

XIII. Erdélyi Tudományos Diákköri Konferencia
Kolozsvár,
2010. május 14–16.

Kódkövetés az oktatásban

TDK dolgozat

Biró Tamás

Sapientia – Erdélyi Magyar Tudományegyetem
Műszaki- és Humántudományok kar, Marosvásárhely

Informatika szak

Témavezető tanár: Fogarasi Ferenc

Sapientia – Erdélyi Magyar Tudományegyetem
Műszaki- és Humántudományok kar, Marosvásárhely
Matematika-informatika tanszék

Kivonat:

A dolgozat célja egy olyan osztott rendszer tervezése és implementálása, amely a programozás elsajátítását könnyíti meg, kezdő programozók számára, de nem csak.

A kiépítendő rendszer egyik nagy előnye, hogy online szolgáltatásként érhető el, anélkül, hogy a felhasználó bármit is kelljen telepítsen a számítógépre. A felhasználókról begyűjtött egyéni és csoportos statisztikai adatok, mint a figyelemkövetés, gyakori hibák, kódolási szokások, gépelési sebesség, stb. felgyorsítják az oktatók reakcióképességét, ezáltal növelve az oktatás minőségét és csökkentve a tananyag elsajátításának idejét.

A fejlesztői környezet a diákok számára könnyen kezelhető flash alapú felületet biztosít, de megenged számos előrehaladott beállítási lehetőséget is, mint például a fordítók paraméterezése. A rendszer ezen túl biztosítja a csapatmunkát a projektek megosztási lehetőségével.

Tartalomjegyzék

Kivonat:.....	2
Tartalomjegyzék.....	3
1. Bevezetés	4
1.1. Számítógépes alkalmazások programozás tanítására.....	4
1.2. Előrehaladott oktatási lehetőség.....	5
2. Online- és oktatóval való tanulás egyszerre.....	5
3. Kitűzött célok.....	6
3.1. Előzmények/ Hasonló alkalmazások.....	6
3.2. Mivel adunk többet az eddig jelenlevő alkalmazásoknál?.....	7
4. BrowseIDE – kódolj bárhol	8
4.1. Böngészéstől tanulásig	8
4.2. Architektúra.....	9
4.2.1. Rendszer-modell	9
4.2.2. Kliens réteg	9
4.2.3. Szerver réteg	12
4.2.4. Adatbázis.....	13
4.2.5. Fordító réteg.....	15
4.3. Folyamatok, avagy az alkalmazás mechanizmusa	15
5. Következtetések	20
6. További célok.....	20
7. Könyvészet.....	20

1. Bevezetés

Manapság az oktatást nem tekinthetjük úgy, mint az előző századokban, ahol iskolában, könyvek által, esetleg tanárok által sajátították el a „tanulók” a tudásukat, hanem ezekhez hozzá kell rendelnünk az öntanítást is az egyre gyorsabban fejlődő számítógépes világ által nyújtotta lehetőségeket kihasználva. Ezentúl azokat a személyeket, akik tanulás elsajátításának szempontjából használják a számítógép és internet által nyújtotta információkat, e-tanulóknak fogom nevezni.

Nagy hangsúlyt kell fektetni olyan szolgáltatások kialakítására, melyek elősegítik az e-tanulók könnyebb fejlődését, és meggyőzni minél több embert, hogy alkalmazzák ezeket. Feltehető a kérdés, hogy milyen feltételeknek kell megfelelnie egy ilyen szolgáltatásnak/alkalmazásnak. Először is vizsgálnunk kell a célcsoportot, és ezeknek szokásait illetve a cél-kategóriát (miben is akarunk segíteni?). Miután kitűztük, hogy kinek akarunk mi egy alkalmazást tervezni, ezt a lehető legegyszerűbb és a felhasználó számára legkönnyebben kezelhető, de mégis az adott területen minden szükségletét kielégítő megoldást kell találnunk.

1.1. Számítógépes alkalmazások programozás tanítására

A fent említett feltevés egy nagyon tág körű kérdést vet fel, ám mi egy jól meghatározott területet szeretnénk alaposabban megvizsgálni. Mivel mi egy olyan csoportot vizsgálunk (programozók), akik a számítógépes világban akarnak előrehaladni – sőt ezt akarják fejleszteni -, így könnyebb dolgunk van. Feltehetjük a kérdést, hogy mire lenne szüksége egy olyan e-tanulónak, aki programozni akar tanulni? Nagyon sok lehetőséget kell, hogy kiszolgáljon egy olyan alkalmazás, amire a kezdő programozók rábízhatják magukat. Néhány fontos dolog, amit egy ilyen jellegű alkalmazásnak tudnia kell:

- Kényelmes programozói felület
- Könnyű projekt-kezelés
- Nyelv-függő kód-színezés
- Szintaxis kiegészítő
- Csapatmunka lehetőség
- Lehető legtöbb segítség nyújtása
- Egyszerű design

- Előrehaladott beállítási lehetőségek
- Testre szabás

Sajnos nem lehet egy olyan alkalmazást készíteni, amely mindenkinek tetszik, de arra kell törekedni, hogy minél több e-tanuló szükségletét kielégítse a könnyebb tanulás érdekében.

1.2. Előrehaladott oktatási lehetőség

Egy kellő alkalmazás nemcsak a programozó illetve e-tanuló dolgát könnyíti meg, hanem az oktató munkáját is egyszerűsíti a kellő környezetben. Már szó esett, hogy a számítógép és internet segítségével egy tanuló sokkal könnyebben elsajátítja a kellő tudást, ám nem kell megfélekedni az ezredek óta alkalmazott, oktatóval való tanulást. Két nagy lehetőség áll hát előttünk: oktató segítségével való tanulás és egyéni „e-tanulás”. Feltevődik a kérdés, mi lenne, ha a kettőt összefűznénk?

A továbbiakban egy olyan alkalmazást akarunk bemutatni, mely kimeríti e két lehetőség együttes alkalmazását, így egy új – nagyon hatékony – tanulási lehetőséget vezetvén be.

