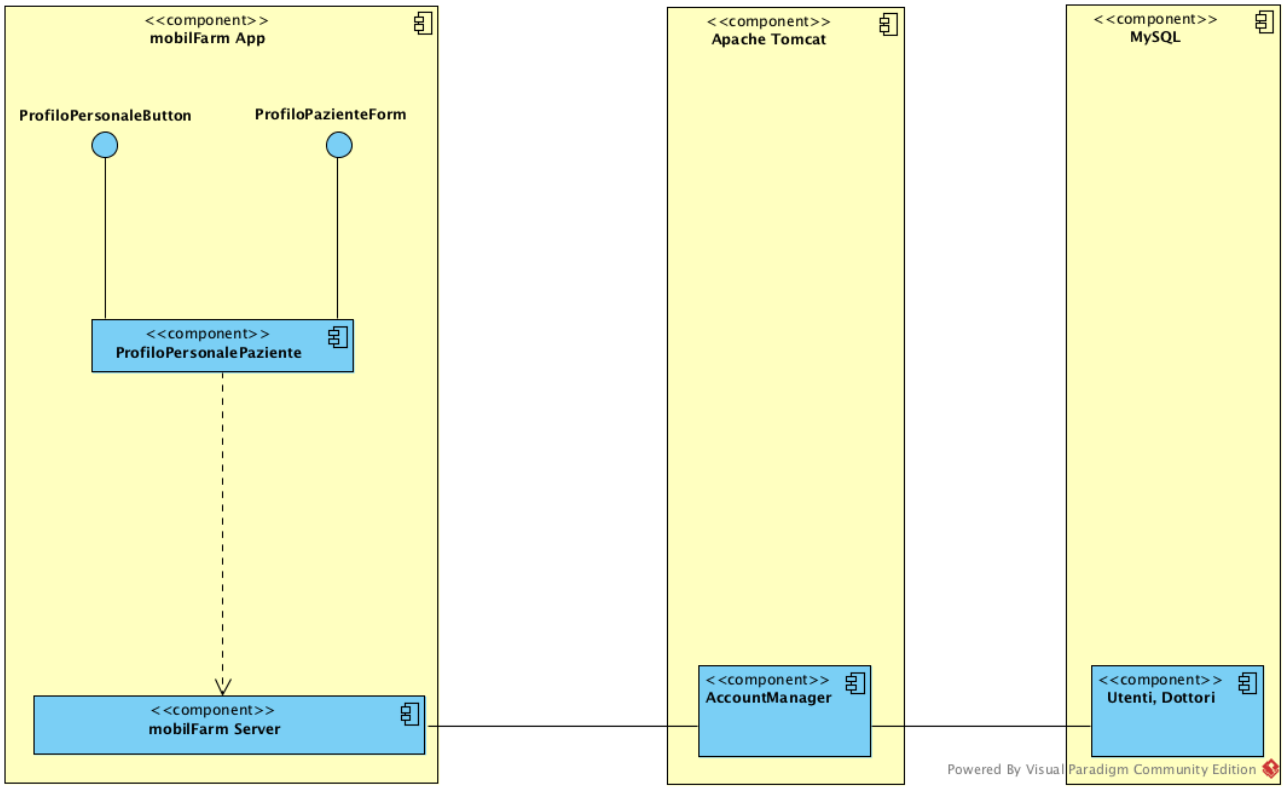
cd_0.3


<<component>>
mobilFarm App
LoginForm
FarmacoInScadenza
<<component>>
LoginHome
<<component>>
DottoreHome
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
FarmacoManager
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
VMF, Farmaci
<<component>>
Utenti
Powered By Visual Paradigm Community Edition

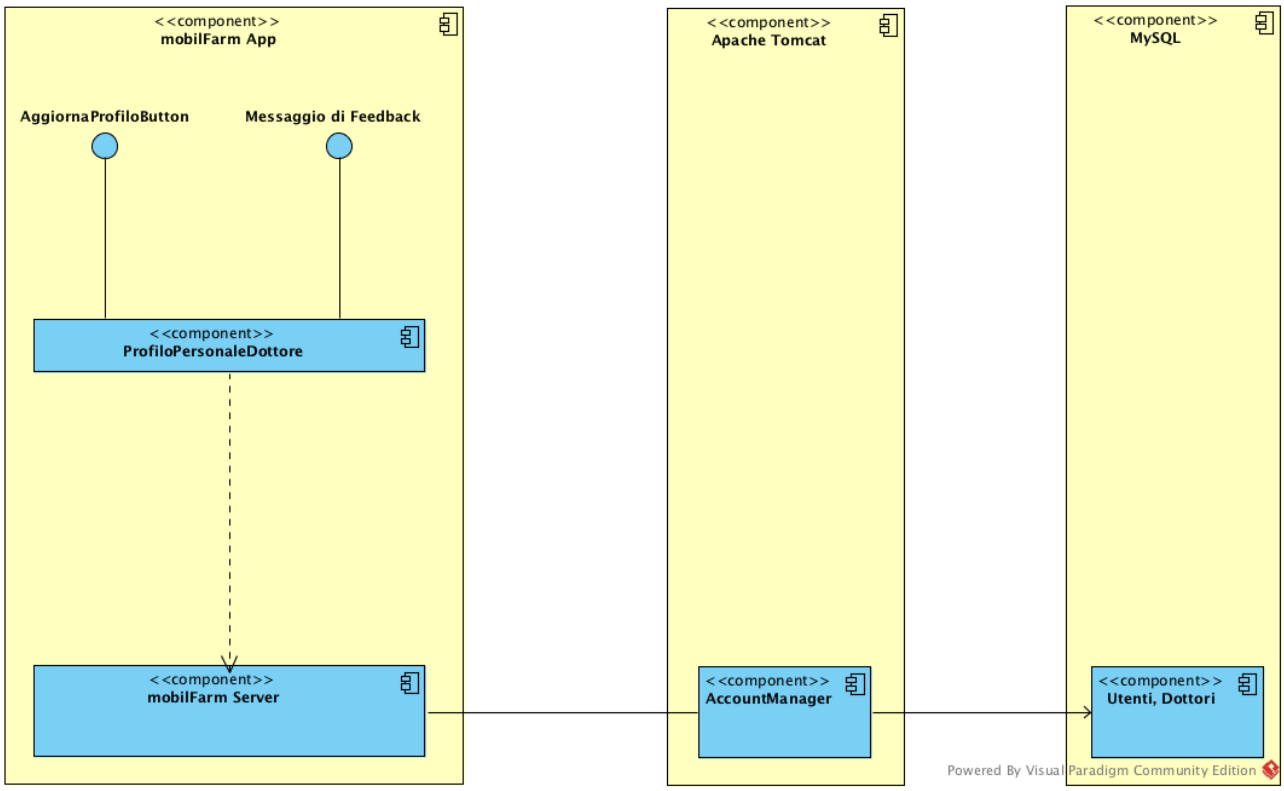
cd_0.4


<<component>>
mobilFarm App
ProfiloPersonaleButton
ProfiloDottoreForm
<<component>>
ProfiloPersonaleDottore
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
Utenti, Dottori
Powered By Visual Paradigm Community Edition

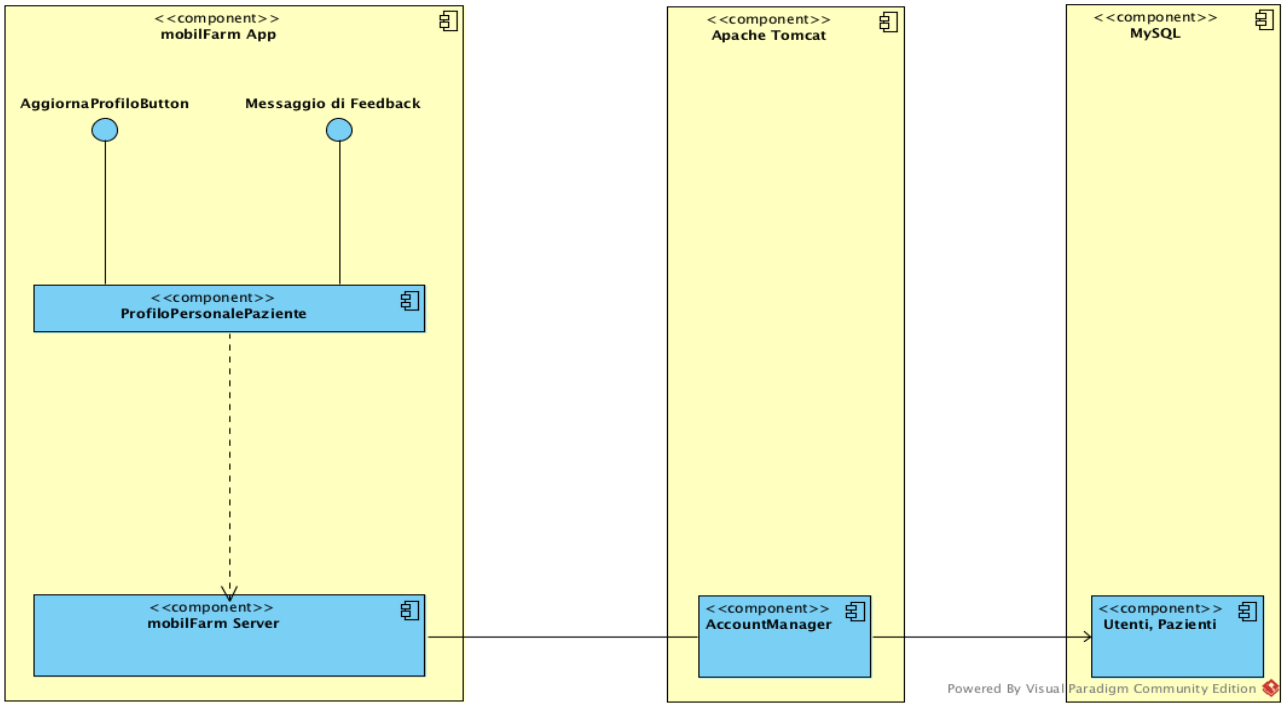
cd_0.5


<<component>>
mobilFarm App
ProfiloPersonaleButton
ProfiloPazienteForm
<<component>>
ProfiloPersonalePaziente
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
Utenti, Dottori
Powered By Visual Paradigm Community Edition

