
MUNKAI DŐ-NYILVÁNTARTÁS ALKALMAZÁS KÉSZÍTÉSE J2EE SEGÍTSÉGÉVEL

Irta: Szőke Ákos

2004. június 1.

Verzió 2.0

Státusz: Lezárt

Tartalomjegyzék

BEVEZETÉS	3
MEGOLDANDÓ PROBLÉMA.....	4
KÖVETELMÉNY SPECIFIKÁCIÓ	5
Rendszer funkcionalitás.....	5
Használati eset lista	5
Használati eset diagramok.....	5
A Beosztott használati eset diagramja.....	6
A Vezető használati eset diagramja	6
Forgatókönyvek.....	7
Munkaidő elszámolási űrlap létrehozása.....	7
Napi munkaidő elszámolásának rögzítése.....	7
Jóváhagyott munkaidő elszámolások lekérdezése	8
Függőben lévő munkaidő elszámolások lekérdezése.....	8
A munkaidő elszámolás elutasítása.....	8
A munkaidő elszámolás jóváhagyása.....	8
Rendszerterv	9
A MEGVALÓSÍTÁS JELLEMZŐI.....	10
Infrastrukturális jellemzők	10
Architekturális jellemzők	11

Bevezetés

Jelen dokumentum célja, hogy definiálja a Szoftver architektúrák c. tárgy féléves implementációs feladatának funkcionális és nem funkcionális jellemzőit.

A féléves feladatként megvalósítandó esettanulmány alapvető célja, hogy egy kiválasztott komponens technológiával – esetünkben a J2EE – implementálva, a magyar informatikai cégeknél még csak elterjedőben lévő munkaidő elszámolásra példát adjon. Általában a munkaidő elszámolás vezetésének célja, hogy a cég egyes munkatársainak a különböző projekteken való részvétele rögzítésre kerüljön, így az ügyfélnek történő számlázáshoz objektív alapot szolgáljon.

Ezen kívül hasznos a cégek fejlesztési folyamatainak méréséhez, melyek alapján az egyes munkatársak fejlesztési-, a különböző projekt megvalósítások- és így a cég működési-hatékonysága mérhetővé, elemezhetővé, és ezért növelhetővé váljon – mely manapság egy kulcstényező a haladó szoftverfejlesztő cégeknél (pl. PSP, TSP, CMM metodikák és metrikák alkalmazása).

A bemutatandó esettanulmány a fenti célkitűzést az alábbi megkötésekkel valósítja meg:

- Két szerepkört különböztet meg: Beosztott és Vezető (nem támogatja a tetszőleges projektszervezeti hierarchiának a kialakítását)
- Csak a projekteken eltöltött idő tárolódik, a feladat nem

Megoldandó probléma

A vállalat különböző munkatársai a munkanapok során számos feladatot ellátnak. Ezek a feladatok döntő többsége különböző projektekhez kapcsolódik. Alapvető szükséglet, hogy a projekt specifikus munkavégzések nyilván legyenek tartva, mely nyilvántartásokból – a kívánt időpontban – lekérdezéseket lehessen készíteni, mely alapot adhat az egyes ügyfelek felé történő számlázáshoz, illetve a későbbi projekt tevékenységek tervezéséhez (idő, költség, ember).

Szükséges megvalósítani, hogy a munkatársak nyilvántartásba vezethessék saját munkaidő ráfordításukat, amely ráfordítások a felettesük jóváhagyásával kerül be ténylegesen a nyilvántartásba.

Fontos elvárás, hogy a munkatársak a Munkaidő-nyilvántartás rendszert a nap bármely időszakában elérhessék, illetve az elérés számos helyről elérhetővé váljon – anélkül, hogy az elérés helye előre ismert lenne. Ezen kívül fontos, hogy az elérés során alkalmazandó rendszer (kliens) heterogenitását – a későbbi továbbfejlesztés érdekében – támogassa az implementáció.

Követelmény specifikáció

Rendszer funkcionalitás

Az előzőekben megadott problémára az alábbi felsorolás körvonalazza a megvalósítandó rendszer által megvalósítandó funkcionalitást:

- Kéttípusú szerepkör kialakítása szükséges:
Beosztott: munkaidő rögzítése, és
Vezető: munkaidő jóváhagyás, riportkészítés.
- A Beosztottak fél órának a többszörösét tudják rögzíteni a rendszerben.
- Beosztottak módosíthassák a bevitt munkaidőt a Vezető általi jóváhagyás előtt.
- Függetlenben lévő nyilvántartások lekérdezése.
- Jóváhagyott nyilvántartások lekérdezése.
- A Vezetők riportokat kérhetnek le a rendszerről: az egyes hónapokban a különböző munkatársak mennyi időt dolgoztak projekteken.
- A Beosztottak és a Vezetők standard Web browser felületen érhessek el a rendszer funkcionalitását.

