System Design Document

**MapYou**

**Giuseppe Fusco**

matricola: 399000091

email: [giuseppe.fusco.666@gmail.com](mailto:giuseppe.fusco.666@gmail.com)

**Giovanni Di Blasio**

matricola: 399000090

email: [diblasiogiovanni48@gmail.com](mailto:diblasiogiovanni48@gmail.com)

Indice

1. Introduzione
   1. Scopo del sistema
   2. Obiettivi di progettazione
   3. Definizioni, acronimi e abbreviazioni
   4. Riferimenti
   5. Panoramica sul resto del documento
2. Architettura software attuale
3. Architettura software proposta
   1. Overview
   2. Decomposizione in sottosistemi
      1. Sottosistema Client
      2. Sottosistema Server
      3. Sottosistema di comunicazione
   3. Mapping Hardware/Software
   4. Gestione dei dati persistenti
   5. Controllo e sicurezza degli accessi
   6. Flusso di controllo globale
   7. Condizioni limite

1. Introduzione

* 1. Obiettivo

L’obiettivo del documento è fornire una descrizione dettagliata dei requisiti per il sistema software “MapYou”.

1.2. Scopo

**Mapyou** è un’applicazione/sistema mobile su piattaforma Android che consente ad un insieme di utenti, appartenenti ad una community (**Mapme**), di raggiungere una location, tramite un percorso, e “guidarli” al raggiungimento della stessa.

L’utilizzo del sistema è consentito solo tramite registrazione.

Mapyou fornisce agli utenti iscritti ad una Mapme un servizio di visione comune degli spostamenti effettuati dagli altri utenti per facilitarne l’indirizzamento verso la location di arrivo.

Tale servizio include:

* una sezione interattiva (**RTI**: real-time interaction):
  + visualizzazione real-time delle posizioni degli utenti iscritti alla Mapme;
  + visualizzazione del percorso della Mapme;
* lo scambio di messaggi tra utenti.

Per fornire tale servizio il sistema necessita di una connessione sia Internet che GPS per ottenere le informazioni geografiche degli utenti.

Lo scambio di messaggi tra utenti è realizzato mediante un sistema di chat interno all’applicazione mobile.

Tutte le informazioni del sistema sono mantenute in un database locato su un server.

L’applicazione potrà essere scaricata da qualsiasi application store o servizi simili.

1.3. Definizioni, acronimi e abbreviazioni

|  |  |
| --- | --- |
| Termine | Definizione |
| Mapyou | Nome del sistema |
| Utente | Qualcuno che interagisce con l’applicazione mobile |
| Location | Locazione fisica di un punto di arrivo/destinazione |
| Mapper | Utente che decide/sceglie/promuove una location di destinazione |
| Mapme | Comunità/gruppo/insieme di utenti |
| Mapped | Utente che fa parte di una mapme |
| RTI | Modalità di interazione real-time con il sistema |
| GPS | Global Positioning System |
| GPS-Navigator | Software installato sul dispositivo mobile che fornisce connessioni GPS |
| Notification-System | Sistema software di terze parti che permette l’invio di notifiche verso dispositivi mobile |
| Application Store | Applicazione installata sul dispositivo mobile che aiuta gli utenti a cercare nuove applicazioni compatibili con il sistema operativo del dispositivo ed effettuarne il download da Internet |

1.4. Riferimenti

1.5. Panoramica sul resto del documento

1. Architettura software attuale

Il sistema è stato implementato dal nulla in quanto nessun sistema avente le stesse caratteristiche esiste in precedenza, per cui questa è la fase di “Greenfield Engineering” e la raccolta dei requisiti è stata effettuata colloquiando direttamente con il cliente, cercando di estrarre i requisiti fondamentali dalle sue richieste.