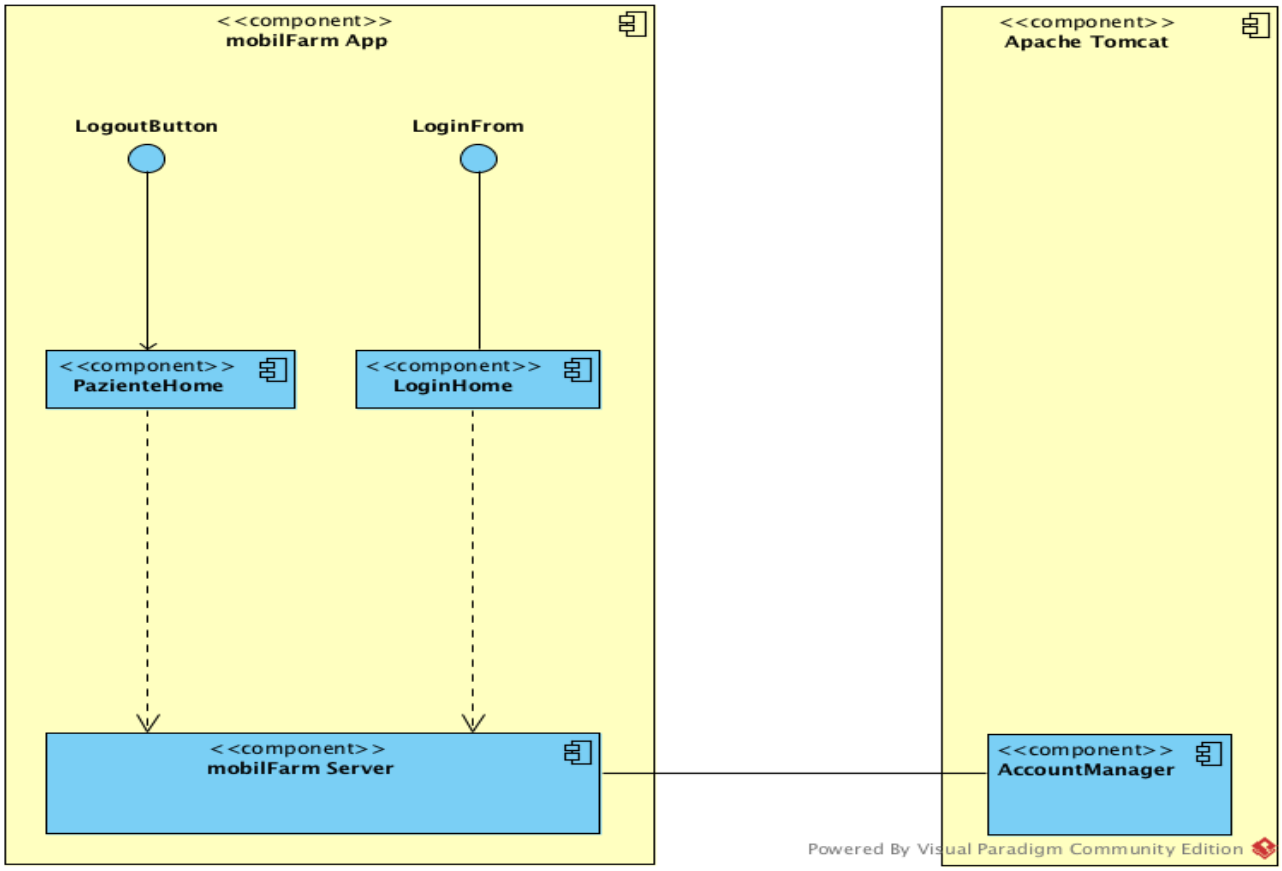
cd_0.6


<<component>>
mobilFarm App
AggiornaProfiloButton
Messaggio di Feedback
<<component>>
ProfiloPersonaleDottore
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
Utenti, Dottori
Powered By Visual Paradigm Community Edition

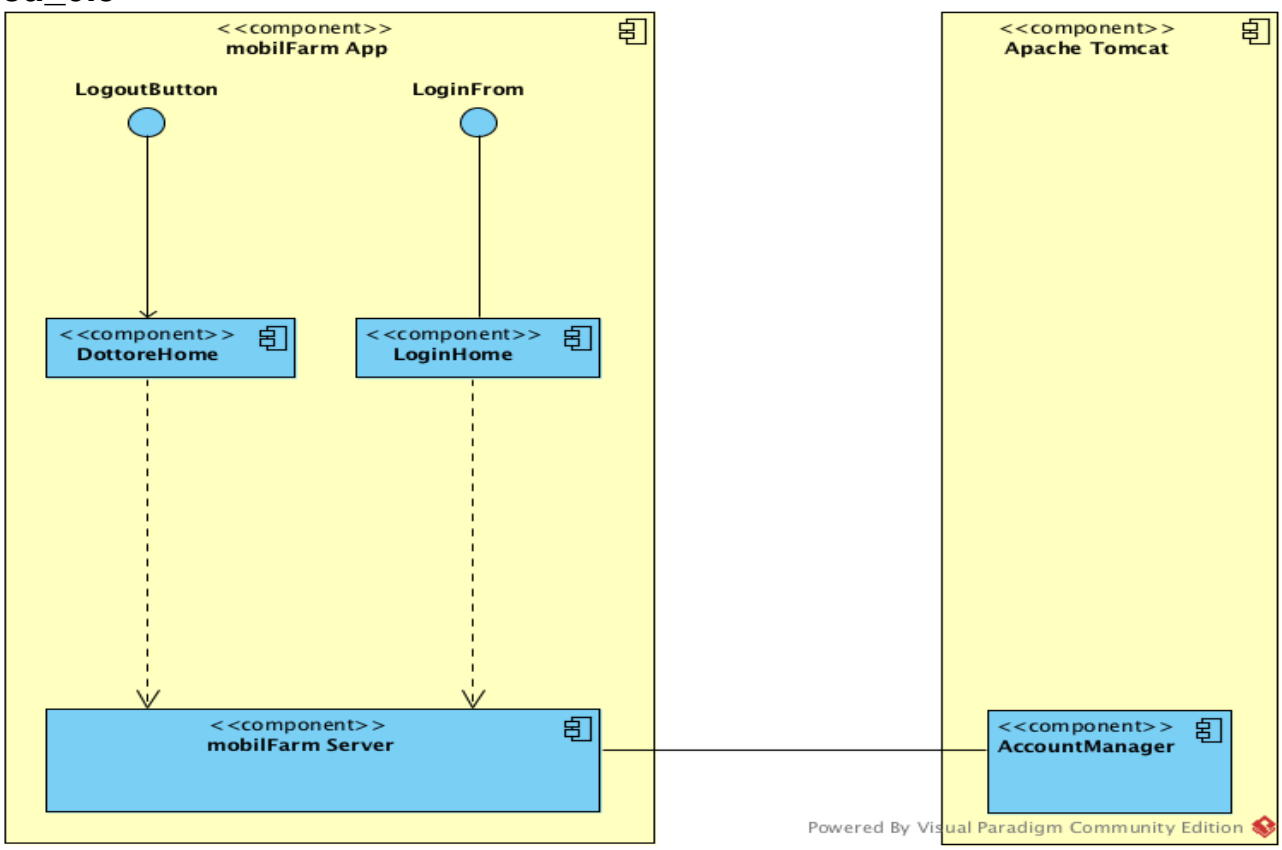
cd_0.7


<<component>>
mobilFarm App
AggiornaProfiloButton
Messaggio di Feedback
<<component>>
ProfiloPersonalePaziente
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
<<component>>
MySQL
<<component>>
Utenti, Pazienti
Powered By Visual Paradigm Community Edition

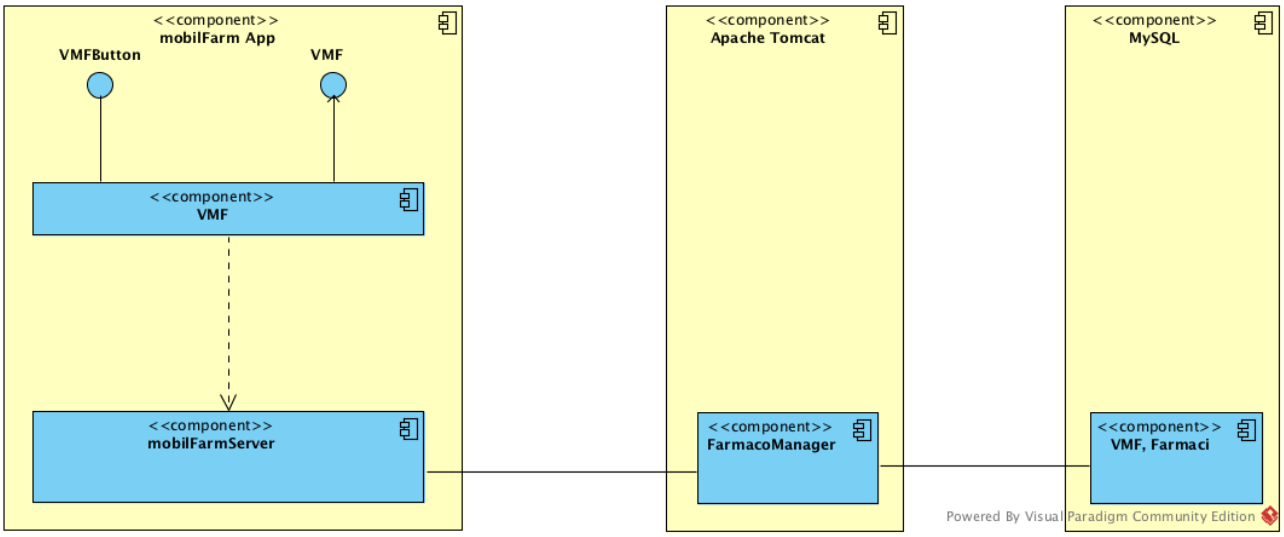
cd_0.8


<<component>>
mobilFarm App
LogoutButton
LoginFrom
<<component>>
PazienteHome
<<component>>
LoginHome
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
Powered By Visual Paradigm Community Edition