Használati eset lista

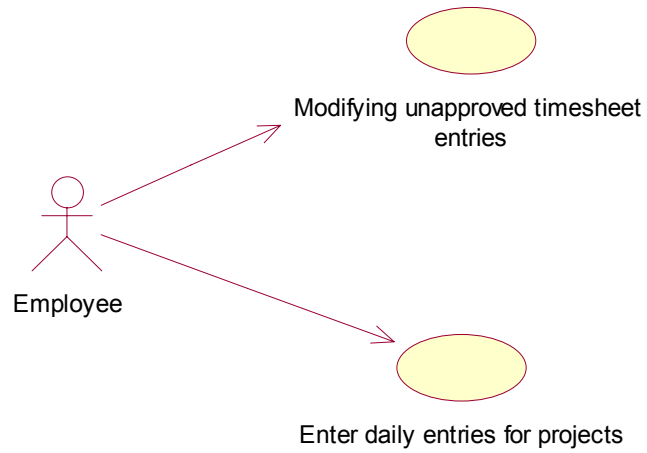
A követelményelemzés során az alábbi használati esetek (Use-case-ek) lettek azonosítva:

- Napi munkaidő elszámolásának rögzítése
- A függetlenben lévő munkaidő elszámolás módosítása
- A munkaidő elszámolás jóváhagyása
- A munkaidő elszámolás elutasítása
- Függetlenben lévő munkaidő elszámolások lekérdezése
- Jóváhagyott munkaidő elszámolások lekérdezése

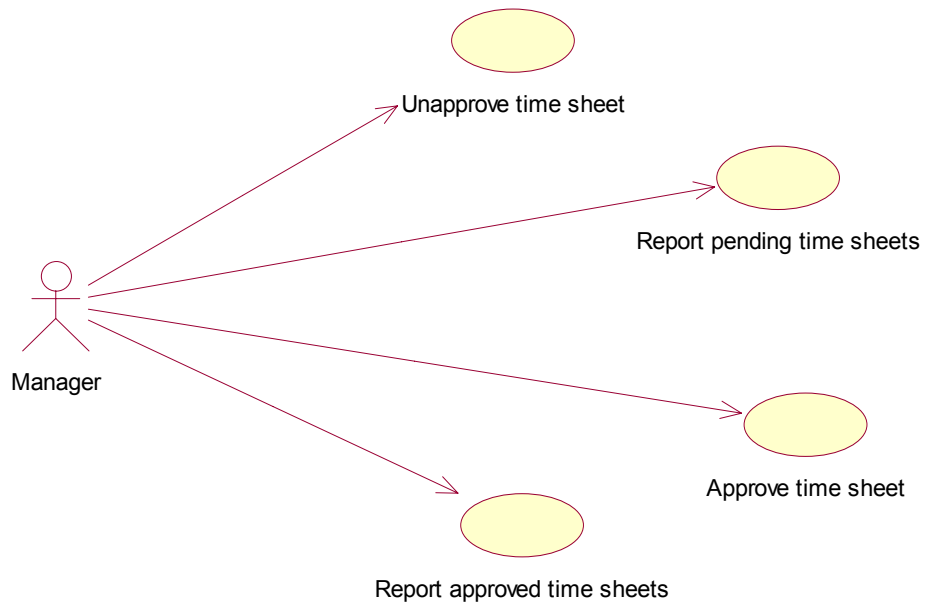
Használati eset diagramok

A követelmény elemzés során azonosított két aktor használati eset diagramjai az alábbiakban látható:

A Beosztott használati eset diagramja



A Vezető használati eset diagramja



Forgatókönyvek

Az alábbi felsorolás adja meg az egyes használati esetek – mint a rendszer statikus funkcionális jellemzői – dinamizmusát, a forgatókönyveket. Az alábbiakban megadott forgatókönyvek a szakterületi (domain) modell legfontosabb elemeit tartalmazzák:

- Employee (Beosztott)
- TimeSheet (Munkaidő elszámolási űrlap)
- TimeSheet entry (Munkaidő elszámolási űrlap egy bejegyzése)
- Project (Projekt)

Munkaidő elszámolási űrlap létrehozása

USE CASE: Munkaidő elszámolási űrlap létrehozása
Előfeltétel: Űrlap létrehozási dátum ismert
Alap eset
<ol style="list-style-type: none"> 1. A rendszer létrehoz egy munkaidő elszámolást, egy Beosztottat és egy dátumot rendel hozzá 2. A rendszer eltárolja a felparaméterezett munkaidő elszámolást

Napi munkaidő elszámolásának rögzítése

USE CASE: Napi munkaidő elszámolásának rögzítése
Alap eset
<ol style="list-style-type: none"> 1. A Beosztott lekér egy munkaidő elszámolást a saját nevére 2. A rendszer leellenőrzi a munkaidő elszámolás létezését <ol style="list-style-type: none"> 2a. Ha létezik, azt megjeleníti a Beosztottnak 2b. Ha nem létezik, a <i>Munkaidő elszámolási űrlap létrehozása</i> -t jeleníti meg 3. Beosztott beállítja a hét napját 4. A Beosztott beállítja a projektet és az óraszámot, amennyit a projekten dolgozott 5. A Beosztott elmenti a beállításait
Alternatív eset
5: A Beosztott abortálja a beállításait

Jóváhagyott munkaidő elszámolások lekérdezése

USE CASE: Jóváhagyott munkaidő elszámolások lekérdezése
Előfeltétel: Jóváhagyott munkaidő elszámolások megléte
Alap eset
1. A rendszer megjeleníti a jóváhagyott lévő munkaidő elszámolásokat (Beosztott, hét bontás)

Függőben lévő munkaidő elszámolások lekérdezése

USE CASE: Függőben lévő munkaidő elszámolások lekérdezése
Alap eset
1. A rendszer megjeleníti a függőben lévő munkaidő elszámolásokat (Beosztott, hét bontás)

A munkaidő elszámolás elutasítása

USE CASE: A munkaidő elszámolás elutasítása
Alap eset
<ol style="list-style-type: none"> 1. A Vezető kiválasztja a munkaidő elszámolást adott Beosztottra és időtartamra vonatkozóan 2. A rendszer megjeleníti a függőben lévő bejegyzéseket 3. A bejegyzés függőbe való állítása
Alternatív eset
2: Nincs megjeleníthető munkaidő elszámolás

A munkaidő elszámolás jóváhagyása

USE CASE: A munkaidő elszámolás jóváhagyása
Alap eset

1. A Vezető kiválasztja a munkaidő elszámolást adott Beosztottra és időtartamra vonatkozóan
2. A rendszer megjeleníti a függőben lévő bejegyzéseket
3. A bejegyzés jóváhagyottra való állítása

