České vysoké učení technické v Praze
Fakulta elektrotechnická

Katedra počítačů

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: Svobodová Lenka

Studijní program: Softwarové technologie a management
Obor: Softwarové inženýrství

Název tématu: Mobilní aplikace pro automatickou detekci epileptiformních výbojů v intraoperační elektrokortikografii

Pokyny pro vypracování:

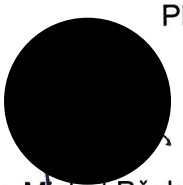
Analyzujte, navrhněte, implementujte, otestujte a dokumentujte vývoj aplikace pro automatickou detekci epileptiformních výbojů v elektrokortikografických záznamech pořízených během intraoperační elektrokortikografie. Při návrhu a implementaci využijte předem připravených bloků kódu. Aplikace by měla mít uživatelsky přívětivé rozhraní optimalizované pro použití na zařízeních s dotykovou obrazovkou. Při prezentaci výsledků detektoru je oproti stávajícímu stavu požadováno prostorově uspořádané kvantitativní hodnocení příslušné používané prostorové montáže snímacích elektrod. Aplikaci implementujte v programovacím jazyce, který může být kompilován pro platformu Windows. V aplikaci vytvořte nástroj pro správu a vizualizaci prostorových montáží kvantitativního hodnocení s možností exportu výsledků do standardních obrazových formátů. Práce by měla obsahovat dokumentaci dostatečnou k udržování aplikace a jejímu dalšímu vývoji.

Seznam odborné literatury:

[1] https://www.lekarskeknihy.cz/produkt/101929-epilepsie-temporalniho-laloku/
[2] Sommerville, Ian: Software Engineering, 9th edition, Addison-Wesley Publishing Company, USA, 2010, ISBN 9780137035151.

Vedoucí: Ing. Petr Ježdík, Ph.D.

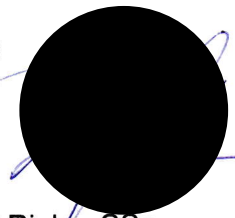
Platnost zadání do konce letního semestru 2017/2018



prof. Dr. Michal Pěchouček, MSc.

vedoucí katedry

L.S.

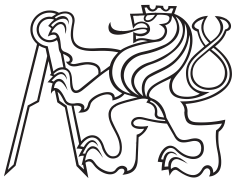


prof. Ing. Pavel Ripka, CSc.

děkan

V Praze dne 31.1.2017

Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra počítačů

Mobilní aplikace pro automatickou detekci epileptiformních výbojů v intraoperační elektrokortikografii

Mobile application for automatic detection of epileptogenic discharges in intrasurgical electrocorticography

Lenka Svobodová

Vedoucí: Ing. Petr Ježdík Ph.D.
Obor: Softwarové inženýrství
Studijní program: Softwarové technologie a management
Květen 2017

Poděkování

Ráda bych zde poděkovala Ing. Petru Ježdíkovi Ph.D. za vedení této práce a své rodině za velkou podporu během celého studia. Dále bych chtěla poděkovat Bc. Martinu Bártovi za pomoc a ochotu při integraci s jeho aplikací Alenka.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracovala samostatně a že jsem uvedla veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 26. května 2017

Abstrakt

Jednou z možností léčby farmakorezistentní epilepsie je resekce poškozené mozkové tkáně. K určení pozice a velikosti této tkáně se během operace měří EEG přímo z povrchu mozku (provádí se tzv. intraoperační elektrokortikografie). V současnosti lékaři polohy elektrod zakreslují na papír do předtištěných schémat mozků. Je to zatím nejrychlejší metoda, ale při změně polohy elektrody musí celý obrázek znovu překreslit. Naměřené EEG vyhodnocují v reálném čase pouze vizuálně, tzn. že vyhodnocení je silně závislé na lékaři.

Cílem této bakalářské práce je navrhnout a implementovat aplikaci, která by lékařům poskytovala prostorově i kvantitativně přehledné a objektivní informace o rozložení epileptiformních výbojů a zobrazovala rozmístění elektrod podobným způsobem jako na papíře.

Aplikace je navržena pro použití na tabletu s OS Windows. K dispozici jsem měla aplikaci Alenka [4], která EEG zpracovává a detekuje spiky v signálu. Alenka je součástí výsledné aplikace a poskytuje data pro statistiky o výbojích.

Pro implementaci je použit framework Qt a deklarativní jazyk QML. Integrace s Alenkou je vyřešena tak, že mezi aplikacemi je možné přepínat, ale zároveň jsou obě aplikace spojeny v jeden celek.

Testování probíhalo na vybraných zařízeních s OS Windows a Ubuntu. Aplikace je připravena pro použití na sále.

Klíčová slova: mobilní aplikace, EEG, intraoperační elektrokortikografie, dotykové ovládání

Vedoucí: Ing. Petr Ježdík Ph.D.