2. Online- és oktatóval való tanulás egyszerre

Egy olyan alkalmazást illetve szolgáltatást építünk ki, melyben a programozók 100% figyelmet kapnak az alkalmazás által, ami közvetíti a tanárnak/csoportvezetőnek is a fellépő hibákat, így felgyorsítva az oktatási folyamatot. A kialakított rendszer teljes kimerítéséhez szükség van három alapvető közreműködőre:

- Programozó
- Általunk szolgáltatott alkalmazás
- Oktató/csoportvezető

Az elképzelés a következő:

- az oktató elmagyarázza a programozó számára a kellő információkat egy illető program megírásához, esetleg leoszt a programozónak egy sablonprojektet
- a programozó létrehoz/felhasznál egy projektet az általunk szolgáltatott keretrendszer szerkesztőjében, és megírja benne a kívánt program forráskódját, majd kompiláltatja¹ a megírt kódot és kéri az eredményezett programot; a szerkesztő mindez idő alatt elemzi a programozó szokásait² és a kompiláláskor esetleges fellépő hibákat és figyelmeztetéseket

¹ Lefordíttatja a forráskódot

² Forráskód bővülésének sebessége és folyamatossága, belépett idő kihasználtsága, másolási kísérletek stb.

vagy a kész programot a programozó számára visszatéríti, az oktatónak viszont statisztikákat számol belőle

- az oktató következtetéseket von le a statisztikai adatokból és jó irányba tereli az e-tanulókat.

3. Kitűzött célok

Egy olyan online szolgáltatást biztosítunk a tanulási intézményeknek (de nem csak), melyben a programozást tanulók gyorsabb tudáselsajátítását biztosítjuk. A keretrendszerünk rendelkezik egy IDE³-vel és egy weboldallal, mely projekt- és csoport-management mellet még jó néhány szolgáltatást biztosít, ahonnan az oktatók számára tervezett statisztikai adatok is elérhetőek és kezelhetőek.

Online szolgáltatásként érhető el, melyet bárki igénybe vehet, egy egyszerű feliratkozás szükségeltetik. Felhasználó oldalon böngészőben, flash-lejátszó⁴ segítségével futó alkalmazásról beszélhetünk, mely egy osztott rendszer azon részét képezi, mellyel a felhasználók közvetlen úton kommunikálnak. Előnye, hogy semmiféle további programot nem kell a felhasználónak telepítenie, a forráskód fordításáért is a szerver⁵ a felelős.

3.1. Előzmények/ Hasonló alkalmazások

Már léteznek részben hasonló szolgáltatások, ám egyik sem érint ekkora területet, mint az általunk fejlesztett rendszer. Amiket gyors rákereséssel találtunk:

- ideone⁶
 - Bemásolhatunk/szerkeszthetünk egyszerű forráskódokat
 - Van szintaxis színezője
 - Sok programozási nyelvet ismer (40 fölött)
- heroku⁷
 - ruby⁸ nyelvben való kódolásra használható

³ IDE - Integrated Development Environment – Integrált fejlesztői környezet

⁴ Flash-player: <http://www.adobe.com/products/flashplayer/>

⁵ Az alkalmazásunk azon része, mely az irányításért felelős

⁶ <http://ideone.com/>

⁷ <http://heroku.com/>

⁸ Ruby: <http://www.ruby-lang.org/>

- ecco⁹
 - Sok mindent ígérő a miénkhez nagyon hasonló alkalmazás, ám úgy tűnik, le van állva a fejlesztése egy kezdetleges állapotnál, és nem érhető el, hogy kipróbálhassuk
- phpanywhere¹⁰
 - Php fejlesztőknek szánt online szerkesztő és kezelő
- amyeditor¹¹
 - Nagyon jó online IDE
 - Csoportos szerkeszthetőségi lehetőség
 - Van szintaxis színezője
 - Nem lehet vele kompilálni
- coderun¹²
 - Gyors és szép felülettel rendelkezik
 - C#, Java, ASP, PHP nyelveket ismeri
 - Online debug¹³-móddal is rendelkezik
 - Nagyon hasonlít a Microsoft Visual Studiora¹⁴
 - IntelliSense szerű segítővel is rendelkezik

A fent említett alkalmazások közül némelyikük előrehaladottabb megoldásokat alkalmaz a projektek kezelésére és fordítására, esetleg gyorsaság szempontjából is előnyösebbek, ám egyikük sem előnyös tanulási szempontokból.

3.2. Mivel adunk többet az eddig jelenlevő alkalmazásoknál?

A mi alkalmazásunk nem csupán egy online, böngészőből elérhető programozási felület, hanem ennél sokkal tágabb teret fed le. A fent említett alkalmazásoktól eltérően, a hangsúly nem a mesterszintű együttműködésen van a lényeg, hanem a kezdő programozók számára otthonos környezetet biztosítani, emellett a tanárok figyelmét a problémás területek fele irányítani.

Nem csupán azért terveztük egy online alkalmazásnak, hogy semmit se kellejen telepítsen egy felhasználó a számítógépére, hanem így számos új lehetőség nyílt, mint például a központi adatbázis, ahonnan nagyon könnyen és kényelmesen szűrhetünk le statisztikai adatokat, emellett

⁹ <http://ecco.sourceforge.net/>

¹⁰ <http://phpanywhere.net/>

¹¹ <http://www.amyeditor.com/>

¹² <http://www.coderun.com/>

¹³ Debug - hibakeresés

¹⁴ <http://msdn.microsoft.com/en-us/vstudio/default.aspx>

lehetőség van a projekteken való együttműködésre is, ha nem is a legoptimálisabb módszerrel. A legfőbb tulajdonsága viszont az, hogy nagyon könnyen, gyorsan és hatékonyan bővíthetjük, fejleszthetjük és javíthatjuk az alkalmazásunkat, és mindezt a „felhasználók” észrevétele nélkül. Kezdetleges elképzelésünket az e-tanulók visszajelzésével is szándékozunk bővíteni.

4. BrowseIDE – kódolj bárhol

Az alkalmazásunk neve, mint ez az utóbbi címből is kiderült: BrowseIDE. A név az alkalmazást felhasználó szempontból nézett két alapvető funkciójából kapta: az eleje „Browse” az angol böngésző – browser – csonkított alakjából és a második részét ugyancsak az angol nyelvben használt rövidítésből – IDE (*Integrated Development Environment*) - kapta, mely a magyar Integrált Fejlesztői Környezetnek felel meg. A mottónk „code anywhere”, avagy „kódolj bárhol” a felhasználó számára a szabadságot sugallja, és nem pedig az „ülj le és tanulj” érzést, ezzel is kedvet hozván a tanulóknak, hogy akár otthonról, akár iskolából, akár akárhonnan fejleszthetik programjaikat és ezzel együtt persze a tudásukat is.