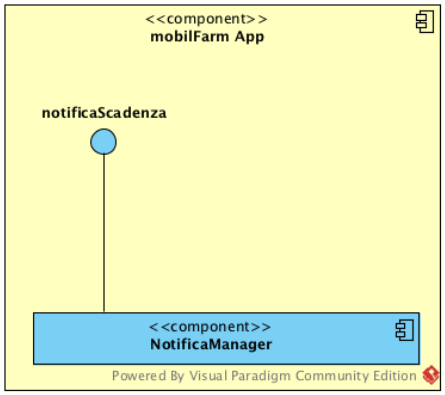
cd_0.9


<<component>>
mobilFarm App
LogoutButton
LoginFrom
<<component>>
DottoreHome
<<component>>
LoginHome
<<component>>
mobilFarm Server
<<component>>
Apache Tomcat
<<component>>
AccountManager
Powered By Visual Paradigm Community Edition

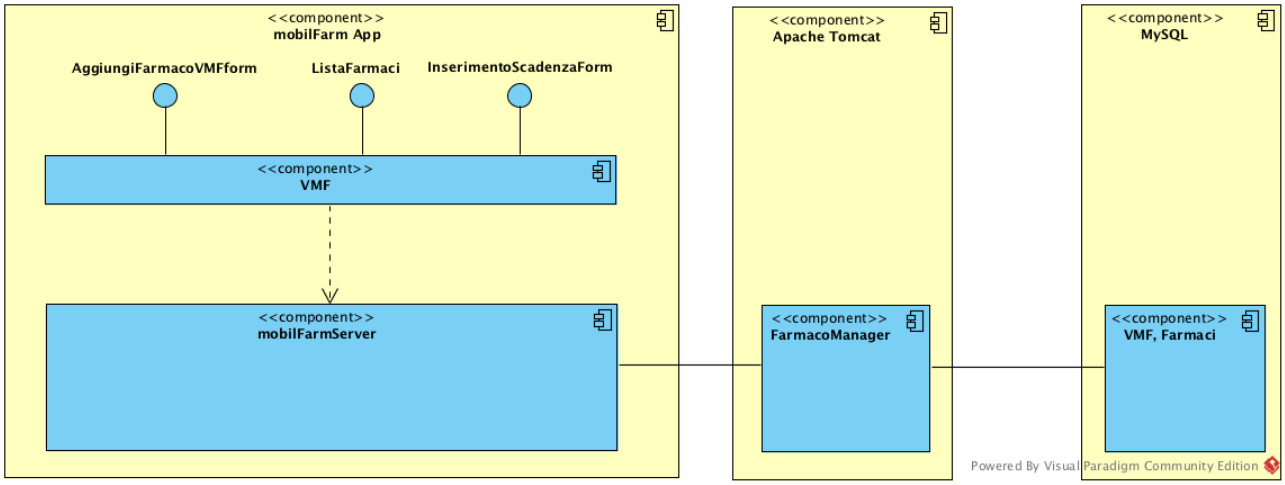
cd_1.1


<<component>>
mobilFarm App
VMFButton
VMF
<<component>>
VMF
<<component>>
mobilFarmServer
<<component>>
Apache Tomcat
<<component>>
FarmacoManager
<<component>>
MySQL
<<component>>
VMF, Farmaci
Powered By Visual Paradigm Community Edition

