



Realizzazione di un prototipo di un software web based per la gestione di un inventario comunale

tesi di laurea

## **Realizzazione di un prototipo di un software web based per la gestione di un inventario comunale**

**Anno Accademico 2009/2010**

relatore

**Ch.mo prof. Porfirio Tramontana**

correlatore

**Ch.mo Ing. Luigi Pontillo**

candidato

**Michele Vitelli**

**Matr. 534 2170**



## Redazione dell'Inventario comunale

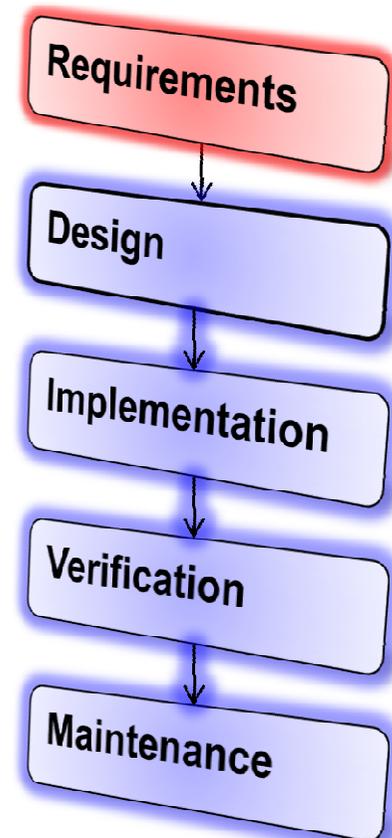
- Operazione necessaria per tenere traccia del Patrimonio
- Necessità di effettuare aggiornamenti
- Normative e leggi che ne regolano modo e forma
- Possibilità di accesso ai dati anche da remoto

### Comprensione del dominio del problema

- Serie di interviste con il committente

- Capire le necessità del cliente
- Identificare i vincoli organizzativi e legislativi

- Class diagram del dominio del problema
- Stipula del documento SRS
- Rappresentazione dei casi d'uso



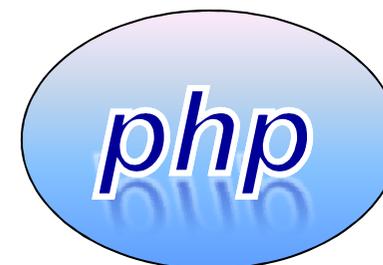




## Scelte progettuali

### ❑ Linguaggio con un buon compromesso tra

- Scalabilità
- Velocità di sviluppo
- Manutenibilità



### ❑ Utilizzo di un DBMS relazionale che offra

- Gestione degli accessi
- Integrità
- Accesso
- Modifica



### ❑ Utilizzo di un framework

- Miglior manutenibilità
- Livello di astrazione alto
- Minor tempo di sviluppo





Realizzazione di un prototipo di un software web based per la gestione di un inventario comunale



Presenza di un gran numero di librerie	Semplicità di configurazione tramite script in PHP	Pochi prerequisiti	Non richiede librerie extra
Utilizzo di un gran numero di design pattern per la realizzazione di strutture eleganti	Presenza di capacità basilari di routing	Piccolo ingombro	Presenza di potenti funzioni a livello di database quali HasOne e HasMany
Documentazione molto completa	Curva di apprendimento bassa	Curva di apprendimento bassa	Funzione di Scaffolding (scaffalatura) molto potente
MVC debole, mancanza di realizzazione della view e poco controllo sulla front page	Layer di database con caratteristiche del modello MVC	Controllo robusto sulle configurazioni	Utilizzo del pattern MVC per ogni livello
Mancanza di script automatizzati per la creazione dell'applicazione, va scritto tutto a mano	Funzionalità iniziale di layout per la creazione di un'interfaccia	Strumenti per Testing, Debugging e per la documentazione	Scripting per effettuare operazioni automatiche da linea di comando



