

Žízeň

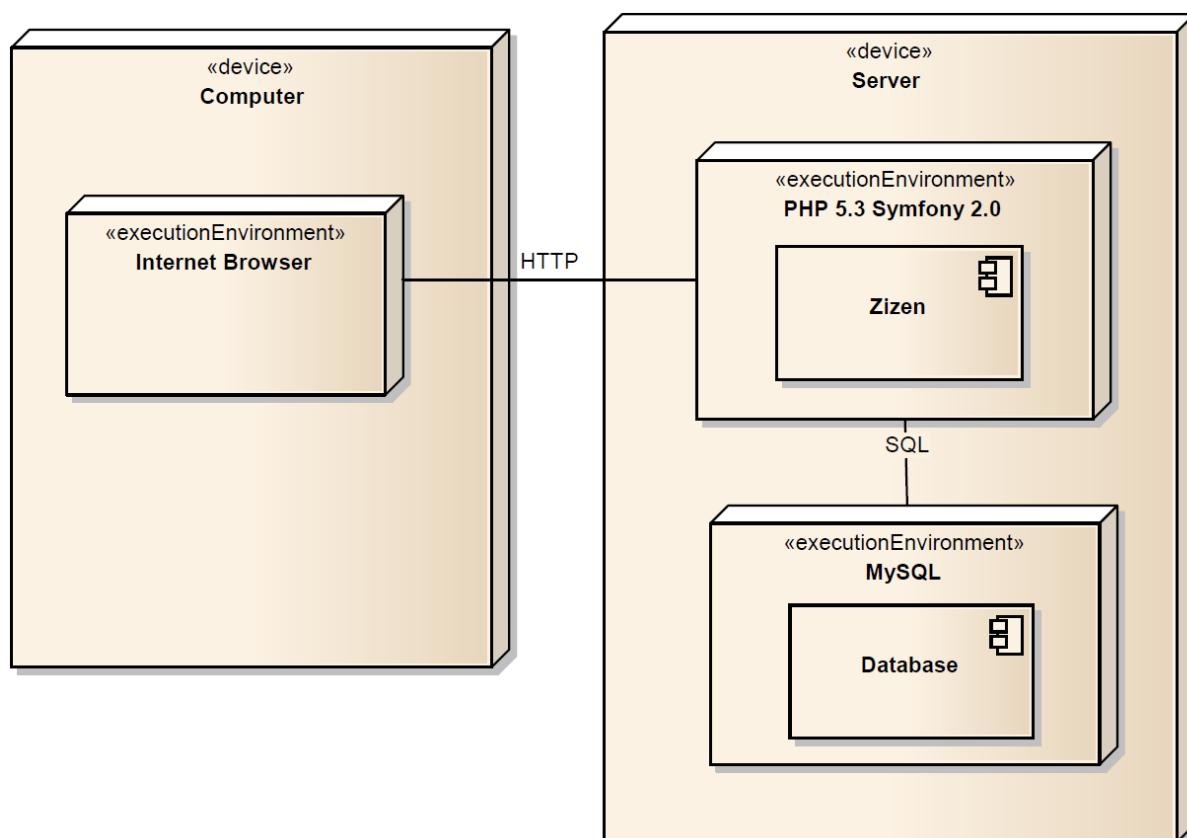
**Semestrální projekt BI-ZSI
Návrhová dokumentace II**

Spolupracovali:

David Tošner
Martin Troup
David Kocík
Miro Hrončok

Model nasazení

Aplikace je navržena jako fyzická architektura server – klient. Klientská část je řešena jako webová aplikace optimalizovaná pro prohlížeče Chrome, Firefox 14.0 a vyšší a Internet Explorer 7.0 a vyšší. Serverová část je rozdělena na dvě vrstvy. První vrstva běží na frameworku Symfony 2.0 v PHP 5.3 a implementuje business vrstvu z návrhového modelu. Tato vrstva komunikuje se stranou klienta a zprostředkovává akce uživatele do druhé, databázové vrstvy. Databázová vrstva je postavena na MySQL a poskytuje data business vrstvě.



Instalace a konfigurace aplikace

Pro studijní účely je výhodné nainstalovat balíček WAMP tento balíček obsahuje potřebné komponenty: MySQL, PHP5.3 a Apache server. Automaticky se s tímto také nainstaluje nástroj pro správu databází phpMyAdmin. Instalační balíček pro windows se dá stáhnout zde: <http://www.wampserver.com/en/>.