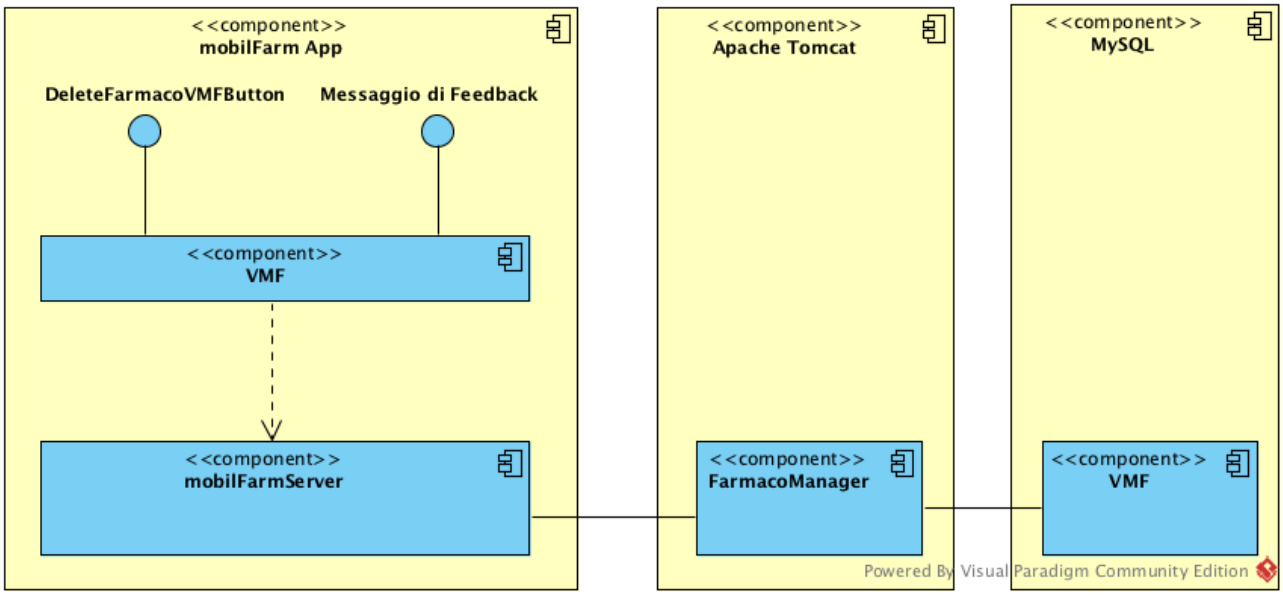
cd_1.2



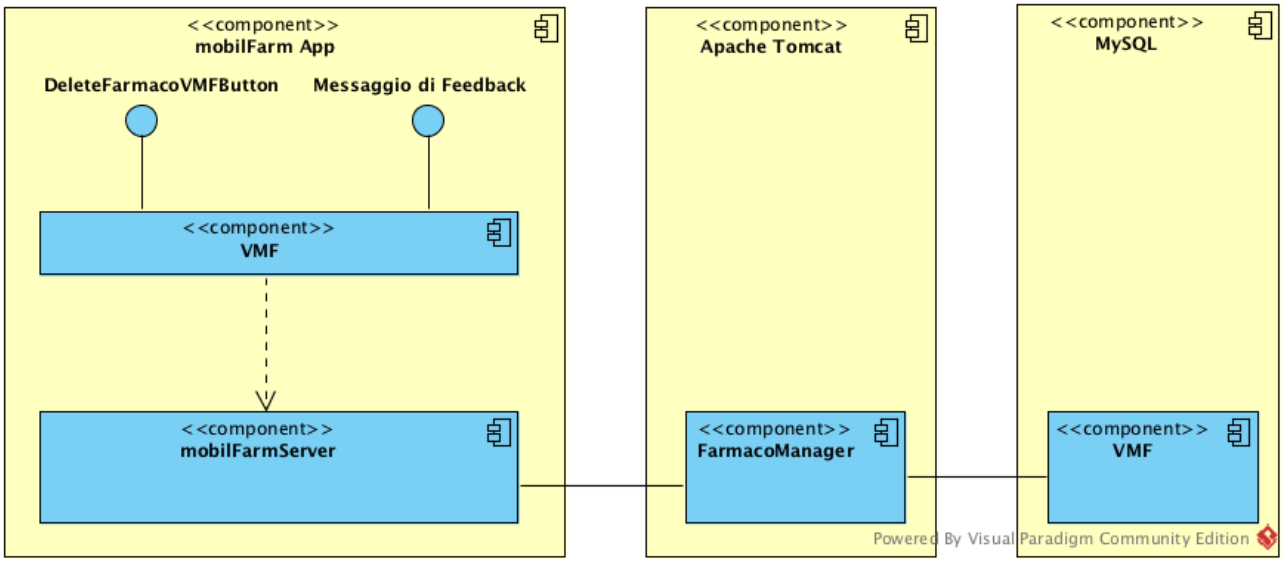
cd_1.3



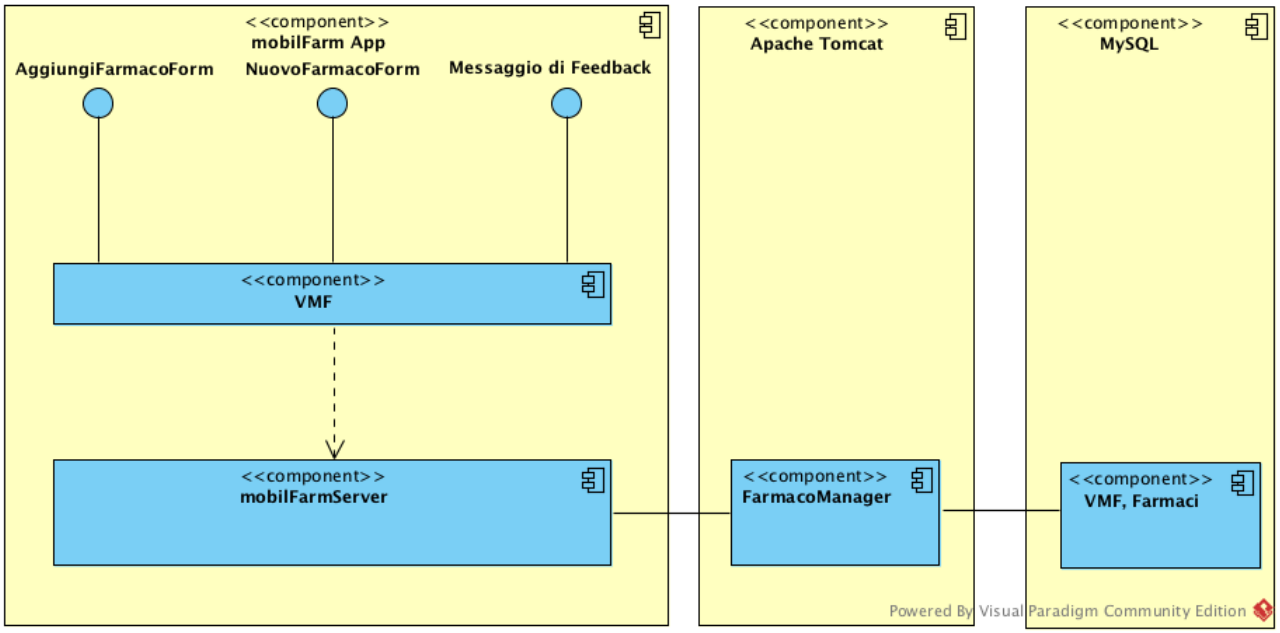
cd_1.4



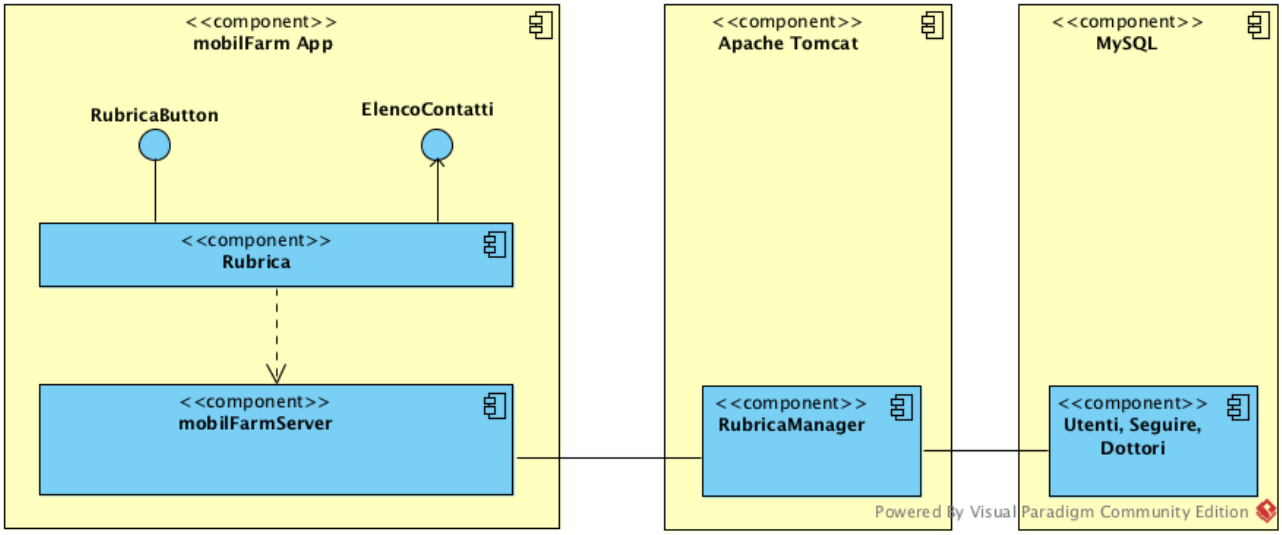
cd_1.5


<<component>>
mobilFarm App
DeleteFarmacoVMFButton
Messaggio di Feedback
<<component>>
VMF
<<component>>
mobilFarmServer
<<component>>
Apache Tomcat
<<component>>
FarmacoManager
<<component>>
MySQL
<<component>>
VMF
Powered By Visual Paradigm Community Edition

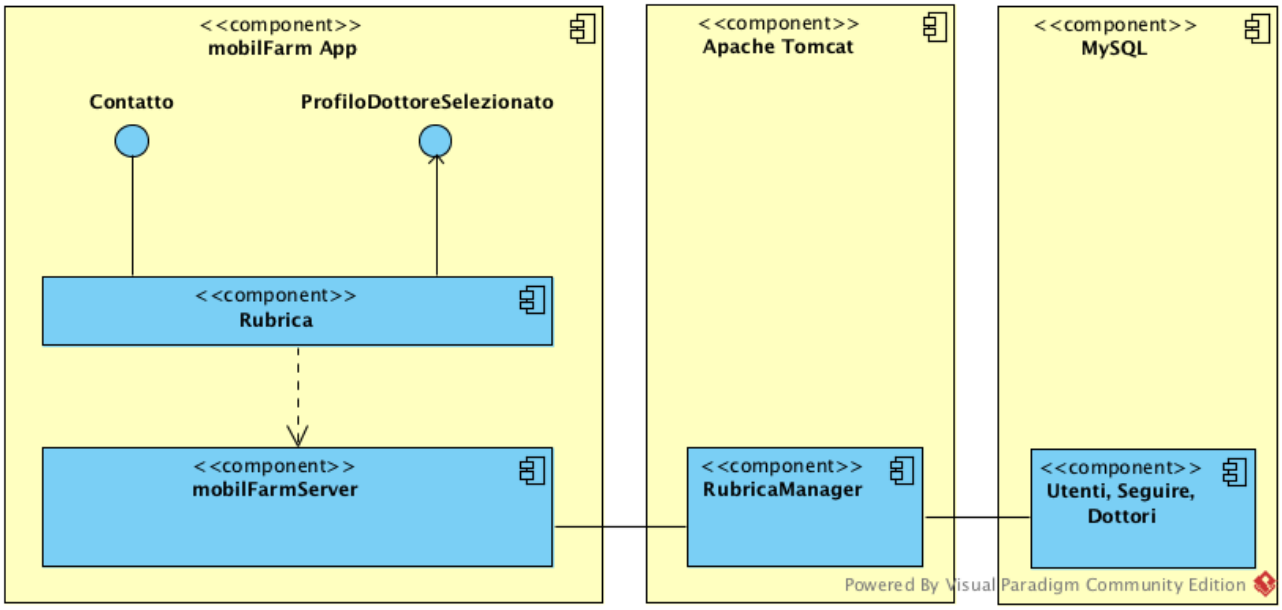
cd_1.6


<<component>>
mobilFarm App
AggiungiFarmacoForm
NuovoFarmacoForm
Messaggio di Feedback
<<component>>
VMF
<<component>>
mobilFarmServer
<<component>>
Apache Tomcat
<<component>>
FarmacoManager
<<component>>
MySQL
<<component>>
VMF, Farmaci
Powered By Visual Paradigm Community Edition

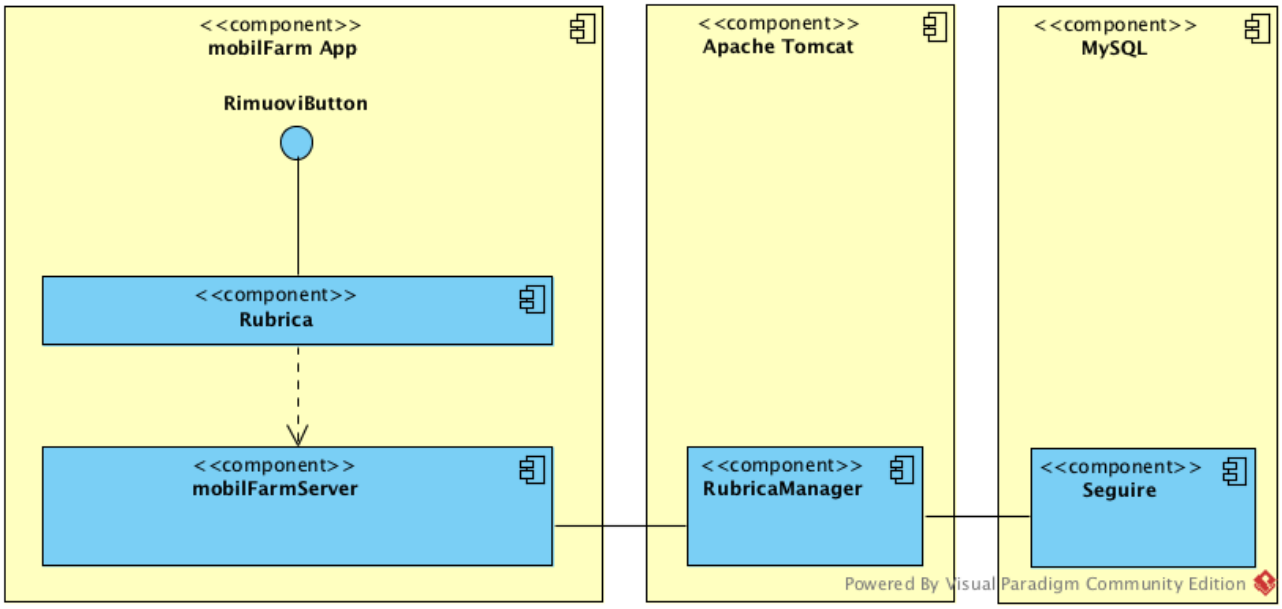
cd_2.1


<<component>>
mobilFarm App
RubricaButton
ElencoContatti
<<component>>
Rubrica
<<component>>
mobilFarmServer
<<component>>
Apache Tomcat
<<component>>
RubricaManager
<<component>>
MySQL
<<component>>
Utenti, Seguire, Dottori
Powered By Visual Paradigm Community Edition

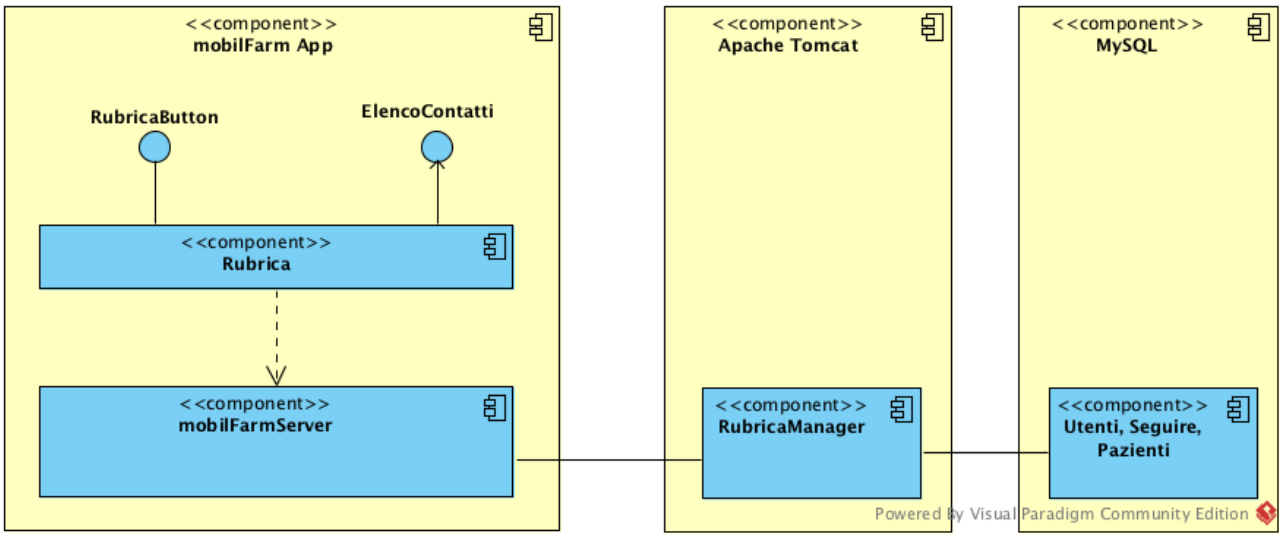
cd_2.2


<<component>>
mobilFarm App
Contatto
ProfiloDottoreSelezionato
<<component>>
Rubrica
<<component>>
mobilFarmServer
<<component>>
Apache Tomcat
<<component>>
RubricaManager
<<component>>
MySQL
<<component>>
Utenti, Seguire, Dottori
Powered By Visual Paradigm Community Edition