A dolgozat további részében az alkalmazásunk részletes felépítését és működését mutatjuk be, vagyis a technikai hátterét ezen osztott rendszernek.

4.1. Böngészéstől tanulásig

Egy böngésző segítségével érjük el az alkalmazásunkat. A cím megadása után egy flash-ben készült weboldalhoz érünk, mely röviden leírja milyen szolgáltatásokat kapnak a feliratkozott személyek. Feliratkozás, majd bejelentkezés után egy menü áll rendelkezésünkre, melynek segítségével hozzáférünk a szolgáltatásokhoz. Itt érhetjük el a programozói felületet, melyben létrehozhatunk új C,C++ valamint Java projekteket, szerkeszthetjük a már meglévőket, és kérhetjük a forráskódok kompilálását. A felhasználói környezet szoros kapcsolatban áll a szerverrel, mely irányítja az összes többi komponenst. A szerkesztő teljes futási ideje alatt információkat gyűjt a benne történő eseményekről, és közli ezt a szerverrel, mely lementi az adatokat a központi adatbázisba. A későbbiekben egy tanár vagy annak a csoportnak a vezetője, melyhez az illető felhasználó is tartozik, ugyancsak a weboldalról el tudja érni ezen adatokból generált statisztikát, melynek tudatában jó irányban tereli a diákokat.

4.2. Architektúra

4.2.1. Rendszer-modell

Az alkalmazás négy fő rétegből áll, melyek szoros kapcsolatban vannak egymással, mindenik egy fontos szerepet betöltve (1. ábra).



1. ábra – Alkalmazás rétegei

A böngésző, illetve a böngészőben futó flash alapú felület biztosítja a felhasználók számára a kezelési felületet, ám ezt felhasználótól függően a szerver irányít. A felület lekéri a szervertől a kívánt adatokat, a szerver feldolgozza a kérést és eldönti, milyen műveletet kell végrehajtson, ami lehet adatbázisból való lekérdezés illetve beszúrás, melynek eredményét visszatéríti a felületnek. A kérés lehet kompilálási kérés is, ilyenkor a szerver kiveszi az adatbázisból az illető projekt tartalmát és átadja a megfelelő kompilátor-szervernek (windows/linux/mac), az kiírja egy ideiglenes könyvtárba a tartalmat majd megpróbálja lefordítani a forráskódokat, a kapott alkalmazást vagy a fordítási hibákat viszont visszatéríti a szervernek, ami beírja az adatbázisba és továbbítja a felületnek is.

4.2.2. Kliens réteg

A kliens réteget képezi mindaz, amit a felhasználó lát, illetve annak a megjelenítési illetve kommunikációt megvalósító háttérét. Böngészőben fut flash lejátszó segítségével.

A technológia, melyre alapszik, az Adobe által biztosított flex SDK¹⁵, a fejlesztést ugyancsak Adobe által szolgáltatott FlexBuilder3¹⁶, melyben kényelmesen lehet felépíteni a flash alapú alkalmazásokat. A programozási nyelv, melyben írva van az mxml¹⁷ és actionscript3¹⁸.

Két modulból áll: a weboldalból és a fejlesztői környezetből, mindkettő kellemes környezetet biztosítva, kényelmesen elérhető funkciókkal.



2. ábra – Weboldal főoldala, bejelentkezett felhasználóval

A weboldal a gyűjtő felület, vagyis innen érjük el az alkalmazás összes szerepét. A felhasználóneveinkre kattintva módosíthatjuk a felhasználói adatainkat, létrehozhatunk

¹⁵ Flex SDK: <http://opensource.adobe.com/wiki/display/flexsdk/>

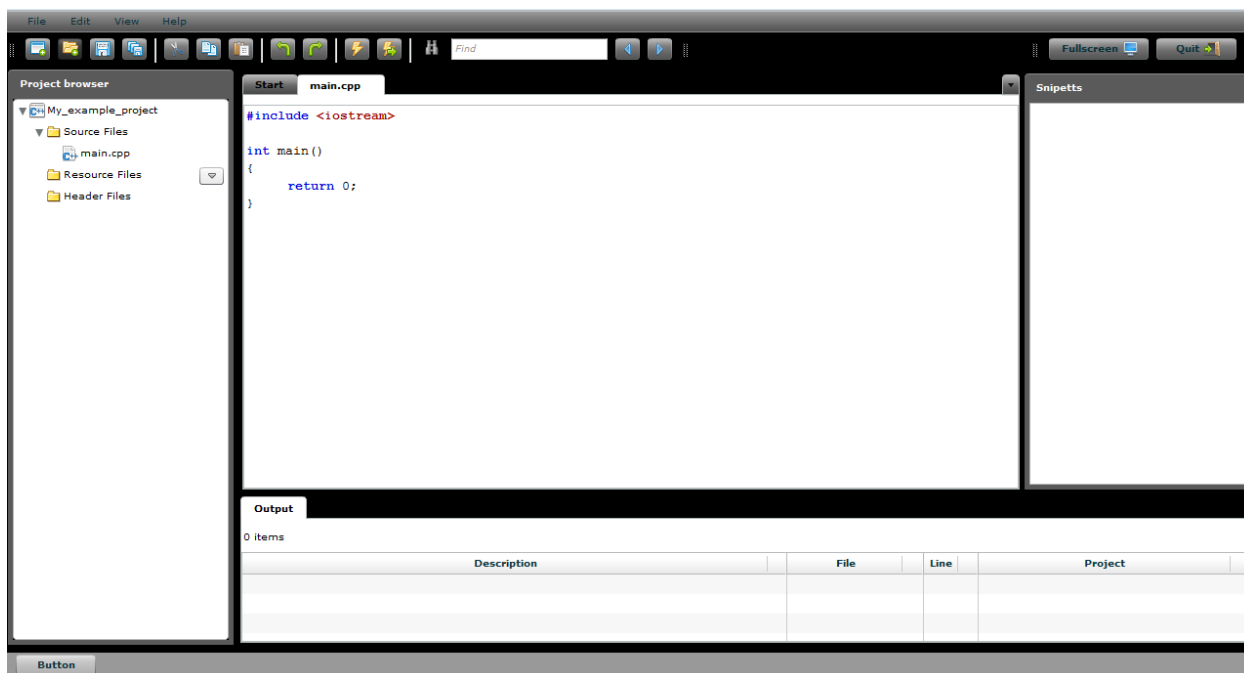
¹⁶ Flex Builder: <http://www.adobe.com/products/flex/>