Abstract

One of the treatment options of pharmacoresistant epilepsy is resection of damaged brain tissue. To determine the position and size of this tissue the EEG is measured directly from the brain surface during the surgery (intrasurgical electrocorticography). At present doctors draw the positions of electrodes on a paper in schematic brain pictures. It is now the fastest method but when position of electrode is changed they have to draw a new picture. Measured EEG is evaluated visually in real time so it is highly dependent on a doctor.

The aim of this bachelor thesis is to design and implement an application that will provide spatially and quantitatively clear and objective information about the distribution of epileptiform discharges. Another goal is to display the distribution of electrodes in a similar way as on paper.

The application is designed for Windows tablets. I had Alenka application [4] at my disposal, which processes EEG and detects spikes in the signal. Alenka is part of the final application and provides data for spikes statistics.

Qt framework and QML are used for implementation. Application is merged with Alenka and it is possible to switch between them. Testing took place on Windows and Ubuntu OS. Application is ready to be used at operating rooms.

Keywords: mobile application, EEG, intrasurgical electrocorticography, touch control

Title: Mobile application for automatic detection of epileptogenic discharges in intrasurgical electrocorticography

Obsah

1 Úvod **1**

2 Popis problematiky **5**

2.1 Terminologie 5

2.2 Epilepsie 6

2.3 Epileptochirurgie 7

2.3.1 Resekční zákroky 7

2.4 Intraoperační elektrokortikografie 8

2.5 Současný stav 9

3 Analýza a návrh **11**

3.1 Cíle 11

3.2 Požadavky 12

3.2.1 Funkcionální požadavky 12

3.2.2 Nefunkcionální požadavky . . . 13

3.3 Případy užití 14

3.3.1 Změnit obrázek 14

3.3.2 Přidat obrázek 15

3.3.3 Smazat obrázek 16

3.3.4 Přidat elektrodu 16

3.3.5 Přiřadit signál k elektrodě . . . 17

3.3.6 Změnit barvu elektrody 17

3.3.7 Přidat identickou elektrodu . 17

3.3.8 Změnit gradient statistiky . . . 18

3.4 Návrh uživatelského rozhraní . . . 19

3.5 Varianty řešení integrace s aplikací Alenka 20

4 Implementace **23**

4.1 Výběr technologií 23

4.1.1 Qt 23

4.1.2 QML 24

4.2 Integrace s aplikací Alenka 24

4.3 Uživatelské rozhraní 26

4.4 Architektura 26

4.5 Další části implementace 28

4.5.1 Podporované formáty a export 28

4.5.2 Použití pro desktop 28

5 Testování **29**

5.1 Testovací prostředí 29

5.2 Manuální testování 30

5.3 Uživatelské akceptační testy 34

6 Závěr **35**

6.1 Zhodnocení splnění cílů 35

6.2 Možná rozšíření a vylepšení 36

Literatura **37**

A Seznam použitých zkratek **41**

B Instalační instrukce **43**

C Návod k použití **45**

D Obsah přiloženého DVD **49**

Obrázky

1.1 Schématické znázornění průběhu epileptochirurgie od diagnostiky epilepsie až po resekci poškozené mozkové tkáně. Vyznačený úsek ukazuje, ve které části se bude používat vyvíjená aplikace. Schéma kraniotomie převzané z [24]. 3

2.1 Schéma uspořádání elektrod - strip, grid 6

2.2 Ukázka intraoperační elektrokortikografie 8

2.3 Současný stav vyhodnocování intraoperační elektrokortikografie na operačním sále (FN v Motole) 10

3.1 Use case model 15

3.2 Ukázka kontextového menu u obrázku 16

3.3 Ukázka dialogu pro přidání nové elektrody 16

3.4 Ukázka nastavení barevného gradientu 18

3.5 Návrh uživatelského rozhraní . . . 20

4.1 Organizace kódu v rámci aplikace Alenka 25

4.2 Ukázka kódu 27

C.1 Obrazovka aplikace Alenka [4] . 45

C.2 Obrazovka Image Manager 46

C.3 Obrazovka Electrode Manager . 47

C.4 Obrazovka Electrode Signal Link 48

C.5 Obrazovka Electrode Placement 48

Tabulky

5.1 Přehled testovacích prostředí . . . 30

5.2 Testovací scénář č.1 30

5.3 Testovací scénář č.2 31

5.4 Testovací scénář č.3 31

5.5 Testovací scénář č.4 32

5.6 Testovací scénář č.5 32

5.7 Testovací scénář č.6 33

5.8 Testovací scénář č.7 33

Kapitola 1

Úvod

Epilepsie je jedním z nejrozšířenějších neurologických onemocnění na světě [23, 3]. Ve většině případů pomůžou nasazené léky, tzv. antiepileptika, které zabraňují vzniku epileptických záchvatů. V některých případech monoterapie, tzn. podávání jednoho druhu léku, nepomůže a potom se předepisuje kombinace více antiepileptik. Pokud ani kombinace léků nesníží počet epileptických záchvatů na přijatelnou míru, je nutno přistoupit k chirurgickému řešení nemoci, které spočívá v resekci mozkové tkáně, která způsobuje epileptické záchvaty.