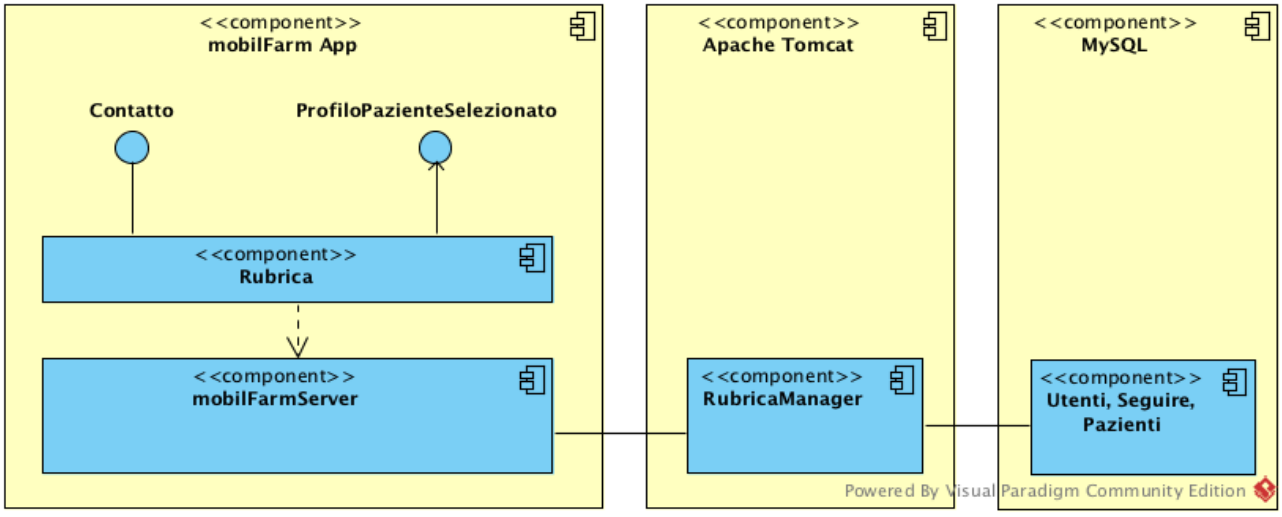
cd_2.3



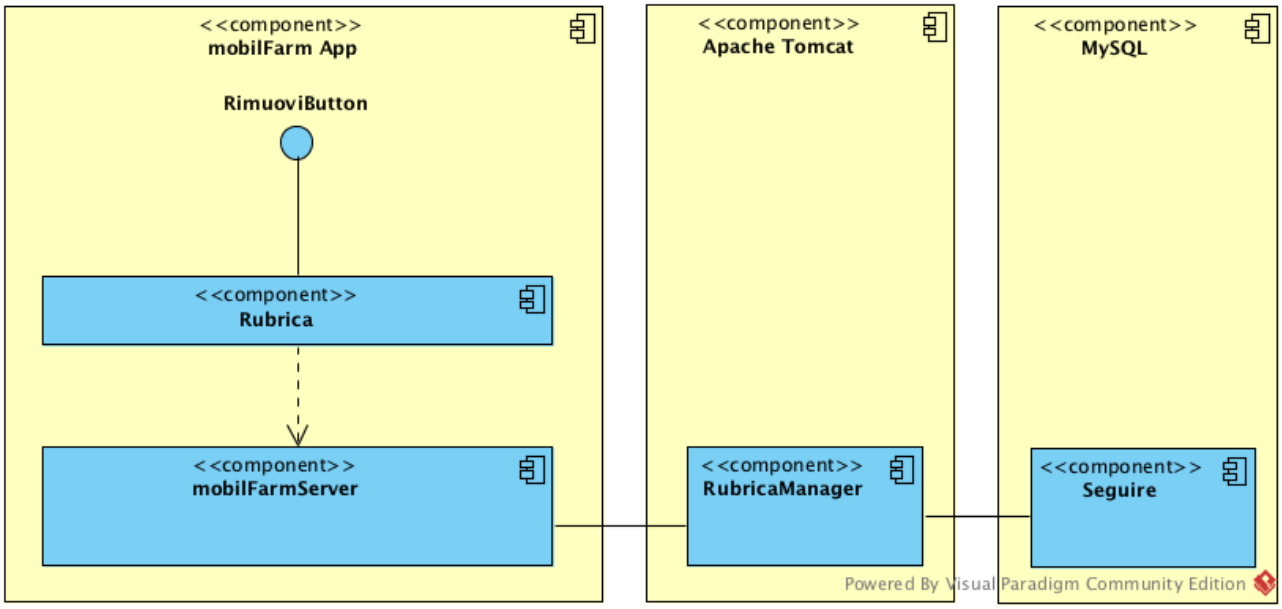
cd_2.4



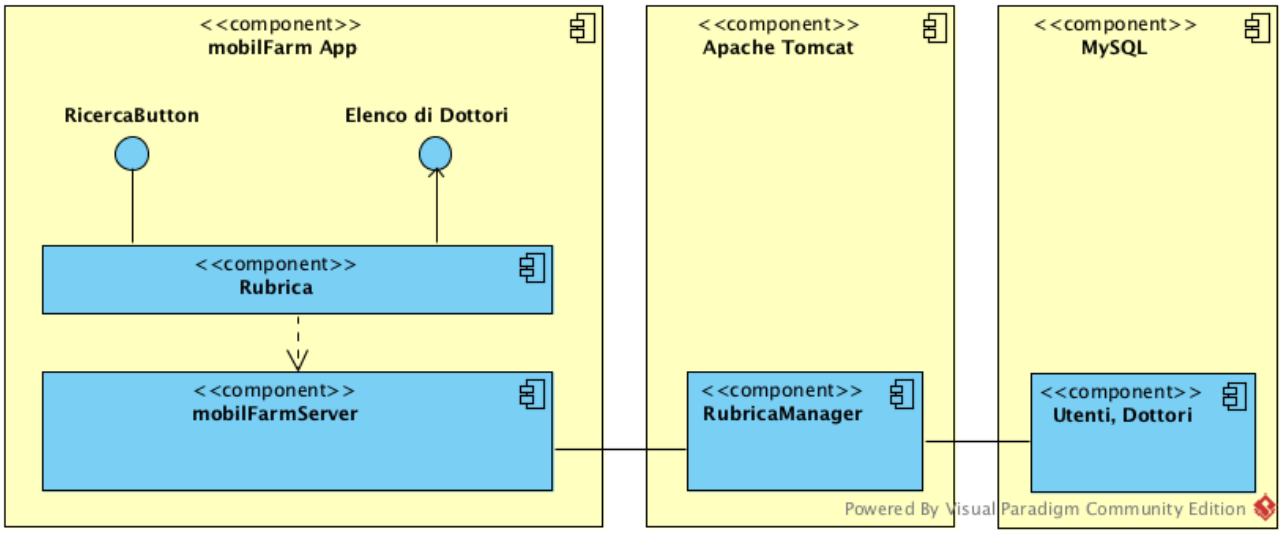
cd_2.5



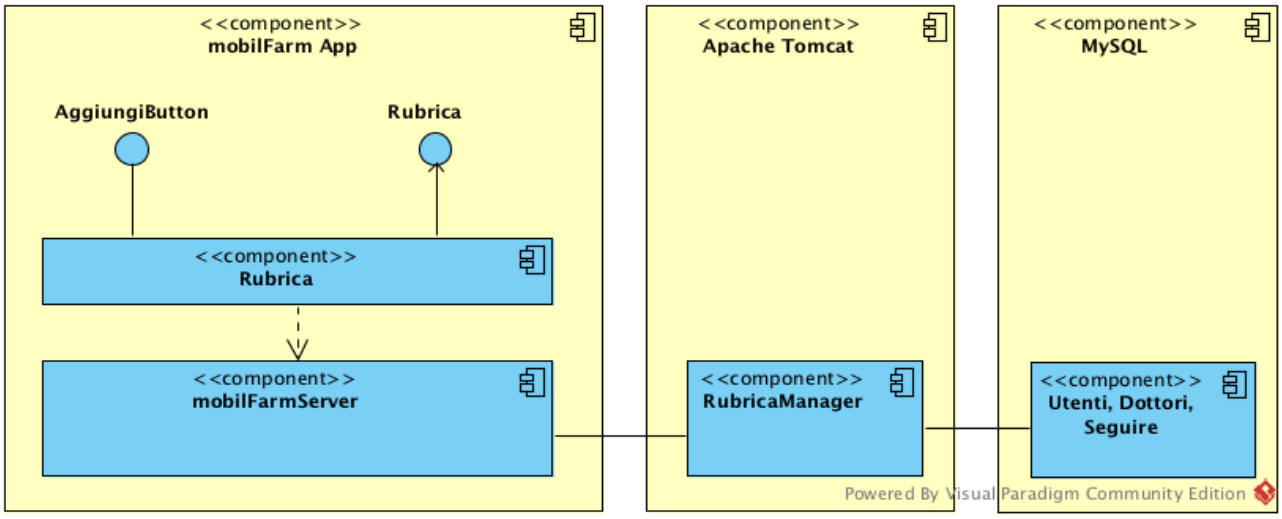
cd_2.6



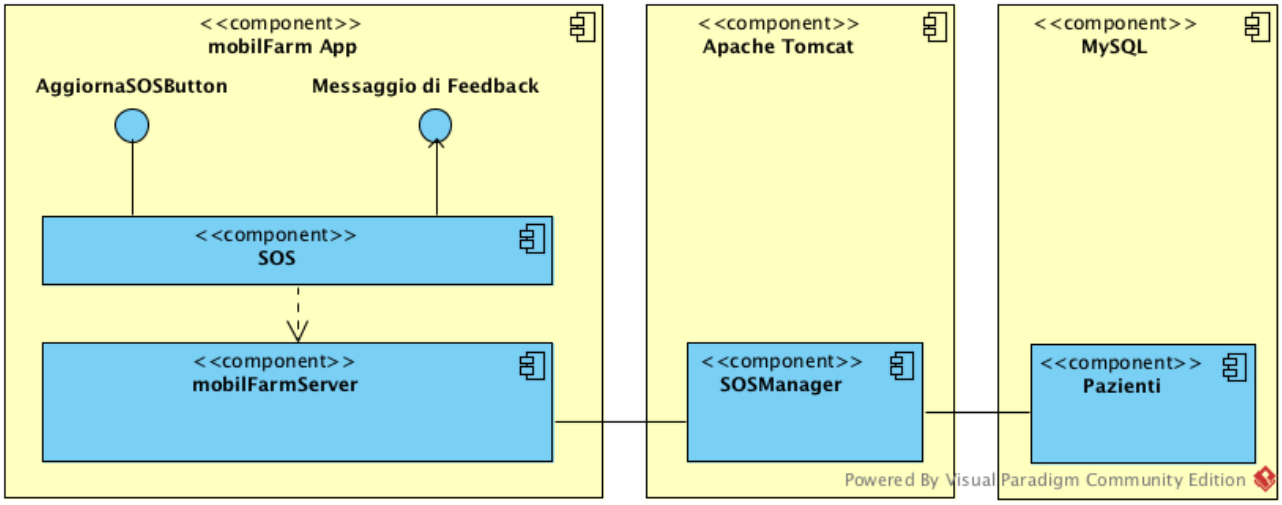
cd_2.7



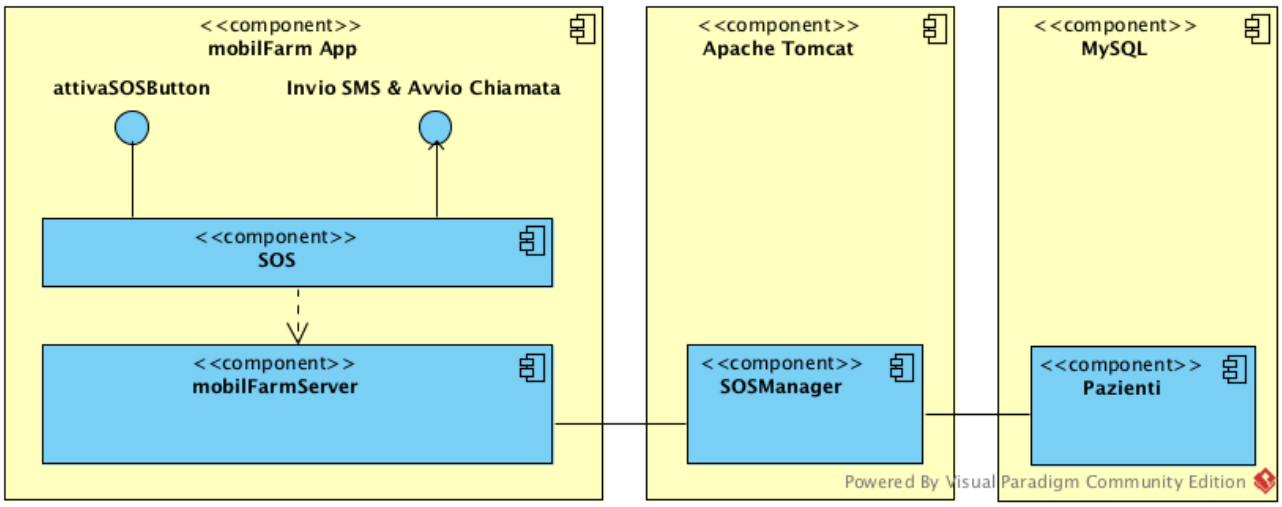
cd_2.8



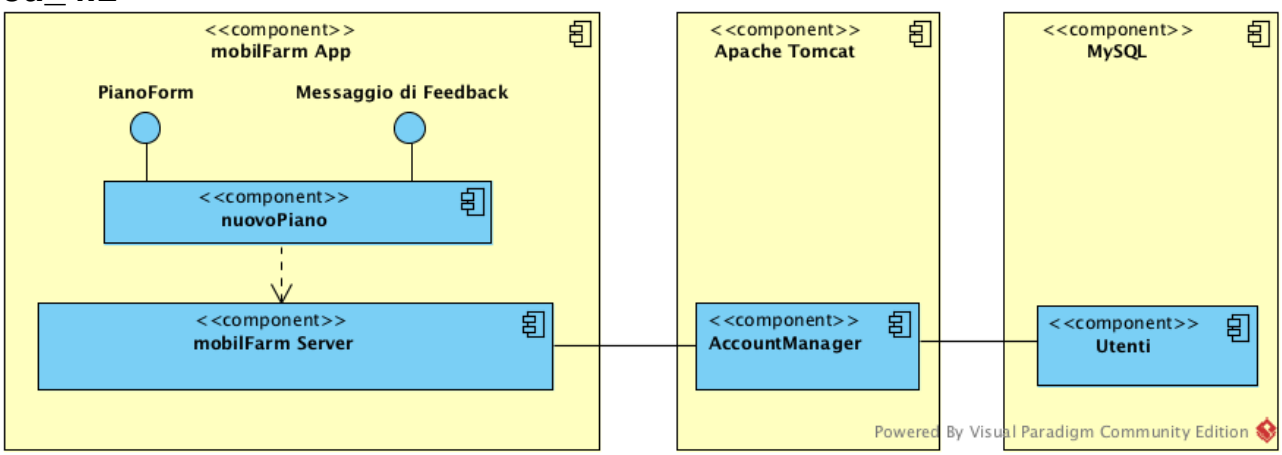
cd_3.1



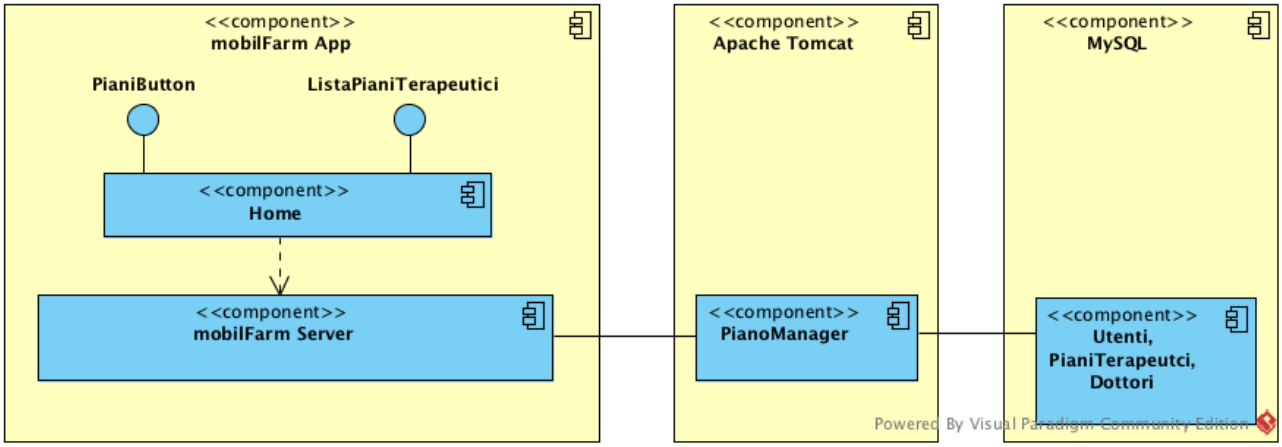
cd_3.2



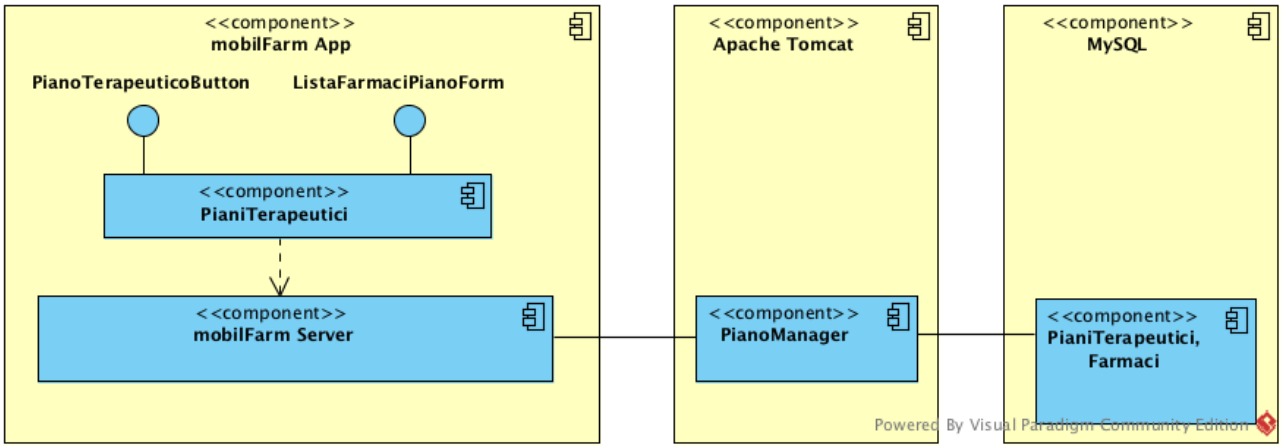
cd_4.1



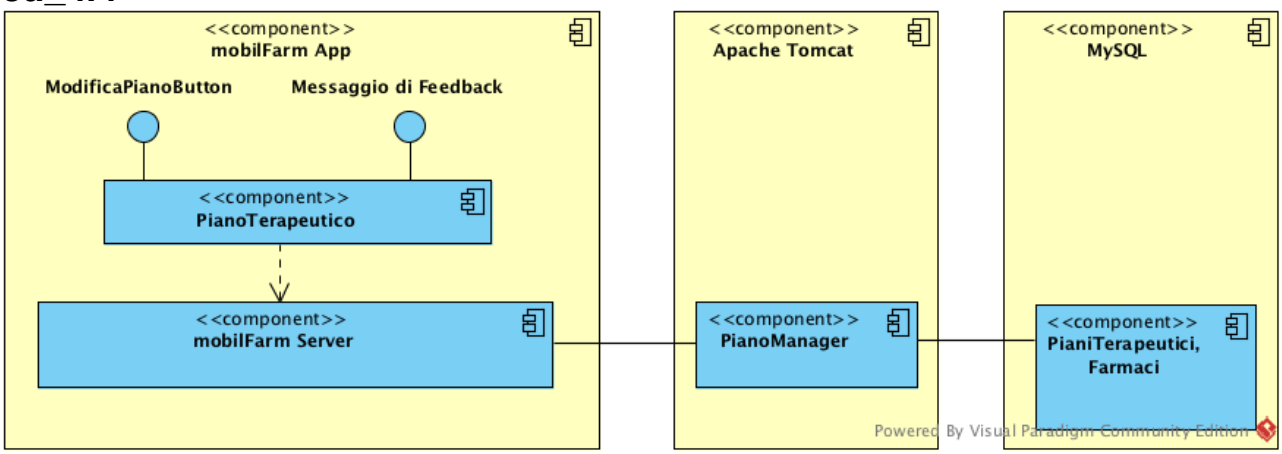
cd_4.2



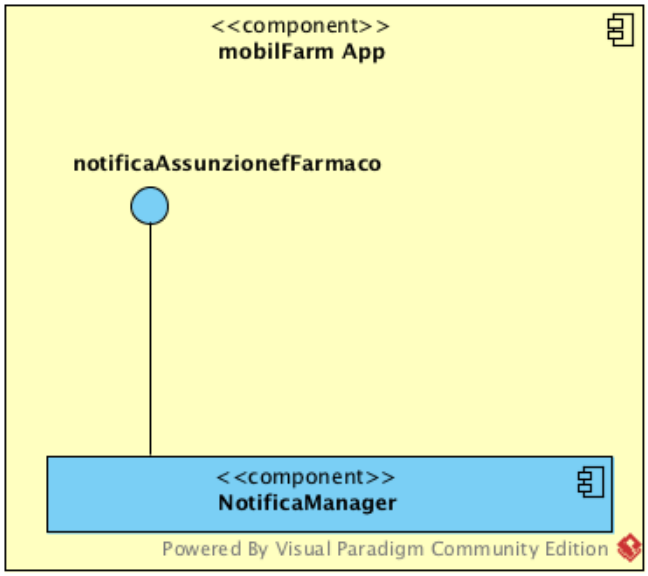
cd_4.3



cd_4.4



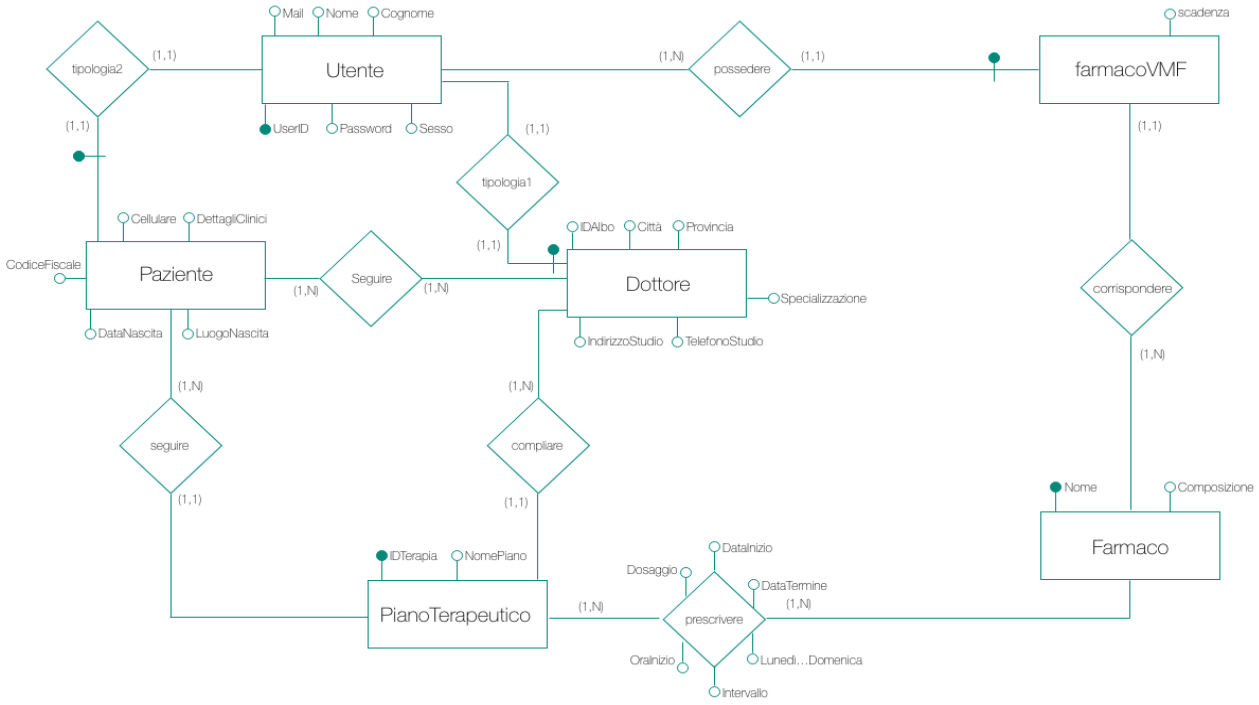
cd_4.5


<<component>>
mobilFarm App
notificaAssunzionefFarmaco
<<component>>
NotificaManager
Powered By Visual Paradigm Community Edition

2.4 Gestione dei dati persistenti

Partendo dal **Class Diagram** rappresentante il modello generale delle classi, presente RAD, abbiamo identificato i **dati persistenti** che verranno gestiti nel Database Server Relazionale MYSQL**.**

2.4.1 Modello Entità Relazioni



2.4.2 Mapping

Utenti (Mail, UserID, Nome, Cognome, Password, Sesso, Tipologia)

Pazienti (UserID, CodiceFiscale, Cellulare, DettagliClinici, DataNascita, LuogoNascita, CittaResidenza, Residenza, NumeroSOS1*, NumeroSOS2*)

Dottori (UserID, IDAlbo, Citta, Provincia, Specializzazione, IndirizzoStudio, TelefonoStudio)

PianiTerapeutici (IDTerapia, NomePiano, CompilatoDa, InviatoA)

VMF (UserID, IDFarmaco, Scadenza)

Farmaci (IDFarmaco, Nome, Composizione*)

Prescrizioni (IDTerapia, IDFarmaco, Dosaggio, DataInizio, DataTermine, OraInizio, Intervallo, Lunedì…Domenica)