¹⁷ MXML: <http://mxml.sourceforge.net/>

¹⁸ Actionsript: <http://www.adobe.com/devnet/actionsript/>

csoportokat vagy az eddigiekhez hívhatunk embereket (kellő jog rendelkezése esetén), illetve ismerősöket vehetünk fel a barátlistánkba, akikkel később megoszthatjuk projektjeinket.

A beállítások pont alatt viszont a felület személyre szabására alkalmas beállítási lehetőségeket találjuk meg. A statisztikák (Statistics) menüpont leginkább az egyes csoportok vezetőinek van kitalálva, ahol eléri egy kiválasztott csoport/csoporton belüli illető statisztikai adatait. Azon személyek számára kik csak sima felhasználók, csak a saját statisztikai adataikat tudják követni. Az utolsó menüpont segítségével kijelentkezhetünk, így biztosíthatjuk, hogy egy osztott gépen az utánunk jövő biztosra ne tudjon bejelentkezni a mi nevünkben.



3. ábra - Szerkesztő

A kódolás (Start coding) menüpont alatt a szerkesztőt érjük el, ami egy hagyományos IDE lehetőségeit nyújtja. Hét különböző csoportot lehet megfigyelni:

- Menüsor – beállításokat és irányítási lehetőségeket találunk meg benne, mint a File, Edit, View, Help és ezekből nyíló számos al-menü
- Gombsor – gyors hozzáférést biztosítanak a gyakran használt funkciókhoz, mint pl. új projekt létrehozása, új állomány létrehozása, mentés, visszalép, előrelép, fordítás (fő projekt fordítása), keresés.
- Projektkezelő – a projektek és azok tartalmát találhatjuk itt, valamint szerkeszthetjük is őket a kurzor fölé vitelével megjelenő menügomb lenyitásával, valamint az állományokat megnyithatjuk a szövegszerkesztőben.
- Szövegszerkesztő – itt szerkeszthetjük az állományok tartalmát; fülek segítségével több megnyitott szövegszerkesztőt választhatunk el, mindenik a forráskód típusa szerint színezi a benne található kódot.

- Kimenetel – fordítási eredményt illetve felhasználó által indítványozott szerverkommunikáció eredményét mutatja.
- Kódtöredékek (Snippets) – a felhasználó által meghatározott kódtöredékek beszúrását teszi lehetővé, mely a programozásban sokat segít.
- Állapotsor – gyors tájékoztatásra szolgál (pl. épp kommunikáció történik a szerverrel, és ennek jellege).

Attól az időponttól, hogy elindítjuk a szerkesztőt, ez percente közli a szerverrel a háttérben begyűjtött adatokat kódolási szokásainkról mindaddig, míg kilépünk vagy megszakad a kapcsolat. Ezek a jelentések mellet fordításkor és mentéskor is feljegyzi a szerver a kellő ismereteket.

A pontos lista, amiket figyelünk egy felhasználóról, ám idővel ez könnyen bővíthető:

- Mennyi ideig van fókuszbán az ablak (figyelemkövetés)
- Milyen gyorsan kódol
- Milyen gyorsan bővül a kód
- Mennyire folyamatosan bővül a kód
- Másolások (másolás/beillesztés) száma és mérete
- Hányszor nyomja le a törlés gombot (kódjavítás)
- Mikor menti el a kódot
- Mikor kompilál
- Milyen kompilálási eredményei vannak (hiba esetén kategorizálva)

A fent említett felmérések jelentését és minden más kommunikációt a felület és a szerver között amfPHP¹⁹ segítségével valósítjuk meg - ami Json²⁰ szerializációt²¹ alkalmaz -, mely gyorsan és hatékonyan továbbítja az üzeneteket.

A felület kinézetét minden felhasználó saját tetszésére testre szabhatja (háttér, színek, megjelenített panelek), ez is elősegítve a kényelmet és szabadságot.

4.2.3. Szerver réteg

A szerverünk egy apache szerverből, 5.3-as verziószámú php-szerverből és amfPHP-ből áll. A szerveret tekinthetjük egy forgalomirányító rétegnek is, ami felelős minden kérését és kapcsolatot teremt a kliens réteg, adatbázis és fordító réteg között. RPC²²-hez hasonlóan történik a kommunikáció, amit az amfPHP old meg. Nekünk a felületről csak a gateway.php-val kell

¹⁹ amfPHP – action message format PHP: <http://amfphp.org/>

²⁰ Json – JavaScript object notation: <http://www.json.org/>

²¹ Szerializáció – objektumok átalakítása byte sorrá, hogy könnyen lehessen akár hálózatban is átküldeni

²² RPC – Remote Procedure Call – távoli függvényhívás

kommunikálnunk, és neki kell átadni a kívánt osztály nevét és a benne levő metódus paramétereivel. Aszinkron módon történik az üzenetcsere: a kliens létrehoz egy kapcsolatot a szerverrel, majd ezen keresztül hívja meg ennek a függvényeit, amire egy figyelőt helyez és folytatja a többi feladatát. Amikor a szerver válaszol, egy eseményt vált ki, így a felületen elkapjuk ezt és lekezeljük a válasz értelmében teendőket. Minden kérésnek meghatározhatunk külön kezelőt, ám ez nem kötelező. Nekünk csak az osztályokat kell létrehoznunk és a benne levő metódusokat, a többit automatikusan végzi az amfPHP. Főbb feladatok, melyeket kezelnie kell:

- Felhasználó azonosítás/létrehozás/módosítás
- Projekt és fájl létrehozás/módosítás
- Beállítások módosítása
- Csoportok kezelése
- Kompilálás leosztás
- Hibakezelés
- Adatbázis kapcsolat fenntartása

