

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

DIPLOMOVÁ PRÁCE

Plzeň, 2013

František Liška

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra informatiky a výpočetní techniky

Diplomová práce

Off-line klient pro EEG/ERP portál

Plzeň, 2013

František Liška

Originál zadání

Prohlášení

Prohlašuji, že jsem diplomovou práci vypracoval samostatně a výhradně s použitím citovaných pramenů.

V Plzni dne 14. května 2013

.....

František Liška

Poděkování

Na tomto místě bych rád poděkoval vedoucímu práce Romanu Moučkovi za jeho odborné rady a všem, kteří se podíleli na testování mé aplikace.

Abstract

Offline client for EEG/ERP portal

Department of Computer Science and Engineering is developing a web application called EEG/ERP portal. It is used for storing and sharing neuroinformatic data gathered from experiments. Some experiments are performed in places without Internet connection.

The objective of this paper is to create a desktop application which allows storing experimental data without connection to the Internet. It will be able to transfer experimental data from the client to the EEG/ERP portal.

This document describes architecture and features of the EEG/ERP portal and web services technology which is used for the data transfer.

Keywords

Web Services, WSDL, SOAP, REST, Apache CXF, EEG/ERP Portal, Neuroinformatics

Obsah

1	Úvod	8
2	Neuroinformatika.....	9
2.1	Elektroencefalografie	9
2.2	Evokované potenciály	9
3	EEG/ERP portál.....	11
3.1	Funkce portálu	12
3.2	Uživatelské role.....	12
3.3	Architektura	13
3.3.1	Datová vrstva	13
3.3.2	Aplikační vrstva	14
3.3.3	Prezentační vrstva.....	14
4	Webové služby	15
4.1	Protokoly	15
4.2	Role	16
4.3	WSDL.....	17
4.4	SOAP.....	19
4.4.1	Envelope element	20
4.4.2	Header element	20
4.4.3	Body element	21
4.4.4	Fault element	22
4.4.5	SOAP a HTTP.....	22
4.5	REST.....	24
4.6	Apache CXF.....	25
5	Specifikace systému	26
6	Návrh Systému	28
6.1	Architektura	28
6.1.1	Perzistentní vrstva.....	29

6.1.2	Servisní vrstva	29
6.1.3	Prezentační vrstva	29
6.2	Uživatelské profily	29
6.3	Stažení dat ze serveru	30
6.4	Nahrání změn na server	31
6.5	Uživatelské rozhraní	32
6.6	Databáze	33
6.6.1	Tabulky	33
6.6.2	ERA Diagram.....	41
7	Implementace	42
7.1	Použité prostředky	42
7.2	Struktura aplikace	43
7.2.1	Server	43
7.2.2	Klient	44
7.3	Komponenty grafického rozhraní.....	47
7.3.1	Validace uživatelských vstupů.....	49
7.4	Přenos dat a servisní objekty	50
8	Testování	52
8.1	Test svázání profilu s účtem	52
8.2	Test přidání dat pro experiment	54
8.3	Test přidání experimentu.....	56
8.4	Test přehrání změn na server	58
9	Zhodnocení dosažených výsledků	60
10	Závěr.....	61
	Seznam zkratk	62
	Literatura.....	64
	Seznam obrázků	67
	Přílohy	70

1 Úvod

Na katedře informatiky a výpočetní techniky Západočeské univerzity v Plzni je vyvíjen EEG/ERP (Electroencephalography/event-related potentials) portál, jehož cílem je uchování a sdílení informací o neuroinformatických experimentech. S rostoucím počtem opakování stejného experimentu roste jeho výpovědní hodnota, proto je cílem portálu sjednotit formu a strukturu dat uložených experimentů tak, aby bylo možné naměřená data analyzovat. Experimenty se provádí po celém světě, a to i na pracovištích, kde není vždy dostupné internetové připojení.

Cílem této práce je navrhnout a vytvořit aplikaci, která umožní zadávat neuroinformatická data v místě, kde není k dispozici připojení k EEG/ERP portálu. Aplikace poskytne při opětovném připojení k webovému portálu možnost dávkového přenosu dat z aplikace do portálu.

V této práci se nejprve seznámíme se základními pojmy z oblasti neuroinformatiky (kapitola 2), s architekturou a funkcemi EEG/ERP portálu (kapitola 3). Dále se zaměříme na obecné principy fungování webových služeb (kapitola 4). Praktická část pojednává o vymezení požadovaných funkcí aplikace (kapitola 5), o jejím návrhu (kapitola 6) a o implementaci (kapitola 7) včetně dávkového přenosu dat z aplikace na webový portál. Poslední část je věnována testování navrženého řešení (kapitola 8).

2 Neuroinformatika

Neuroinformatika je vědní obor zabývající se integrací informací na všech úrovních neurověd. Jejich cílem je porozumění funkcím mozku a využití získaných znalostí k určení příčin nemocí ze zjištěných symptomů. Neuroinformatika zahrnuje vývoj standardů a infrastruktury pro sběr dat a jejich ukládání, sdílení, publikování, modelování, simulování a vizualizaci [1].

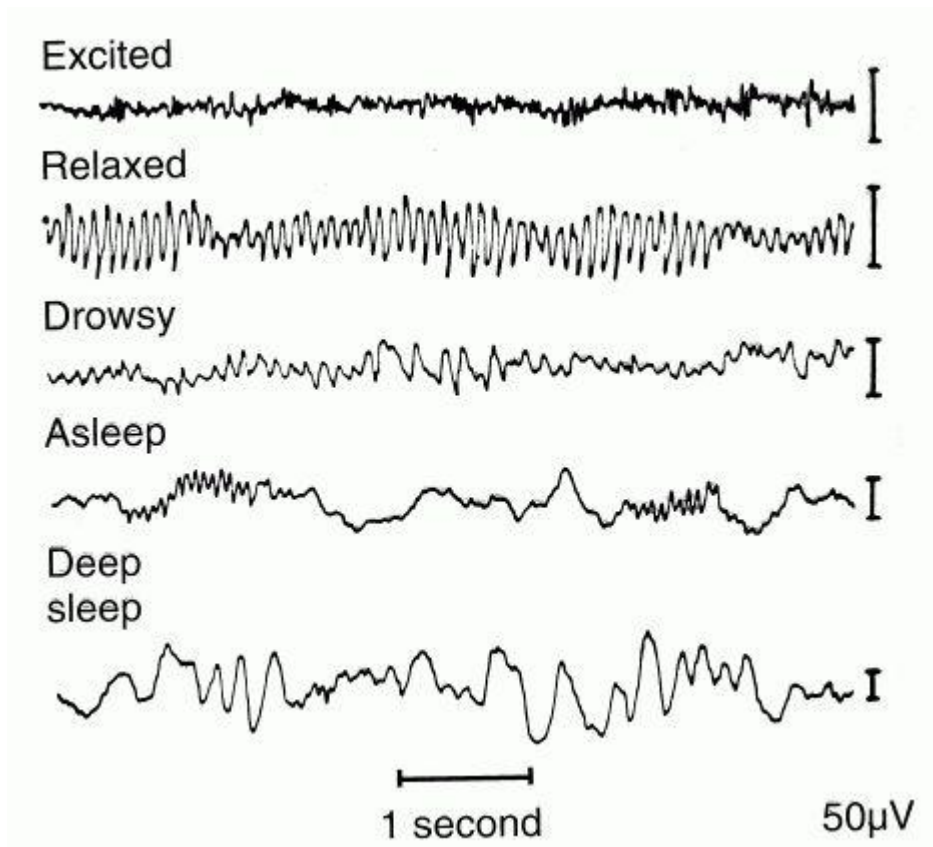
2.1 Elektroencefalografie

Elektroencefalografie (EEG) je jednou z metod měření aktivity lidského mozku. Principem je snímání změny elektrických potenciálů elektrodami z povrchu hlavy a následné vyhodnocení naměřeného EEG signálu. Vzhledem k nízké ceně, neinvazivnosti a rutinnímu postupu při měření je tato metoda velmi rozšířená. Nevýhodou je hrubost výsledného obrazu mozkové aktivity, jelikož EEG zachycuje velké množství zdrojů neuronální aktivity. Proto je těžké určit odpovídající neurokognitivní procesy z naměřené mozkové aktivity. Specifické neuronální odpovědi spojené s konkrétními smyslovými, kognitivními nebo motorickými událostmi je možné extrahovat z celkové naměřené aktivity [14].

2.2 Evokované potenciály

Evokovaný potenciál je odpověď nervové soustavy na stimulaci receptorů. Tato odpověď se projevuje změnou elektrického napětí v nervové tkáni. Kognitivní evokované potenciály či potenciály vázané na událost (ERP - Event-Related Potentials) jsou změny elektrické aktivity mozku vyvolané identifikovatelnými podněty nebo událostmi z vnějšího prostředí [7].

Příklad měření EEG aktivity je na obrázku 2.1. Měření bylo provedeno v různých stavech testovaného subjektu. V jednotlivých stavech byl subjekt rozrušený, uvolněný, ospalý, spící a v hlubokém spánku. Během rozrušení mají mozkové vlny vysokou frekvenci s malou amplitudou, kdežto během spánku mají frekvenci nízkou s velkou amplitudou [9].



Obrázek 2.1: Měření EEG aktivity v různých stavech testované osoby [9]

3 EEG/ERP portál

EEG/ERP portál je webová aplikace (viz Obrázek 3.1) navržená pro ukládání a sdílení dat získaných z měření EEG aktivity. Systém sjednocuje různé formáty dat pro ukládání EEG/ERP experimentů, umožňuje ukládat jejich binární data a ty popisovat na základě definovaných dodatečných informací (metadat). Experimenty jsou ukládány v rámci výzkumných skupin, ale lze je i sdílet. Každý experiment je proveden v rámci určitého scénáře (postupu experimentu). Je možné si nadefinovat informace např. ohledně počasí během experimentu, použitím hardwaru či testovaných osob [18].

EEGbase No user logged | [Register](#)

Home

Welcome to EEGbase 2.0

EEG base is a system for storage and management of EEG/ERP resources - data, metadata, tools and materials related to EEG/ERP experiments. EEG base advances electrophysiology research by enabling access to public data, tools and results of research groups.

Features:

- Management of EEG/ERP data and metadata
- Management of EEG/ERP experimental design (experimental scenarios)
- Management of data related to tested subjects
- Sharing of knowledge and working within groups
- Signal processing tools
- Content management system
- Fulltext search

For continuing on this website you need to log in. If you don't have an account, you can create one.

[Register](#)

Discover the EEGbase 2.0

Login
[Forgotten password?](#)

E-mail

Password

Remember me on this computer.

[Log in](#)

OR

[Login](#)

[Log in with LinkedIn](#)

Partners

REGISTERED WITH NIF

incf
National Institute of
Czech Republic

EEGbase - database for data gained in encephalography research.
Copyright © [The University of West Bohemia](#) 2008-2013

Obrázek 3.1: Domovská stránka EEG/ERP portálu

3.1 Funkce portálu

Aktuální funkce portálu zahrnují [15]:

- registraci nového uživatele
- ukládání, editaci a stažení dat experimentů a jejich dodatečných informací
- sdílení dat a metadat mezi výzkumnými skupinami
- zobrazení historie stahování (pro administrátory skupin a systémové správce)
- přidávání globálních článků (pro systémové správce)
- přidávání skupinových článků (pro administrátory skupin)
- komentování článků
- fulltextové a pokročilé vyhledávání v EEG databázi
- vystavení webových služeb pro ostatní aplikace

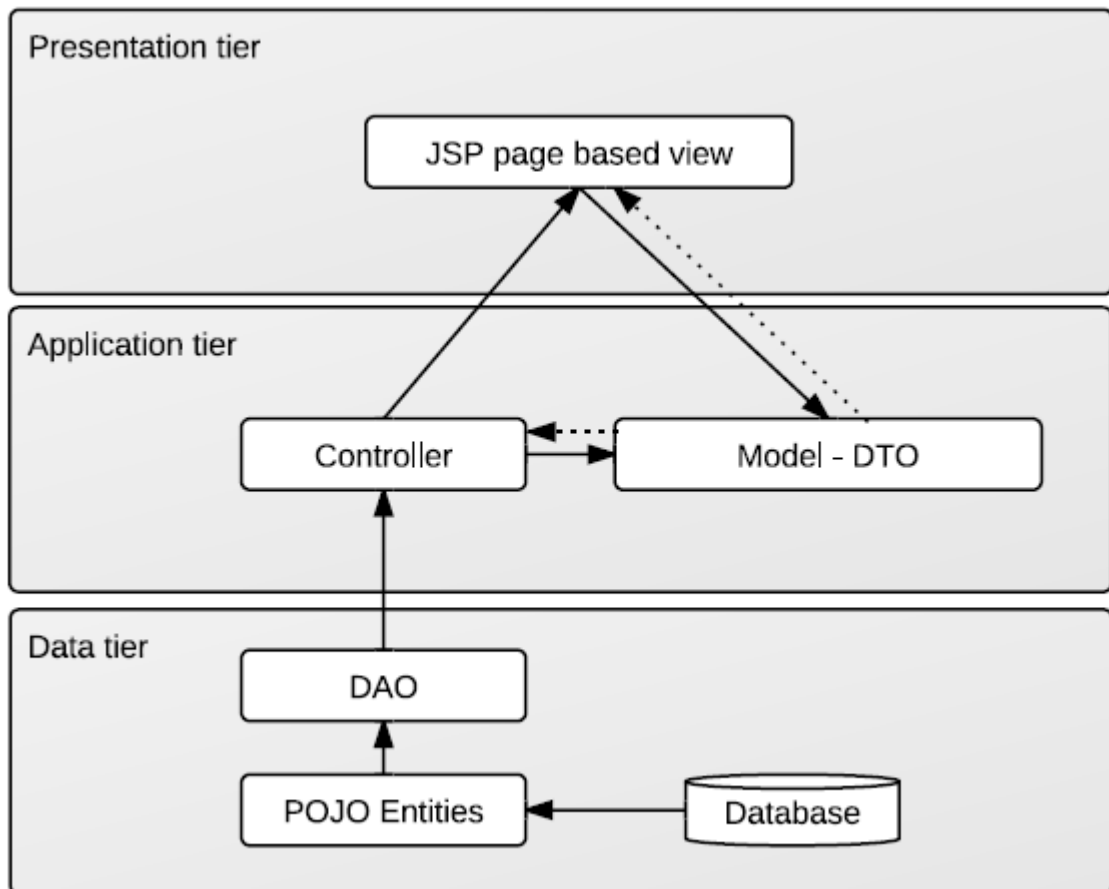
3.2 Uživatelské role

Uživatelé portálu jsou rozděleni podle rolí. Role uživatelů jsou následující [15]:

- **Anonymní** (anonymous) – Nepřihlášený uživatel. Přístupná je pouze domovská stránka s přihlášením a stránka s registrací uživatele.
- **Čtenář** (reader) – Uživatel vytvořený jiným uživatelem při přidávání testované osoby u experimentu. Má minimální práva. Může se pouze přihlásit, být přidán do výzkumné skupiny a má právo na prohlížení veřejných experimentů, scénářů a článků.
- **Uživatel** (user) – Rozšiřuje pravomoci čtenáře o možnost vytvářet vlastní výzkumné skupiny a možnost požádat o přidělení vyšší role.
- **Experimentátor** (experimenter) – Má právo na vkládání a správu vlastních experimentů a scénářů. Uživatel se stane experimentátorem po schválení jeho požadavku správcem na přidělení této role.
- **Admin** (supervisor) – Globální administrátor aplikace. Má právo na správu uživatelských účtů a má přístup ke všem datům v aplikaci.

3.3 Architektura

Architektura portálu je vrstevná (viz Obrázek 3.2). V aplikační a prezentační vrstvě je použit architektonický vzor MVC (Model-View-Controller) [15]. MVC rozděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent, takže změna jedné z nich má jen minimální vliv na ostatní komponenty [13]. Portál je postavený na platformě Java EE (Java Platform, Enterprise Edition), která je určena pro vývoj přenositelných, robustních, škálovatelných serverových aplikací v jazyce Java. Součástí platformy Java je také knihovna pro tvorbu webových služeb [23].



Obrázek 3.2: Architektura portálu [15]

3.3.1 Datová vrstva

Všechna aplikační data jsou uložena v Oracle 11g databázi. Pro přístup k databázi je použit ORM (Object-Relational Mapping) framework Hibernate. Jednotlivé tabulky jsou mapovány k jednoduchým objektům POJO (Plain Old Java Object), v nichž jsou definována omezení a mapovací parametry přes anotace. Pro přístup k těmto objektům slouží třídy typu DAO (Data Access Object) [15].

3.3.2 Aplikační vrstva

Portál je postaven na Spring Framework 3.0 s integrovanými moduly Spring Core, MVC, Security a AOP (Aspect-oriented Programming). Spring umožňuje implementaci bez těsných vazeb mezi jednotlivými vrstvami a jejich komponentami. Uživatelské akce jsou ověřovány validátorem, o který se starají třídy controller. Controller volá DAO pro ukládání nebo načítání dat z databáze a těmito daty naplňuje DTO (Data Transfer Object) pro zobrazení v prezentační vrstvě [15].

3.3.3 Prezentační vrstva

Na prezentační vrstvě používá portál technologii JSP (Java Server Pages) pro zobrazení do standardních HTML (HyperText Markup Language) webových stránek. Proto lze k portálu přistoupit odkudkoliv přes webový prohlížeč bez potřeby dodatečných pluginů. Pro vkládání dat určených k zobrazení je použita technologie JSTL (JavaServer Pages Standard Tag Library) [15].

Vzhled prezentační vrstvy je ukázán na obrázku 3.3, na kterém je stránka pro zobrazení experimentů vytvořených aktuálním uživatelem.

The screenshot shows the EEGbase web application interface. At the top, there is a logo for EEGbase and a navigation menu with items: Home, Artides, Experiments, Scenarios, Groups, People, Lists, History. A user is logged in as 'fiska@students.zcu.cz' with links for 'My account' and 'Log out'. A search bar is present on the right. The main content area is titled 'My experiments' and displays a table of experiments. The table has columns: No., Scenario title, Date, Gender, Year of birth, Services, Detail, and Download. The table contains 15 rows of experiment data. Below the table, there are navigation controls for the page, including '<<', '<', 'Page 1 of 1', '>', and '>>'.

No.	Scenario title	Date	Gender	Year of birth	Services	Detail	Download
260	eScenario	09.04.2013, 01:05	M	2013	Not Available	Detail	Download
255	eScenario	08.04.2013, 18:00	M	1987	Not Available	Detail	Download
258	medovo 21	07.04.2013, 01:00	F	2010	Not Available	Detail	Download
257	tomcattomcat	07.04.2013, 01:00	F	2010	Not Available	Detail	Download
256	medovo 21	07.04.2013, 01:00	F	2010	Not Available	Detail	Download
265	medovo 21	05.04.2013, 01:00	F	2010	Not Available	Detail	Download
261	Kscenario	03.04.2013, 01:00	M	1985	Not Available	Detail	Download
248	medovo 21	02.04.2013, 01:00	F	2010	Not Available	Detail	Download
238	medovo 21	01.12.2012, 01:00	F	2010	Not Available	Detail	Download
236	delScenar	27.11.2012, 01:00	F	2010	Not Available	Detail	Download
237	eScenario	27.11.2012, 01:00	M	2011	Not Available	Detail	Download
235	Nazev scenare	29.10.2012, 21:05	M	1990	Not Available	Detail	Download
234	Nazev scenare	29.10.2012, 21:00	M	1990	Not Available	Detail	Download
232	delScenar	22.05.2012, 14:56	M	1987	Not Available	Detail	Download

Obrázek 3.3: Prezentační vrstva portálu

4 Webové služby

Webové služby (Web Services) vznikly za účelem sjednocení komunikace programů běžících na různých platformách. Jsou založené na W3C (World Wide Web Consortium) standardech, lze je použít nezávisle na programovacím jazyce, typu procesoru či operačního systému [27].