Seguire (PazienteID, DottoreID)

Utenti

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
Mail	varchar (50)	✓	✗	Indirizzo mail corrispondente alla mail personale dell'utente	✗
UserID	varchar (32)	✗	✗	Indirizzo mail criptato con algoritmo md5 che rappresenta l'identificativo dell'utente	✗
Nome	varchar (20)	✗	✗	Nome dell'utente	✗
Cognome	varchar (20)	✗	✗	Cognome dell'utente	✗
Password	varchar (32)	✗	✗	Stringa alfanumerica scelta dall'utente con al più 15 caratteri Criptata in md5	✗
Sesso	char (1)	✗	✗	Singolo carattere a scelta tra F e M per indicare il sesso dell'utente	✗
Tipologia	char (8)	✗	✗	Singolo carattere a scelta tra P e D per indicare rispettivamente paziente o dottore	✗

Pazienti

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
UserID	**varchar (32)**	✓	✓	**Indirizzo mail criptato con algoritmo md5 che rappresenta l'identificativo dell'utente**	✗
CodiceFiscale	**varchar (16)**	✗	✗	**Stringa di 16 caratteri corrispondente al codicefiscale dell'utente**	✗
Cellulare	**varchar (12)**	✗	✗	**Stringa composta da +39 e 10 cifre corrispondenti al numero di cellulare personale dell'utente**	✗
DettagliClinici	**varchar (50)**	✗	✗	**Area di testo a disposizione dell'utente dove elencare le proprie patologie, intolleranze e/o eventuali allergie**	✗
LuogoNascita	**varchar (20)**	✗	✗	**Stringa di caratteri che identifica il luogo di nascita dell'utente**	✗
DataNascita	**varchar(10)**	✗	✗	**Valore espresso nel formato YYYY/MM/DD**	✗
CittaResidenza	**varchar(50)**	✗	✗	**Stringa di caratteri che indentifica la città di** residenza dell'utente	✗