4.2.4. Adatbázis

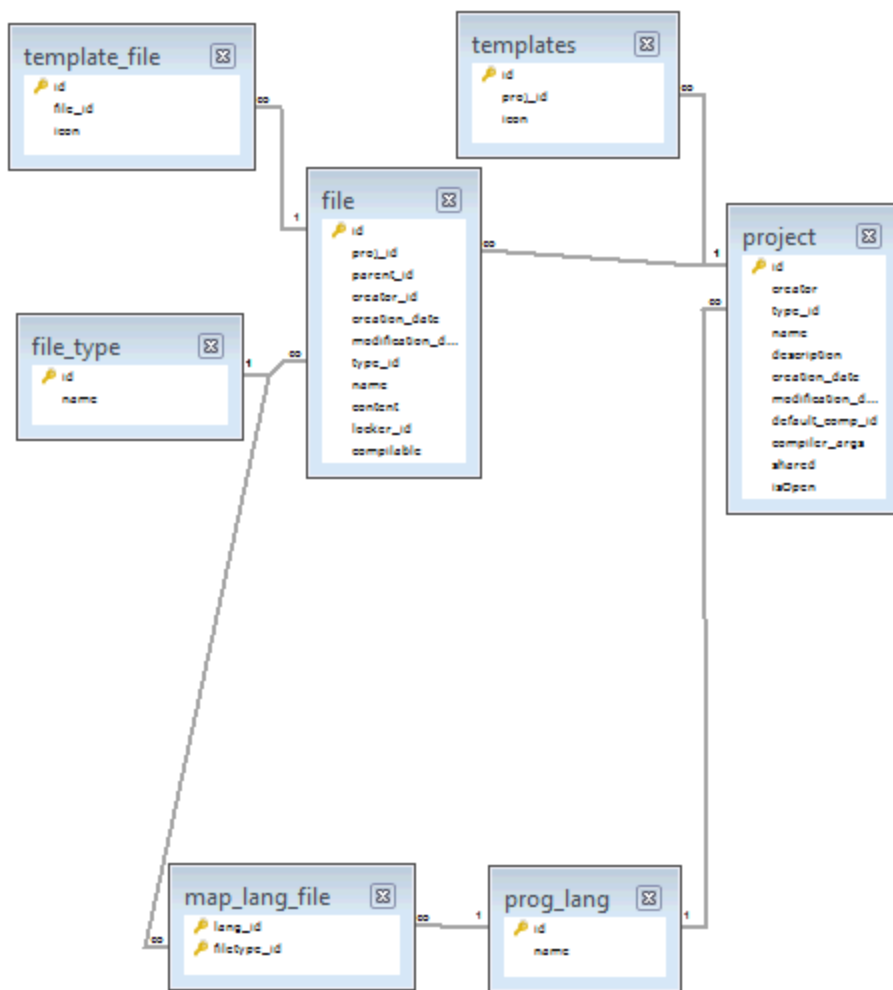
Az adatbázis szerepét egy MySQL adatbázis tölti be, mellyel a php kommunikál közvetlenül. Itt van tárolva az összes adat, úgy a működési beállítások, mint a felhasználók által szerkesztett tartalmak.

Az adatbázis 21 táblát tartalmaz, ha kategóriákba soroljuk akkor 7 csoportot tudunk elkülöníteni:

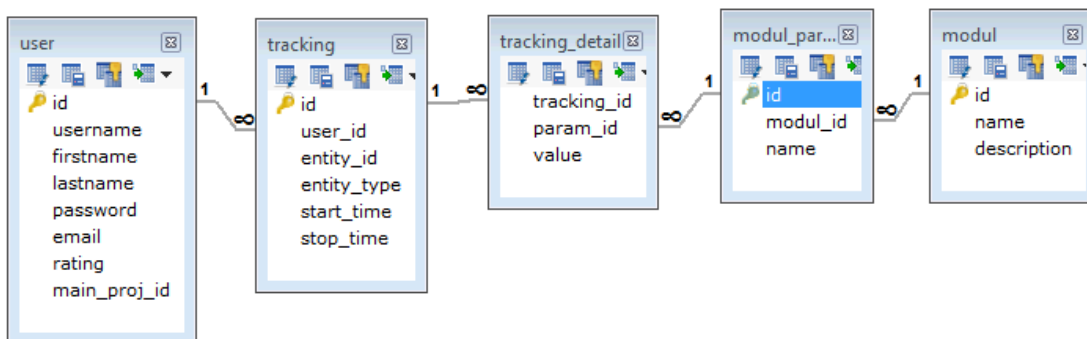
- Felhasználó és beállításai
- Projektek és sablon projektek
- Állományok és sablonok
- Programozási nyelvek és kompilátorok²³
- Csoportok
- Felhasználó-figyelés
- Naplózás

A projektek, programozási nyelvek és fájlok közötti kapcsolatokat a 4. ábrán láthatjuk. A felhasználó a követő táblákkal való kapcsolata az 5. ábrán látható, ahol a USER a felhasználó, a MODUL az, amit figyelünk, a MODUL_PARAMS egy bizonyos figyelt eseménynek a paraméterlistája, a TRACKING tartalmazza az effektív méréseket, mely biztosítja a kapcsolatot a felhasználóval is.

²³ Fordító programok



4. ábra – Projekt és fájlok kapcsolata az adatbázisban



5. ábra – Felhasználó és a követő kapcsolata az adatbázisban

4.2.5. Fordító réteg

A fordítási rétegben találkozunk két új fogalommal: MinGW²⁴ (Minimalist GNU for Windows), Perl²⁵. Az effektív fordítást a MinGW segítségével valósítjuk meg. A szerver aszerint, hogy milyen platform alapú kódot akarunk fordítani (windows/linux) átadja a kiválasztott kompiláló szervernek a fordítandó projektet, ez kiírja egy ideiglenes könyvtárba a projekt tartalmát, majd meghív egy Perl szkriptet, mely előkészíti a projekt fordítását. Végül meghívódik nyelv szerint az a fordító, amilyen kódot épp fordítani akarunk, mely lekompilálja a forrásállományokat és az eredményt visszaírja az adatbázisba. Abban az esetben, ha csak kompilálási kérés volt és nem eredmény-program visszakérése is, akkor az adatbázisba csak a fordítás által visszaadott üzenetek kerülnek vissza az adatbázisba (siker/hibák), illetve a felhasználó tudomására, viszont ha az eredményprogram is kérve volt, ha sikerül a kompilálás, akkor ez is beíródik az adatbázisba, innen majd a szerver továbbküldi a felhasználónak.