Pro co nejpřesnější lokalizaci poškozené tkáně se mimo jiných metod používá také intraoperační elektrokortikografie. Při této metodě se během operace snímá elektrická aktivita přímo z povrchu mozku a z naměřených EEG záznamů se určí (upřesní) poloha epileptogenního ložiska.

Právě vyhodnocování EEG naměřeného při intraoperační elektrokortikografii se týká tato bakalářská práce. Konkrétně má za cíl nasimulovat prostorové uspořádání elektrod a automaticky zpracovat EEG, jehož vyhodnocení se zobrazí v prostorovém uspořádání.

Při operaci tak komplikovaného a důležitého orgánu jako je mozek, si musí být lékaři každým svým rozhodnutím naprosto jisti a případně si ho umět zpětně obhájit v případě pozdějších komplikací. Jakékoliv špatné rozhodnutí založené např. na nedostatečným informacích, mohou mít pro pacienta doživotní následky. Proto má tato aplikace lékařům poskytnout další a přesnější data, aby při vyhodnocování EEG z elektrokortikografie nebyly odkázáni pouze na informace typu „Na elektrodě XY je hodně/málo epileptiformních výbojů.“, ale měli k dispozici prostorově i kvantitativně přehledné informace o rozložení epileptických výbojů.

Proces epileptochirurgie od diagnostiky epilepsie u pacienta až po resekci epileptogenní zóny je schématicky zobrazen na obr. 1.1. Na obrázku je rovněž zvýrazněno místo použití aplikace vyvíjené v rámci této BP. Jednotlivé kroky

na schématu jsou velmi zjednodušeny a zobecněny, více se danému tématu věnuji v kapitole 2.

Tato bakalářská práce je rozdělena do 6 kapitol - *Úvod*, *Popis problematiky*, *Analýza a návrh*, *Implementace*, *Testování* a *Závěr*.

Vzhledem k tomu, že je tato práce úzce spjata s medicínou a s ní spojenými odbornými termíny, v kapitole *Popis problematiky* definuji pojmy, které v této práci používám. Dále se na základní úrovni věnuji epilepsii a intraoperační elektrokortikografii, při které bude tato aplikace používána.

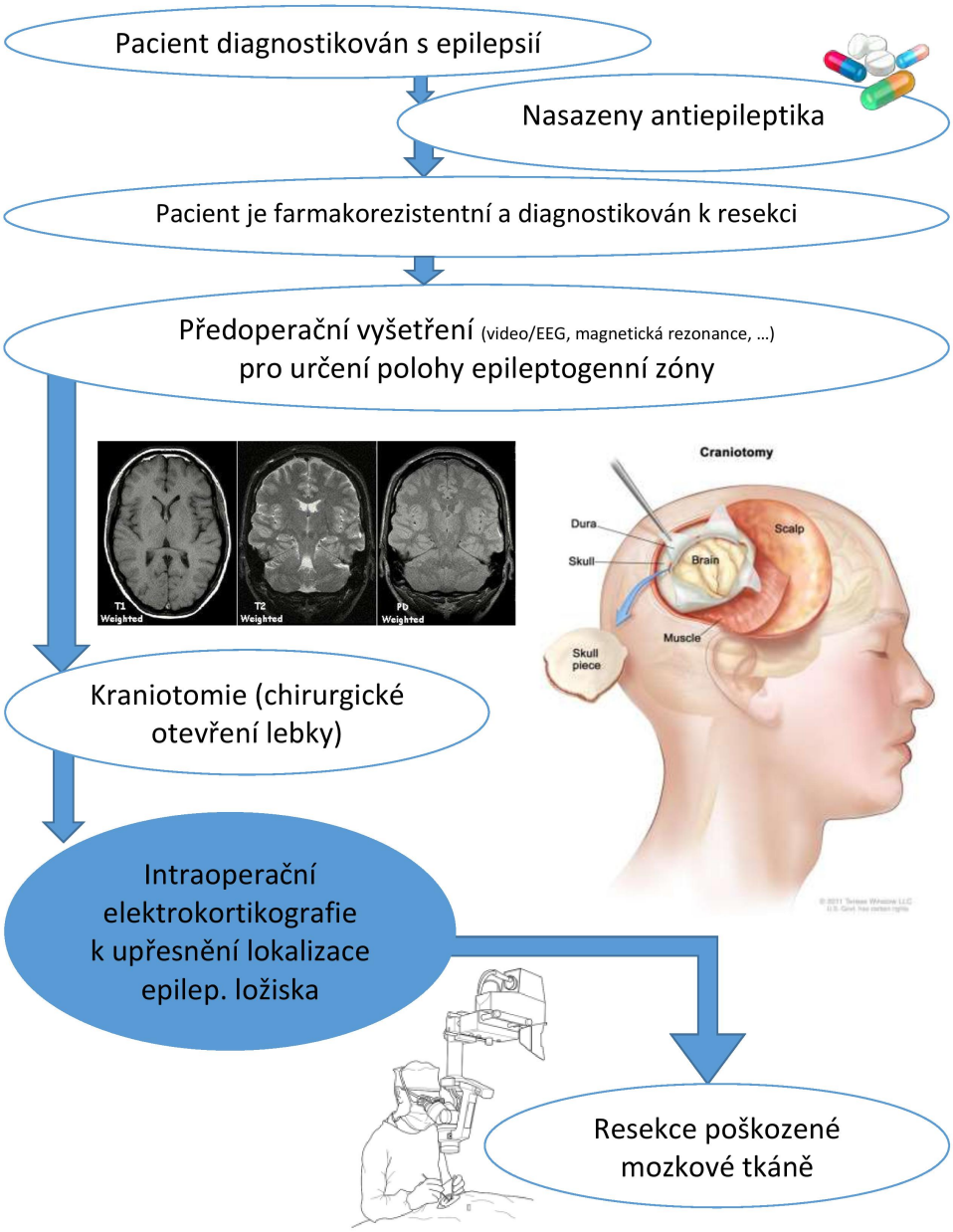
V kapitole *Analýza a návrh* definuji detailní požadavky na aplikaci, a to jak funkcionální, tak nefunkcionální. Dále se zabývám detailní analýzou a návrhem uživatelského rozhraní.