Residenza	**varchar(50)**	✗	✗	**Stringa di caratteri che indentifica la via di residenza** dell'utente	✗
NumeroSOS1	**varchar (13)**	✗	✗	**Stringa composta da +39 e 10 cifre corrispondenti al numero di cellulare di un familiare indicato dall'utente**	✓
NumeroSOS2	**varchar (13)**	✗	✗	**Stringa composta da +39 e 10 cifre corrispondenti al numero di cellulare di un familiare indicato dall'utente**	✓

PianiTerapeutici

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
IDTerapia	**int**	✓	✗	**Numero di identificazione univoco che rappresenta la terapia**	✗
NomePiano	**varchar (50)**	✗	✗	**Stringa corrispondente al nome della terapia**	✗
CompilatoDa	**varchar (32)**	✗	✓	**Id di riferimento del dottore**	✗
InviatoA	**varchar (32)**	✗	✓	**Id di riferimento del paziente**	✗

Dottori

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
UserID	**varchar (32)**	✓	✓	**Indirizzo mail criptato con algoritmo md5 che rappresenta l'identificativo dell'utente**	✗
IDAlbo	**varchar (8)**	✗	✗	**Stringa alfanumerica di identificazione del numero di iscrizione all'albo**	✗
Città	**varchar (50)**	✗	✗	**Stringa di identificazione della Città di ubicazione dell'ambulatorio**	✗
Provincia	**char (2)**	✗	✗	**Sigla della provincia di ubicazione dell'ambulatorio**	✗
Specializzazione	**varchar (32)**	✗	✗	**Caratteri preimpostati che rappresentano la specializzazione del dottore**	✗
IndirizzoStudio	**varchar (50)**	✗	✗	**Indirizzo di ubicazione dell'ambulatorio**	✗
TelefonoStudio	**varchar (12)**	✗	✗	**Stringa composta da +39 e da 10 cifre che rappresentano il numero di telefono del dottore**	✗