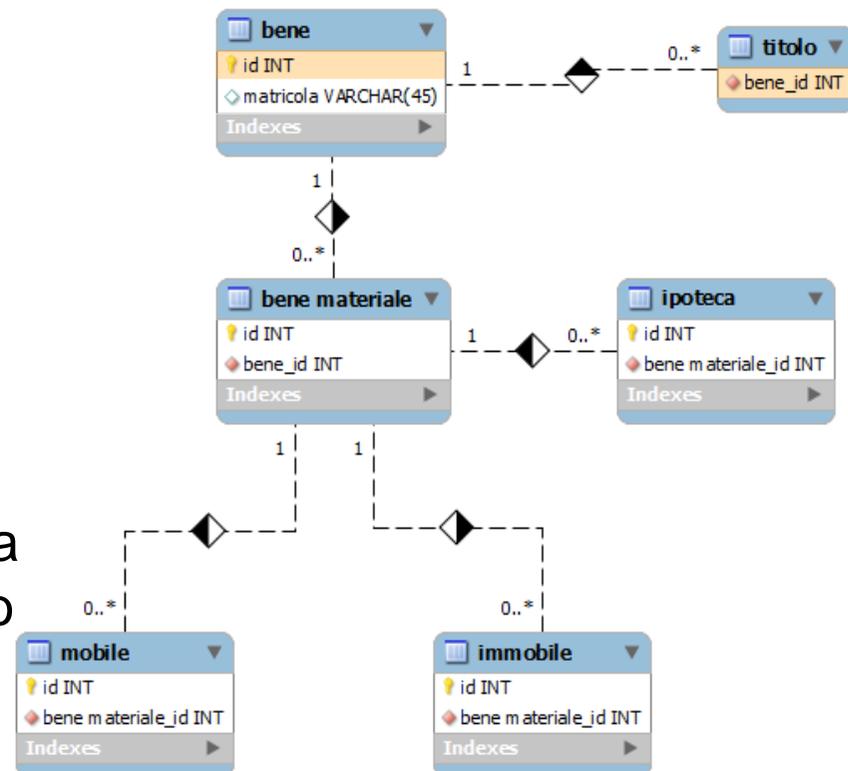
## Progettazione

### □ Pattern

- Scelta del modello MVC come design pattern più adatto al contesto
  - Maggior coesione
  - Minor accoppiamento
  - Maggior flessibilità
  - Possibilità di riuso

### □ Database

- Progettazione concettuale del database sulla base del class diagram del dominio del problema
- Realizzazione dell'archivio storico
- Traduzione del modello concettuale in modello logico