Jde o technologii pro volání vzdálených procedur (RPC – Remote Procedure Call), využívající jazyka XML (Extensible Markup Language) k přenosu zpráv přes protokol HTTP (HyperText Transfer Protocol). Pro definici typů dat je využíván standard XML Schema, k identifikaci objektů pak XML namespace [27].

Výhody webových služeb

- vystavení existující funkcionality přes počítačovou síť pro jiné aplikace
- na bázi XML – možnost spojení více různých aplikací při zachování platformní a technologické nezávislosti
- standardizovaný protokol
- nízké náklady na komunikaci (lze použít i levné internetové připojení pro poskytování webových služeb)
- volně vázané – změna v implementaci rozhraní webové služby nevyžaduje změny v programech uživatelů webové služby
- podpora synchronní (klient čeká na vykonání služby) i asynchronní komunikace (výsledek lze vrátit i později) [29]

4.1 Protokoly

Architektura webových služeb z pohledu používaných protokolů zahrnuje čtyři hlavní vrstvy [29].

Service transport

Transportní vrstva je zodpovědná za přenos zpráv mezi aplikacemi. Do této vrstvy spadají protokoly HTTP, SMTP (Simple Mail Transfer Protocol), FTP (File Transfer Protocol) a novější protokoly jako například BEEP (Blocks Extensible Exchange Protocol) [29].

XML messaging

Tato vrstva je zodpovědná za zakódování zpráv do XML formátu tak, aby obě strany komunikace byly schopné rozpoznat obsah posílaných zpráv. Patří sem XML-RPC a SOAP (Simple Object Access Protocol) [29].

Service description

Poskytuje jazyk WSDL (Web Service Description Language) pro popis rozhraní webové služby [29].

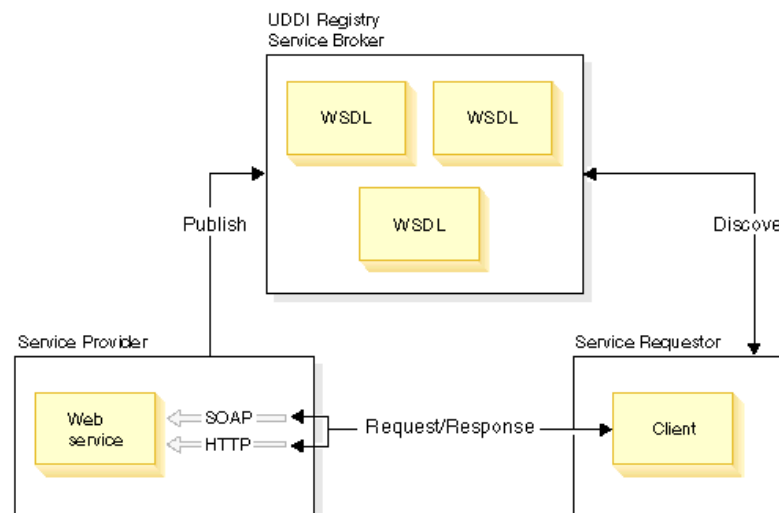
Service discovery

Tato vrstva centralizuje poskytované webové služby do registru služeb UDDI (Universal Description, Discovery and Integration), který umožňuje vystavit webovou službu nebo nalézt dostupné webové služby [29].

4.2 Role

V architektuře webových služeb se nachází tři uživatelské role (viz Obrázek 4.1) [29].

- **Service Provider** (poskytovatel služby) – Implementuje rozhraní služby a vystavuje ji přes síť.
- **Service Requestor/Consumer** (žadatel služby) - Uživatel webové služby. Otevírá síťové připojení k webové službě a posílá XML požadavek.
- **Service Registry/Broker** (registr služby) – Místo, kde lze najít dostupné služby nebo zaregistrovat nové služby.



Obrázek 4.1: Architektura webové služby [2]

4.3 WSDL

WSDL (Web Service Description Language) je jazyk na bázi XML pro popis webových služeb a přístupu k webovým službám. WSDL je velmi často použit v kombinaci s XML Schema a SOAP pro poskytování webových služeb přes Internet. Klientský program připojující se k webové službě si může přečíst WSDL dokument na serveru a zjistit tak, které funkce server poskytuje. Poté může použít protokol SOAP k zavolání funkce vypsaných v WSDL dokumentu [30].

WSDL dokument

Jedná se o XML dokument, který specifikuje umístění a vystavené operace (metody) webové služby. Pro popis webové služby slouží tyto elementy (viz Obrázek 4.2) [30]:

- **Definition** – Kořenový element. Popisuje název služby a jmenné prostory.
- **Data types** – Kontejner definic datových typů používaných webovou službou.
- **Message** – Definice dat použitých v posílaných zprávách.
- **Operation** – Abstraktní definice operací používaných ve zprávách.
- **Port type** – Soubor operací podporovaných jedním nebo více endpointy¹.
- **Binding** – Specifikuje protokol a datový formát pro konkrétní typ portu.
- **Port** – Poskytuje cílovou adresu pro komunikaci se službou.
- **Service** – Kolekce souvisejících endpointů obsahujících definici služby.

```
<definitions>
  <types>
    data type definitions.....
  </types>
  <message>
    definition of the data being communicated....
  </message>
  <portType>
    set of operations.....
  </portType>
  <binding>
    protocol and data format specification....
  </binding>
</definitions>
```

Obrázek 4.2: Struktura WSDL dokumentu [30]

¹ endpoint – místo pro komunikaci se službou specifikovanou URI (Uniform Resource Identifier) [26]

Následující příklad WSDL dokumentu (viz Obrázek 4.3) demonstruje ukázkou jednoduché služby *HelloService*. Tato služba poskytuje metodu s názvem *sayHello*. Tato funkce očekává vstupní parametr typu string a vrací řetězec s pozdravem, kde je použit vstupní parametr (Např. je-li vstupním parametr *world*, pak vrací *Hello, world!*). Služba je dostupná na adrese *http://www.examples.com/SayHello/* (viz element *service*) [30].

```
<definitions name="HelloService"
  targetNamespace="http://www.examples.com/wsd1/HelloService.wsd1"
  xmlns="http://schemas.xmlsoap.org/wsd1/"
  xmlns:soap="http://schemas.xmlsoap.org/wsd1/soap/"
  xmlns:tns="http://www.examples.com/wsd1/HelloService.wsd1"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <message name="SayHelloRequest">
    <part name="firstName" type="xsd:string"/>
  </message>
  <message name="SayHelloResponse">
    <part name="greeting" type="xsd:string"/>
  </message>

  <portType name="Hello_PortType">
    <operation name="sayHello">
      <input message="tns:SayHelloRequest"/>
      <output message="tns:SayHelloResponse"/>
    </operation>
  </portType>

  <binding name="Hello_Binding" type="tns:Hello_PortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="sayHello">
      <soap:operation soapAction="sayHello"/>
      <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded"/>
      </input>
      <output>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded"/>
      </output>
    </operation>
  </binding>

  <service name="Hello_Service">
    <documentation>WSDL File for HelloService</documentation>
    <port binding="tns:Hello_Binding" name="Hello_Port">
      <soap:address
        location="http://www.examples.com/SayHello/">
    </port>
  </service>
</definitions>
```

Obrázek 4.3: Příklad WSDL dokumentu [30]

4.4 SOAP

SOAP (Simple Object Access Protocol) je jednoduchý a otevřený komunikační protokol na bázi XML sloužící k výměně informací mezi počítači. Může být používán různými transportními protokoly, ale nejčastější je přenos vzdálených procedur (či dokumentů) přes protokol HTTP. Výhodou použití protokolu HTTP je jednoduchost, stabilita, vysoká rozšířenost a fakt, že ve většině firewallů je HTTP provoz povolen [20].

Protokol SOAP umožňuje klientským aplikacím se jednoduše připojit ke vzdáleným službám a spustit vzdálené metody. SOAP zprávy jsou napsány celé v XML, jsou tedy nezávislé na platformě či použitém programovacím jazyce.

SOAP zpráva je obyčejný XML dokument (viz Obrázek 4.4) obsahující následující elementy [20]:

- **Envelope** (obálka) – Definiuje počátek a konec zprávy.
- **Header** (hlavička) – Obsahuje aplikačně specifické informace o SOAP zprávě.
- **Body** (tělo) – Obsahuje aplikační data posílané zprávy.
- **Fault** (chyba) – Poskytuje informace o chybách během zpracování zprávy.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
  ...
  </soap:Header>

  <soap:Body>
  ...
    <soap:Fault>
    ...
    </soap:Fault>
  </soap:Body>

</soap:Envelope>
```

Obrázek 4.4: Struktura SOAP zprávy [20]

4.4.1 Envelope element

Envelope element slouží jako obálka pro SOAP zprávu (viz Obrázek 4.5). Příjemce podle tohoto elementu pozná začátek a konec vysílání zprávy. Jedná se o povinný element, který musí obsahovat element *body*. Může obsahovat *header* element, který se nachází před tělem. Struktura obálky je závislá na použité verzi SOAP. Při příjmu zprávy s jinou verzí je vyhozena chyba [20].

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
  ...
  Message information goes here
  ...
</soap:Envelope>
```

Obrázek 4.5: Příklad obálky ve verzi 1.2 [20]

4.4.2 Header element

Header je nepovinný element poskytující možnost specifikovat dodatečné požadavky na aplikační úrovni (viz Obrázek 4.6). Příkladem může být specifikace digitálního podpisu pro služby, které jsou chráněny heslem [20].

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Header>
    <m:Trans xmlns:m="http://www.w3schools.com/transaction/"
      soap:mustUnderstand="1">234
    </m:Trans>
  </soap:Header>
  ...
  ...
</soap:Envelope>
```

Obrázek 4.6: Příklad hlavičky SOAP dokumentu [20]

SOAP definuje ve standardním jmenném prostoru (<http://www.w3.org/2001/12/soap-envelope>) tři atributy [20]:

- **EncodingStyle** – Definuje datový typ použitý v dokumentu. Může se vyskytovat u kteréhokoliv SOAP elementu.
- **MustUnderstand** – Indikuje, zda je příjemce povinen zpracovat *header* element.
- **Actor** – Určuje cílový endpoint pro příjem *header* elementu (zpráva může procházet na cestě i přes více endpointů k příjemci).

4.4.3 Body element

Body je povinný element, který obsahuje samotnou SOAP zprávu určenou pro příjemce. Každý potomek elementu *body* musí mít specifikovaný jmenný prostor. Následující příklad (viz Obrázek 4.7) ukazuje poslání zprávy webové službě pro zjištění ceny jablek.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPrice xmlns:m="http://www.w3schools.com/prices">
      <m:Item>Apples</m:Item>
    </m:GetPrice>
  </soap:Body>

</soap:Envelope>
```

Obrázek 4.7: Příklad těla SOAP požadavku [20]

Odpověď serveru na tuto zprávu by mohla vypadat jako na obrázku 4.8.

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">

  <soap:Body>
    <m:GetPriceResponse xmlns:m="http://www.w3schools.com/prices">
      <m:Price>1.90</m:Price>
    </m:GetPriceResponse>
  </soap:Body>

</soap:Envelope>
```

Obrázek 4.8: Příklad odpovědi na SOAP požadavek [20]

4.4.4 Fault element

Fault je nepovinný element, který v případě výskytu značí, že se stala chyba. Element se v případě výskytu musí nacházet uvnitř elementu *body*. V celé SOAP zprávě se může vyskytovat pouze jednou [20].

Fault může obsahovat následující elementy [20]:

- **Faultcode** – Kód identifikující chybu.
- **Faultstring** – Popis chyby.
- **Faultactor** – Informace o tom, kdo způsobil chybu.
- **Detail** – Obsahuje specifické informace související s *body* elementem.

Kódy identifikující chybu (*faultcode*) mohou být následující [20]:

- **VersionMismatch** – Indikuje použití neplatného jmenného prostoru pro obálku.
- **MustUnderstand** – Potomek hlavičky s *mustUnderstand* atributem nastaveným na jedna nebyl korektně zpracován.
- **Client** – Vyslaná zpráva měla nesprávnou strukturu nebo informaci.
- **Server** – Na serveru nastal problém a zpráva nemohla být přijata ke zpracování.

4.4.5 SOAP a HTTP

HTTP protokol komunikuje přes TCP/IP (Transmission Control Protocol/Internet Protocol). HTTP klient se připojí k serveru a po připojení pošle HTTP požadavek serveru. Server ho zpracuje a odešle odpověď klientovi. Odpověď obsahuje stavový kód požadavku [20].

SOAP protokol používá HTTP požadavky/odpovědi ve formátu, který vyhovuje SOAP pravidlům. SOAP požadavek může být například HTTP *POST* nebo HTTP *GET* požadavek. U HTTP *POST* je potřeba specifikovat dvě HTTP hlavičky: *Content-Type* a *Content-Length* (viz Obrázek 4.9) [20].

```
POST /item HTTP/1.1
Content-Type: application/soap+xml; charset=utf-8
Content-Length: 250
```

Obrázek 4.9: Příklad HTTP požadavku [20]

Content-Type

Definuje MIME² formát zprávy a kódování pro XML tělo požadavku/odpovědi [20].

Content-Length

Hlavička *Content-Length* specifikuje počet bytů v těle požadavku nebo odpovědi.

V následujícím příkladu je zaslán požadavek na zjištění ceny akcií (viz Obrázek 4.10).

SOAP požadavek má parametr *StockName* (název akcie) [20].

```
POST /InStock HTTP/1.1
Host: www.example.org
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPrice>
    <m:StockName>IBM</m:StockName>
  </m:GetStockPrice>
</soap:Body>
</soap:Envelope>
```

Obrázek 4.10: Příklad SOAP požadavku [20]

SOAP Odpověď vrací parametr *Price* (cena).