Poté je potřeba na server nainstalovat Framework Symfony 2.0. Tento Framework se stáhne na stránce <http://symfony.com/download> a nainstaluje dle instrukcí na stránce. V případě že se vám po zobrazení stránky *localhost/Symfony/web/app_dev.php* zobrazí hláška *You are not allowed to access this file. Check app_dev.php for more information.* Je nutné nastavit vyšší práva tj.

```
cd app
find cache logs -type d -exec chmod 777 {} \;
find cache logs -type f -exec chmod 666 {} \;
```

Poté je potřeba vytvořit Bundle žízeň, ve kterém pracujeme tj příkaz:

```
app/console generate:bundle --namespace=ZSI/ZizenBundle --format=yml
```

Do vytvořeného bundle ve složce *Symfony/src/bundles/ZizenBundle*. Nakopírujeme obsah našeho bundlu.

Vytvoříme databázi pomocí phpMyadmin. Další věcí kterou je potřeba udělat je nastavit připojení k databázi v souboru *app/config/parameters.ini*. Vygenerování tabulek v databázi provedeme pomocí příkazů:

```
app-console doctrine:schema:update --dump-sql
app-console doctrine:schema:update --force
```

Nyní je aplikace připravena k použití.

Spuštění aplikace

Spuštění aplikace se provádí přes webový prohlížeč. Počáteční stránka se zobrazí odkazem *localhost/Symfony/web/app_dev.php/Zizen/index*.

Popis implementace

Celá ukázková aplikace, popřípadě její další rozšíření je programováno v jazyce PHP

5.3 za pomoci Symfony 2.0 frameworku. Pro komunikaci s MySQL databází byl použit framework Doctrine, který již zmíněný framework Symfony 2.0 podporuje.

Celá aplikace je stavěna na architektuře MVC, která je opět součástí již zmíněného frameworku. V rámci ukázkové aplikace byly v adresářovém stromě frameworku vytvořeny tyto soubory:

pro business vrstvu:

/ZizenBundle/Controller/
-FirstPageController.php
-SecurityController.php
-UserController.php

pro datovou vrstvu:

/ZizenBundle/Entity/
-Filters.php
-Location.php
-Pub.php
-Thirst.php
-User.php

pro prezenční vrstvu:

/ZizenBundle/Resources/views
-firstPageLayout.html.twig
-layout.html.twig
/ZizenBundle/Resources/views/FirstPage/
-about.html.twig
-index.html.twig
-logged.html.twig
/ZizenBundle/Resources/views/Form/
-addFavouritePubForm.html.twig
-addLocationForm.html.twig
-createAccountForm.html.twig
-editFiltersForm.html.twig
-editRecordForm.html.twig
-searchFriendForm.html.twig
-selectDefaultLocForm.html.twig
-setThirstForm.html.twig

formuláře:

/ZizenBundle/Resources/views/Form/

-AddFavouritePubForm.php

-AddLocationForm.php

-CreateAccountForm.php

-EditFiltersForm.php

-EditRecordForm.php

-SearchFriendForm.php

-SelectDefaultLocForm.php

-SetThirstForm.php

Při implementaci bylo prioritou dodržet pravidla z návrhového modelu a použít architekturu MVC. Výhodou frameworku byla ulehčená práce zejména z hlediska validace formulářů a manipulace se sessions.

Momentálně se aplikace nachází ve stavu testování a je připravena pro další rozšíření, která by byla pro zavedení potřebná.

Součástí implementace je základní stylování v podobě css souboru. Podrobnější dokumentace k implementační části byla vygenerována pomocí PHP Documentor 2.

Trigger

Jak již bylo zmíněno v předchozích částech dokumentace budeme využívat triggerů. Což nám umožňuje neřešit tyto části přímo v aplikační vrstvě, nýbrž o vrstvu výš.