V kapitole *Implementace* se věnuji použitým technologiím, které jsem při implementaci použila a také samotnému procesu implementace.

V kapitole *Testování* popisuji jakými technikami jsem ověřila správnost implementace.

V poslední kapitole *Závěr* zhodnotím splnění cílů a navrhuji možná rozšíření a vylepšení aplikace do budoucna.

Zadavatelem této práce je výzkumná skupina ISARG (Intracranial Signal Analysis Research Group) [9], která se zabývá analýzou EEG signálů u pacientů s epilepsií. V této skupině spolupracují specialisté z Fakulty elektrotechnické ČVUT (včetně vedoucího této práce) s lékaři z 2. lékařské fakulty Univerzity Karlovy a z Fakultní nemocnice v Motole a odborníky z Fyziologického ústavu Akademie věd ČR.



Obrázek 1.1: Schématické znázornění průběhu epileptochirurgie od diagnostiky epilepsie až po resekci poškozené mozkové tkáně. Vyznačený úsek ukazuje, ve které části se bude používat vyvíjená aplikace. Schéma kraniotomie převzané z [24].

Kapitola 2

Popis problematiky

Tato bakalářská práce je zpracovávána na Fakultě elektrotechnické, ale týká se medicíny, a proto je tato kapitola věnována lékařské problematice, která s prací přímo souvisí. Kapitola by měla obsahovat všechny potřebné informace z oblasti medicíny, aby čtenář pochopil souvislosti BP bez předchozích znalostí z tohoto oboru.

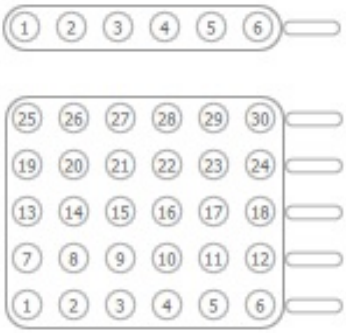
Definuji terminologii, kterou dále v práci používám a seznámím čtenáře se základními věcmi, které se týkají epilepsie a epileptochirurgie. Dále popíši průběh intraoperační elektrokortikografie, během které by měla být aplikace využívána a nakonec nastíním současný stav, který by měla tato práce změnit. V závěru v krátkosti uvedu dostupný program na analýzu EEG, jehož výsledky bude aplikace zpracovávat a na který tato BP navazuje.

2.1 Terminologie

Pojmy jsou seřazeny v abecedním pořadí.

- **Alenka** - program na zpracování EEG vyvíjený v rámci bakalářské práce [4] a dále diplomové práce na FIT ČVUT .
- **EEG -** metoda snímání elektrické aktivity mozku prostřednictvím elektrod. Může být neivazivní, kdy jsou elektrody umístěny na povrchu hlavy, nebo invazivní, kdy je elektrická aktivita snímána prostřednictvím elektrod přímo umístěných na povrchu mozku.
- **Elektrokortikografie -** měření EEG pomocí elektrod umístěných

přímo na povrchu mozku. Rozděluje se na intraoperační (měření během operace, viz kapitola 2.4) a extraoperační (měření mimo operaci).
- **Elko** - pracovní název aplikace vyvíjené v rámci této bakalářské práce, konkrétně tedy „mobilní aplikace pro automatickou detekci epiletiformních výbojů v intraoperační elektrokortikografii“.
- **Epilepsie -** neurologické záchvatovité onemocnění. Více viz kapitola 2.2.
- **Epileptochirurgie -** chirurgické řešení epilepsie. Více viz kapitola 2.3.
- **Grid -** elektroda s kontakty uspořádanými do mřížky. Schéma viz obr. 2.1.
- **iEEG -** intrakraniální elektroencefalografie nebo také elektrokortikografie. Více viz kapitola 2.4.
- **Intraoperační** - prováděné během operace
- **Intraoperační elektrokortikografie -** EEG snímané přímo z povrchu mozku během operace. Více viz kapitola 2.4.
- **Resekce -** chirurgické odstranění poškozené části mozkové tkáně.
- **Strip -** elektroda s kontakty uspořádanými do řádku. Schéma viz obr. 2.1.