```
HTTP/1.1 200 OK
Content-Type: application/soap+xml; charset=utf-8
Content-Length: nnn
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
<soap:Body xmlns:m="http://www.example.org/stock">
  <m:GetStockPriceResponse>
    <m:Price>34.5</m:Price>
  </m:GetStockPriceResponse>
</soap:Body>
</soap:Envelope>
```

Obrázek 4.11: Příklad SOAP odpovědi [20]

² MIME (Multipurpose Internet Mail Extension) – standard, který umožňuje přes elektronickou poštu posílat zprávy v různých formátech [30]

4.5 REST

REST (Representational State Transfer) je architektura webových služeb založená na sadě principů, které popisují, jak jsou definovány a adresovány síťové zdroje. Aplikace, které využívají tyto principy, jsou nazývány jako RESTful nebo REST-style. Každý síťový zdroj je v RESTful architektuře adresovatelný přes sadu příkazů. Typicky za použití HTTP příkazů *GET*, *POST*, *PUT* nebo *DELETE*. Protokol je bezstavový, typu klient-server s podporou cachování [8].

Obrázek 4.12 ilustruje princip přenosu zprávy v architektuře REST. Nejprve je na URI adresu, která specifikuje umístění požadovaného zdroje, vyslán požadavek na vrácení informací o firemním účtu. Jako odpověď je vrácen XML s požadovanými údaji [8].



Obrázek 4.12: Přenos zprávy technologií REST [8]

Výhody

- nezávislost na programovacím jazyce či platformě
- jednodušší vývoj než v SOAP
- snáze naučitelný
- nepotřebuje další vrstvu pro posílání zpráv [26]

Nevýhody

- předpokládá point to point komunikaci – není použitelný v distribuovaném prostředí
- chybí podpora rozličných standardů, proto je vývoj sofistikovanějších služeb náročnější
- svázán s HTTP transportním protokolem [26]

4.6 Apache CXF

Apache CXF (Celtic XFire) je open source framework určený pro tvorbu webových služeb. Pro vytvoření webové služby využívá CXF různé API, například JAX-WS (Java API for XML Web Services) nebo JAX-RS (Java API for RESTful Web Services). Tyto služby mohou používat různé protokoly jako například SOAP či RESTful HTTP [4].

Apache CXF se funkcí zaměřuje převážně na následující oblasti [4]:

- **Standardy webových služeb** – CXF podporuje různé standardy pro webové služby včetně SOAP, WSDL, WS-Addressing, WS-Policy, WS-Security aj.
- **Frontendy** – CXF podporuje různé druhy frontendů (např. JAX-WS).

JAX-WS implementace v CXF je oproti referenční specifikaci výrazně snazší na používání. Umí automaticky generovat kód pro beans reprezentující požadavky a odpovědi. V jednoduchých případech nepotřebuje ani WSDL. V ostatních případech CXF podporuje jak generování programového kódu při znalosti WSDL, tak generování WSDL z kódu napsaného v Javě [4].

5 Specifikace systému

Navrhovaný systém se bude jmenovat EEGClient a bude umožňovat zadávání a prohlížení neuroinformatických dat a přenos těchto dat mezi klientem a EEG/ERP portálem.

Klient bude implementovat tyto funkce:

- zadávání neuroinformatických dat
- prohlížení neuroinformatických dat
- dávkový přesun dat mezi serverem a klientem
 - stáhnutí vstupních dat ze serveru
 - nahrání nových změn na server
- připojení/odpojení od serveru
- vytvoření/přihlášení na profil

Pro zadávání či prohlížení neuroinformatická dat bude možné použít následující funkce (dostupné na serveru v době tvorby tohoto klienta):

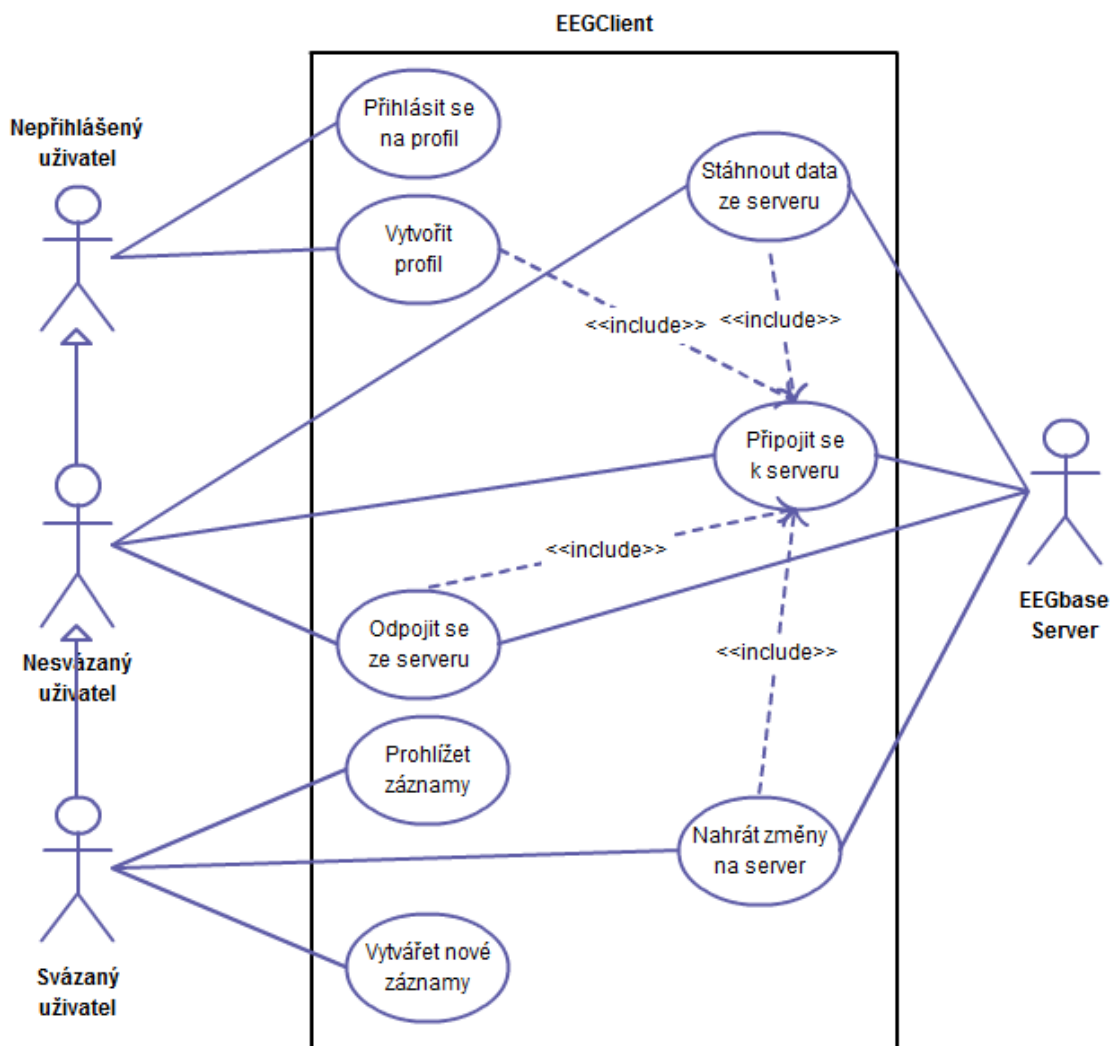
- vytvořit experiment (zahrnuje přidání souboru experimentu)
- prohlédnout vytvořené experimenty
 - přidat dodatečné parametry experimentu či další soubory
 - prohlédnout soubory příslušející experimentu
 - přidat dodatečné informace o souboru
- prohlédnout/vytvořit scénáře
- prohlédnout/vytvořit výzkumné skupiny
 - přidat osobu do skupiny
- prohlédnout/vytvořit osoby
 - přidat dodatečné parametry k osobám
- prohlédnout/přidat číselníky (seznamy definovaných parametrů) pro konkrétní výzkumné skupiny
 - použitý hardware

- počasí během experimentu
- definice dodatečného parametru pro osoby
- definice dodatečného parametru pro experimenty
- definice dodatečných informací o souboru

Systém bude rozpoznávat tři typy uživatelů:

- nepřihlášený uživatel (uživatel ihned po spuštění programu)
- nesvázaný uživatel (poprvé přihlášený na svůj profil bez stažených vstupních dat)
- svázaný uživatel (se staženými vstupními daty, propojeným profilem se serverovým účtem, s možností prohlížení a vytváření neuroinformatických dat)

Popisované funkce jsou zobrazeny na obrázku 5.1.



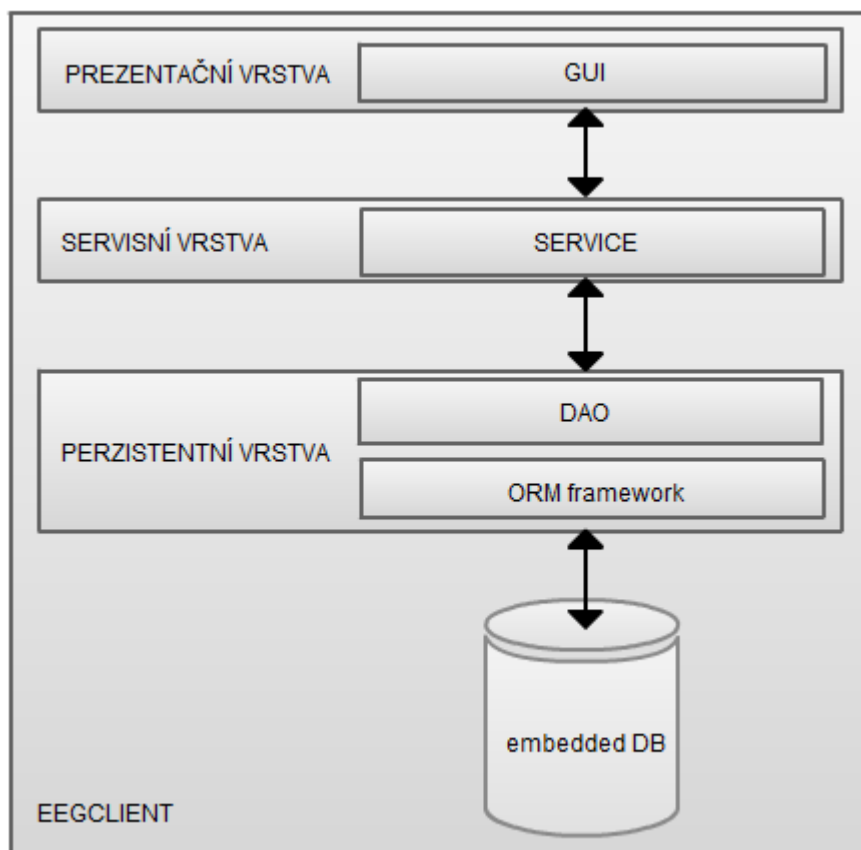
Obrázek 5.1: Diagram případů užití

6 Návrh Systému

V této kapitole je uveden návrh architektury klientské aplikace, návrh přihlášení do klienta, návrh obousměrného přenosu dat klient-server, uživatelského rozhraní a databáze.

6.1 Architektura

Aplikace je navržena jako tlustý klient. Tlustý klient je architektura typu klient-server, kde klient umožňuje vykonávat funkce i bez připojení k serveru [25]. Komunikace mezi klientem a serverem je popsána v kapitolách 6.3 a 6.4. Architektura samotného klienta je vrstevná. Komunikovat spolu mohou pouze sousední vrstvy (viz Obrázek 6.1).



Obrázek 6.1: Architektura klienta

6.1.1 Perzistentní vrstva

Perzistentní vrstva poskytuje servisní vrstvě data z databáze. Pro perzistenci dat byl zvolen ORM (Object-Relational Mapping) přístup. Výhodou tohoto přístupu je možnost na aplikační úrovni pracovat s objektovým modelem i přesto, že aplikace ukládá do relační databáze. Nevýhodou může být vyšší režie při mapování, a tedy potenciálně menší výkon. Databáze je navržena jako jednouživatelská (nevelké nároky na výkon), proto lze tuto nevýhodu zanedbat. Jednotlivé tabulky (viz kapitola 6.6) jsou mapovány na POJO objekty, ke kterým se přistupuje přes DAO objekty [19].

6.1.2 Servisní vrstva

Aplikační logika je reprezentována servisní vrstvou, která se stará o komunikaci mezi prezentační a perzistentní vrstvou. Obsahuje servisní metody pro všechny případy užití.

6.1.3 Prezentační vrstva

Prezentační vrstva slouží k zobrazení informací pro uživatele formou grafického uživatelského rozhraní. Na této úrovni probíhá kontrola vstupů od uživatelů. Návrh uživatelského rozhraní je uveden v kapitole 6.5.

6.2 Uživatelské profily

Klient bude podporovat uživatelské profily. Profilem se rozumí účet, ke kterému se lze přihlásit bez nutnosti zadávat heslo. Profil bude vytvořen se stejným názvem jako uživatelský účet použitý při prvním připojení k serveru. Důvodem pro použití profilu bez hesla je nemožnost získání zapomenutého hesla v případě, že k dispozici není internetové připojení. V tomto případě by nebylo možné zadávat neuroinformatická data do klienta, což je součástí zadání této práce. Použitím tohoto řešení také odpadá problém se změnou hesla na serverovém účtu oproti klientskému.

6.3 Stažení dat ze serveru

Pro zadávání neuroinformatických dat v klientovi bez internetového připojení jsou potřeba alespoň základní vstupní data uložená na serveru (např. při zadávání nového experimentu je potřeba scénář postupu). Proto je nutné po založení nového profilu nejprve tyto data stáhnout. Mezi vstupní data budeme řadit osoby, scénář, záznamy o existujících výzkumných skupinách a seznamy definovaných parametrů. Po stažení vstupních dat je již možné klienta používat pro zadávání neuroinformatických dat bez připojení k serveru. Proces prvního stažení potřebných vstupních dat ze serveru budeme nazývat jako svázání (profilu se serverovým účtem).

Stažení vstupních dat odpovídá určitému pořadí tak, aby bylo možné správně přiřadit vazby mezi jednotlivými záznamy v databázi. Pořadí stažení tabulek (přehled tabulek je v kapitole 6.6.1) je následující:

1. EDUCATION LEVEL
2. PERSON
3. SCENARIO
4. RESEARCH GROUP
 - a. HARDWARE
 - b. WEATHER
 - c. PERSON_OPT_PARAM_DEF
 - d. EXPERIMENT_OPT_PARAM_DEF
 - e. FILE_METADATA_PARAM_DEF
 - f. RESEARCH_GROUP_MEMBERSHIP
5. PERSON_OPT_PARAM_VAL

Při opětovném připojení k internetu bude možné znovu synchronizovat data, dostupná na serveru, za použití stejné funkce jako při stažení vstupních dat. Tento proces nejprve přes webovou službu stáhne záznamy ze serverových tabulek, porovná, zda existují tyto záznamy v klientovi a v případě neexistence je vytvoří. Každý stažený záznam si ponezá informaci o primárním klíči (tzv. *REMOTE_ID*) stejného záznamu na serveru.

6.4 Nahrání změn na server

Změny provedené v klientovi budou zaznamenávány do tabulky *CHANGE* (viz kapitola 6.6.1). Klient nebude podporovat jiné změny než typu přidání, ale do budoucna se zde počítá s rozšířením o editaci nebo smazání záznamu. Každá změna bude náležet profilu, na kterém se odehrála.

Nahrání změn na server bude provedeno přes webovou službu spuštěnou na serveru. Při přidání záznamu do tabulky na serveru bude jako návratová hodnota vrácen nově vytvořený primární klíč daného záznamu. Tato návratová hodnota se uloží do *REMOTE_ID* příslušného záznamu v klientovi. Změna bude po zpracování z tabulky odebrána. Pokud nastane přerušení při přenosu, pak tabulka *CHANGE* obsahuje informace o změnách, které ještě nebyly úspěšně odeslány. Proto půjde proces znovu restartovat a po vyřešení problémů (např. s připojením) zbylé změny na server nahrát.

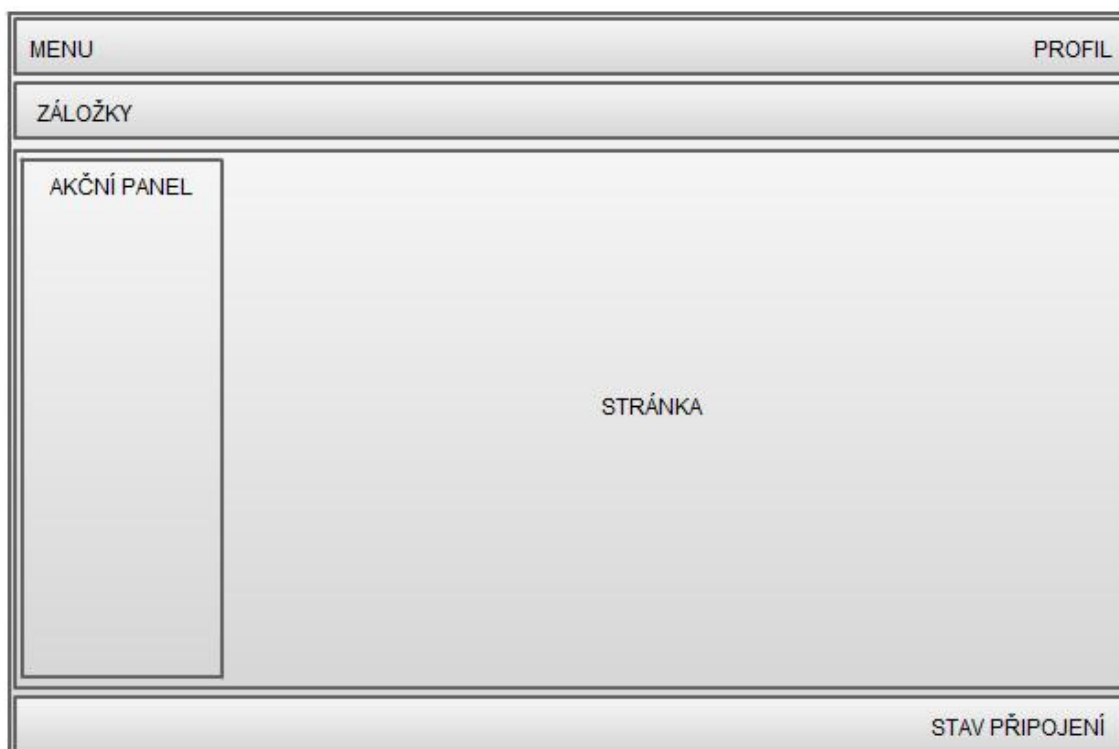
Obdobně jako při stažení dat ze serveru je nutné změny ukládat v určitém pořadí. Tabulky se budou nahrávat na server v pořadí následujícím:

1. RESEARCH_GROUP
2. HARDWARE
3. WEATHER
4. PERSON_OPT_PARAM_DEF
5. PERSON
6. PERSON_OPT_PARAM_VAL
7. RESEARCH_GROUP_MEMBERSHIP
8. FILE_METADATA_PARAM_DEF
9. SCENARIO
10. EXPERIMENT_OPT_PARAM_DEF
11. EXPERIMENT
12. DATA_FILE
13. FILE_METADATA_PARAM_VAL
14. EXPERIMENT_OPT_PARAM_VAL

6.5 Uživatelské rozhraní

Uživatelské rozhraní klienta klade velký důraz na podobnost s uživatelským rozhraním serveru (viz Obrázek 6.2). Důvodem je snadnost použití při znalosti serverového rozhraní. Rozhraní se bude skládat z následujících částí:

- **Menu** – Pro vyvolání některých funkcí programu (připojení, přenos dat, ukončení programu apod.).
- **Profil** – Pro zobrazení aktuálně přihlášeného profilu/pro změnu profilu.
- **Záložky** – Pro přechod k množině stránek se společnou tématikou (např. přidávání/prohlížení experimentů). Při výběru záložky se zobrazí vždy první stránka z dané množiny.
- **Akční panel** – Obsahuje navigační tlačítka mezi stránkami v rámci jedné záložky.
- **Stránka** – Pro zobrazení jednotlivých funkcí (např. prohlížení experimentu).
- **Stav připojení** – Zobrazuje stav připojení k serveru.



Obrázek 6.2: Části uživatelského rozhraní

6.6 Databáze

Klientská data budou uložena v embedded databázi. Výhodou embedded databáze je pro uživatele odstranění nutnosti instalace lokální databáze, jejího nastavení, spuštění a údržby. Další výhodou je použití jedné konkrétní verze, pro kterou je aplikace odladěna, a proto nevznikají potíže při použití verze jiné. Databáze musí být lokálního charakteru, jelikož klient je určen primárně pro zadávání dat bez síťového připojení.

Databáze se bude automaticky generovat při prvním spuštění klienta. Naplnění daty proběhne až po prvním připojení k serveru a následném stažení vstupních dat.