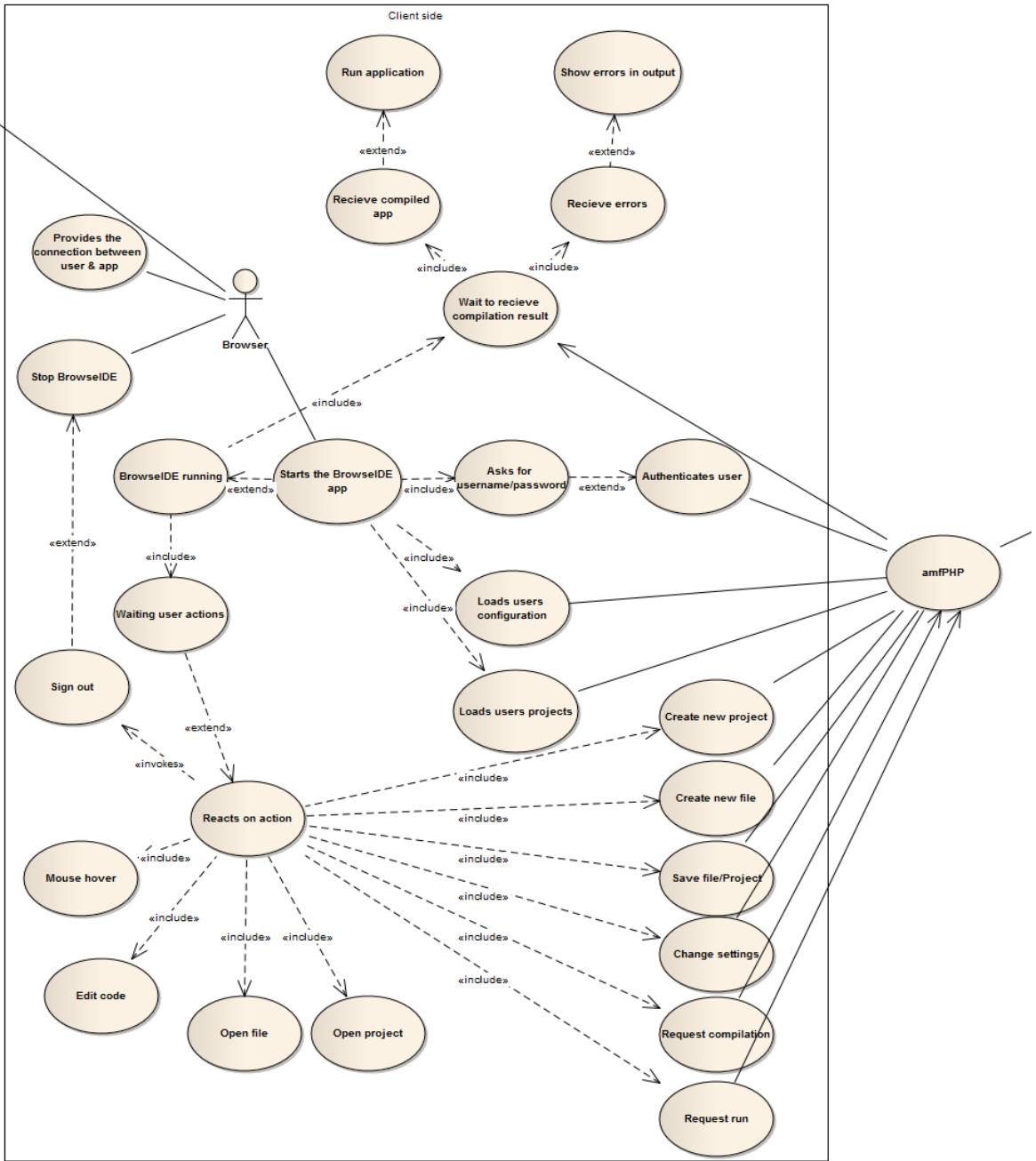
4.3. Folyamatok, avagy az alkalmazás mechanizmusa

Az alkalmazás rétegei között üzenetcserevel kommunikálnak. Ennek egyetlen hátránya, hogy folytonos internet kapcsolatban kell legyenek. A kérések mindig a felhasználó irányából jönnek, ezekre a válaszok ennek irányába haladnak. A felülettől jön egy kérés, ez lehet adatbázisba való bejegyzés, adatbázisból való lekérdezés vagy kompilálási kérés. Ez utóbbi esetben nem csak a felület, szerver és adatbázis között folyik a kommunikáció, hanem bejön a fordító réteg is, amit egy másik szerveren találunk, eszerint, hogy milyen platformra akarunk fordíttatni (windows,linux).