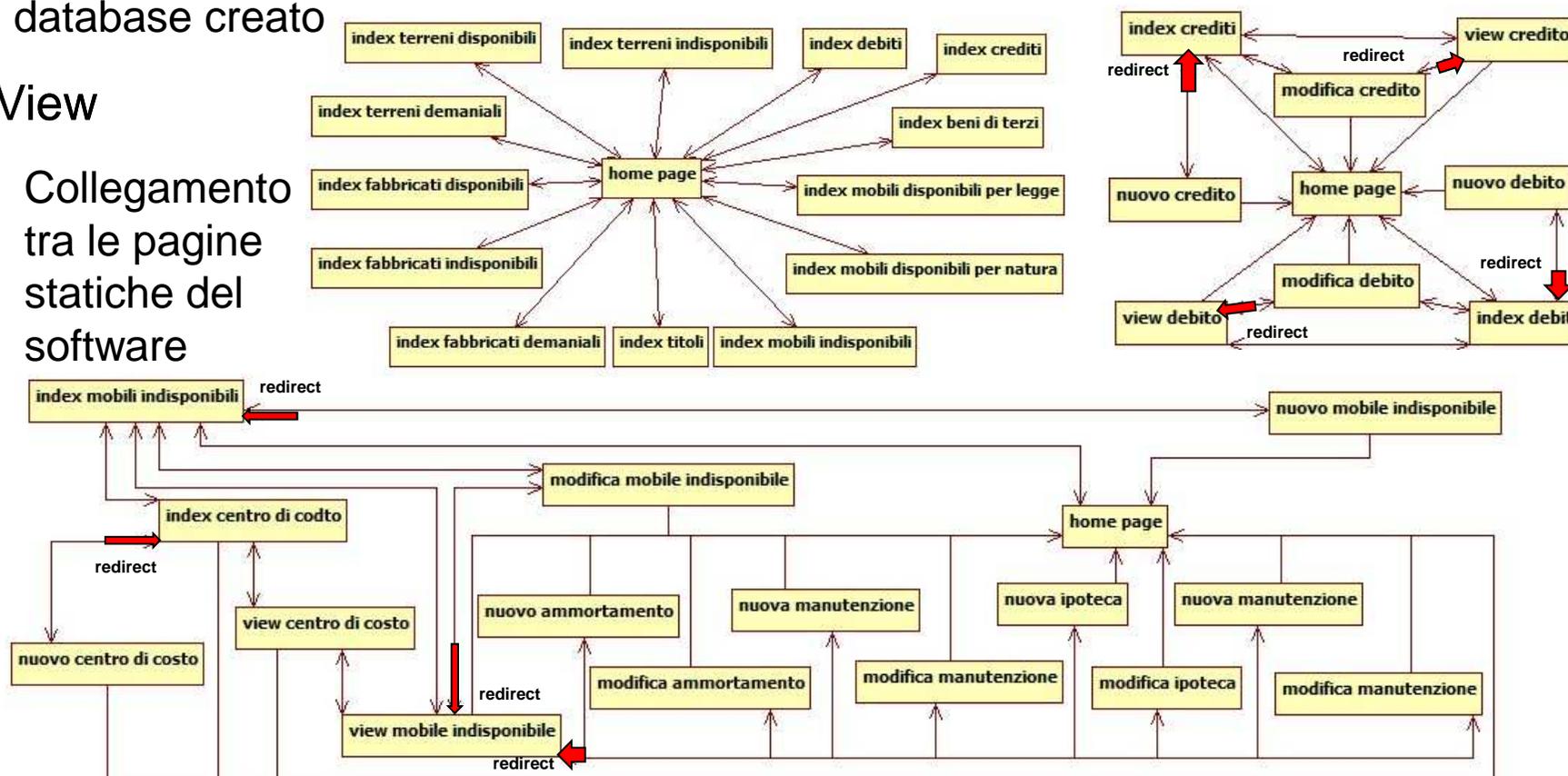
## Progettazione

### Model

- Contiene i metodi per accedere ai dati del database
- Viene generato in modo automatico da CakePHP configurando l'accesso al database creato

### View

- Collegamento tra le pagine statiche del software

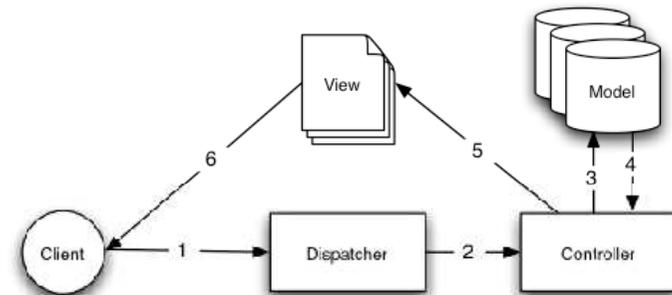




## Progettazione

### □ Controller

- Funzionalità per ogni occorrenza del modello a cui ci si riferisce:
  - creazione
  - modifica
  - eliminazione
  - visualizzazione
- Funzionalità per tutte le occorrenze
  - elenco



## Implementazione del software: CakePHP e pattern MVC

- **View:** File (Pagine) .ctp (CakePHP Template) contenenti la logica per la presentazione dei dati passati dal controller
- **Controller:** File .PHP contenenti la logica del software
- **Model:** File .PHP contenenti criteri di validazione, relazioni e comportamento dei dati, automaticamente collegati al database adottato



Realizzazione di un prototipo di un software web based per la gestione di un inventario comunale