Obrázek 2.1: Schéma uspořádání elektrod - strip, grid

2.2 Epilepsie

Epilepsie je jedno z nejčastějších neurologických onemocnění a to jak u dětí, tak dospělých. V současnosti jí trpí zhruba 50 miliónů lidí po celém světě, z čehož většina jich žije v zemích s nízkým nebo středním příjmem [23]. Nemoc se projevuje tzv. epileptickými záchvaty. Jedná se o náhlé poruchy pohybu, vědomí, které jsou způsobeny spontánní elektrickými výboji v šedé kůře mozkové. Záchvaty můžeme rozdělit na parciální (postihují část mozku) a generalizované (postihují celý mozek) [19].

Epilepsie se nejčastěji léčí tzv. antiepileptiky, což jsou léky na potlačení epileptických záchvatů. Pomocí monoterapie (užívání jednoho antiepileptika) se daří dosáhnout kompenzace až u více než poloviny pacientů [8]. Pokud nepomůže monoterapie, přejde se na kombinaci více antiepileptik nebo se

nemoc diagnostikuje jako tzv. farmakorezistentní epilepsie. Je otázka jak dlouho by se mělo čekat než se může říci, že farmakologická léčba nezabrala. Mezinárodní liga proti epilepsii (ILAE) definovala farmakorezistentní epilepsii jako *neuspokojivou kompenzaci záchvatů po vyzkoušení dvou tolerovaných, vhodně vybraných a správně dávkovaných antiepileptik (ať již v monoterapii nebo v kombinaci)* [11].

Velkou část pacientů s farmakorezistetní epilepsií představují pacienti s epilepsií temporálního (spánkového) laloku (TLE) [3], která je zároveň nejčastějším typem epilepsie [2]. Termínem „epilepsie temporálního laloku“ ale není z klinického hlediska nemoc přesně definovaná. Podle rozdílných klinických obrazů, patologicko-anatomického podkladu i průběhu léčby můžeme TLE dále dělit na např. Meziotemporální nebo Meziotemporopolární epilepsii [2].

2.3 Epileptochirurgie

Epileptochirurgie, jinými slovy chirurgická léčba epilepsie, patří do oblasti neurochirurgie a až na výjimky při ní nejde přímo o záchranu života, ale o zlepšení jeho kvality. Epileptochirurgické zákroky můžeme rozdělit na kurativní a paliativní.

Mezi kurativní výkony patří zejména výkony resekční a jejich cílem je úplné odstranění epileptických záchvatů. Oproti tomu cílem paliativních výkonů není úplné vyléčení epilepsie, ale pouze zmírnění jejích projevů, tzn. snížení epileptických záchvatů nebo odstranění záchvatů, které pacienta nejvíce ohrožují. K paliativní léčbě se přistupuje v situaci, kdy není možné provést resekční výkon nebo se předchozím resekčním výkonem nedosáhlo uspokojivého výsledku [3]. Z paliativních zákroků se používá především stimulace nervus vagus (stimulace bloudivého nervu, vagový stimulátor) [2].