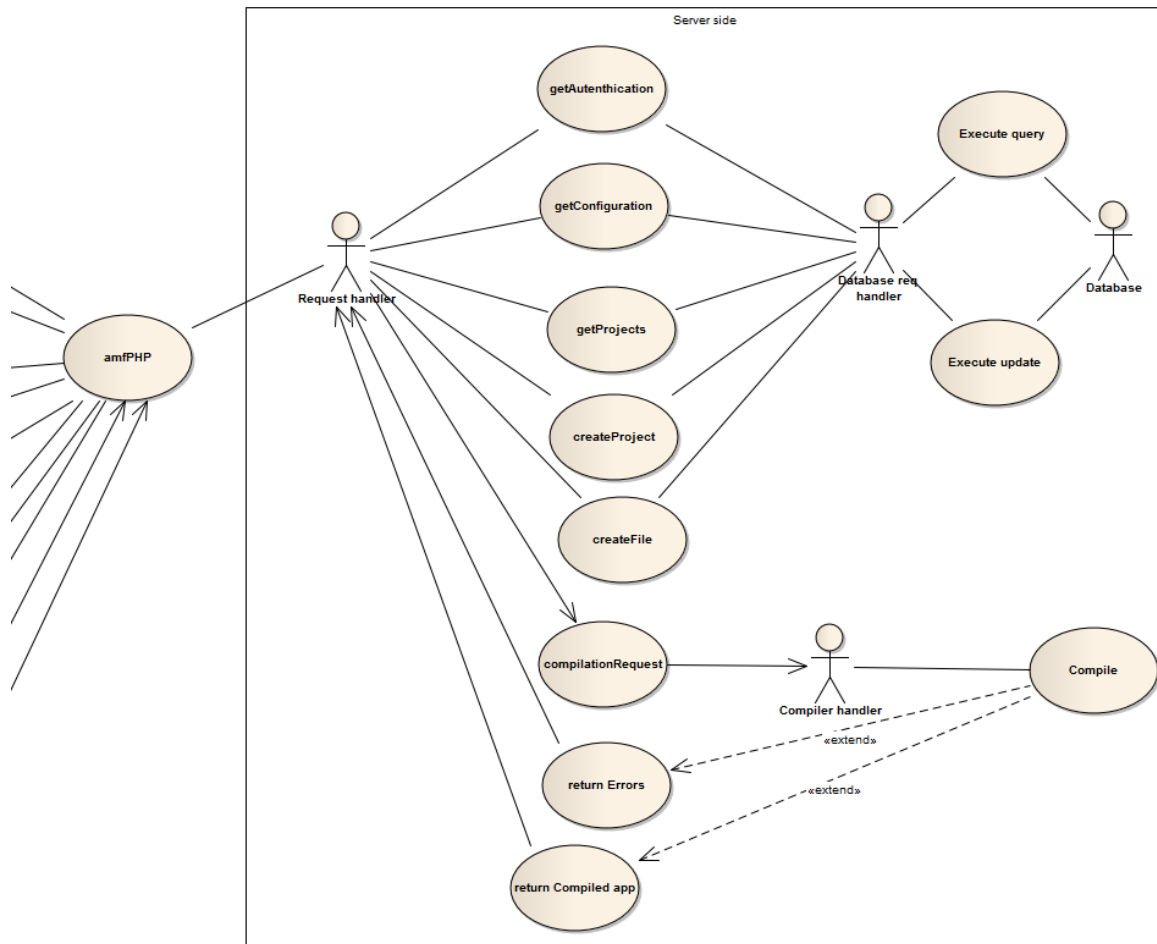
A következőkben az alkalmazás használati eset diagramját és az adatkapcsolati diagramját láthatjuk.

²⁴ MinGW – <http://mingw.org>

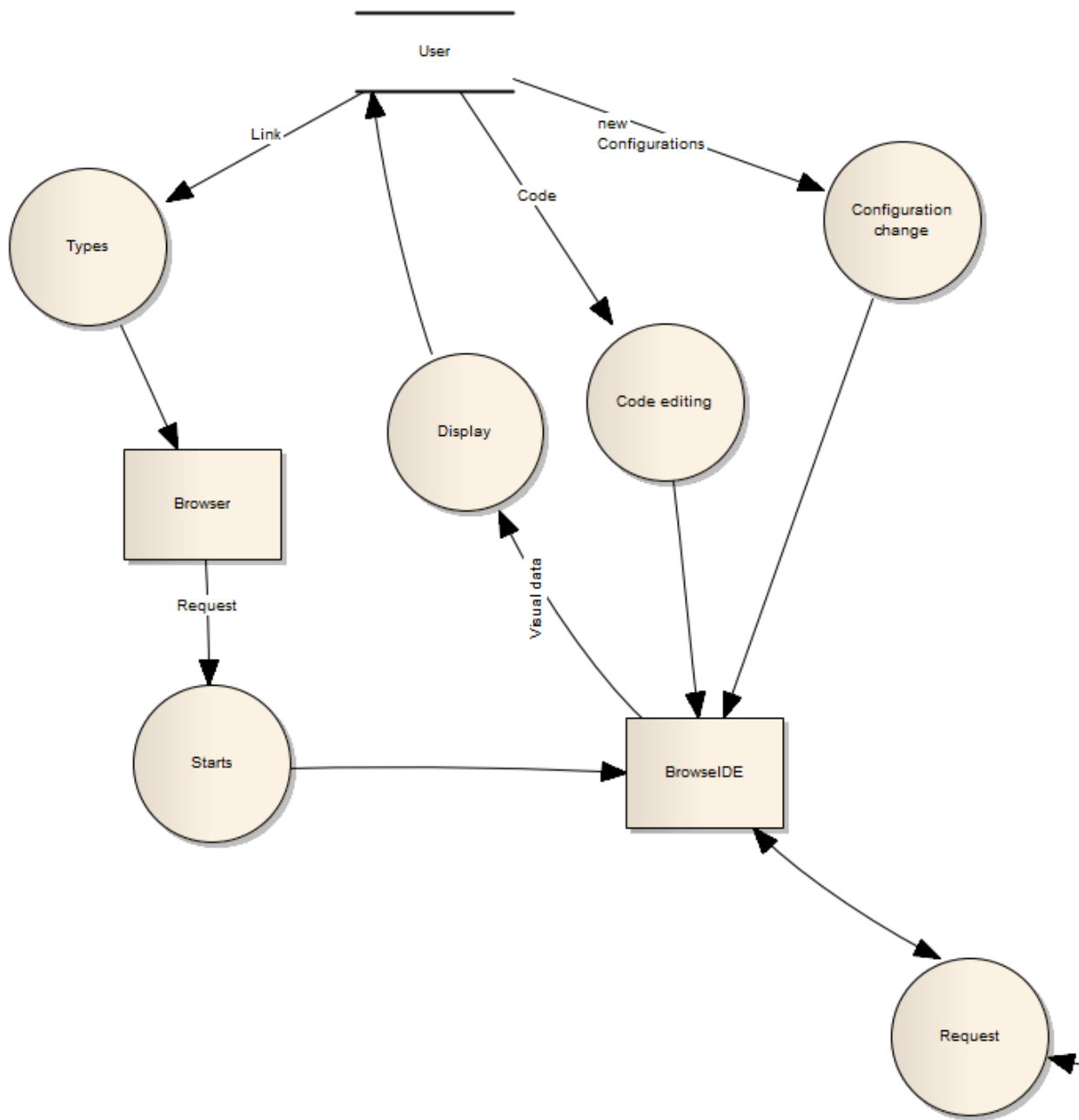
²⁵ Perl programozási nyelv - <http://www.perl.org/>