6.6.1 Tabulky

Většina tabulek v lokální databázi bude mít atributy i názvy shodné s tabulkami v EEG portálu, jelikož jde o zadání dat do jedné databáze (klientské) a přehrání dat ve stejné podobě do databáze druhé (na server). Přibudou pouze dvě nové tabulky. Následuje výčet tabulek v abecedním pořadí spolu s popisem jejich atributů.

Tabulka CHANGE

Nová tabulka uchovávající informace o změnách, které se odehrály v klientské databázi.

- CHANGE_ID – primární klíč
- RECORD_LABEL – popis změny (např. název scénáře při jeho přidání)
- RECORD_TABLE – tabulka, ve které se odehrála změna
- RECORD_ID – primární klíč změněného záznamu
- TYPE – typ změny (s hodnotami 'ADD', 'EDIT', 'DELETE')
- PROFILE_ID – cizí klíč odkazující na primární klíč profilu, na kterém se odehrála změna

Tabulka DATA_FILE

Tabulka, která ukládá údaje o nahraném souboru pro experiment.

- DATA_FILE_ID – primární klíč
- EXPERIMENT_ID – odkazuje na experiment, ke kterému soubor patří
- DESCRIPTION – informace o datech uložených v souboru

- FILE_CONTENT – soubor uložený jako BLOB (Binary Large Object)
- MIMETYPE – formát souboru
- FILENAME – název souboru
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka EDUCATION_LEVEL

Číselník ukládající informace o dosaženém vzdělání osoby.

- EDUCATION_LEVEL_ID – primární klíč
- TITLE – vzdělání osoby
- IS_DEFAULT – hodnota 1, pokud je záznam společný pro všechny skupiny, jinak 0
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka EXPERIMENT

Tabulka s informacemi o experimentu (měření).

- EXPERIMENT_ID – primární klíč
- WEATHER_ID – odkaz na počasí v době experimentu
- SUBJECT_PERSON_ID – testovaná osoba
- OWNER_PERSON_ID – zakladatel experimentu
- SCENARIO_ID – scénář měření
- RESEARCH_GROUP_ID – odkazuje na výzkumnou skupinu, v rámci které bylo provedeno měření
- START_TIME – čas počátku měření
- END_TIME – čas konce měření
- TEMPERATURE – teplota při měření experimentu

- ENVIRONMENT_NOTE – poznámka k prostředí, ve kterém se měření provádělo
- PRIVATE – zda je měření soukromé nebo viditelné pro všechny
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka EXPERIMENT_OPT_PARAM_DEF

Číselník ukládající definice dodatečných parametrů experimentu.

- EXPERIMENT_OPT_PARAM_DEF_ID – primární klíč
- PARAM_NAME – název parametru
- PARAM_DATA_TYPE – datový typ parametru
- EXPERIMENT_OPT_PARAM_GROUP_REL – M:N rozkladová tabulka mezi RESEARCH_GROUP a touto tabulkou
- IS_DEFAULT – hodnota 1, pokud je záznam společný pro všechny skupiny, jinak 0
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka EXPERIMENT_OPT_PARAM_VAL

Číselník ukládající hodnoty dodatečných parametrů experimentu.

- EXPERIMENT_OPT_PARAM_VAL_ID – primární klíč
- EXPERIMENT_ID – experiment, ke kterému patří daná hodnota parametru
- EXPERIMENT_OPT_PARAM_DEF_ID – odkaz na definici daného parametru
- PARAM_VALUE – hodnota parametru

Tabulka FILE_METADATA_PARAM_DEF

Číselník ukládající informace o souboru s měřením.

- FILE_METADATA_PARAM_DEF_ID – primární klíč
- PARAM_NAME – název parametru

- PARAM_DATA_TYPE – datový typ parametru
- FILE_METADATA_PARAM_GROUP_REL – M:N rozkladová tabulka mezi RESEARCH_GROUP a touto tabulkou
- IS_DEFAULT – hodnota 1, pokud je záznam společný pro všechny skupiny, jinak 0
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka FILE_METADATA_PARAM_VAL

Číselník ukládající metadata pro soubory s experimentem.

- FILE_METADATA_PARAM_VAL_ID – primární klíč
- DATA_FILE_ID – soubor s měřením, ke kterému patří daná hodnota parametru
- FILE_METADATA_PARAM_DEF_ID – odkaz na definici daného parametru
- METADATA_VALUE – hodnota parametru

Tabulka HARDWARE

Číselník, který ukládá informace o použitém hardwaru.

- HARDWARE_ID – primární klíč
- TITLE – nadpis pro hardware
- TYPE – typ hardwaru
- DESCRIPTION – popis hardwaru
- HARDWARE_GROUP_REL – M:N rozkladová tabulka mezi RESEARCH_GROUP a touto tabulkou
- HARDWARE_USAGE_REL – M:N rozkladová tabulka mezi touto tabulkou a tabulkou EXPERIMENT
- IS_DEFAULT – hodnota 1, pokud je záznam společný pro všechny skupiny, jinak 0

- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka PERSON

Tabulka s informacemi o osobách v systému (uživatel či testovaný subjekt).

- PERSON_ID – primární klíč
- GIVENNAME – jméno osoby
- SURNAME – příjmení
- DATE_OF_BIRTH – datum narození
- GENDER – pohlaví
- PHONE_NUMBER – telefonní číslo
- REGISTRATION_DATE – datum registrace do systému
- NOTE – další poznámky k osobě
- USERNAME – uživatelské jméno (e-mail)
- LATERALITY – udává dominanci jedné ruky (pravák/levák)
- EDUCATION_LEVEL_ID – odkaz na dosažený stupeň vzdělání
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka PERSON_OPT_PARAM_DEF

Číselník ukládající definice dodatečných parametrů osob.

- PERSON_OPT_PARAM_DEF_ID – primární klíč
- PARAM_NAME – název parametru
- PARAM_DATA_TYPE – datový typ parametru
- PERSON_OPT_PARAM_GROUP_REL – M:N rozkladová tabulka mezi RESEARCH_GROUP a touto tabulkou
- IS_DEFAULT – hodnota 1, pokud je záznam společný pro všechny skupiny, jinak 0

- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka PERSON_OPT_PARAM_VAL

Číselník ukládající hodnoty dodatečných parametrů osob.

- PERSON_OPT_PARAM_VAL_ID – primární klíč
- PERSON_ID – osoba, ke které patří daná hodnota parametru
- PERSON_OPT_PARAM_DEF_ID – odkaz na definici daného parametru
- PARAM_VALUE – hodnota parametru

Tabulka PROFILE

Nová tabulka představující uživatelský profil.

- PROFILE_ID – primární klíč
- PROFILE_NAME – název uživatelského profilu
- PROFILE_LINKED – true, pokud je profil svázán (tj. jsou nahrána vstupní data) s uživatelským serverovým účtem, jinak false
- LINKED_PERSON_ID – cizí klíč odkazující na svázanou osobu

Tabulka RESEARCH_GROUP

Tabulka s informacemi o výzkumných skupinách v systému.

- RESEARCH_GROUP_ID – primární klíč
- OWNER_ID – odkaz na vlastníka skupiny (osoba)
- TITLE – název skupiny
- DESCRIPTION – popis skupiny
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

Tabulka RESEARCH_GROUP_MEMBERSHIP

Tabulka vyjadřuje členství v rámci výzkumné skupiny.

- RESEARCH_GROUP_MEMBERSHIP_ID – primární klíč
- RESEARCH_GROUP_ID – odkaz na skupinu, ke které je daný člen přiřazen
- PERSON_ID – odkaz na osobu
- AUTHORITY – práva pro daný typ členství

Tabulka SCENARIO

Tabulka popisuje scénář experimentu.

- SCENARIO_ID – primární klíč
- OWNER_ID – odkaz na zakladatele scénáře (osoba)
- RESEARCH_GROUP_ID – odkaz na skupinu, ve které se daný scénář vyskytuje
- TITLE – název scénáře
- SCENARIO_LENGTH – doba trvání scénáře
- DESCRIPTION – popis scénáře
- PRIVATE – zda je scénář veřejný nebo soukromý
- SCENARIO_NAME – název souboru se scénářem
- MIMETYPE – formát souboru
- FILE_CONTENT – samotný soubor scénáře uložený jako BLOB
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

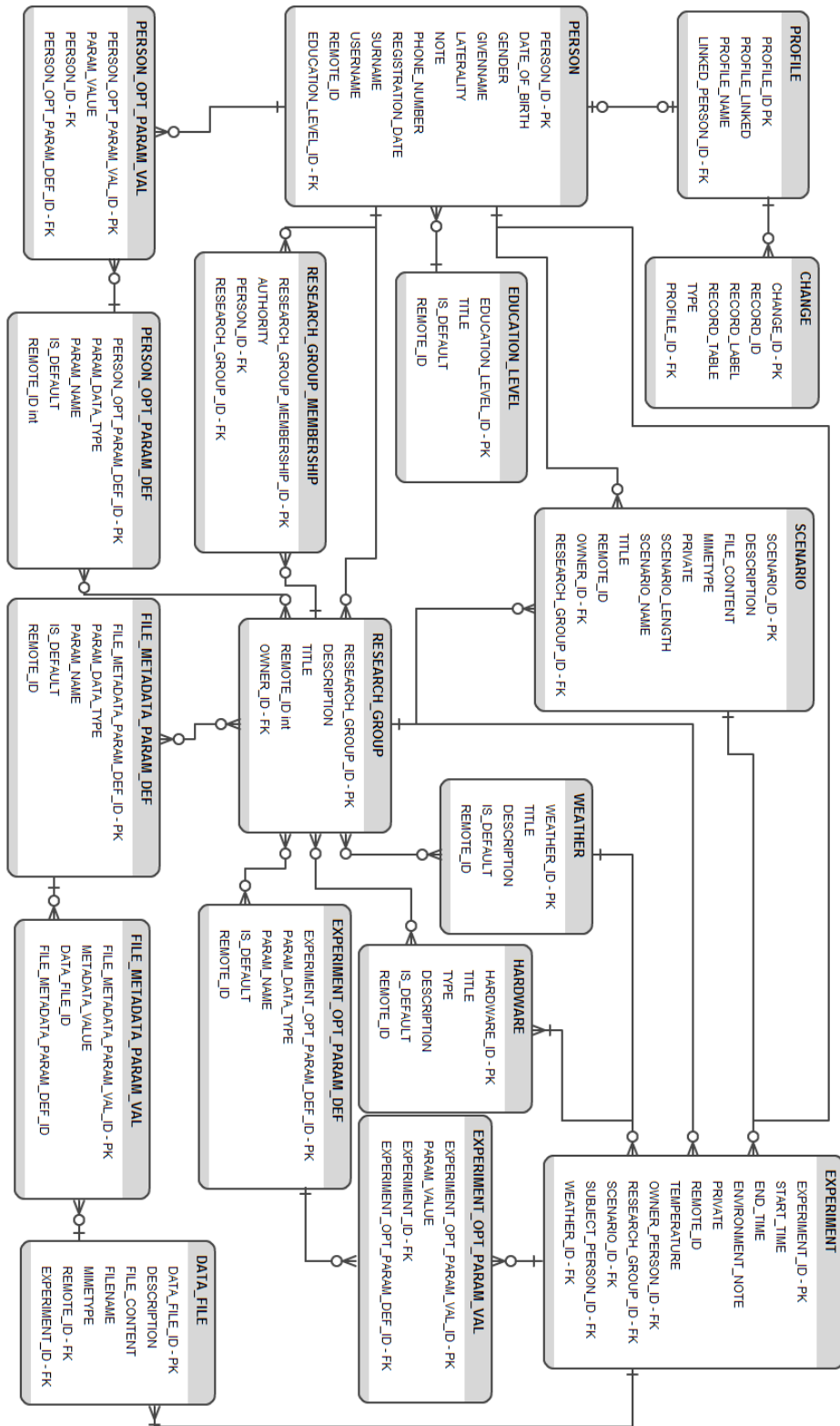
Tabulka WEATHER

Číselník, který ukládá informace o počasí při měření.

- WEATHER_ID – primární klíč
- TITLE – název pro počasí
- DESCRIPTION – popis počasí
- WEATHER_GROUP_REL – M:N rozkladová tabulka mezi RE-SEARCH_GROUP a touto tabulkou
- IS_DEFAULT – hodnota 1, pokud je záznam společný pro všechny skupiny, jinak 0
- REMOTE_ID – nový sloupec s odkazem na primární klíč stejného záznamu na serveru (pokud existuje)

6.6.2 ERA Diagram

Na obrázku 6.3 je znázorněn ERA (Entity Relationship Attribute) model klientské data-báze.



Obrázek 6.3: ERA diagram

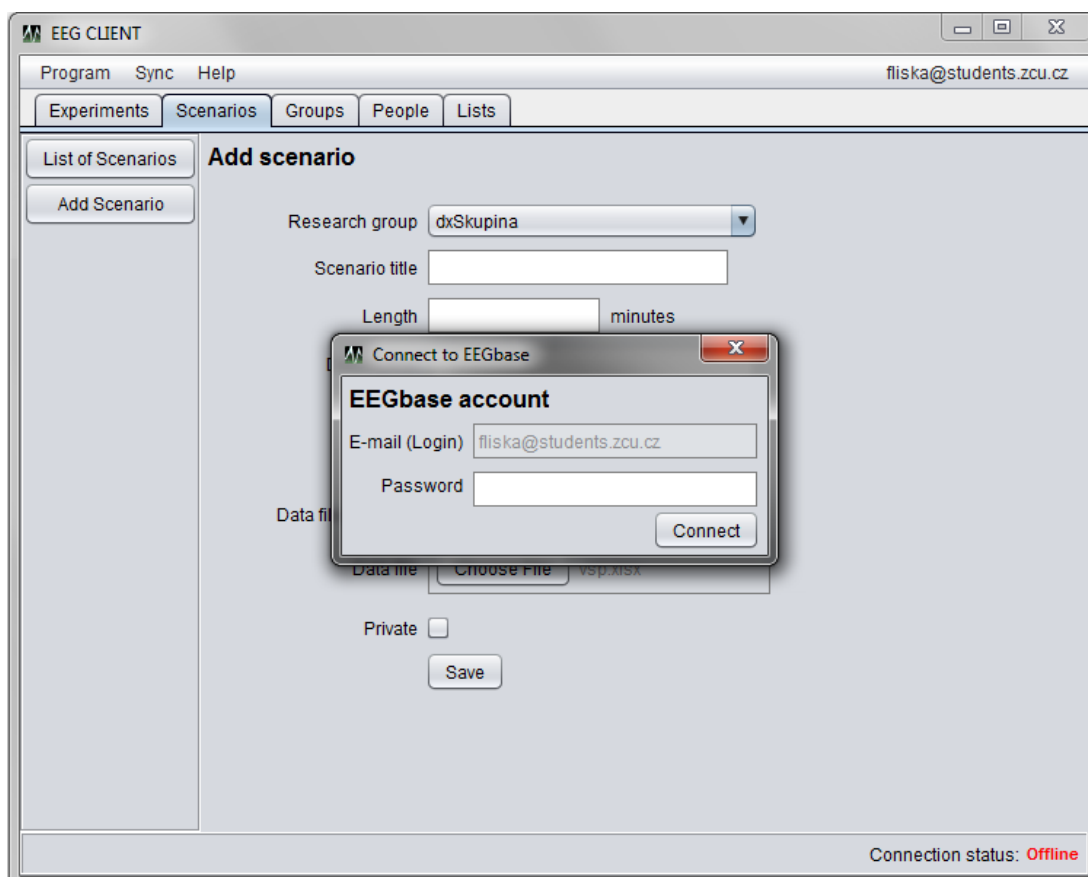
7 Implementace

Tato kapitola popisuje technologie použité při tvorbě klienta, strukturu kódu serveru a klienta, komponenty grafického rozhraní, validaci uživatelských vstupů a servisní objekty použité při realizaci přenosu dat mezi klientem a serverem.

7.1 Použité prostředky

Hlavním kritériem pro výběr použitých technologií byla jednoduchost vstupu nového programátora do projektu při znalosti technologií serveru. Výhodou tohoto přístupu je to, že některé serverové části kódu jsou znovupoužitelné v klientovi. Programátoři, kteří budou pokračovat na tomto projektu, budou především z řad studentů Fakulty aplikovaných věd Západočeské univerzity v Plzni, proto byl jako programovací jazyk zvolen Java SE 1.6 (Java Platform, Standard Edition).

Uživatelské rozhraní klienta je postaveno na knihovně Swing, která je součástí Javy. Jako standardní vzhled pro aplikaci je použit vzhled Nimbus (viz Obrázek 7.1), který byl představen v Java SE 1.6 update 10.



Obrázek 7.1: Vzhled Nimbus

Jako embedded databázi jsem zvolil Apache Derby 10, jelikož je open source, jednoduchá na instalaci a zahrnuje v sobě i JDBC (Java Database Connectivity) driver. Sama také zabírá málo místa na disku (kolem 2.6 MB) a je snadná na použití [5].

Následující technologie se z velké části nachází i na serveru, proto jsem se rozhodl je použít i v klientovi. O objektově relační mapování se stará framework Hibernate 3.6, který je dnes standardem mezi ORM frameworky používanými společně s Javou. Umožňuje například vytvářet novou databázi (při neexistenci) po spuštění klienta. Výhodou je například i to, že lze většinu POJO objektů a některých částí DAO objektů dostupných na serveru použít i v klientovi. V konfiguračním souboru lze jednoduše změnit databázi, kterou klient používá, bez nutnosti většího zásahu do kódu [11].

Klient (stejně jako server) je postaven na frameworku Spring (ve verzi 3.2.2), který odstraňuje závislosti POJO objektů a vrstev použitím návrhového vzoru IoC (Inversion of Control). V modulu ORM obsahuje podporu pro framework Hibernate. Spring usnadňuje konfiguraci různých komponent, jelikož sdružuje jejich konfigurace na jedno místo [17].

O závislosti na externích knihovnách se stará Apache Maven 4 [6]. Maven je použit pro automatizaci buildu projektu, pro generování kódu pro třídy definující požadavky a odpovědi u webových služeb prostřednictvím knihovny Apache CXF [4].

7.2 Struktura aplikace

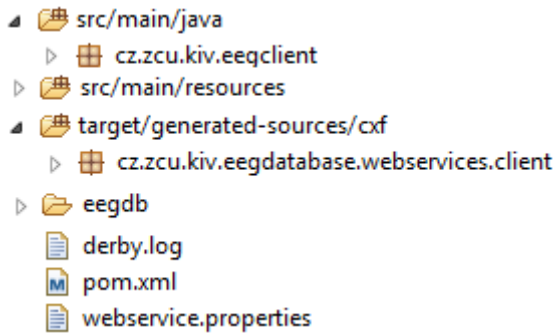
Aplikace je rozdělena na dvě části. První část je implementace klienta, druhá část je implementace webové služby na serveru.

7.2.1 Server

Implementace a rozhraní serverové části (projekt *eegdatabase*) se nachází v balíku *cz/zcu/kiv/eegdatabase/webservices/client*, který obsahuje balík s kontejnery pro posílání data (balík *wrappers*), rozhraní webové služby (*ClientService.java*) pro přidávání/čtení dat ze serverové databáze a její implementaci.

7.2.2 Klient

Projekt *eegclient* je ve formátu typickém pro maven projekty (viz Obrázek 7.2).

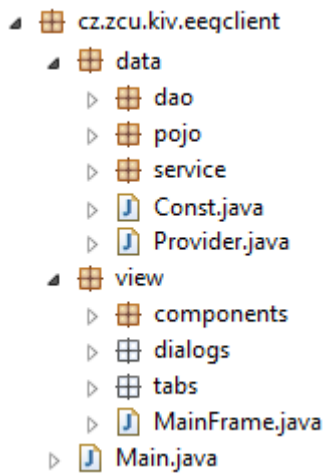


Obrázek 7.2: Struktura aplikace (klient)

Následuje popis balíků klienta a jejich význam:

- `src/main/java` – Obsahuje hlavní balík `cz/zcu/kiv/eegclient`, který má uvnitř další balíky a soubory s příponou `.java`.
- `src/main/resources` – Obsahuje konfigurační soubory, které jsou při buildu projektu součástí `jar` souboru.