Farmaci

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
Nome	**varchar (20)**	✗	✗	**Nome completo del farmaco**	✗
IDFarmaco	**int**	✓	✗	**Numero di identificazione univoco del farmaco**	✗
Composizione	**int**	✗	✗	Valore intero che rappresenta i gr di composizione del farmaco	✓

Seguire

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
PazienteID	**varchar (32)**	✓	✓	**Id di riferimento del paziente**	✗
DottoreID	**varchar (32)**	✓	✓	**Id di riferimento del dottore**	✗

Prescrizioni

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
IDTerapia	**int**	✓	✓	**Numero di identificazione univoco che rappresenta la terapia**	✗
IDFarmaco	**int**	✓	✓	**Numero di identificazione univoco del farmaco**	✗
Dosaggio	**varchar(10)**	✗	✗	**Quantità in dosi del farmaco da assumere**	✓
DataInizio	**varchar(10)**	✗	✗	Valore espresso in YYYY/MM/DD	✓
DataTermine	**varchar(10)**	✗	✗	Valore espresso in YYYY/MM/DD	✓
OraInizio	**varchar(5)**	✗	✗	Valore espresso in HH:MM	✓
Intervallo	**int**	✗	✗	Valore intero che rappresenta la quantità di ore tra un'assunzione e l'altra.	✓
Lunedi	**boolean**	✗	✗	**Se vero il**	✓

				giorno Lunedì è selezionato	
Martedi	**boolean**	✗	✗	**Se vero il giorno Martedì è selezionato**	✓
Mercoledi	**boolean**	✗	✗	**Se vero il giorno Mercoledì è selezionato**	✓
Giovedi	**boolean**	✗	✗	**Se vero il giorno Giovedì è selezionato**	✓
Venerdi	**boolean**	✗	✗	**Se vero il giorno Venerdì è selezionato**	✓
Sabato	**boolean**	✗	✗	**Se vero il giorno Sabato è selezionato**	✓
Domenica	**boolean**	✗	✗	**Se vero il giorno Domenica è selezionato**	✓

VMF

Nome Attributo	Tipo	Chiave Primaria	Chiave Esterna	Descrizione	Opzionale
UserID	**varchar (32)**	✓	✓	**Indirizzo mail criptato con algoritmo md5 che rappresenta l'identificativo dell'utente**	✗
IDFarmaco	**int**	✓	✓	**Numero di identificazione univoco del farmaco**	✗
Scadenza	**varchar(10)**	✓	✗	Valore espresso in YYYY/MM/DD	✗

2.5 Sicurezza e controllo degli accessi

mobilFarm è un sistema che permette l'accesso alle informazioni a diversi tipi di utenti realizzeremo **tre tipi di politiche di accesso** a seconda della tipologia di utente (dottore, paziente, utente non identificato).
Gli utenti che hanno già un account (**dottore e paziente**) potranno identificarsi attraverso la schermata di login e inserendo la propria mail e la relativa password, successivamente avranno accesso alle proprie informazioni e potranno interagire con il sistema in base alla matrice degli accessi sottostante.
Gli utenti **non identificati**, invece, non potranno visualizzare alcun dato e non potranno interagire con le varie parti del sistema tranne quelle relative alla registrazione e al login (così come descritto nella matrice degli accessi seguente).

AM_GAT_1 - GestoreAccount

Oggetto / Attore	GestioneAccount	GestioneRubrica
Dottore	**profiloPersonaleDottore()** **aggiornaProfiloDottore()** **logout()**	**visualizzaRubricaPazienti()** **visualizzaProfiloPazienteSelezionato()** **rimuoviPaziente()**
Paziente	**profiloPersonalePaziente()** **aggiornaProfiloPaziente()** **impostaSOS()** **modalitaSOS()** **logout()**	**visualizzaRubricaDottori()** **visualizzaProfiloDottoreSelezionato()** **rimuoviDottore()** **ricercaDottore()** **aggiungiDottore()**
Utente non identificato	**registraAccountDottore()** **registraAccountPaziente()** **controlloCredenziali()** **login()**	**-**

AM_GAT_2 - GestoreFarmaco

Attore \ Oggetto	GestioneFarmaco	GestioneVMF
Dottore	creaFarmaco() ricercaFarmaco()	visualizzaVMF() aggiungiFarmacoVMF() eliminaFarmacoVMF() elencoFarmaciScaduti() elencoFarmaciInScadenza()
Paziente	ricercaFarmaco()	visualizzaVMF() elencoFarmaciInScadenza() elencoFarmaciScaduti() aggiungiFarmacoVMF() eliminaFarmacoVMF()
Utente non identificato	-	-

AM_GAT_3 – GestorePiano

Attore \ Oggetto	GestionePianoTerapeutico
Dottore	creaPianoTerapeutico() aggiungiFarmaco() rimuoviFarmaco() selezionaPaziente() aggiornaDettagliAssunzioneFarmaco() visualizzaDettagliAssunzioneFarmaco() visualizzaPianoTerapeutico() visualizzaPianiTerapeutici()
Paziente	visualizzaPianoTerapeutico() elencoAssunzioni() visualizzaPianiTerapeutici() visualizzaDettagliAssunzioneFarmaco()
Utente non identificato	-

2.6 Controllo globale del software

mobilFarm prevede un controllo del flusso **esplicito centralizzato** di tipo **Event-Driven**.
I dispatcher saranno le classi che rappresentano il cuore di ogni sottosistema lato front-end: GestioneAccount, GestionePianoTerapeutico, GestioneVMF.
Queste dopo aver verificato la coerenza della richiesta, svolgerà il metodo e effettuerà una chiamata alla classe chiamante per indicare la riuscita o meno dell'operazione.
Inoltre, per evitare la ricezione di richieste costruite ad-hoc al server, le classi ManagerAccount, ManagerPianoTerapeutico e ManagerVMF effettueranno lo stesso tipo di controllo svolto dalle classi gemelle.

2.6.1 Sequenza delle richieste

Le richieste si attivano quando l'utente preme sul boundary presente nella UI dell'applicazione mobile.
La sequenza è la seguente:
Boundary -> Classe dell'Activity -> Classe Gestione -> Classe Manager -> Query DB

2.6.2 Sincronizzazione e problemi di concorrenza

Il problema della sincronizzazione e della concorrenza viene risolto in maniera trasparente da AWS EC2.
Questo ci permetterà di concentrarci sulla logica di business.

2.7 Casi Limite

2.7.1 Configurazione, Start-up, Shut-down & Reboot

Grazie alla console di gestione di AWS potremo configurare, effettuare l'attivazione, lo spegnimento, il riavvio dei server in maniera efficace e veloce.
La gestione quindi viene effettuata tramite la console e non all'interno del sistema stesso, proprio per questo non è necessaria la definizione di casi d'uso specifici per queste operazioni.

2.7.2 Eccezioni, Problemi di Connessione & Problemi di Servizio

Per quanto riguarda i problemi servizio, AWS garantisce l'uptime minimo del servizio al 99.95%.
In caso di down-time o problemi di connessione causati dal network l'utente visualizzerà un messaggio di errore relativo alla mancanza di connessione.
Infine, nel caso di possibili errori relativi alla logica applicativa, l'utente visualizzerà un messaggio d'errore relativo al problema riscontrato.

3. Servizi dei sottosistemi

SS_1 GestoreAccount

Questo sottosistema comprende le funzionalità relative sia agli account dei medici che dei pazienti, offrendo metodi utili alla registrazione, modifica, visualizzazione dei propri dati e visualizzazione dei dati relativi ai propri dottori o ai propri pazienti. Inoltre si occupa della verifica delle credenziali durante l'accesso all'applicazione e della modalità SOS.

Servizi	Descrizione
registraAccountDottore()	**Permette al dottore di registrarsi al sistema**
registraAccountPaziente()	**Permette al paziente di registrarsi al sistema**
login()	**Permette ad un dottore o un paziente registrato di poter accedere al servizio** tramite le credenziali inserite all'atto **della registrazione**
profiloPersonaleDottore()	**Permette al dottore di accedere al proprio profilo**
profiloPersonalePaziente()	**Permette al paziente di accedere al proprio profilo**
logout()	**Permette ad un dottore o un paziente loggato di poter uscire dal sistema**
impostaSOS()	
modalitaSOS()	
controlloCredenziali()	**Verifica se le credenziali inserite al** momento dell'accesso sono corrette, **in caso fossero corrette viene invocato il metodo login()**

SS_2 GestoreFarmaco

Questo sottosistema comprende le funzionalità relative ai farmaci pubblici e quelli relativi al proprio virtualmobilFarm, offrendo metodi di inserimento, ricerca, notifica di scadenza e creazione (da parte dei soli dottori) dei farmaci.

Servizi	Descrizione
visualizzaFarmacoVMF()	Visualizza l'elenco dei farmaci **presenti nel VMF**
aggiungiFarmacoVMF()	**Permette di inserire un farmaco nel VMF compreso di scadenza**
ricercaFarmacoVMF()	**Permette di ricercare nel database pubblico dei farmaci quello da inserire nel proprio VMF**
creaFarmaco()	**Permette al dottore di inserire un nuovo farmaco nel database pubblico dei farmaci**
eliminaFarmacoVMF()	Permette l'eliminazione di un farmaco **dal proprio VMF**

SS_3 GestorePiano

Questo sottosistema comprende tutte le funzionalità relative alla gestione dei piani terapeutici, offrendo classi e relativi metodi inerenti alla creazione di un nuovo piano, alla modifica e alla predisposizione di notifiche negli orari di assunzione.

Servizi	Descrizione
creaPianoTerapeutico()	**Permette al dottore di creare un nuovo piano terapeutico**
aggiungiFarmaco()	**Permette di inserire un farmaco da assumere**
rimuoviFarmaco()	**Permette di rimuovere un farmaco da assumere**
selezionaPaziente()	**Permette di selezionare il paziente a cui è destinata la terapia**
modificaDettagliAssunzioneFarmaco()	**Permette di modificare i dettagli** sull'assunzione del farmaco
salvaDettagliAssunzioneFarmaco()	**Salva i cambiamenti effettuati dalla modifica effettuata con modificaDettagliAssunzioneFarmaco()**
visualizzaPianoTerapeutico()	
visualizzaPianiTerapeuticiDiUnPaziente()	
visualizzaDettagliAssunzioneFarmaco()	