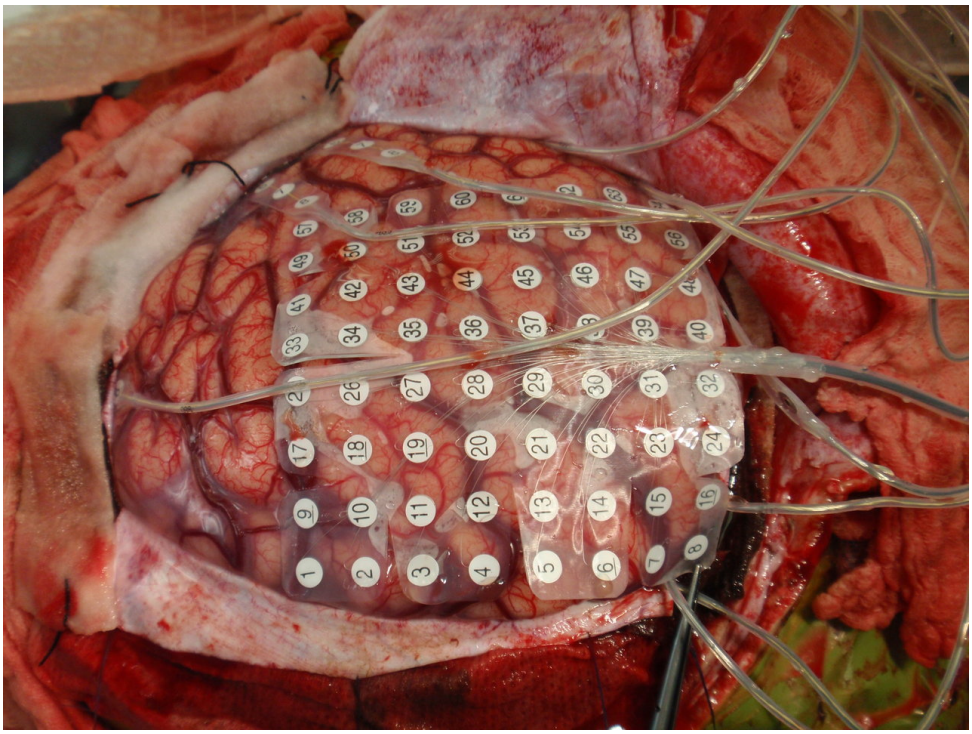
2.3.1 Resekční zákroky

Hlavním cílem resekčního zákroku je odstranění tzv. epileptogenní zóny bez negativního ovlivnění motorických, senzorických nebo kognitivních funkcí. Epileptogenní zóna je definovaná jako část funkčně změněné mozkové tkáně, která je přímo zodpovědná za generování záchvatů [3]. K určení co nejpřesnější pozice a velikosti této zóny se provádějí nejrůznější předoperační vyšetření jako je např. video/EEG, magnetická rezonance nebo traktografie [10]. Během operace se pořizují záznamy z intraoperační elektrokortikografie (ECoG, viz kapilota 2.4).

2.4 Intraoperační elektrokortikografie

Intraoperační elektrokortikografie (ECoG) neboli intrakraniální elektroencefalografie (iEEG) slouží k upřesnění lokalizace a rozměrů epileptogenního ložiska určeného k resekci [21]. Jedná se o metodu snímání elektrické aktivity mozku přímo z kůry mozkové během epileptochirurgické operace (viz obr. 2.2) Aktivita se snímá pomocí stripových nebo gridových elektrod (viz obr. 2.1). Vzhledem k tomu, že je k vyšetření nutná kraniotomie, tzn. chirurgické otevření lebky, řadí se ECoG mezi invazivní vyšetření.

Výhodou elektrokortikografie oproti elektroencefalografii (EEG) je větší přesnost, protože signál není utlumen při průchodu lebkou. Dále také není potřeba žádný vodivý gel, vodivost zajišťují tělní tekutiny. Výhodou oproti jiným invazivním vyšetření je možnost měnit polohu elektrod při vyšetření a možnost snímat elektrickou aktivitu opakovaně, před a po resekci poškozené tkáně [2, 3].



Obrázek 2.2: Ukázka intraoperační elektrokortikografie

2.5 Současný stav

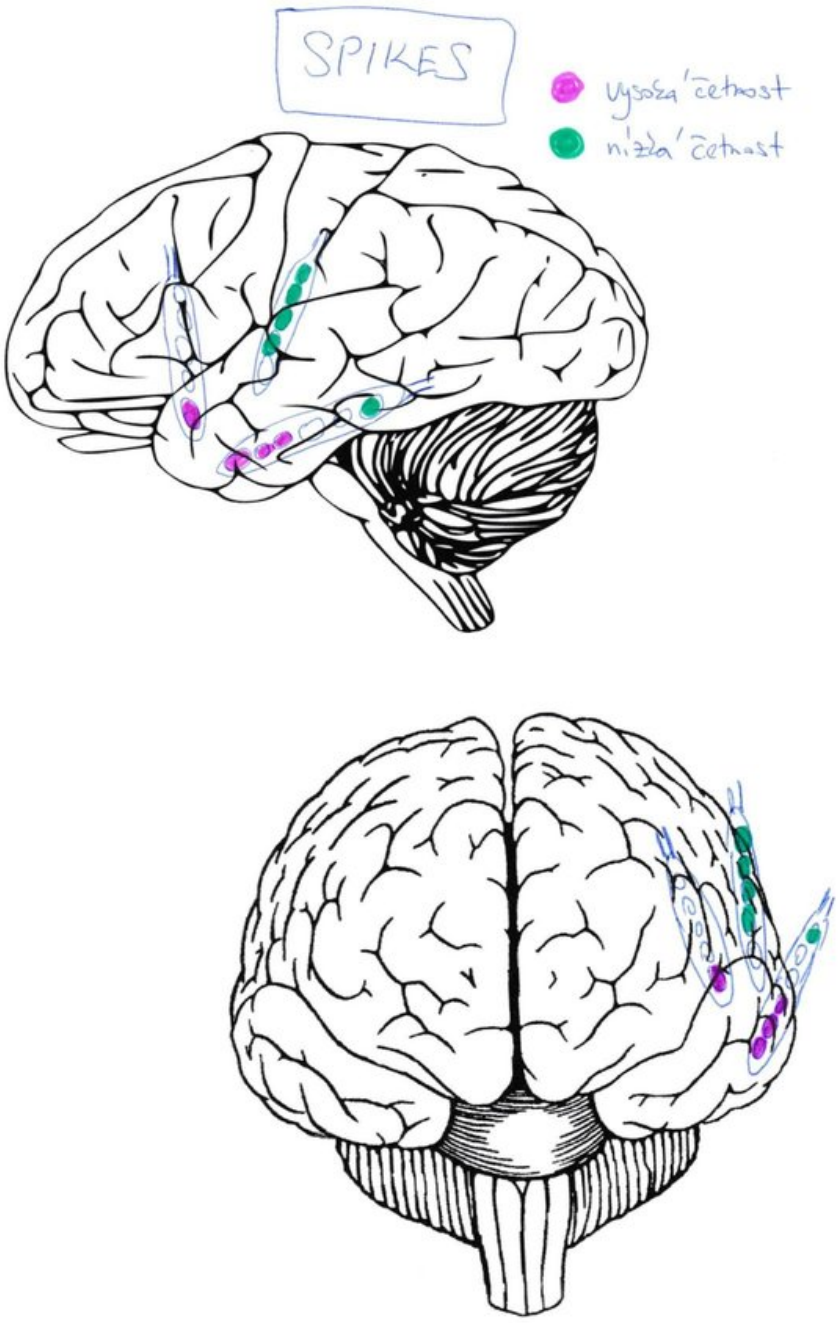
V současné době lékaři na sále při intraoperační kortikografii (konkrétně na Neurochirurgické klinice dětí a dospělých 2. LF UK a FN Motol) nepoužívají žádnou aplikaci, která by automaticky zpracovávala EEG a na které by měli možnost nasimulovat prostorové uspořádání elektrod. Pro zachycení uspořádání elektrod na mozku používají papír s předtištěnými schématy mozku a tužku, kterou kreslí jak jsou elektrody na mozku položeny. Snímané EEG vyhodnocují pouze vizuálně a v reálném čase. Do elektrod poté barevnými fixami zakreslují svá pozorování (viz obr. 2.3).