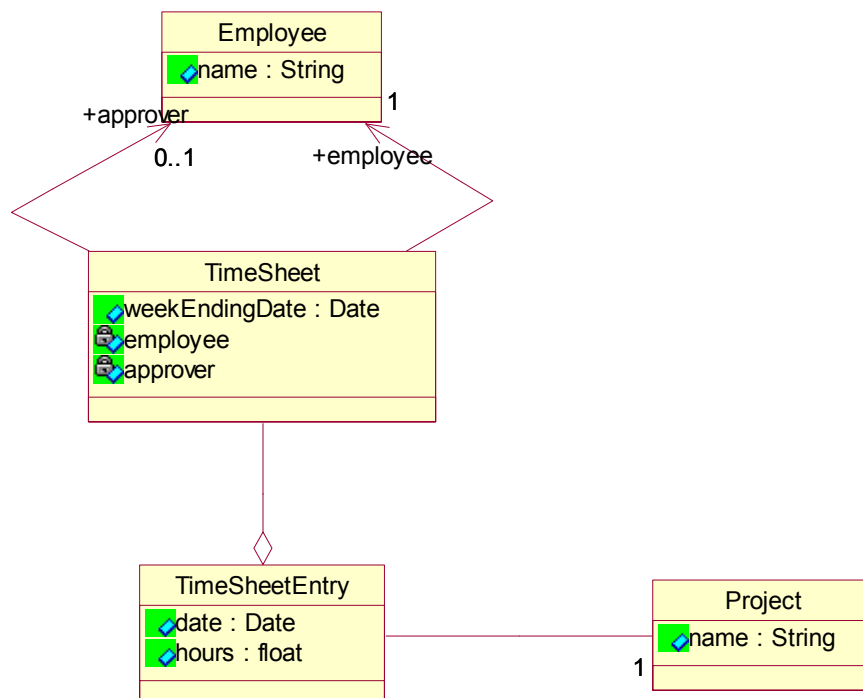
Alternatív eset

2: Nincs megjeleníthető munkaidő elszámolás

Rendszerterv

Jelen esetben megvalósítandó rendszer tervét nem célszerű túlságosan részletezni, hiszen nem egy nagy bonyolultságú alkalmazásról van szó. Azonban azon kulcsfontosságú elemekre vonatkozóan érdemes tervet készíteni, melyek az alkalmazás központi részét jelentik. Az esettanulmányban a Munkaidő elszámolási űrlapot (TimeSheet) és a hozzá kapcsolódó többi objektumok (Beosztott /Employee/, Munkaidő elszámolási űrlap bejegyzés /TimeSheetEntry/, és a Projekt /Project/) kapcsolatát érdemes meggondolni.

Az alábbi osztálydiagram a fent említett elemek kapcsolatait írja le:



A megvalósítás jellemzői

Az implementáció – mivel a követelmények semmilyen megkötést nem tettek erre – az alábbi jellemzőkkel valósul meg:

Infrastrukturális elemek

- TCP/IP és HTTP protokollok
- IBM WebSphere 5.0.1 alkalmazás szerver
- IBM DB2 8.1 adatbázis szerver
- IBM HTTP Server 1.3.8 web szerver
- Standard Web browser (pl. MS Internet Explorer 6.0)

Technológia

- J2EE 1.3
- Java Server Pager (JSP) 1.2
- Java Servlets
- Struts framework (Apache Software Foundation)
- JDBC

Fejlesztő eszköz

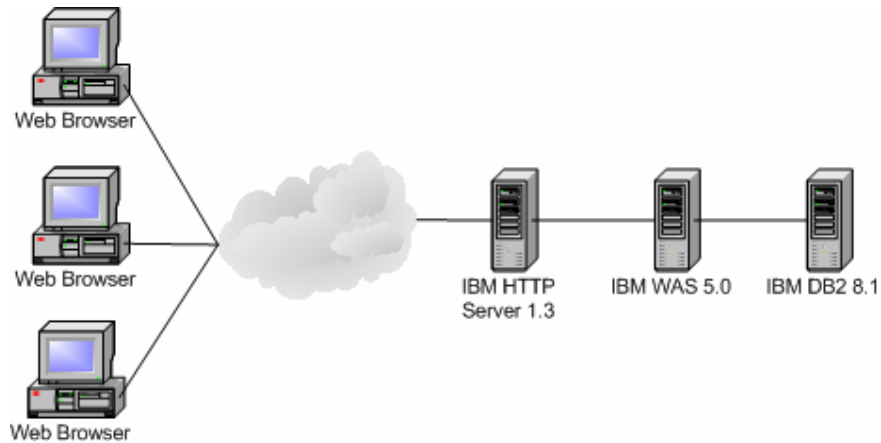
- IBM WebSphere Studio Application Developer 5.0

Egyéb implementációs jellemzők

- Gateway Servlet design
- Exception handling
- Logging

Infrastrukturális jellemzők

A követelmény elemzés során felmerült igények alapján az alábbi infrastrukturális felépítést célszerű megvalósítani:

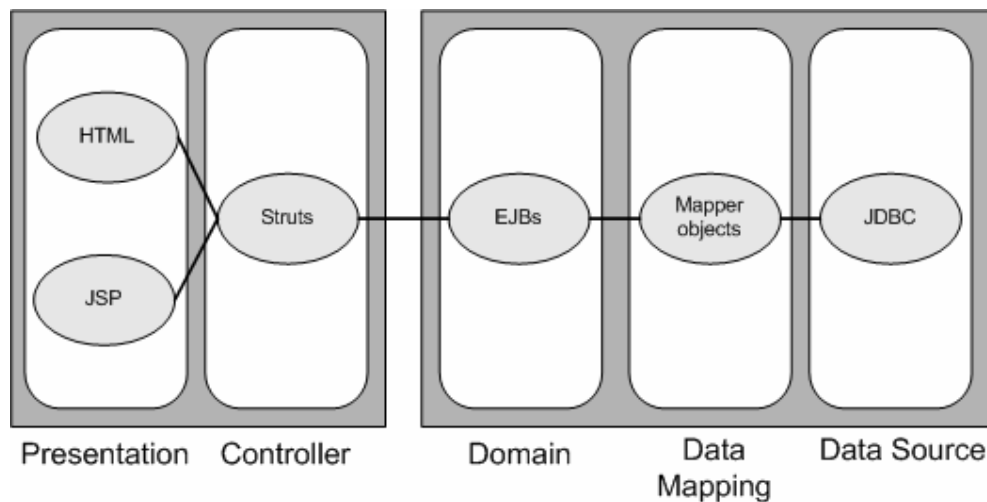


Architekturális jellemzők

A megvalósítandó alkalmazást réteges (layered) felépítésben célszerű megvalósítani – különösen a felhasználói interfészt – annak érdekében, hogy a későbbiekben a Web browser-es kliensek mellett - többek között - tenyérgepekkel (handheld PC), és mobil telefonokkal is kapcsolódni lehessen a rendszerhez. Ez az elvárás a Controller réteg kialakításával lesz támogatva.

Emellett célszerű az adatbázis esetleges cseréjének könnyű megvalósítását is támogatni, melyhez a Data Mapping réteget célszerű bevezetni.

Ennek érdekében a szokásos 3 rétegű felépítés helyett 5 rétegű megvalósítást célszerű kialakítani, a Controller és Data Mapping rétegek bevezetésével:



A megvalósított rendszer

Jelen fejezet célja, hogy a megtervezett alkalmazás megvalósítását dokumentálja, nagyvonalú fejlesztési dokumentációt szolgáltatva ezzel.

Mivel a megoldás az MVC (Model-View-Controller) architektúrát követi, ezért az implementáció bemutatása is ezt követi.

- Model— JavaBeans and EJBs
- View— JSPs
- Controller— Servlets

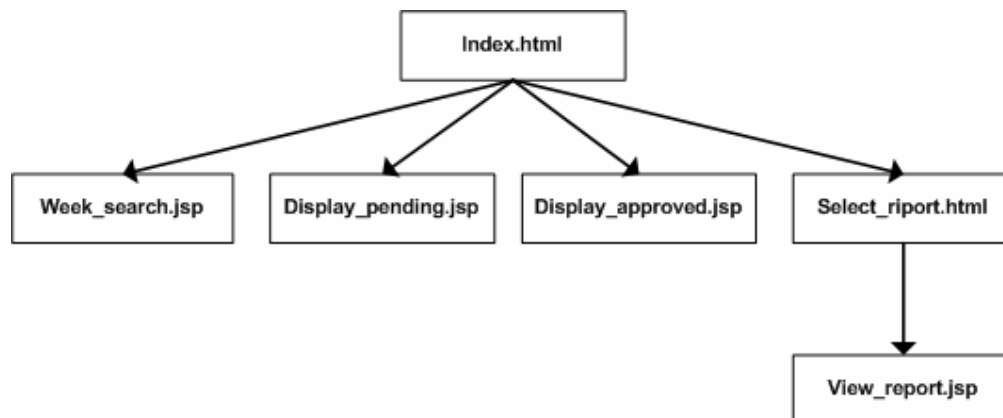
TimeSheetClient bemutatása

Ez az Web modul (WAR) tartalmazza az alkalmazás kliens oldalának funkcióit. Tartalmazza a prezentációs felületet (JSP, HTML), valamint a felhasználói felület dinamikus generálásához szükséges controller réteget is (Servlet). Ez a modul valójában a rendszer egyik leglényegesebb része, mivel a bemenet és megjelenítés mellett az input validációt (Struts Framework), formok közötti navigációt is tartalmazza.

A következőkben ezt a modult – mint az alkalmazás legjelentősebb részét – kicsit részletesebben áttekintjük.