- `target/generated-sources/cxf` – Obsahuje třídy vygenerované z WSDL přes Apache CXF plugin. Třídy jsou standardně generovány do balíku s obdobným umístěním jako na serveru (`cz/zcu/kiv/eegdatabase/webservices/client`).
- `eegdb` – Adresář s vytvořenou Derby databází.
- `derby.log` – Logovací soubor pro databázi Derby.
- `pom.xml` – Hlavní konfigurační soubor mavenu.
- `webservice.properties` – Konfigurační soubor pro endpoint webové služby.

Hlavní balík *cz.zcu.kiv.eegclient* je rozdělen na dvě části (viz Obrázek 7.3). Balík *data* obsahuje datovou a servisní vrstvu. Balík *view* obsahuje komponenty frameworku a jednotlivé části uživatelského rozhraní.



Obrázek 7.3: Struktura hlavního balíku (klient)

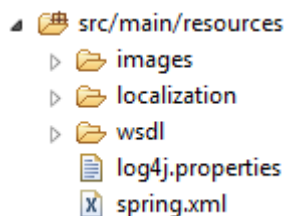
Následuje popis jednotlivých částí hlavního balíku:

- *data/dao* – Obsahuje rozhraní a implementace přístupových objektů pro záznamy uložené v databázi.
- *data/pojo* – Jednoduché objekty reprezentující tabulkové záznamy v databázi.
- *data/service* – Obsahuje rozhraní a implementace servisních objektů.
- *data/Const.java* – Třída obsahující důležité konstanty programu.
- *data/Provider.java* – Třída obsahující stavové proměnné klienta (typu přihlášený uživatel, nastavený jazyk). Poskytuje klientskou část pro připojení k webovým službám a aplikační kontext.
- *view/components* – Obsahuje veškeré části frameworku pro prezentační vrstvu (viz kapitola 7.3).
- *view/dialogs* – Balík obsahující dialogy hlavního okna programu (pro připojení k serveru, stahování, nahrávání dat a přihlášení se na profil).
- *view/tabs* – Obsahuje balíky s jednotlivými záložkami ve struktuře kopírující používání programu.
- *view/MainFrame.java* – Třída s hlavním oknem programu.
- *Main.java* – Třída pro spuštění a zobrazení hlavního okna programu.

Servisní vrstva (implementovaná v balíku *data/service*) obsahuje rozhraní:

- *DataService* – Obsahuje metody pro manipulaci s daty uloženými v lokální databázi (čtení, ověřování existence, přidávání).
- *DownloadService* – Poskytuje metodu pro stažení dat ze serveru.
- *UploadService* – Poskytuje metodu pro nahrání změn na server (pro konkrétní profil).
- *WebServiceClient* – Klient pro připojení k webové službě serveru (*ClientService*).

Konfigurační soubory jsou uloženy v *src/main/resources* (viz Obrázek 7.4: **Struktura konfiguračních souborů**).



Obrázek 7.4: Struktura konfiguračních souborů

Následuje popis struktury konfiguračních souborů:

- *images* – Obrázky programu (např. logo).
- *localization* – Obsahuje soubory s texty, které se vypisují v programu v konkrétním jazyce (standardně v anglickém).
- *wsdl* – Obsahuje WSDL dokument (získaný ze serveru), z kterého se generují data pro servisní objekty (přes *mvn generace-sources*).
- *log4j.properties* – Nastavení logování programu.
- *spring.xml* – Konfigurační soubor pro Spring.

7.3 Komponenty grafického rozhraní

V balíku *view/components* se nachází framework pro uživatelské rozhraní postavený nad knihovnou Swing. Balík obsahuje přednastavené komponenty, které dědí od komponent z balíku *javax.swing.JComponent* (lze tedy používat i rodičovské komponenty). Komponenty frameworku jsou označeny prefixem *Custom*. Usnadňují tvorbu uživatelského rozhraní klienta. Cílem je minimalizovat množství potřebného kódu pro jejich použití. Balík obsahuje rozšířené verze standardních komponent (např. Button, Table, Text Field aj.), ale i komponenty nové.

Následuje výčet nových komponent:

CustomPage

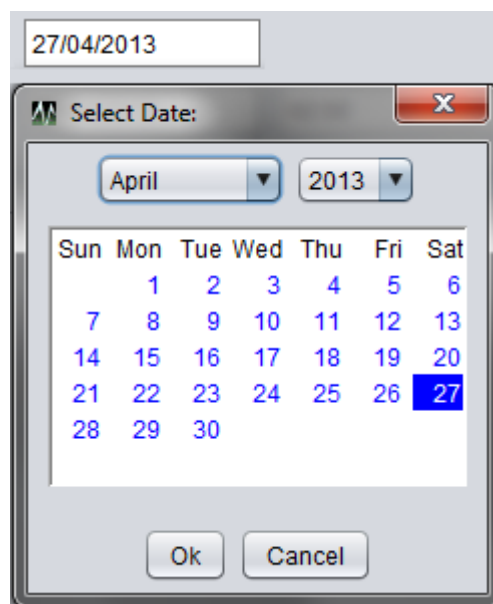
- Představuje panel pro konkrétní stránku na záložce (např. *Add experiment*).

CustomTab

- Reprezentuje panel pro záložku (např. *Experiments*), který umožňuje volbu konkrétní stránky z dané záložky.

CustomDateField

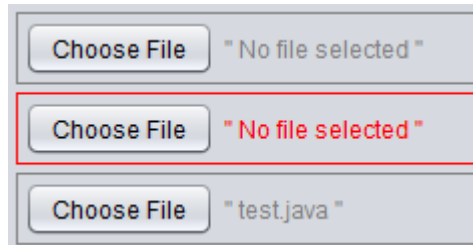
- Speciální typ textového pole, které po kliknutí zobrazí dialog s kalendářem pro výběr data (viz Obrázek 7.5).



Obrázek 7.5: Pole pro výběr data

CustomFileChooser

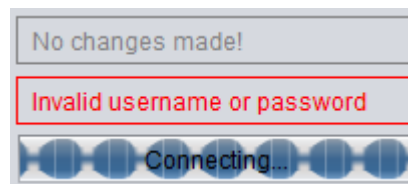
- Panel pro výběr souboru (*JFileChooser*) s textovým popiskem přizpůsobeným pro validaci (viz Obrázek 7.6).
- V případě dlouhého názvu souboru je text zkrácen.



Obrázek 7.6: Panel pro výběr souboru

CustomInfoPanel

- Panel pro zobrazení informací (defaultně skrytý).
- Poskytuje metodu pro zobrazení informačního textu, chybového výpisu či *JProgressBar* s informací (viz Obrázek 7.7).



Obrázek 7.7: Druhy výpisů informačního panelu

CustomLink

- Tlačítko, které vzhledově napodobuje hypertextový odkaz (viz Obrázek 7.8).

[View detail of scenario](#)

Obrázek 7.8: Tlačítko se vzhledem odkazu

CustomAction

- Jedná se o abstraktní akci, která nejprve provede validaci, poté nastaví parametry, provede akci (při korektním vstupu) v samostatném vlákně a nakonec zobrazí výsledek/chybový výpis.

7.3.1 Validace uživatelských vstupů

Pro validaci uživatelského vstupu slouží třída *Validator*. Chybový výpis zobrazuje do *CustomInfoPanel* objektu, který je zadán jako parametr u konkrétní metody validace. Validátor poskytuje validaci pro dvě různé skupiny komponent.

Do první skupiny spadají komponenty, do kterých lze zadat textový vstup. Tyto prvky implementují rozhraní *Verifiable*. V případě nevalidního vstupu se okraj *Verifiable* komponenty zbarví dočervena.

U *Verifiable* komponent lze validovat:

- neprázdný text
- prázdný/neprázdný text vyhovující regulárnímu výrazu
- záznam existující v databázi
- záznam neexistující v databázi

Validátor poskytuje základní regulární výrazy pro vstupy složené z písmen, číslic, bez mezery, s mezerou, pro telefonní číslo, email, datum a čas.

Rozhraní *Verifiable* implementují komponenty:

- *CustomTextField*
- *CustomDateField*
- *CustomPasswordField*
- *CustomTextArea*

Druhou skupinu tvoří komponenty implementující rozhraní *Selectable*. Pro tuto skupinu lze kontrolovat, zda je v komponentě vybrán záznam.

Rozhraní *Selectable* implementují komponenty:

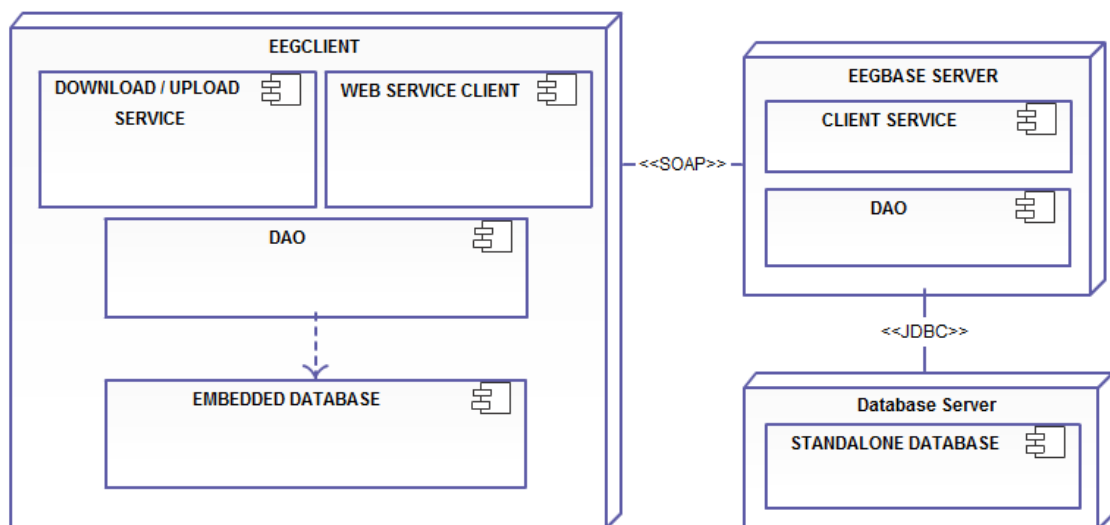
- *CustomFileChooser*
- *CustomList*
- *CustomTable*

Speciálním případem je *CustomComboBox*, který implementuje obě rozhraní a lze validovat všemi zmíněnými způsoby.

7.4 Přenos dat a servisní objekty

Přenos dat mezi serverem a klientem je řešen přes protokol SOAP. Ten je již na serveru implementován kvůli komunikaci s jinými projekty. Posílaná data se před odesláním vkládají do obálky.

Na serverové straně se nachází rozhraní webové služby s názvem *ClientService* (viz Obrázek 7.9), jejíž implementace obsahuje metody pro ukládání konkrétních záznamů do serverové databáze a metody pro vrácení seznamů s obálkami požadovaných záznamů.

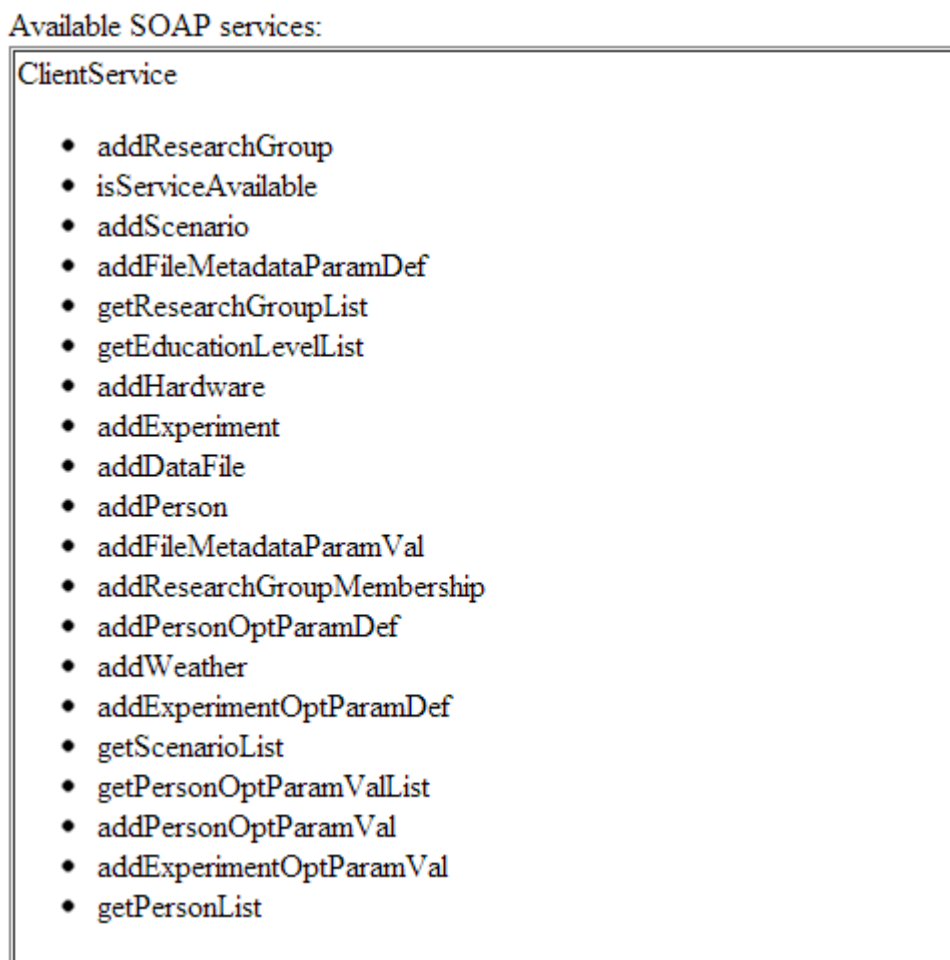


Obrázek 7.9: Diagram nasazení

Na klientské straně se nachází *WebServiceClient*, který se stará o připojení k webové službě serveru a poskytuje serverovou službu *ClientService* pro servisní objekty klienta. Implementace rozhraní *DownloadService* poskytuje metody pro stažení dat ze serveru. O opačný přenos, tedy z klienta na server, se stará *UploadService*, která ukládá přenesené informace do databáze na serveru. Obě tyto služby využívají k operacím nad klientskou databází třídu *DataService*, která operuje nad DAO konkrétních POJO objektů.

Implementace rozhraní *ClientService* je zveřejněna jako webová služba na serveru použitím anotace `@javax.jws.WebService`. Tato anotace obsahuje parametr (*endpointInterface*), jehož hodnota je `cz.zcu.kiv.eegdatabase.webservices.client.ClientService`.

Po spuštění serveru lze zobrazit registr poskytovaných webových služeb (UDDI) na adrese `http://,,adresa_serveru:port“/EEGDatabase/webservice/` (viz Obrázek 7.10).



Obrázek 7.10: UDDI registr EEG/ERP portálu

Ke každé poskytované SOAP webové službě je k dispozici WSDL dokument, který je dostupný z hypertextového odkazu uvedeného v UDDI registru u příslušné služby. Je zde uvedena i přístupová adresa k dané službě a její jmenný prostor (viz Obrázek 7.11).



Obrázek 7.11: Informace o *ClientService* v UDDI registru

8 Testování

Fáze testování zahrnovala vytvoření funkčních testů, které mapují všechny případy užití. Tyto testy byly předloženy třem různým testerům s různým stupněm vzdělání v oblasti počítačů. Testeré na jejich základě pracovali samostatně podle předem připraveného postupu. Jejich poznatky a případné dotazy byly zaznamenány. Na základě těchto testů byly provedeny opravy jak funkční, tak i úpravy usnadňující práci s programem.

Skupina testerů vypadala následovně:

- žena - 21 let, studentka vysoké školy (humanitně zaměřené), základní znalost kancelářských programů typu Word a Excel
- muž - 27 let, programátor, vystudované technické vzdělání se zaměřením na informatiku, dobrá znalost programování a návrhu uživatelského rozhraní
- muž - 26 let, zaměstnanec ve strojírenské firmě, vystudované technické vzdělání zaměřené na strojírenství, dobrá znalost práce s počítačem

Následuje výpis testů funkčnosti. Každý test má vlastní název, cíl, předpoklady a postup. Testy na sebe navazují, takže splnění předpokladů samotného testu je podmíněno splněním testů předcházejících. Pod každým testem jsou uvedeny poznatky od testerů a jejich řešení v rámci programových oprav. Anglické názvy jednotlivých komponent uživatelského rozhraní programu jsou značeny kurzívou.

8.1 Test svázání profilu s účtem

Cílem testu je ověření funkčnosti připojení k serveru (vytvoření profilu) a stažení dat potřebných pro off-line zadávání nových experimentů.

Předpoklady:

- funkční připojení k serveru, vytvořený účet na serveru

Postup:

1. Spustíte program *eegclient.bat*.
2. Zobrazí se přihlašovací dialog s výběrem profilů.
 - 2.1 Stiskněte tlačítko *New Profile*.
3. Zobrazí se dialogové okno pro připojení k serveru.

- 3.1 Vyplňte přihlašovací jméno a heslo.
- 3.2 Stiskněte tlačítko *Connect*.
4. Při úspěšném připojení se zobrazí dialogové okno potvrzující připojení (*Connection status* v pravém dolním rohu se změní na *Online*), v opačném případě je vyhozeno chybové hlášení. Budete také automaticky přihlášení na nově vytvořený profil (viz pravý horní roh menu) s názvem shodným s účtem na serveru.
5. Stiskněte tlačítko *Ok* a zobrazí se dialogové okno pro stažení dat ze serverového účtu.
6. Stiskněte tlačítko *Download*.
7. Zobrazí se ukazatel postupu, který vypisuje stahování dat ze serveru.
8. Po úspěšném stažení dat se zobrazí dialog s potvrzením o svázání profilu s účtem.
9. Stiskněte tlačítko *Ok* a všechny záložky se změní na označitelné.
10. Ověřte, zda se data (vyjma experimentů) přenesla ze serveru do klienta.
 - 10.1 Klikněte na záložku *Scenarios* a ověřte existenci scénářů (zda jsou zobrazené v seznamu).
 - 10.2 Klikněte na záložku *Groups* a ověřte existenci výzkumných skupin.
 - 10.3 Klikněte na záložku *People* a ověřte existenci uživatelů/testovaných subjektů.
 - 10.4 Klikněte na záložku *Lists* a ověřte existenci číselníků.
 - 10.4.1 V levém menu projděte všechny možnosti.
 - 10.4.2 V případě, že se vaše osoba nachází ve více skupinách, ověřte existenci číselníků pro všechny skupiny výběrem z roletového menu u všech číselníků.