Tento současný stav je nenáročný jak časově, tak i na prostředky. Má ale několik nevýhod, konkrétně:

- nemožnost úpravy pozic elektrod bez toho, aniž by se musel celý obrázek překreslit,
- omezený výběr schémat mozku, k dispozici jsou v daném okamžiku pouze předtištěná schémata,
- nemožnost použít fotografii přímo z dané operace jako podklad pro rozmístění elektrod a
- možnost vlivu lidského faktoru, kdy lékař může v EEG něco přehlédnout.

Aplikace vyvíjená v rámci této BP si klade za cíl všechny tyto nedostatky odstranit.

Nyní existuje program pro zobrazování biologických signálů pracovně nazvaný Alenka [4], který je rozšiřován o další moduly v rámci diplomové práce na FIT ČVUT. Jedná se *Specializovaný systém pro zobrazování biologických signálu pacientu zařazených do epilepto-chirurgického programu.* Program zpracovává EEG signály ve formátech .edf, .gdf a .mat a umožňuje nad nimi provádět další operace (např. sčítání signálů). Mimo jiné obsahuje spike detektor [5], který na EEG označí „spikes“, tzn. epilepticky významné špičky v signálu. Výstupy tohoto detektoru jsou používány jako data pro zobrazení statistik v této bakalářské práci.



Obrázek 2.3: Současný stav vyhodnocování intraoperační elektrokortikografie na operačním sále (FN v Motole)

Kapitola 3

Analýza a návrh

V této kapitole jsou specifikovány základní cíle aplikace, dále jsou detailně popsány požadavky funkcionální i nefunkcionální a vybrané případy užití. V rámci analýzy a návrhu je také důležité navrhnout vzhled aplikace, který je popsán v podkapitole 3.4 Návrh uživatelského rozhraní a rozebrat varianty řešení integrace s aplikací Alenka.

3.1 Cíle

Tato bakalářská práce a její výsledná aplikace si klade za cíl zautomatizovat vyhodnocování EEG při intraoperační elektrokortikografii, aby lékaři nebyli při vyhodnocování odkázáni pouze na svůj subjektivní úsudek, který je závislý na jejich znalostech a zkušenostech, ale mohli si své výsledky porovnat s objektivními výsledky zpracované aplikací. Lékaři tak budou mít při rozhodování další pohled na vyhodnocení naměřeného EEG.

Dalším přínosem projektu je možnost přesnějšího nasimulování rozmístění elektrod, aby uživatelé aplikace získali lepší přehled o prostorovém uspořádání elektrod na mozku při operaci. Nyní jsou schéma mozku a elektrod znázorňována na papír, kde není možnost pozdější úpravy.

3.2 Požadavky

Nyní definuji požadavky na funkce a vlastnosti aplikace podle požadavků zadavatele (vedoucího práce). V seznamu je uveden kompletní přehled požadavků včetně těch, které vznikaly v průběhu vývoje aplikace. Požadavky jsem rozdělila na funkcionální a nefunkcionální podle [20]. Funkcionální požadavky (nebo jinak funkční) definují co bude aplikace uživateli umožňovat, jak se bude systém chovat při různých vstupech a různých situacích. Nefunkcionální požadavky (výstižněji kvalitativní požadavky a omezení) se týkají celé aplikace a jde o požadavky na kvalitu a případná omezení aplikace. Mezi nefunkcionální požadavky se řadí například uživatelská přívětivost, dokumentace, ale také požadavky na bezpečnost a výkon.