Home page

**Immobili**

Fabbricati

- Fabbricati indisponibili
- Fabbricati disponibili
- Fabbricati demaniali

Terreni

- Terreni indisponibili
- Terreni disponibili
- Terreni demaniali

**Mobili**

- Mobili indisponibili
- Mobili disponibili per legge
- Mobili disponibili per natura

**Non materiali**

titoli

- Crediti
- Debiti

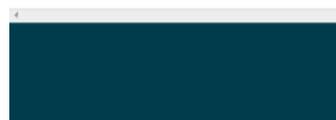


- Azioni
- Nuovo fabbricato indisponibile
  - Elenca unita fabbricati
  - Nuova unita fabbricati

**Fabbricati indisponibili**

Matricola	Data Modifica	Sezione	Zona	Microzona	Categoria	Classe	Consistenza	Rendita	Dati Derivanti Da	Uso Speciale	Descrizione	Annotazione	Via
FABB001	2010-12-15 18:11:34	9	5	4	7	a	23	30.00		caserma	palazzo di 2 piani		mazz

Page 1 of 1, showing 1 records out of 1 total, starting on record 1, ending on 1

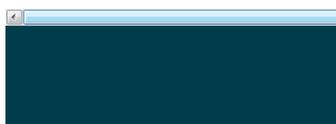


- Azioni
- Modifica fabbricato indisponibile
  - Elimina fabbricato indisponibile
  - Elenca fabbricati indisponibili
  - Nuovo fabbricato indisponibile
  - Elenca unita fabbricati
  - Nuova unita fabbricati

**Fabbricato indisponibile**

Matricola	FABB001
Data Modifica	2010-12-15 18:11:34
Sezione	9
Zona	5
Microzona	4
Categoria	7
Classe	a
Consistenza	23
Rendita	30.00
Dati Derivanti Da	
Uso Speciale	caserma
Descrizione	palazzo di 2 piani
Annotazione	
Via	mazzini
Foglio	1
Particella	R

Indice dei fabbricati indisponibili



Visualizzazione di un fabbricato indisponibile



## Testing

Il testing effettuato è di tipo comportamentale, o Black box. Sono state considerate classi di equivalenza per cercare di coprire tutte le categorie di input progettate

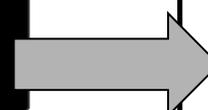
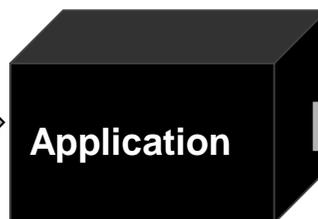
### Input:

#### Classi valide:

- Tipo e valore corretto

#### Classi non valide

- Valore negativo per gli interi
- Valore non univoco dove ciò è richiesto
- Valore nullo
- Tipo errato



### Output:

#### Risultati possibili

- Funzioni errate o mancanti
- Errori di interfaccia
- Errori nella struttura dati o all'accesso al database esterno
- Errori comportamentali

**I casi di test effettuati non hanno rilevato malfunzionamenti**



## Conclusioni

- ❑ Implementazione di un software per la gestione di un inventario comunale come web application, con tutti i benefici che tale scelta comporta  
Compatibilità cross-platform, aggiornamento continuo, immediatezza di accesso, facilità di prova, meno requisiti di memoria, meno bug, minor prezzo, dati online, utenti multipli nello stesso tempo, sicurezza dei dati, maggior libertà sul linguaggio di programmazione
- ❑ Utilizzo di un linguaggio che offre un' ottima scalabilità e velocità di sviluppo
- ❑ Utilizzo di un framework facile da apprendere, rapido con potenti strumenti da riga di comando che facilitano il lavoro di prototipazione
- ❑ Creazione automatica di un archivio storico di tutte le operazioni e modifiche effettuate
- ❑ Facilmente estensibile a nuove funzionalità grazie all'utilizzo di un valido design pattern: MVC

## Sviluppi futuri

- ❑ Stampa dei report in formato pdf
- ❑ Possibilità di personalizzazione delle viste