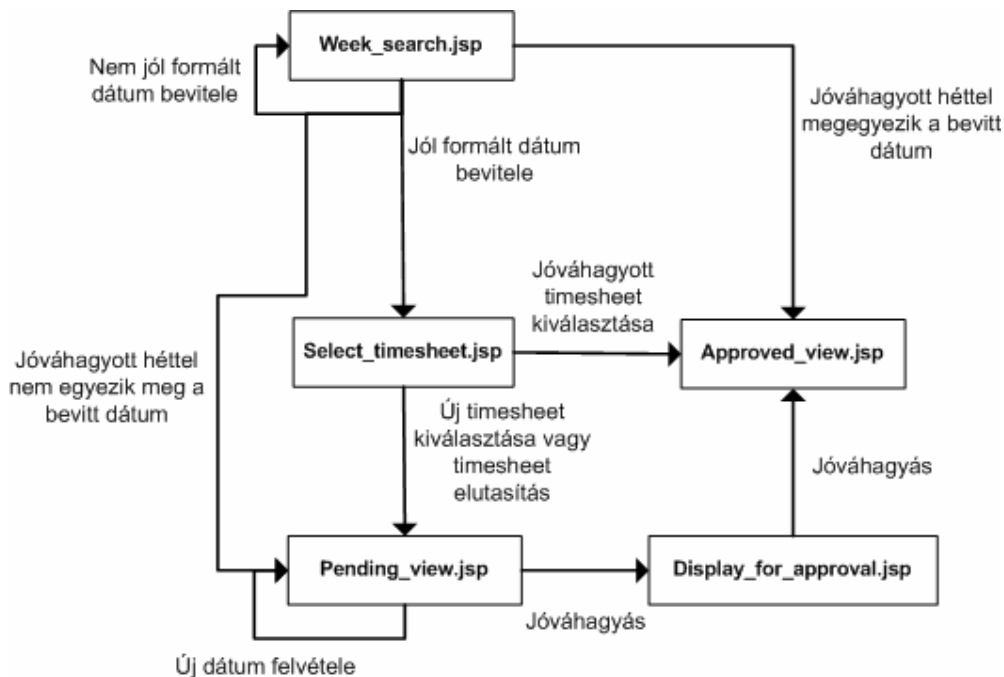
A rendszerbe történő belépés

Az alábbi diagram megadja, hogy a rendszerbe történő belépést követően a felhasználó milyen funkciókat választhat ki és ezen funkciókat mely rendszer elemek (.jsp és html) szolgálják ki.



Munkaidő nyilvántartás feldolgozása (create, update, approve)

Az alábbi diagram a rendszer leginteraktívabb és legkomplikáltabb részét írja le. Ez a diagram az előző diagramhoz a Week_search.jsp oldalon keresztül kapcsolódik.



A Week_search.jsp implementálása során az alábbi tervezői döntés született: ne drop-down listás megoldás szülessen, mert nagyon sok lehet a lista elemeinek a száma (egy adott évben 52, illetve sajnos a calendar control nincs HTML-ben, csak ActiveX-ben, vagy Applet-ben implementálható, ami viszont a vékony kliens elv feladását vonta volna maga után – ezért elvetettük).

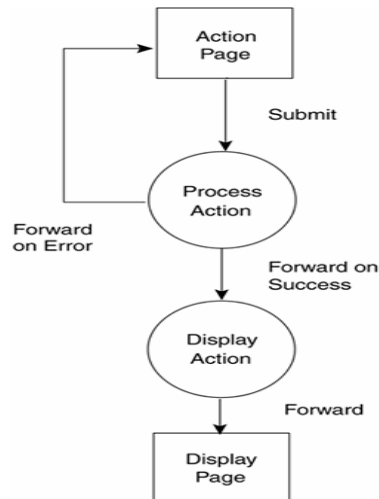
Struts Framework alkalmazása

A megoldásban gyakran alkalmazott megoldás a Struts framework, ezért ennek rövid ismertetése szükséges a megértéshez.

A Struts valójában néhány installálható .jar állományt tartalmaz – amely a servlet-es projektek gyakori problémáira nyújtanak segítséget:

- Validáció
- Hibakezelés (a felhasználó is értesüljön a hibákról)
- HTTP paraméterek JavaBean paraméterekre történő leképezése (HTTP POST form-ról jsp:usebean és jsp:setProperty-kre).

A Struts framework használata az alábbi működési módot támogatja (best practice):



A működés folyamatának nagyvonalú leírása

A `Week_search.jsp` Struts validációs logikát tartalmaz: `<html:form action="EmployeeTimeSheet.do">` - sortól kezdődően a bevitt dátumra vonatkozóan (valós dátum a kívánt formátumban).

Ebben a jsp-ben a Struts `html:options` tag lett használva a felhasználók drop down listában való megjelenítésére. A `html:select` lett használva arra, hogy a `TimeSheetForm` `ActionForm` `submitterId` tulajdonsága beállításra kerüljön.