Výsledky testování:

Všechna data se testerům správně uložila pod jejich profil. Jeden z testerů objevil chybu, kdy je možné zadat do textového pole pro připojení text o neomezené délce znaků. Opravou bylo nastavení povoleného defaultního počtu znaků s možností změny pro konkrétní textové pole.

8.2 Test přidání dat pro experiment

Cílem testu je ověření funkčnosti přidání nového scénáře, výzkumné skupiny, osoby a číselníků v případě, kdy není k dispozici připojení k serveru.

Předpoklady:

- přihlášení na profil, svázaný profil

Postup:

1. Pokud jste připojeni k serveru, odpojte se přes menu *Sync* kliknutím na *Disconnect*.
2. V záložce *Groups* stiskněte tlačítko *Create group* v levém panelu.
 - 2.1. Zobrazí se stránka pro přidání výzkumné skupiny.
 - 2.2. Vyplňte název a popis skupiny.
 - 2.3. Stiskněte tlačítko *Save*.
 - 2.4. Zobrazí se stránka s výpisem skupin a nově přidanou skupinou.
3. V záložce *Scenarios* stiskněte tlačítko *Add Scenario* v levém panelu.
 - 3.1. Zobrazí se stránka pro přidání scénáře.
 - 3.2. Vyplňte název, dobu trvání a popis scénáře.
 - 3.3. Zaškrtněte přidání souboru a zvolte soubor tlačítkem *Choose File*.
 - 3.4. Stiskněte tlačítko *Save*.
 - 3.5. Zobrazí se stránka s výpisem scénářů a nově přidaným scénářem.
4. V záložce *People* stiskněte tlačítko *Add Person* v levém panelu.
 - 4.1. Zobrazí se stránka pro přidání osoby.
 - 4.2. Vyplňte jméno, příjmení a email osoby (ostatní volitelné).
 - 4.3. Stiskněte tlačítko *Save*.
 - 4.4. Zobrazí se stránka s detailem nově vytvořené osoby.
 - 4.5. Klikněte na odkaz *Detail* v tabulce u vytvořené osoby.
 - 4.5.1. Zobrazí se stránka s informacemi o osobě.
 - 4.5.2. Stiskněte tlačítko *Add optional parameter*.

- 4.5.3. Zobrazí se stránka pro přidání parametru pro osobu.
 - 4.5.4. Zvolte vytvořený datový typ a vyplňte hodnotu.
 - 4.5.5. Stiskněte tlačítko *Add optional parameter*.
 - 4.5.6. Zobrazí se stránka s informacemi o osobě s přidáním parametrem v tabulce *Optional Parameters*.
5. V záložce *Lists* stiskněte tlačítko *Hardware* v levém panelu.
 - 5.1. Zobrazí se stránka s výpisem hardwaru.
 - 5.2. Zvolte vytvořenou výzkumnou skupinu z roletového menu.
 - 5.3. Stiskněte tlačítko *Add hardware definition*.
 - 5.4. Vyplňte název, typ a popis hardwaru a stiskněte tlačítko *Save*.
 - 5.5. Zobrazí se výpis s nově vytvořeným záznamem.
 6. Obdobným způsobem jako v bodě 4 přidejte záznamy pro všechny další položky z levého menu v záložce *Lists*.

Výsledky testování:

Při testování se přišlo na možnost optimalizace ovládání aplikace, jejímž cílem bylo odstranění přebytečné uživatelské interakce pro rychlejší dosažení požadované informace. Při procházení seznamů *Lists*, ve kterých se nachází roletové menu pro výběr výzkumné skupiny, se vždy vracel výběr této skupiny na první skupinu z daného menu při přechodu na jiný seznam. Nově si každý seznam pamatuje vybranou skupinu i po přechodu na jiný seznam, takže při přidávání dat pro konkrétní skupinu se proces značně urychlil.

Jedna z poznámek od testera byla, že při přidávání osoby by přednastavená hodnota pro *Laterality* měla být *right* namísto *left*, jelikož je skupina lidí s dominantní pravou rukou větší [3].

Další zjištěný poznatek se týkal řazení výsledků v seznamu osob, kdy nebyly výsledky řazeny abecedně (podle příjmení), tudíž se špatně vyhledávala konkrétní osoba. Vzhledem k tomuto poznatku byla opravena i řazení záznamů v jiných částech aplikace.

Byly také zjištěny problémy s validací jednotlivých textových polí a tyto problémy byly odstraněny.

8.3 Test přidání experimentu

Cílem testu je ověřit funkčnost přidání experimentu a jeho dodatečných parametrů v případě, že není k dispozici připojení k serveru.

Předpoklady:

- přihlášení na profil, svázaný profil, alespoň jeden záznam vytvořený v každé záložce (scénář, výzkumná skupina, lidé, skupiny, číselníky) vyjma experimentů

Postup:

1. V záložce *Experiments* stiskněte tlačítko *Add experiment* v levém panelu.
2. Zobrazí se první stránka pro přidání experimentu.
 - 2.1 Zvolte výzkumnou skupinu, nastavte datum a čas pro začátek/konec experimentu a zvolte testovanou osobu.
 - 2.2 Stiskněte tlačítko *Next*.
3. Zobrazí se druhá stránka pro přidání experimentu.
 - 3.1 Zvolte scénář experimentu, hardware (přes ctrl + kliknutí lze označit více najednou) a počasí během experimentu. Ostatní vyplňte libovolně.
 - 3.2 Stiskněte tlačítko *Next*.
4. Zobrazí se třetí stránka pro přidání souboru k experimentu.
 - 4.1 Zvolte frekvenci vzorkování souborů.
 - 4.2 Zvolte soubor na disku přes tlačítko *Choose File*.
 - 4.3 Klikněte na odkaz *Attach next file* a opakujte bod 4.2.
 - 4.4 Stiskněte tlačítko *Finish*.
5. Zobrazí se stránka *My experiments* s nově přidaným experimentem.
 - 5.1 Klikněte na odkaz *Detail* u vytvořeného experimentu v seznamu experimentů.
 - 5.2 Ověřte existenci informací zadaných v předchozích bodech.
 - 5.3 Stiskněte tlačítko *Add optional parameter*.
 - 5.4 Zobrazí se stránka pro přidání dodatečného parametru k experimentu.

- 5.4.1 Zvolte typ parametru a vyplňte jeho hodnotu.
- 5.4.2 Stiskněte tlačítko *Add optional parameter*.
- 5.5 Nový parametr se zobrazí v detailu experimentu v tabulce *Optional parameters*.
- 5.6 V detailu experimentu stiskněte tlačítko *Add data file*.
 - 5.6.1 Opakujte scénář z bodu 4.1 a 4.2.
 - 5.6.2 Stiskněte tlačítko *Add data file*.
- 5.7 Přidaný soubor se zobrazí v detailu experimentu v tabulce *Data files*.
- 5.8 Klikněte na odkaz *Detail* u vytvořeného souboru.
- 5.9 Zobrazí se stránka pro přidání metadat.
 - 5.9.1 Zvolte typ parametru a vyplňte jeho hodnotu.
 - 5.9.2 Stiskněte tlačítko *Add*.
 - 5.9.3 Vytvořený parametr se zobrazí v tabulce *Metadata*.

Výsledky testování:

U tohoto testu byla odhalena funkční chyba, kdy se nesprávným způsobem ukládal atribut *Weather note* při vkládání poznámky o počasí k novému experimentu.

Dále byl odhalen přebytečný panel pro zobrazení stejné chybové informace, jelikož komponenta *CustomFileChooser* již informační panel obsahuje. Tento panel byl tedy odstraněn a chyba u počasí opravena.

8.4 Test přehrání změn na server

Cílem testu je ověřit funkčnost nahrání dat (včetně souborů) přidanych v předchozích testech z klienta na server.

Předpoklady:

- přihlášení na profil, svázaný profil, alespoň jeden záznam vytvořený v každé záložce v klientovi

Postup:

1. Připojte se k serveru přes menu *Sync* zvolením položky *Connect*.
 - 1.1 Vyplňte heslo a stiskněte tlačítko *Connect*.
2. Zvolte položku *Upload Changes* z horního menu *Sync*.
 - 2.1 Zobrazí se dialog pro nahrání změn na server s tabulkou, ve které jsou vypsané jednotlivé změny provedené na klientovi.
 - 2.2 Stiskněte tlačítko *Upload*.
 - 2.3 Zobrazí se ukazatel postupu nahrávání dat na serveru. Při nahrávání budou jednotlivé položky ze seznamu změn postupně mizet.
 - 2.4 Po úspěšném nahrání změn se zobrazí dialog s potvrzením.
 - 2.5 Stiskněte tlačítko *Ok*.
3. Přejděte na stránku EEG/ERP portálu ve webovém prohlížeči.
 - 3.1 Přihlaste se na serverový účet, se kterým je svázaný profil v klientovi.
4. Zvolte záložku *Experiments* a v levém menu zvolte *My experiments*.
 - 4.1 Ověřte, že vytvořený experiment z klienta je v tabulce *My experiments*.
 - 4.2 Klikněte na odkaz *Detail* u vytvořeného experimentu.
 - 4.3 Ověřte existenci zadaných údajů.
 - 4.4 Klikněte na odkaz *Detail* u souboru, u kterého jste přidali metadata.
 - 4.4.1 Ověřte existenci parametru experimentu zadaného v klientovi.
 - 4.4.2 Stiskněte tlačítko *Download file*.

4.4.3 Zkontrolujte, zda stažený soubor odpovídá původně nahranému souboru z klienta.

5. Obdobně jako v klientovi zkontrolujte vytvořené záznamy v záložkách *Scenario*, *Research Group*, *Lists* a *People*.

6. Ověřte existenci parametru vytvořené osoby z klienta kliknutím na odkaz *Detail* u osoby v záložce *People*.

Výsledky testování:

V tomto testu byly nalezeny chyby zejména z hlediska rychlosti algoritmu nahrání dat na server. Algoritmus zjišťoval veškeré změny ze všech tabulek postupně (vzhledem ke stanovenému pořadí přenosu tabulek k zajištění správných vazeb mezi daty), takže množství dotazů do databáze bylo větší, než je zapotřebí. Řešením bylo porovnávat existenci názvu tabulky aktuální odesílané změny v seznamu změn, který zahrnoval ještě neodeslané změny (při zachování stanoveného pořadí). Nárůst celkové rychlosti přenosu změn byl velmi znatelný.

Také bylo znemožněno stisknout tlačítko pro přenos, pokud žádné změny nenastaly.

9 Zhodnocení dosažených výsledků

V průběhu realizace programu jsem se z velké části učil nové technologie, takže jsem nejprve vyvinul prototyp, který jsem posléze kompletně přeprogramoval. Po vytvoření prototypu se objevilo několik nedostatků zejména v nízké rychlosti použitých algoritmů, které byly následně optimalizovány. Také bylo nutné stažení vstupních dat ze serveru po založení nového profilu, proto byl systém profilů zcela přeprogramován s doplněním přenosu dat oběma směry.

U synchronizace dat mezi klientem a serverem bylo potřeba vyřešit problém s primárními klíči jednotlivých záznamů. Webové služby nabízely v tomto ohledu řešení problému zpětným přenosem primárního klíče nově vytvořeného záznamu ze serveru na klienta. Další problém se objevil v návaznosti tabulek. Ten byl úspěšně vyřešen zvolením vhodného pořadí tabulek při přenosu dat.

Nejvíce chyb bylo testery odhaleno při validaci uživatelských vstupů, proto jsem se rozhodl vytvořit vlastní systém validace, který je velmi snadno modifikovatelný podle potřeby. Nově navržené komponenty nad knihovnou Swing značně urychlují vytváření nových stránek, usnadňují validaci a výpis informací pro uživatele.

Struktura aplikace je navržena tak, aby jí bylo jednoduché porozumět a dále rozšiřovat. Stávající aplikace umožňuje pouze přidávání nových záznamů. Je možné ji rozšířit o editaci či mazání, ale je nutné řešit i synchronizaci těchto akcí.

10 Závěr

Hlavním cílem této diplomové práce bylo navrhnout a implementovat klienta k webovému EEG/ERP portálu, který poskytuje možnost zadávat neuroinformatická data při absenci internetového připojení a umožňuje tato data dávkovým způsobem přenést do webového portálu.

Při realizaci práce jsem se nejprve seznámil s návrhem EEG/ERP portálu a specifikoval jsem rozsah funkcí, které bylo nezbytné implementovat v klientovi. Při návrhu grafického rozhraní klienta jsem se inspiroval vzhledem EEG/ERP portálu, čímž je ovládání klienta pro uživatele portálu velmi intuitivní.

Nastudoval jsem obecné principy webových služeb, které jsou vhodným řešením přenosu dat mezi klientem a webovým portálem. Implementace zahrnovala vytvoření webové služby na serveru a klienta, který je schopen se k ní vzdáleně připojit a používat její vystavené funkce. Klient po připojení k webové službě poskytuje možnost obousměrné synchronizace dat. Při založení nového profilu v klientovi je nutné připojení k webovému portálu z důvodu prvotní synchronizace dat s uživatelským účtem. Tím jsou zajištěny základní vstupy pro zadávání neuroinformatických dat v klientovi i bez nutnosti připojení.