1. Architettura software proposta
   1. Overview

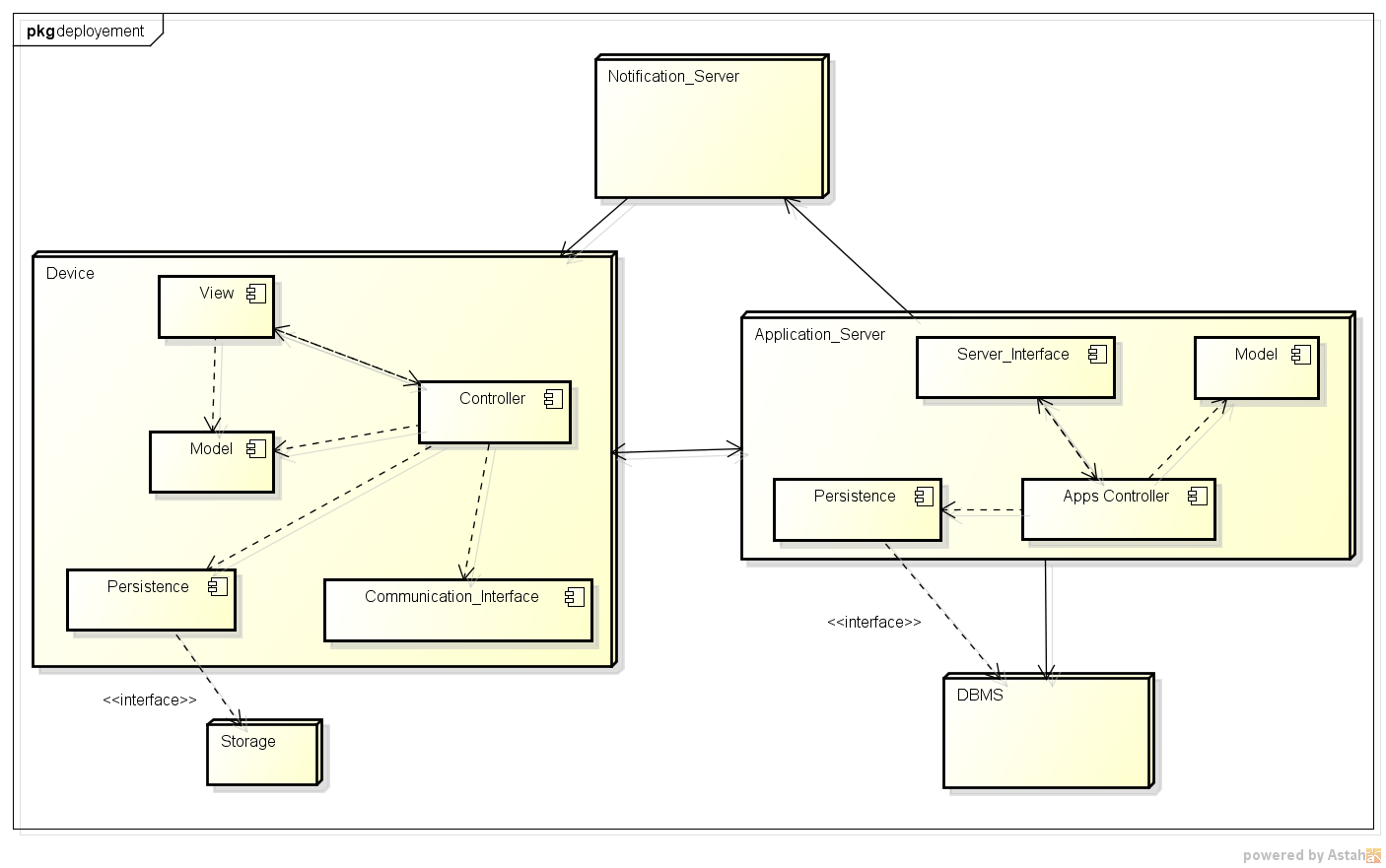
Il sistema è costituito da un nucleo centrale contenente l’archivio di tutte le informazioni sulle MapMe ed sugli utenti. Gli utenti possono interagire con il sistema centrale da qualsiasi postazione. Le richieste devo essere sempre fatte al sistema centrale. L’architettura più idonea per il sistema è di tipo Client/Server.

Il sistema permette la creazione, la gestione delle partecipazione alle MapMe, lo scambio di messaggi privati tra i vari utenti. Quando un utente richiederà un determinato servizio, la richiesta sarà inoltrata al Server, il quale tramite un’interazione con il DBMS, preleva l’informazione, la quale sarà inoltrata all’utente che ne ha fatto richiesta.

* 1. Decomposizione del sistema in sottosistemi

Il sistema software è stato decomposto in sottosistemi indipendenti. I moduli dei sottosistemi sono stati progettati evitando un forte accoppiamento tra di essi ed incrementandone la coesione. La progettazione dei vari sottosistemi e delle comunicazioni tra moduli è stata effettuata adottando una serie di pattern comportamentali, strutturali e creazionali, per agevolare lo sviluppo di nuovi moduli, la portabilità del sistema, i vari interventi di manutenzione e aggiornamento.

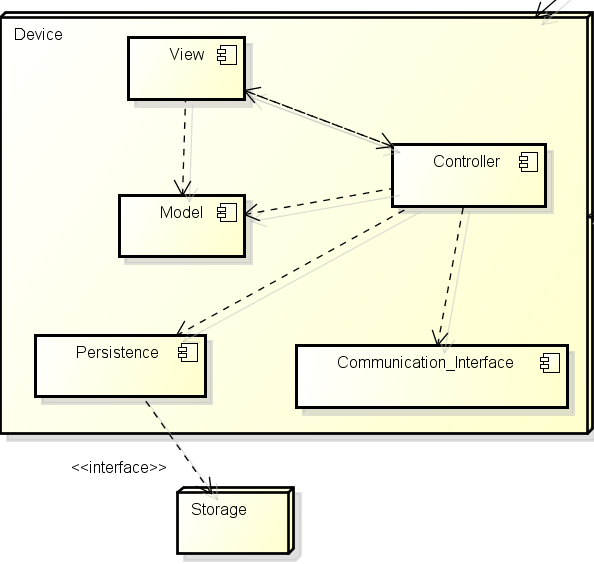
Di seguito è illustrato il diagramma di deployment con i componenti suddivisi per ogni nodo.