A `TimeSheetSearchAction` tölti fel az adatokkal a session-t, melyet a `TimeSheetProcessor` kért le korábban.

A Struts a validációt a submit-ot követően végzi el, egy `TimeSheetForm` létrehozásával, melynek a `validate()` funkciójának meghívásával.

Ha a validáció eredményeképpen hiba fordul elő a bekérő form újra megjelenik, különben a `EmployeeTimeSheetAction` metódus hívódik meg.

TimeSheetEJB bemutatása

Ez az EJB modul tartalmazza az alkalmazás szerver oldalának legfontosabb funkcióját: EJB-Relational Database mapping-ot (Entity bean-ek leképését).

TimeSheetUtility bemutatása

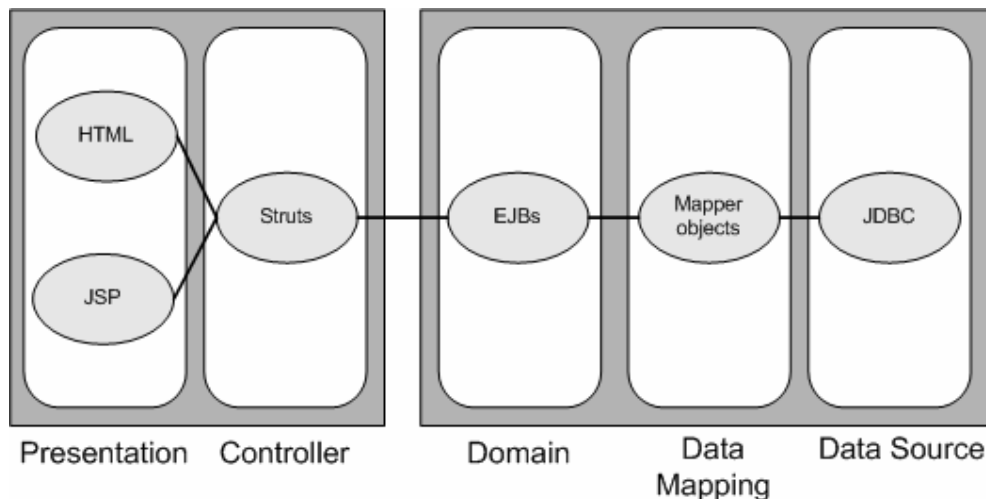
Egy enterprise alkalmazás legfontosabb részei az EJB és Web modulok. Azonban gyakran vannak olyan részek egy ilyen alkalmazás fejlesztése során, melyeket több helyen használunk fel (pl. Alkalmazás specifikus kivételkezelés). Ezen közös Java file-okat a deploy során érdemes közös JAR-ba (jar-ba) rakni. (Fontos, hogy az EAR Deployment Descriptor-ában kell ezeket feltüntetni, így ezeket az ún. Utility JAR-okat bármely modul használhatja).

OIDGenerator bemutatás

Mivel minden entity bean 1-1 sort reprezentál a perzisztens tárolón (pl. adatbázis), ezért minden egyes sornak egyedinek kell lennie. Ezeknek az egyedi primary key-k kezelését valamely módon meg kell oldani. Az IBM DB2 (sok más adatbáziskezelővel egyetemben ezt automatikusan megoldja, ha új sort szűrünk be az adatbázisba, azonban a WSAD nem. Ennek a kezelését oldja meg ez az EJB JAR modul.

Architekturális felépítés

Az implementáció – a korábbi nagyvonalú terveknek megfelelően – az alábbi jellemzőkkel valósul meg:



A kifejlesztett alkalmazás két fő modulja (a többi modul jelentős részét a WSAD generálta) az alábbi módon helyezkedik el:

- A **TimeSheetClient** modul a HTML, JSP és Struts részeket tartalmazza.
- A **TimeSheetUtility** modul az EJBs, Mapper Objects és JDBC részeket tartalmazza.

Konfigurációs jellemzők

A kifejlesztett alkalmazás az alábbi elemekből épül fel:

- TimeSheet (EAR)
- TimeSheetEJB.jar (EJB module)
- TimeSheetClient.war (Web module)
- TimeSheeUtility.jar (utility JAR)
- OIDGenerator (EJB modul)

A WebSphere alkalmazás szerver az alábbi konfigurációban üzemel (ld. még WAS5.wsi):

- Hostname: localhost
- HTTP port: 9080