Současná verze EEG/ERP portálu se za dobu realizace této práce rozšířila o nové funkce, proto je vhodným rozšířením tyto nové funkce včetně synchronizace implementovat i do klienta.

Seznam zkratek

AOP - Aspect-oriented Programming

API - Application Programming Interface

BEEP - Blocks Extensible Exchange Protocol

CXF - Apache Celtic XFire

DAO - Data Access Object

DTO - Data Transfer Object

EEG - Electroencephalography

ERA - Entity Relationship Attribute

ERP - Event-Related Potentials

FTP - File Transfer Protocol

HTML - HyperText Markup Language

HTTP - HyperText Transfer Protocol

IoC - Inversion of Control

Java EE - Java Platform, Enterprise Edition

Java SE - Java Platform, Standard Edition

JAX-RS - Java API for RESTful Web Services

JAX-WS - Java API for XML Web Services

JDBC - Java Database Connectivity

JSP - Java Server Pages

JSTL - JavaServer Standard Tag Library

MIME - Multipurpose Internet Mail Extension

MVC - Model-View-Controller

ORM - Object-Relational Mapping

POJO - Plain Old Java Object

RPC - Remote Procedure Call

SOAP - Simple Object Access Protocol

SMTP - Simple Mail Transfer Protocol

REST - Representational State Transfer

TCP/IP - Transmission Control Protocol/Internet Protocol

UDDI - Universal Description, Discovery and Integration

URI - Uniform Resource Identifier

VM – Virtual Machine

W3C - World Wide Web Consortium

WSDL - Web Service Description Language

XML - Extensible Markup Language

Literatura

- [1] About INCF. *International Neuroinformatics Coordinating Facility* [online]. 2013 [cit. 2013-05-04]. Dostupné z: <http://www.incf.org/about/>
- [2] About Web service standards. *IBM* [online]. 2004 [cit. 2013-04-24]. Dostupné z: <http://publib.boulder.ibm.com/infocenter/radhelp/v7r5/index.jsp?topic=%2Forg.eclipse.jst.ws.doc.user%2Fconcepts%2Fcwsstandards.html>
- [3] ABRAMS, Daniel a Mark PANAGGIO. [online]. Royal Society Publishing, 2012 [cit. 2013-04-10]. Dostupné z: <http://rsif.royalsocietypublishing.org/content/early/2012/04/24/rsif.2012.0211>
- [4] Apache CXF: An Open-Source Services Framework. *Apache* [online]. 2012 [cit. 2013-04-25]. Dostupné z: <http://cxf.apache.org/index.html>
- [5] Apache Derby. *Apache* [online]. 2012 [cit. 2013-04-25]. Dostupné z: <http://db.apache.org/derby/>
- [6] Apache Maven Project. *Apache* [online]. 2012 [cit. 2013-04-25]. Dostupné z: <http://maven.apache.org/>
- [7] BAREŠ, Martin. Kognitivní evokované potenciály. *Česká a slovenská neurologie a neurochirurgie* [online]. 2011 [cit. 2013-05-04]. Dostupné z: <http://medical-dictionary.thefreedictionary.com/electroencephalogram>
- [8] BARRY, Douglas K. Web Services Explained. *Web Services and Service-Oriented Architectures* [online]. 2013 [cit. 2013-05-04]. Dostupné z: http://www.service-architecture.com/web-services/articles/web_services_explained.html
- [9] Dorland's Medical Dictionary for Health Consumers. *The Free dictionary* [online]. 2007 [cit. 2013-05-04]. Dostupné z: <http://medical-dictionary.thefreedictionary.com/electroencephalogram>
- [10] GRAND, Mark. Mime Overview. *Mindspring* [online]. 1993 [cit. 2013-05-04]. Dostupné z: <http://mgrand.home.mindspring.com/mime.html>
- [11] Hibernate advantages and disadvantages. *Rose India* [online]. 2010 [cit. 2013-04-25]. Dostupné z: <http://www.roseindia.net/hibernate/examples/hibernate-advantages-and-disadvantages.html>

- [12] MAUTNER, Pavel. Neuroinformatika. *ZČU* [online]. 2007 [cit. 2013-05-04].
Dostupné z:
<http://www.kiv.zcu.cz/studies/predmety/uir/predn/P6/Neuroinformatika.pps>
- [13] Model-View-Controller. *Object Oriented Tips* [online]. 1998 [cit. 2013-04-15].
Dostupné z: <http://ootips.org/mvc-pattern.html>
- [14] MOUČEK, Roman a Pavel MAUTNER. Pozornost řidiče při dvojí zátěži - EEG/ERP experiment. *Department of Applied Informatics* [online]. [cit. 2013-05-04]. Dostupné z: <http://dai.fmph.uniba.sk/events/kuz2009/prispevky-pdf/moucek.pdf>
- [15] NOVOTNÝ, Jiří. *EEG/ERP portal security in new technologies*. Plzeň, 2012. Diplomová práce. ZČU.
- [16] PERGLER, Jindřich. *Performance optimization and security of EEG/ERP portal*. Plzeň, 2012. Diplomová práce. ZČU.
- [17] PICHLÍK, Roman. Spring Framework. *Interval* [online]. 2005 [cit. 2013-05-04]. Dostupné z: <http://interval.cz/clanky/spring-framework-predstaveni-j2ee-lightweight-kontejneru/>
- [18] Portál EEG/ERP experimentů. *ZČU* [online]. 2011 [cit. 2013-05-04]. Dostupné z: https://www.kiv.zcu.cz/index.php?id=557&produkt_id=72
- [19] SCHAEFERLE, Martin. 4 Benefits of Object-Relational Mapping (ORM). *Learn Now Online* [online]. 2012 [cit. 2013-05-04]. Dostupné z: <http://blogs.learnnowonline.com/blog/bid/210553/4-Benefits-of-Object-Relational-Mapping-ORM>
- [20] SOAP Tutorial. *W3schools* [online]. 2013 [cit. 2013-04-30]. Dostupné z: <http://www.w3schools.com/soap/default.asp>
- [21] Spring Framework. *Spring Source* [online]. 2013 [cit. 2013-04-16]. Dostupné z: <http://www.springsource.org/spring-framework>
- [22] STRBAČKA, Martin. *Vytvoření publikačního standardu v oblasti evokovaných potenciálů*. Plzeň, 2012. Diplomová práce. ZČU.
- [23] The Java EE 5 Tutorial. *Oracle* [online]. 2010 [cit. 2013-04-15]. Dostupné z: <http://docs.oracle.com/javaee/5/tutorial/doc/bnaaw.html>

- [24] The Java EE 6 Tutorial. *Oracle* [online]. 2012 [cit. 2013-04-24]. Dostupné z:
<http://docs.oracle.com/javaee/6/tutorial/doc/giqsx.html>
- [25] Thick Client. *Tech Terms* [online]. 2013 [cit. 2013-04-23]. Dostupné z:
<http://www.techterms.com/definition/thickclient>
- [26] Web Services. SPIES, Brennan. *Ajaxonomy* [online]. 2008 [cit. 2013-04-24].
Dostupné z: <http://www.ajaxonomy.com/2008/xml/web-services-part-1-soap-vs-rest>
- [27] Web Service Architecture. *The World Wide Web Consortium* [online]. 2004
[cit. 2013-04-24]. Dostupné z: <http://www.w3.org/TR/ws-arch/>
- [28] Web Services Glossary. *W3C* [online]. 2004 [cit. 2013-04-29]. Dostupné z:
<http://www.w3.org/TR/ws-gloss/>
- [29] Web Services Tutorial. *W3schools* [online]. 2013 [cit. 2013-04-30]. Dostupné
z: <http://www.w3schools.com/webservices/default.asp>
- [30] WSDL Tutorial. *W3schools* [online]. 2013 [cit. 2013-04-30]. Dostupné z:
<http://www.w3schools.com/wsdl/default.asp>

Seznam obrázků

Obrázek 2.1: Měření EEG aktivity v různých stavech testované osoby [9]	10
Obrázek 3.1: Domovská stránka EEG/ERP portálu	11
Obrázek 3.2: Architektura portálu [15].....	13
Obrázek 3.3: Prezentační vrstva portálu	14
Obrázek 4.1: Architektura webové služby [2]	16
Obrázek 4.2: Struktura WSDL dokumentu [30].....	17
Obrázek 4.3: Příklad WSDL dokumentu [30]	18
Obrázek 4.4: Struktura SOAP zprávy [20]	19
Obrázek 4.5: Příklad obálky ve verzi 1.2 [20].....	20
Obrázek 4.6: Příklad hlavičky SOAP dokumentu [20].....	20
Obrázek 4.7: Příklad těla SOAP požadavku [20]	21
Obrázek 4.8: Příklad odpovědi na SOAP požadavek [20].....	21
Obrázek 4.9: Příklad HTTP požadavku [20]	22
Obrázek 4.10: Příklad SOAP požadavku [20]	23
Obrázek 4.11: Příklad SOAP odpovědi [20]	23
Obrázek 4.12: Přenos zprávy technologií REST [8].....	24
Obrázek 5.1: Diagram případů užití	27
Obrázek 6.1: Architektura klienta.....	28
Obrázek 6.2: Části uživatelského rozhraní	32
Obrázek 6.3: ERA diagram.....	41
Obrázek 7.1: Vzhled Nimbus.....	42
Obrázek 7.2: Struktura aplikace (klient).....	44
Obrázek 7.3: Struktura hlavního balíku (klient)	45
Obrázek 7.4: Struktura konfiguračních souborů.....	46
Obrázek 7.5: Pole pro výběr data.....	47

Obrázek 7.6: Panel pro výběr souboru.....	48
Obrázek 7.7: Druhy výpisů informačního panelu.....	48
Obrázek 7.8: Tlačítko se vzhledem odkazu	48
Obrázek 7.9: Diagram nasazení	50
Obrázek 7.10: UDDI registr EEG/ERP portálu	51
Obrázek 7.11: Informace o <i>ClientService</i> v UDDI registru.....	51
Obrázek A.1: Dialog pro přihlášení na profil	71
Obrázek A.2: Dialog pro připojení	72
Obrázek A.3: Dialog pro stažení dat.....	72
Obrázek A.4: Menu klienta.....	73
Obrázek A.5: Seznam experimentů	73
Obrázek A.6: První fáze přidání experimentu	74
Obrázek A.7: Druhá fáze přidání experimentu	74
Obrázek A.8: Třetí fáze přidání experimentu	75
Obrázek A.9: Detail experimentu	75
Obrázek A.10: Přidání parametru experimentu	76
Obrázek A.11: Detail souboru	76
Obrázek A.12: Přehled scénářů	77
Obrázek A.13: Detail scénáře	77
Obrázek A.14: Přidání scénáře	78
Obrázek A.15: Přehled výzkumných skupin	78
Obrázek A.16: Přidání výzkumné skupiny	79
Obrázek A.17: Detail výzkumné skupiny.....	79
Obrázek A.18: Přehled osob	80
Obrázek A.19: Detail osoby.....	80
Obrázek A.20: Přidání parametru osoby.....	81

Obrázek A.21: Přidání osoby.....	81
Obrázek A.22: Přehled hardwaru.....	82
Obrázek A.23: Přidání hardwaru	82
Obrázek A.24: Dialog pro nahrání změn na server	83
Obrázek A.25: Ukazatel postupu přenosu dat	83
Obrázek A.26: Nahraná data z klienta na portál.....	84

Přílohy

A. Uživatelská Dokumentace

Server

Podporovaná verze EEG/ERP portálu ve formátu *war* spolu s postupem instalace je součástí příloženého CD. Portál byl odladěn na aplikačním serveru Apache Tomcat 6. Portál spolupracuje s databázovým Oracle serverem verze 11g. Testovací databáze je dostupná po přihlášení na uživatele *EEGTEST* na adrese:

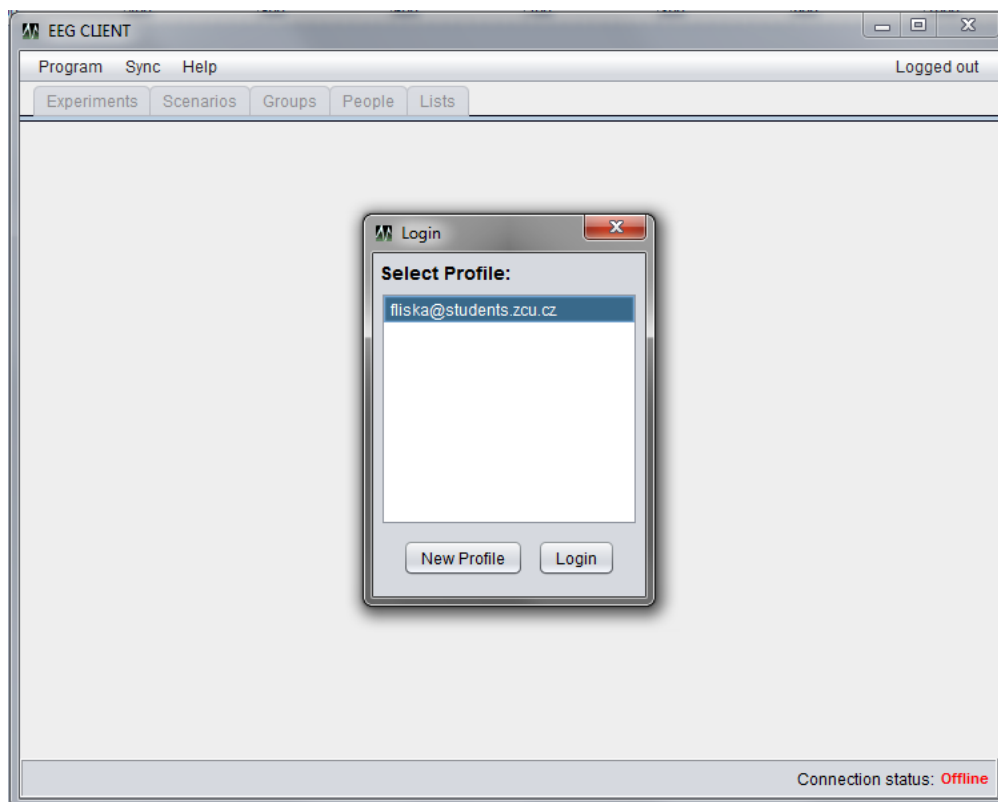
students.kiv.zcu.cz:1521/EEGERP.

Klient

Klient se spouští souborem *eegclient-1.0.bat*, který je spolu se zdrojovými kódy a textem práce součástí příloženého CD. Pro správnou funkci klienta je nutné mít vytvořený účet na serveru. Zdrojové kódy klienta jsou dostupné i v repositáři na adrese:

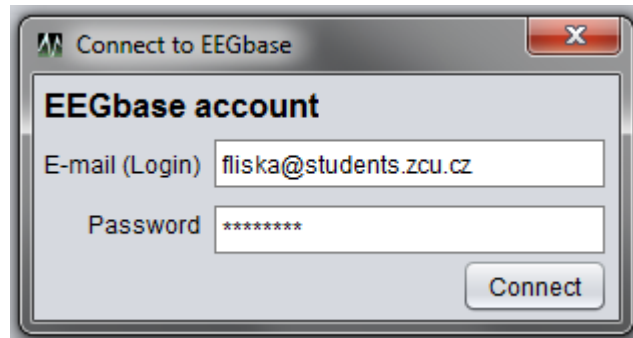
https://svn.code.sf.net/p/eegclient/code/trunk

Po spuštění klienta je zobrazen dialog s výběrem profilu (viz Obrázek A.1). Pokud máte již založený profil, pak se lze přihlásit výběrem v seznamu a stisknutím tlačítka *Login*. V opačném případě je nejprve nutné vytvořit profil stisknutím tlačítka *New Profile*.



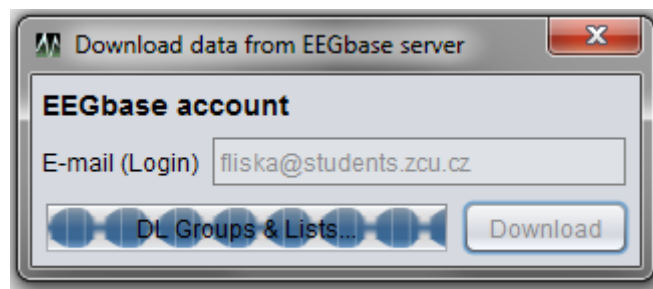
Obrázek A.1: Dialog pro přihlášení na profil

Po stisknutí se zobrazí dialog pro připojení k EEG/ERP portálu (viz Obrázek A.2). Pro připojení k serveru je nutné zadat uživatelské jméno a heslo registrovaného účtu na portálu. Po připojení je automaticky vytvořen nový profil v klientovi.



Obrázek A.2: Dialog pro připojení

Zobrazí se dialog pro stažení vstupních dat (viz Obrázek A.3). Po stisknutí tlačítka *Download* začne stahování, jehož postup se zobrazí spolu s popisem stahovaných dat.



Obrázek A.3: Dialog pro stažení dat

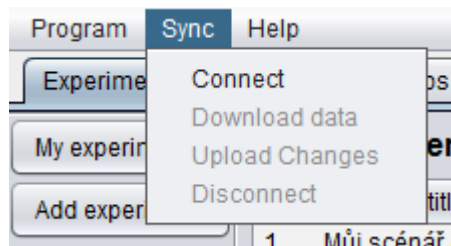
Po úspěšném stažení dat je již možné zadávat neuroinformatická data bez nutnosti připojení.

V horní části aplikace se nachází následující záložky:

- **Experiments** – Pro prohlížení a přidávání experimentů.
- **Scenarios** – Pro prohlížení a přidávání scénářů.
- **Groups** – Pro prohlížení a přidávání výzkumných skupin.
- **People** – Pro prohlížení a přidávání osob.