I sottosistemi fondamentali possono essere divisi su due macro livelli, il primo riguarda i sottosistemi lato Client ed il secondo riguarda i sottosistemi lato Server, i quali comunicano mediante un sottosistema di comunicazione. Un altro componente chiave per la comunicazione tra i dispositivi è il Notification System: è un sistema software di terze parti che permette l’invio di notifiche verso dispositivi mobile.

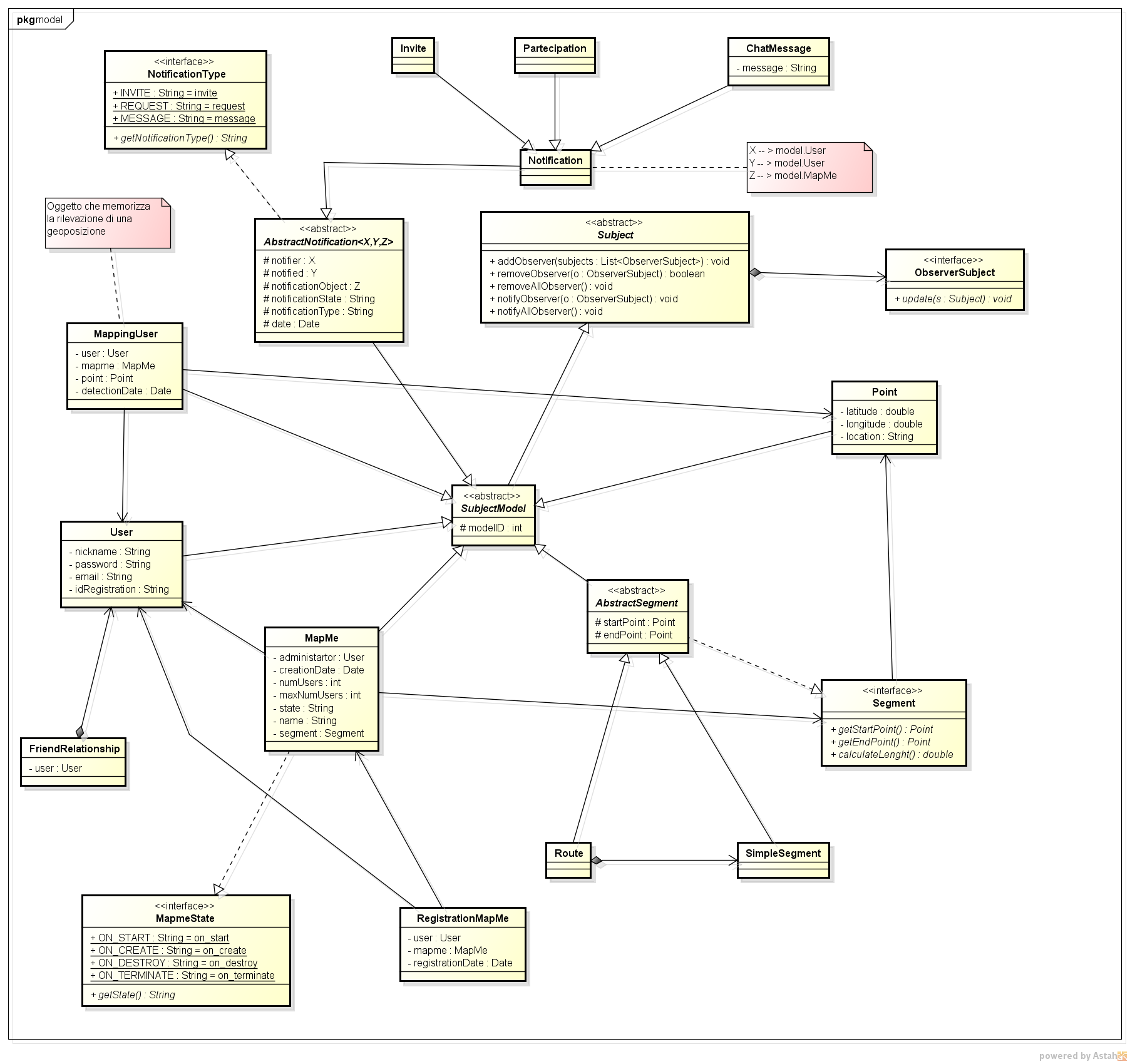
3.2.1 Sottosistema Client

Il sottosistema Client è stato progettato utilizzando il pattern architetturale MVC (Model-View-Controller).



* **Model:** componente che gestisce gli oggetti di dominio del sistema.

In Fig… è mostrato il Class Diagram di design.



I pattern utilizzati nella progettazione sono i seguenti:

* **Composite:** è stato utilizzato per la gestione dei segmenti di un percorso, in quanto un generico percorso è composto da un solo segmento, il quale può, o non, essere a sua volta decomposto in molteplici segmenti.
* **Abstract Factory:** è stato utilizzato per la gestione dei messaggi di notifica (Invito – Richiesta di partecipazione – Messaggi tra utenti).
* **Observer:** è stato utilizzato per gli aggiornamenti degli oggetti del componente View, che visualizzano le informazioni inerenti al componente Model.
* Persistence
* Controller
* View
* Communication Interface: tale componente verrà trattata nel paragrafo 3.2.3.