Zde je uveden příklad triggeru, který bude zajišťovat nalezení žízní.

```
CREATE OR REPLACE FUNCTION filter_sex_by_filter_id(fid IN integer) IS
BEGIN
    RETURN (SELECT filter_sex FROM filters WHERE filter_id = fid);
END
/

CREATE OR REPLACE FUNCTION sex_by_user_id(uid IN integer) IS
BEGIN
    RETURN (SELECT sex FROM user WHERE user.user_id = uid);
END
/

CREATE OR REPLACE FUNCTION age_min_by_filter_id(fid IN integer) IS
BEGIN
    RETURN (SELECT age_min FROM filters WHERE filter_id = fid);
END
/

CREATE OR REPLACE FUNCTION age_max_by_filter_id(fid IN integer) IS
BEGIN
    RETURN (SELECT age_max FROM filters WHERE filter_id = fid);
END
/

-- Podobným způsobem deklarovat další použité funkce:
-- user_follows_user(u1,u2)
-- friend_radius_by_filter_id(fid)
-- other_radius_by_filter_id(fid)
-- geo_by_location_id(lid)
-- geo_delta(g1,g2)

CREATE OR REPLACE TRIGGER find_matching_thirsts_on_insert
AFTER INSERT ON thirst
FOR EACH ROW
TYPE thirst_id_table_type IS TABLE OF thirst.thirst_id%TYPE INDEX BY PLS_INTEGER;
thirst_id_table thirst_id_table_type;
my_id_table thirst_id_table_type;
BEGIN
    --Jen nad aktivní žízní
    IF :new.thirst_status=1 THEN
        SELECT :new.thirst_id, thirst_id INTO thirst_ids_table1 FROM thirst
```

```

WHERE thirst_status=1 -- aktivní
AND user_id <> :new.user_id -- není jeho vlastní
AND (filter_sex_by_filter_id(:new.filter_id) = NULL OR
filter_sex_by_filter_id(:new.filter_id) = sex_by_user_id(user_id)) --sex filtr odpovídá
případnému filtru protistrany
AND (filter_sex_by_filter_id(filter_id) = NULL OR filter_sex_by_filter_id(filter_id) =
sex_by_user_id(:new.user_id)) -- sex filtr protistrany odpovídá případnému filtru
AND age_min_by_filter_id(:new.filter_id) <= age_by_user_id(user_id) -- protistrana
není mladší než limit
AND age_max_by_filter_id(:new.filter_id) >= age_by_user_id(user_id) -- protistrana
není starší než limit
AND age_min_by_filter_id(filter_id) <= age_by_user_id(:new.user_id) -- nejsme
mladší než limit protistrany
AND age_max_by_filter_id(filter_id) >= age_by_user_id(:new.user_id) -- nejsme
starší než limit protistrany
AND (
    (user_follows_user(:new.user_id,user_id) AND
    friend_radius_by_filter_id(:new.filter_id) >=
    geo_delta(geo_by_location_id(:new.location_id),
    geo_by_location_id(location_id))) -- sledujeme ho, tak použijeme friend
    radius
    OR
    (other_radius_by_filter_id(:new.filter_id) >=
    geo_delta(geo_by_location_id(:new.location_id,location_id))) --
    nesledujeme, použijeme other radius
)
AND (
    (user_follows_user(user_id,:new.user_id) AND
    friend_radius_by_filter_id(filter_id) >=
    geo_delta(geo_by_location_id(:new.location_id),
    geo_by_location_id(location_id)))) -- sleduje nás, tak použijeme friend
    radius
    OR
    (other_radius_by_filter_id(filter_id) >=
    geo_delta(geo_by_location_id(:new.location_id,location_id))) -- nesleduje,
    použijeme other radius
); -- END SELECT

FOR idx IN..(SELECT COUNT(*) FROM thirst_ids_table)
    my_id_table(idx):=:new.thirst_id;
END FOR;

FORALL idx IN 1..(SELECT COUNT(*) FROM thirst_ids_table)
    INSERT INTO thirst_match (thirst1,thirst2) VALUES (my_id_table(idx),
    thirst_id_table(idx));

END IF;
END find_matching_thirsts_on_insert;
/

```

Připomínky k ukázkové implementaci

Ukázková aplikace je implementována s omezenou funkcí. Již vytvořené funkce vyžadují řadu dalších rozšíření, která by byla pro reálné zavedení nutná implementovat.

Z pohledu uživatele by bylo nezbytné aplikaci implementovat intuitivněji, pokud by měla mít reálnou šanci na úspěch v praxi. K realizaci této připomínky by však stačilo lépe navrhnout jednotlivé funkce v prezenční vrstvě aplikace.

Pro zavedení by bylo také třeba aplikaci rozšířit o verzi pro mobilní telefony a reálně pak pracovat s daty GPS lokací.

Další možností by bylo přihlašování přes další sociální sítě jako je Facebook, či Twitter.

Závěr naší práce

Předmět ZSI byl pro nás přínosem zejména z hlediska problematiky analýzy systému. Tvoření jednotlivých diagramů nás postupně vedlo k zvážení jednotlivých funkcí naší aplikace.

Snažili jsme se na tvorbu systému pohlížet z obou stran, jak strany klienta, tak strany analytika/programátora. V postupném vývoji došlo k mnohým změnám, které vedly až k finálnímu návrhovému modelu, který byl z obou stran uspokojivý.

Rádi bychom podotkli, že výsledek naší práce je především podklad pro rozšířenou implementaci, která by vedla k finálnímu vytvoření aplikace a jejímu zavedení do praxe. Naše ukázková aplikace, ač není kompletní a plně funkční, je založena na pevném základu, z kterého lze v budoucnu vycházet.