- **Lists** – Pro prohlížení a přidávání seznamů (počasí, hardware, definice dodatečných parametrů).

Menu programu (viz Obrázek A.4) obsahuje čtyři položky:

- **Program** – Pro ukončení programu.
- **Sync** – Pro připojení k serveru, stažení vstupních dat, nahrání změn na server a odpojení od serveru.
- **Help** – Informace o verzi programu.
- **Název profilu** – Nachází se v pravé části menu. Slouží pro změnu přihlášeného profilu.



Obrázek A.4: Menu klienta

Záložka Experiments

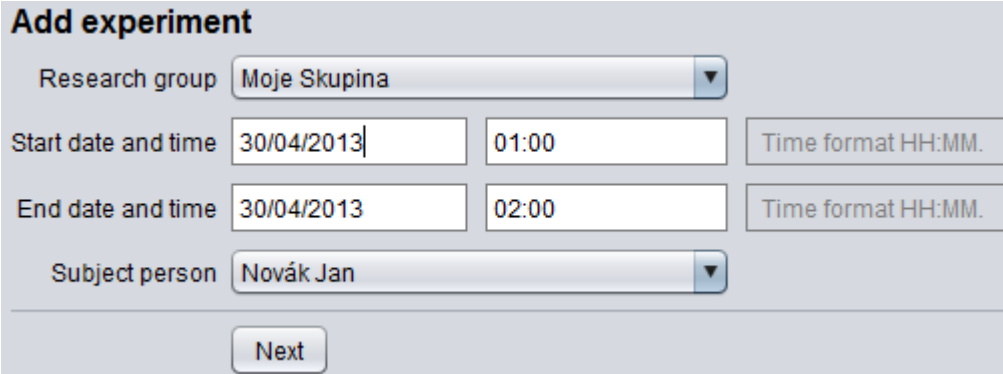
Základní stránka záložky *Experiments* zobrazuje experimenty vytvořené na tomto profilu (viz Obrázek A.5). V tabulce experimentů lze zobrazit podrobnosti o zvoleném experimentu kliknutím na odkaz *Detail*.

The image shows a web application interface with a tabbed menu at the top containing 'Experiments', 'Scenarios', 'Groups', 'People', and 'Lists'. The 'Experiments' tab is active. On the left side, there is a sidebar with 'My experiments' and 'Add experiment' buttons. The main content area is titled 'My experiments' and contains a table with the following data:

No.	Scenario title	Date	Gender	Year of birth	
1	Můj scénář experimentu	30.30.2013, 01:00	M	1987	Detail

Obrázek A.5: Seznam experimentů

V levém menu lze přecházet mezi stránkami v této záložce. Stránka *Add experiment* poskytuje funkci pro přidání nového experimentu. První fáze přidání (viz Obrázek A.6) zahrnuje výběr skupiny, testované osoby a doby trvání experimentu.



Add experiment

Research group: Moje Skupina

Start date and time: 30/04/2013 01:00 Time format HH:MM.

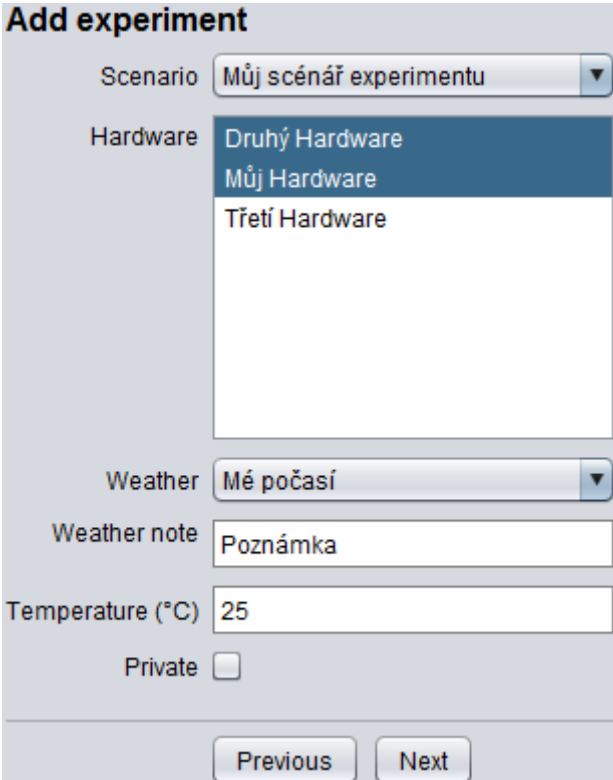
End date and time: 30/04/2013 02:00 Time format HH:MM.

Subject person: Novák Jan

Next

Obrázek A.6: První fáze přidání experimentu

Po stisknutí tlačítka *Next* se zobrazí druhá fáze přidání experimentu (viz Obrázek A.7). Tato část zahrnuje výběr scénáře, hardwaru (i více najednou přes CTRL + kliknutí), počasí a nastavení, zda se jedná o soukromý experiment.



Add experiment

Scenario: Můj scénář experimentu

Hardware: Druhý Hardware, Můj Hardware, Třetí Hardware

Weather: Mé počasí

Weather note: Poznámka

Temperature (°C): 25

Private:

Previous Next

Obrázek A.7: Druhá fáze přidání experimentu

Závěrečná fáze zahrnuje přidání souboru s naměřeným experimentem (viz Obrázek A.8). Po stisknutí odkazu *Attach next file* je možné přidat i více souborů současně. Tlačítkem *Finish* se dokončí vytvoření nového experimentu.

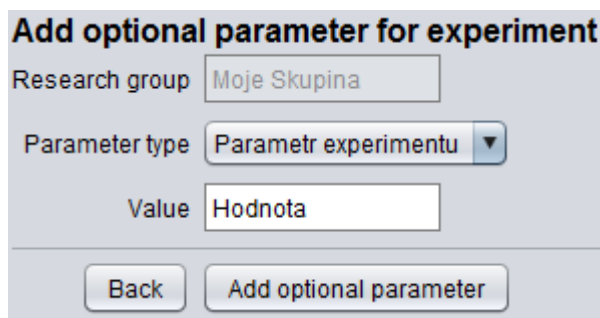
Obrázek A.8: Třetí fáze přidání experimentu

Následně je zobrazen detail vytvořeného experimentu (viz Obrázek A.9). Detail zobrazuje informace o experimentu, umožňuje zobrazit detail osoby a scénáře po kliknutí na příslušný odkaz.

Obrázek A.9: Detail experimentu

K experimentu je možné přidat dodatečné soubory po kliknutí na *Add data file*. Zobrazí se stránka podobná stránce jako v třetí fázi přidání experimentu.

Po kliknutí na *Add optional parameter* lze přidat k experimentu dodatečné parametry (viz Obrázek A.10), které se zobrazí v tabulce *Optional parameters*.



Add optional parameter for experiment

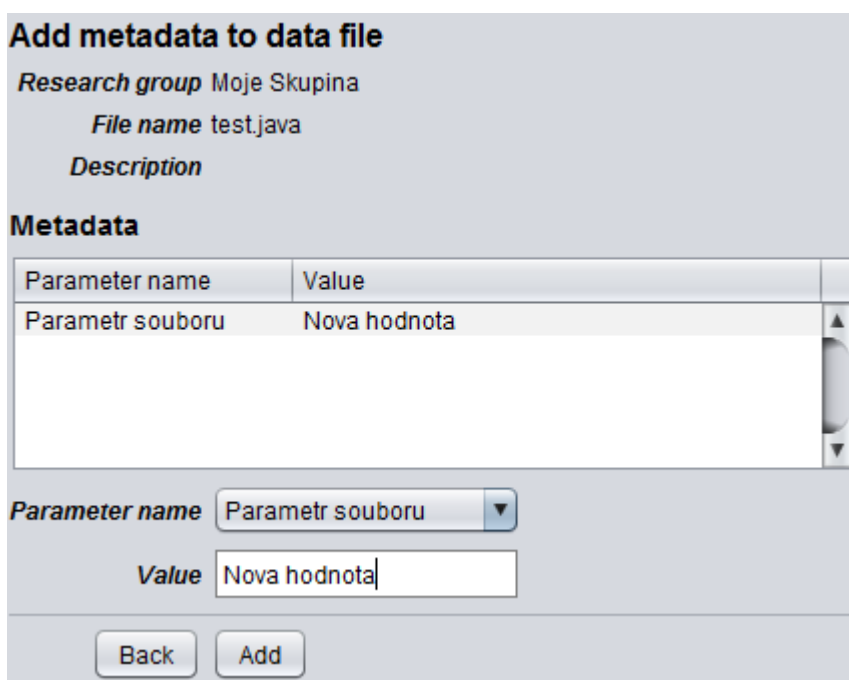
Research group

Parameter type

Value

Obrázek A.10: Přidání parametru experimentu

V tabulce *Data Files* lze zobrazit detail souboru přes kliknutí na odkaz. U detailu souboru (viz Obrázek A.11) je možné přidat metadata k zobrazenému souboru.



Add metadata to data file

Research group Moje Skupina

File name test.java

Description

Metadata

Parameter name	Value
Parametr souboru	Nova hodnota

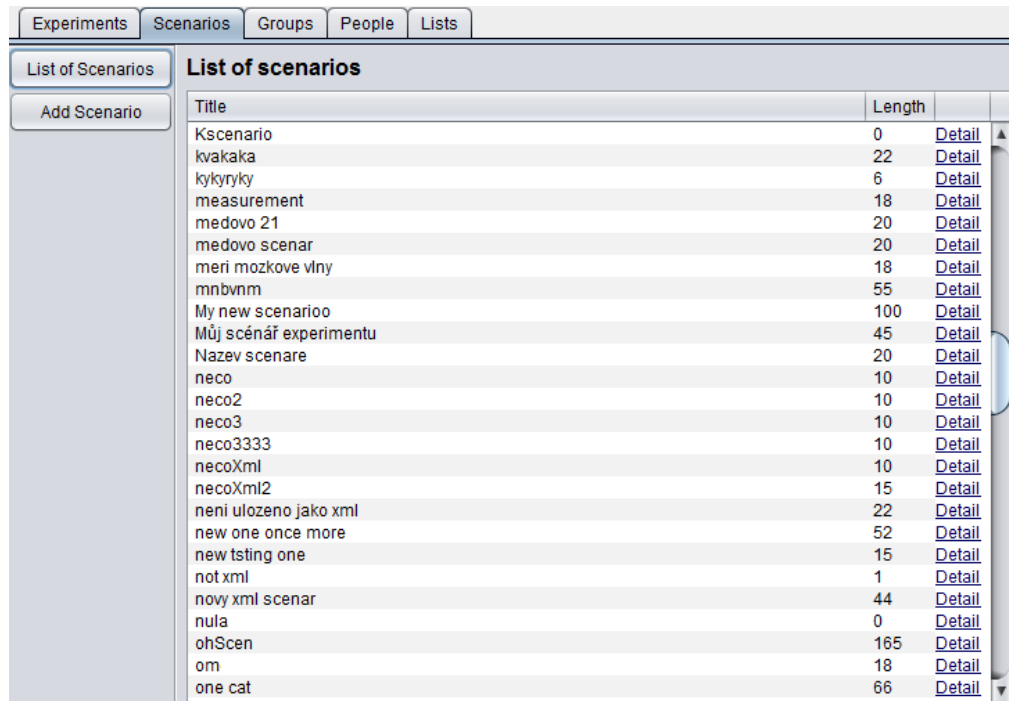
Parameter name

Value

Obrázek A.11: Detail souboru

Záložka Scenarios

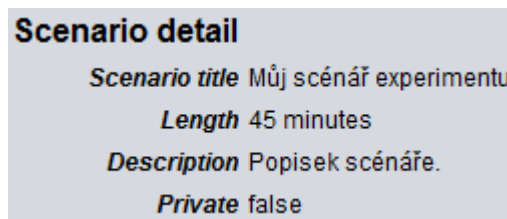
Po kliknutí na druhou záložku *Scenarios* se zobrazí přehled scénářů (viz Obrázek A.12).



Title	Length	
Kscenario	0	Detail
kvakaka	22	Detail
kykyryky	6	Detail
measurement	18	Detail
medovo 21	20	Detail
medovo scenar	20	Detail
meri mozbove vlny	18	Detail
mnbvnm	55	Detail
My new scenario	100	Detail
Můj scénář experimentu	45	Detail
Název scenare	20	Detail
neco	10	Detail
neco2	10	Detail
neco3	10	Detail
neco3333	10	Detail
necoXml	10	Detail
necoXml2	15	Detail
neni ulozeno jako xml	22	Detail
new one once more	52	Detail
new tsting one	15	Detail
not xml	1	Detail
novy xml scenar	44	Detail
nula	0	Detail
ohScen	165	Detail
om	18	Detail
one cat	66	Detail

Obrázek A.12: Přehled scénářů

V tabulce scénářů lze zobrazit podrobnosti o zvoleném scénáři (viz Obrázek A.13) kliknutím na odkaz *Detail*.



Scenario detail
Scenario title Můj scénář experimentu
Length 45 minutes
Description Popisek scénáře.
Private false

Obrázek A.13: Detail scénáře

Po kliknutí na tlačítko *Add Scenario* v levém menu lze přidat nový scénář (viz Obrázek A.14). Přidání zahrnuje výběr souboru scénáře, popisu, délky a názvu.

Add scenario

Research group: Moje Skupina

Scenario title: Můj scénář experimentu

Length: 45 minutes

Description: Popisek

Data file available:

Data file: Choose File "test.java"

Private:

Save

Obrázek A.14: Přidání scénáře

Záložka Groups

Základní stránka záložky *Groups* zobrazuje výzkumné skupiny (viz Obrázek A.15).

Experiments Scenarios **Groups** People Lists

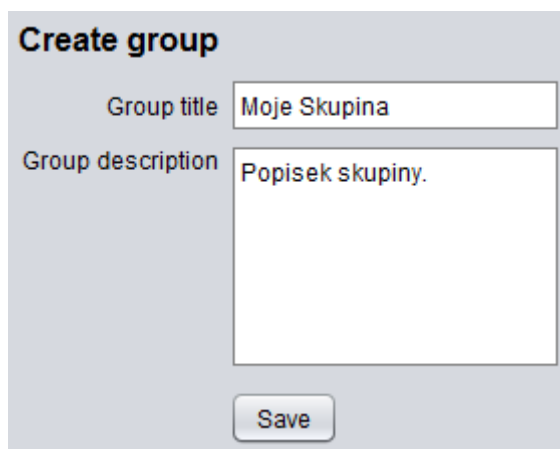
List of Groups

Create group

Group title	Description	
Fandovo	Fandovo	Detail
Ktitle	Ktitle	Detail
Moje Skupina	Popisek skupiny	Detail
Shade test group	testovaci	Detail
Shade test group 2	asdf	Detail
delThis	delThis	Detail
dxSkupina	dxSkupina	Detail
eGroup	eGroup	Detail
faGroup	faGroup	Detail
kaGroup	kaGroup	Detail
ppGroup	ppGroup	Detail
roman	roman	Detail
toGroup	toGroup	Detail

Obrázek A.15: Přehled výzkumných skupin

Kliknutím na *Create group* z levého menu se zobrazí stránka pro vytvoření nové výzkumné skupiny (viz Obrázek A.16).



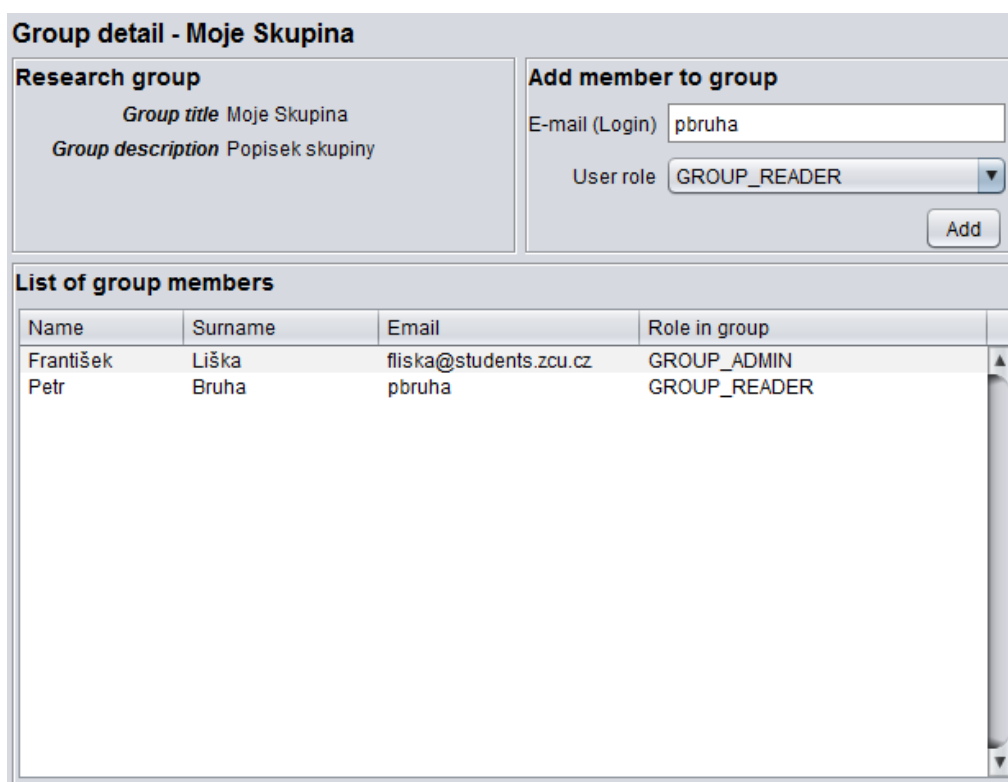
Create group

Group title:

Group description:

Obrázek A.16: Přidání výzkumné skupiny

V tabulce výzkumných skupin lze zobrazit podrobnosti o zvolené skupině kliknutím na odkaz *Detail*. Slouží i pro přidávání uživatelů do zvolené skupiny (viz Obrázek A.17).



Group detail - Moje Skupina

Research group
Group title: Moje Skupina
Group description: Popisek skupiny

Add member to group
E-mail (Login):
User role:

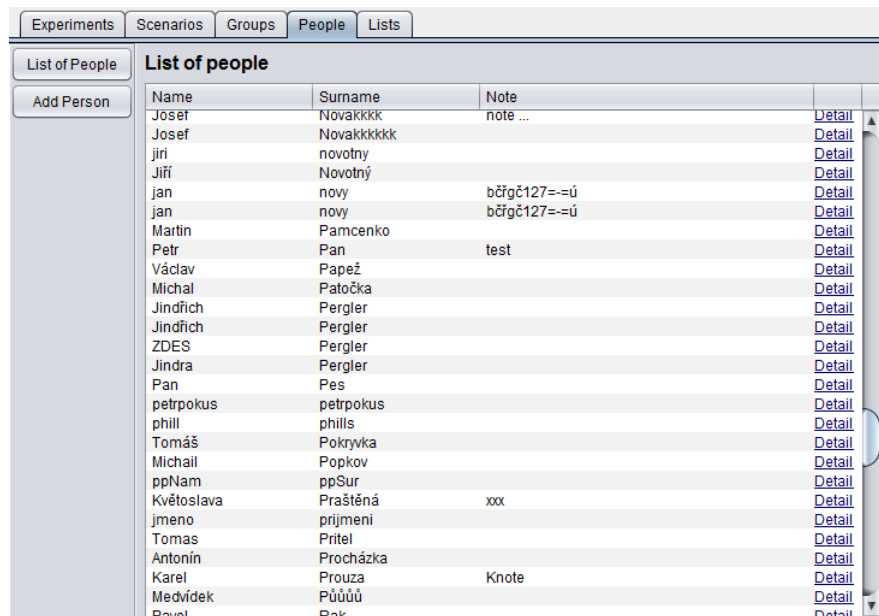
List of group members

Name	Surname	Email	Role in group
František	Liška	fliska@students.zcu.cz	GROUP_ADMIN
Petr	Bruha	pbruha	GROUP_READER

Obrázek A.17: Detail výzkumné skupiny

Záložka People

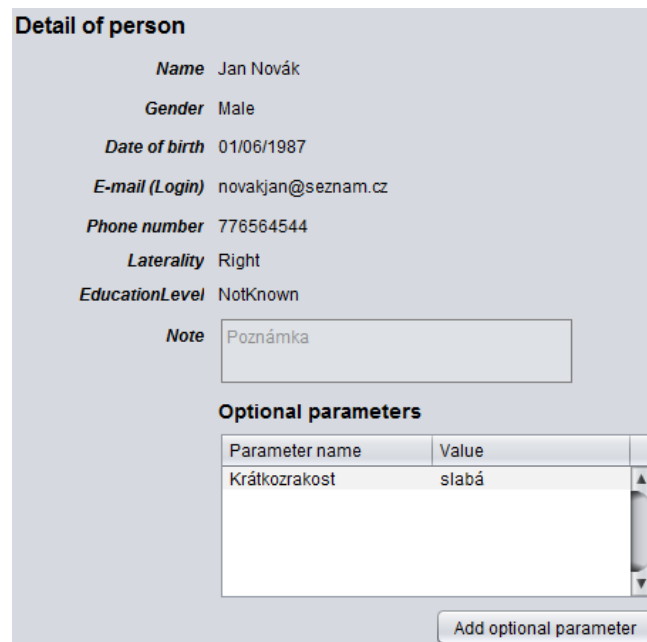
Základní stránka záložky *People* zobrazuje existující osoby (viz Obrázek A.18). Tyto osoby jsou uživatelé registrovaní na portále, popřípadě slouží jako subjekty v měřených experimentech.



Name	Surname	Note	
Josef	Novákkkk	note ...	Detail
Josef	Novákkkkkk		Detail
Jiří	novotny		Detail
Jiří	Novotný		Detail
jan	novy	bčfgč127=-ú	Detail
jan	novy	bčfgč127=-ú	Detail
Martin	Pamcenko		Detail
Petr	Pan	test	Detail
Václav	Papež		Detail
Michal	Patočka		Detail
Jindřich	Pergler		Detail
Jindřich	Pergler		Detail
ZDES	Pergler		Detail
Jindra	Pergler		Detail
Pan	Pes		Detail
petrpokus	petrpokus		Detail
phill	phills		Detail
Tomáš	Pokryvka		Detail
Michail	Popkov		Detail
ppNam	ppSur		Detail
Květoslava	Praštěná	xxx	Detail
jmeno	prijmeni		Detail
Tomas	Pritel		Detail
Antonín	Procházka		Detail
Karel	Prouza	Knote	Detail
Medvídek	Půůů		Detail
Paul	Polk		Detail

Obrázek A.18: Přehled osob

Po kliknutí na *Detail* u vybrané osoby se zobrazí stránka s podrobnostmi o osobě³ (viz Obrázek A.19).



Detail of person

Name Jan Novák

Gender Male

Date of birth 01/06/1987

E-mail (Login) novakjan@seznam.cz

Phone number 776564544

Laterality Right

EducationLevel NotKnown

Note

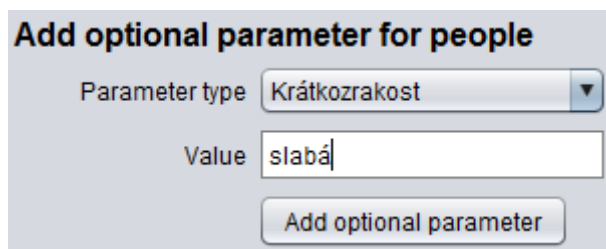
Optional parameters

Parameter name	Value
Krátkozrakost	slabá

Obrázek A.19: Detail osoby

³ Poznámka: Všechna uvedená data jsou smyšlená a nereprezentují skutečnou osobu.

Tabulka *Optional parameters* zobrazuje dodatečné parametry osoby. Po kliknutí na tlačítko *Add optional parameter* se zobrazí stránka pro přidání dodatečného parametru (viz Obrázek A.20).



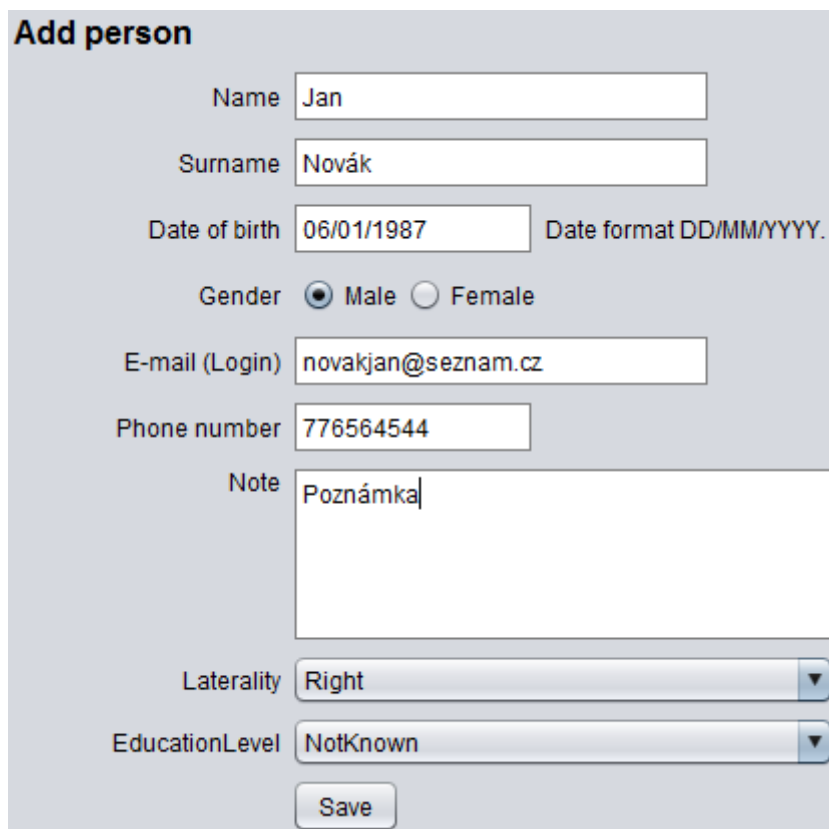
Add optional parameter for people

Parameter type

Value

Obrázek A.20: Přidání parametru osoby

Z levého menu lze kliknutím na tlačítko *Add Person* zobrazit stránku pro přidání osoby (viz Obrázek A.21).



Add person

Name

Surname

Date of birth Date format DD/MM/YYYY.

Gender Male Female

E-mail (Login)

Phone number

Note

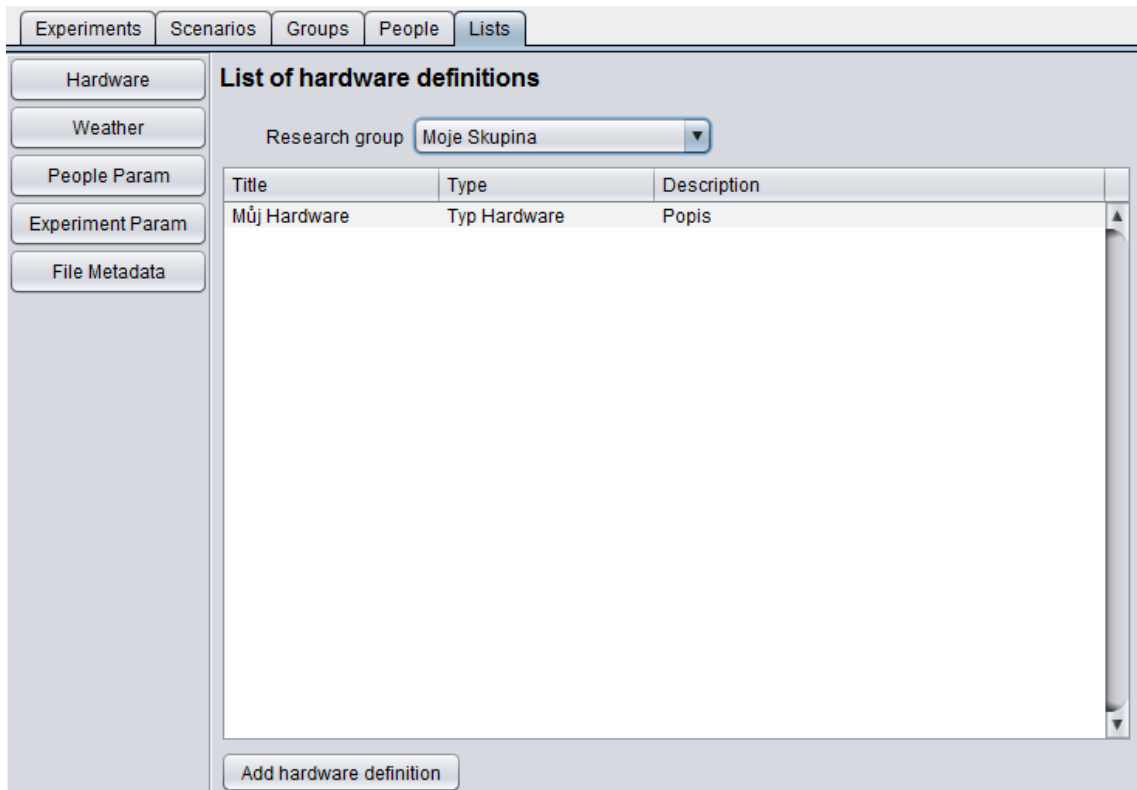
Laterality

EducationLevel

Obrázek A.21: Přidání osoby

Záložka Lists

Po kliknutí na záložku *Lists* se zobrazí stránka s přehledem hardwaru (viz Obrázek A.22). Z levého menu lze vybrat i další seznamy (počasí, definice dodatečných parametrů pro osoby, experimenty a soubory). Záznamy jsou spojeny s výzkumnou skupinou, takže při zvolení jiné skupiny z roletového menu se zobrazí jiné záznamy.



Obrázek A.22: Přehled hardwaru

Na každé stránce s přehledem záznamů je k dispozici tlačítko pro přidání záznamu do vybrané skupiny. Například přidání hardwaru (viz Obrázek A.23) je dostupné po kliknutí na tlačítko *Add hardware definition*.

Add hardware definition

Research group: Moje Skupina

Title: Můj Hardware

Type: Typ Hardware

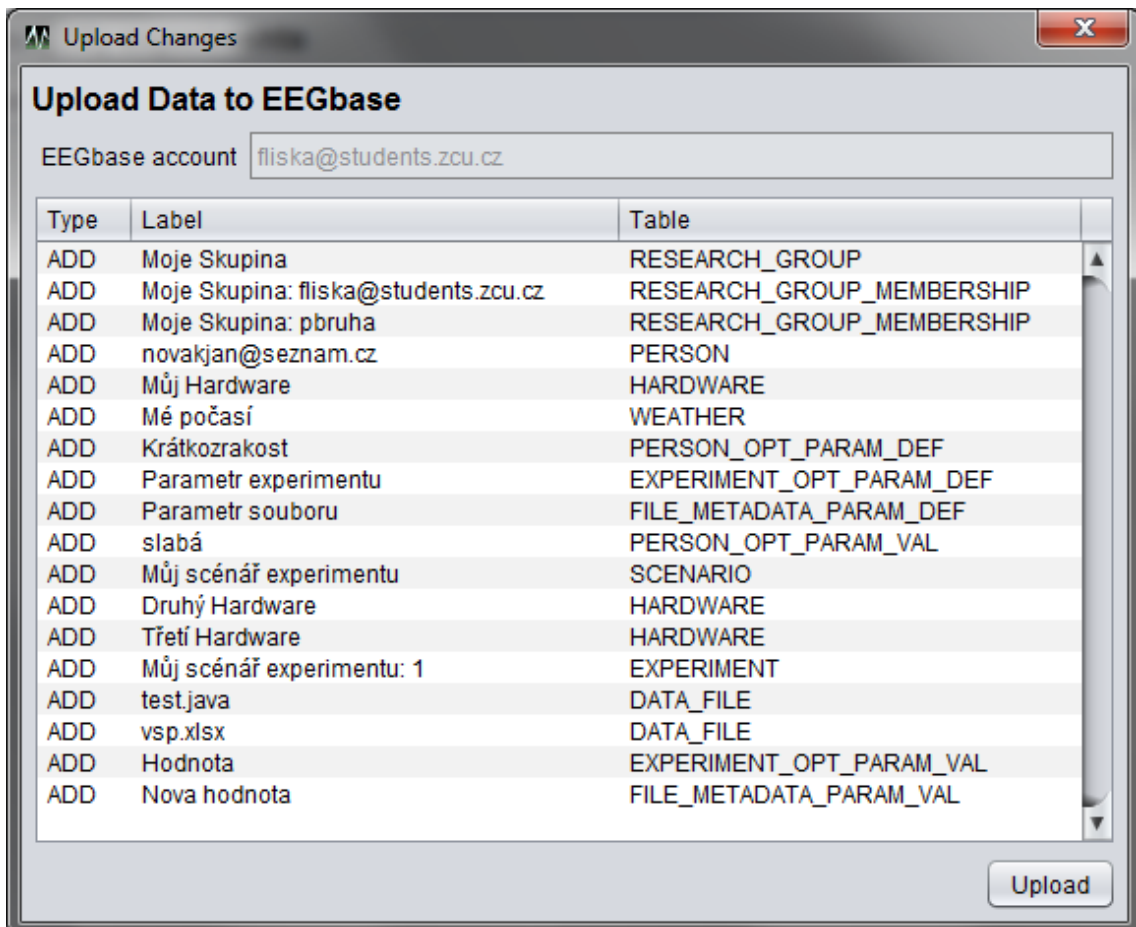
Description: Popis

Save

Obrázek A.23: Přidání hardwaru

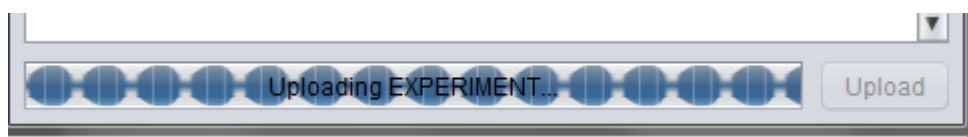
Nahrání změn na server

Z menu *Sync* je po přidání neuroinformatických dat možné tyto změny nahrát na server kliknutím na položku *Upload Changes*. Zobrazí se dialog (viz Obrázek A.24), který popisuje, ke kterému účtu se budou nahrávat data. Dialog obsahuje tabulku, ve které jsou vypsané změny v pořadí, v jakém se odehrály. V tabulce jsou uvedeny sloupce označující typ změny (například přidání), popisek změny (např. název výzkumné skupiny) a tabulku, u které se změna odehrála.



Obrázek A.24: Dialog pro nahrání změn na server

Po kliknutí na *Upload* se začnou postupně zasílat změny na server a po úspěšném přenosu změny zmizí tato změna z tabulky. Zobrazí se ukazatel postupu, který ukazuje název odesílané tabulky/souboru. (viz Obrázek A.25)



Obrázek A.25: Ukazatel postupu přenosu dat

Po úspěšném přehrání dat na EEG/ERP portál lze tato data zkontrolovat například na stránce s detailem experimentu (viz Obrázek A.26) přihlášením na vlastní uživatelský účet.

Experiment detail

Experiment ID	276
Beginning of experiment	30.04.2013, 01:00
End of experiment	30.04.2013, 02:00
Temperature	25
Weather	Mé počasí
Weather note	Poznámka
Private	false

Subject person

Gender	Male
Year of birth	1987

Scenario

Scenario title	Můj scénář experimentu
View detail of scenario	

Used hardware

Title	Type
Můj Hardware	Typ Hardware
Druhý Hardware	Intel

Optional parameters

Parameter name	Value
Parametr experimentu	Hodnota

Data files

File name	Description
test.java	Detail
vsp.xlsx	Detail

Obrázek A.26: Nahraná data z klienta na portál