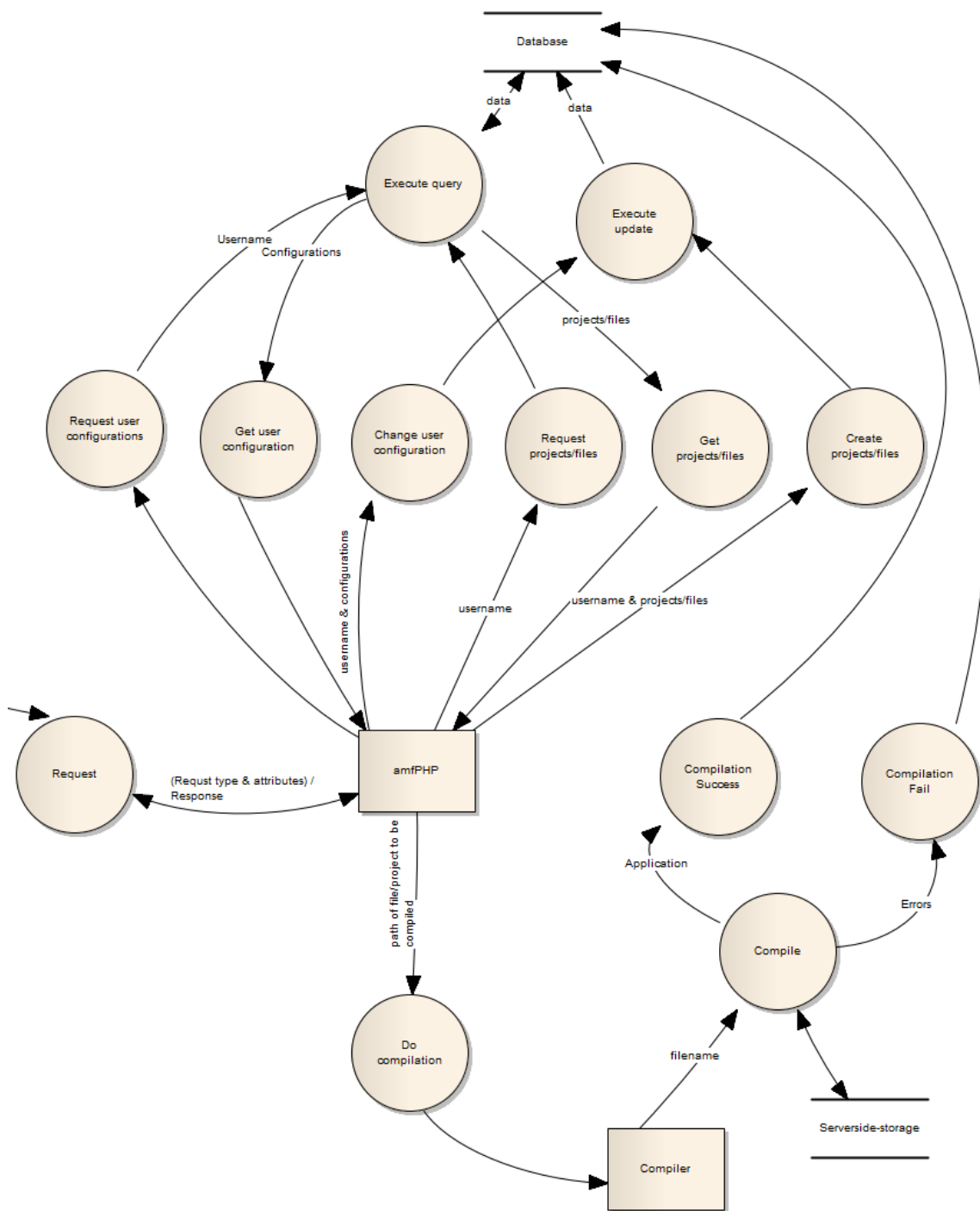
6. ábra – Az alkalmazás használat-eseti diagramjának bal oldala



7. ábra – Az alkalmazás használat-eseti diagramjának jobb oldala



8. ábra – Az alkalmazás adatfolyam diagramjának bal oldala



9. ábra – Az alkalmazás adatfolyam diagramjának jobb oldala

5. Következtetések

A számítógépes tanítási módszerek egyre jobban kielégítik az emberek elvárásait, emiatt gyorsabban lehet komoly szakembereket képezni. Ezen alkalmazás segítségével a diákok, de egyúttal a tanárok fejlődése is elősegíthető.

Egyelőre méréseket még nem sikerült begyűjteni, de pár hónapos teszteléssel ki lehet alakítani egy átlagot és lehet majd kérni a tanulóktól egy adott szint elérését.

6. További célok

További céljaink a rendszer tökéletesítése olyan szintre, hogy professzionális környezetben is alkalmazható lehessen. Bevezetni több nyelv ismeretét es kód kibővítő funkció implementálása, emellett diákok tesztelése és átlagok felállítása volna a cél. Több ember bevonásával a projektbe egy minőségi, ingyenes, diákok és tanárok számára könnyen használható szoftver-terméket szeretnénk fejleszteni, mely egy nagy mérföldkönek tekinthető módszer lesz a programozók minél egyszerűbb és emelt minőségű betanításában.

7. Könyvészet

Julita Vassileva, „Toward Social Learning Environments”, IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 1, NO. 4, OCTOBER-DECEMBER 2008

Dr. Sarmad Mohammad (PhD Computer Science), “Effectiveness of E-Learning System” , 2009 International Conference on Computer Engineering and Technology

Elizabeth J. Brown, Member, IEEE, Timothy J. Brailsford, Tony Fisher, and Adam Moore, “Evaluating Learning Style Personalization in Adaptive Systems: Quantitative Methods and Approaches”, IEEE TRANSACTIONS ON LEARNING TECHNOLOGIES, VOL. 2, NO. 1, JANUARY-MARCH 2009

Wikipedia, online, <http://wikipedia.com>

Adobe labs, online, <http://labs.adobe.com/technologies/flex/>