3.2.1 Funkcionální požadavky

Aplikace se bude používat při operaci, tzn. požadavky se vztahují k použití u jednoho konkrétního pacienta a jeho naměřeného EEG.

1. **Nahrání vlastního obrázku (např. fotografie mozku).** Systém bude uživateli umožňovat nahrát vlastní soubor ve standardním obrazovém formátu a následně ho použít jako jeden z pohledů na mozek.
2. **Změna nahraného obrázku.** Systém bude uživateli umožňovat již nahraný obrázek později změnit.
3. **Odstranění obrázku.** Uživatel bude mít možnost odstranit kterýkoliv z obrázků (předem nadefinované i nahrané).
4. **Navrhnutí a vytvoření elektrody podle vlastních rozměrů.** Uživatel bude mít možnost navrhnout vlastní rozměry elektrody a následně ji použít stejně jako ostatní ze základního výběru.
5. **Export do standardních obrazových formátů.** Aplikace bude umožňovat export stránky s rozmístěním elektrod na schématech mozku do standardních obrazových formátů (.jpg, .bmp, .png). Export bude možné udělat i pokud bude stránka přiblížená.
6. **Fixace pozic jednotlivých elektrod.** Uživatel bude mít při rozmístění elektrod možnost zafixovat pozici každé elektrody jednotlivě.
7. **Rozlišení elektrod.** Aplikace bude umožňovat rozlišit od sebe elektrody a to změnou barvy.

8. **Fixace pozic všech elektrod.** Uživatel bude mít při rozmístění elektrod možnost zafixovat pozice všech elektrod najednou.

9. **Přiřazení signálů k jednotlivým kontaktům na elektrodě.** Uživatel bude mít možnost přiřadit k jednotlivým kontaktům na elektrodě naměřené EEG výstupy podle toho jak jsou připojeny ve skutečnosti.

10. **Zobrazení statistiky spike detektoru.** Uživatel bude mít možnost zobrazit na elektrodách statistiku spike detektoru. Množství detekovaných špiček (spikes) v signálu bude rozlišeno pomocí škály barev.

11. **Umístění jedné elektrody na několik míst zároveň.** Při rozmisťování elektrod na schémata mozků bude mít uživatel možnost umístit jednu elektrodu na více míst zároveň, aby měl možnost vidět elektrodu z různých úhlů, tzn. „zkopírovat“ elektrodu a umístit ji na odlišný obrázek, aby ji viděl pod jiným úhlem.

12. **Změna gradientu pro zobrazení statistiky.** Uživatel bude mít možnost změnit barevný gradient, který je používán pro zobrazení statistiky na elektrodách. Aplikace bude umožňovat změnit barvy i meze (minimum a maximum).

13. **Zobrazení tabulky statistiky**. Uživatel bude mít možnost zobrazit tabulku s přiřazenými kanály a k nim odpovídající počet spiků ze spike detektoru.

14. **Přepnutí do aplikace Alenka.** Z aplikace bude mít uživatel možnost vrátit se zpět do aplikace Alenka.

15. **Uložit stav aplikace**. Aplikace bude umožňovat uložit výsledné prostorové uspořádání elektrod (tzn. použité elektrody, jejich umístění, přiřazené EEG kanály a nastavení rozmezí barevného gradientu).

16. **Načíst stav aplikace.** Uživatel bude mít možnost načíst uložený stav aplikace, tzn. použité elektrody, jejich umístění, přiřazené EEG kanály a nastavení rozmezí barevného gradientu.

3.2.2 Nefunkcionální požadavky

1. **Mobilní aplikace.** Aplikace bude vyvíjena pro tablet a bude využívat možnost intuitivního dotykového ovládání.

2. **Uživatelská přívětivost.** Aplikace bude uživatelsky přívětivá a jednoduchá na ovládání, aby ji byli schopni ovládat i uživatelé bez speciálního technického vzdělání.

3. **Podpora dotykového ovládání.** Aplikace bude podporovat dotykové ovládání, konkrétně zoom a rotaci pomocí dvou prstů na vybraných položkách.

4. **Kompilace pro operační systém Windows.** Aplikace bude spustitelná na operačním systému Windows 8 a 10.

5. **Dokumentace.** Zdrojový kód bude obsahovat dokumentaci pro lepší pochopení a orientaci. Dokumentace bude psaná v anglickém jazyce.

6. **Rozšiřitelnost aplikace.** Aplikaci bude možné rozšířit o nové funkcionality a upravovat funkce stávající.

7. **Integrace s programem Alenka.** Aplikace bude spojena s programem Alenka v rámci jedné aplikace a bude možné mezi nimi přepínat. Aplikace bude používat výstupy spike detektoru, který je implementován v Alence.

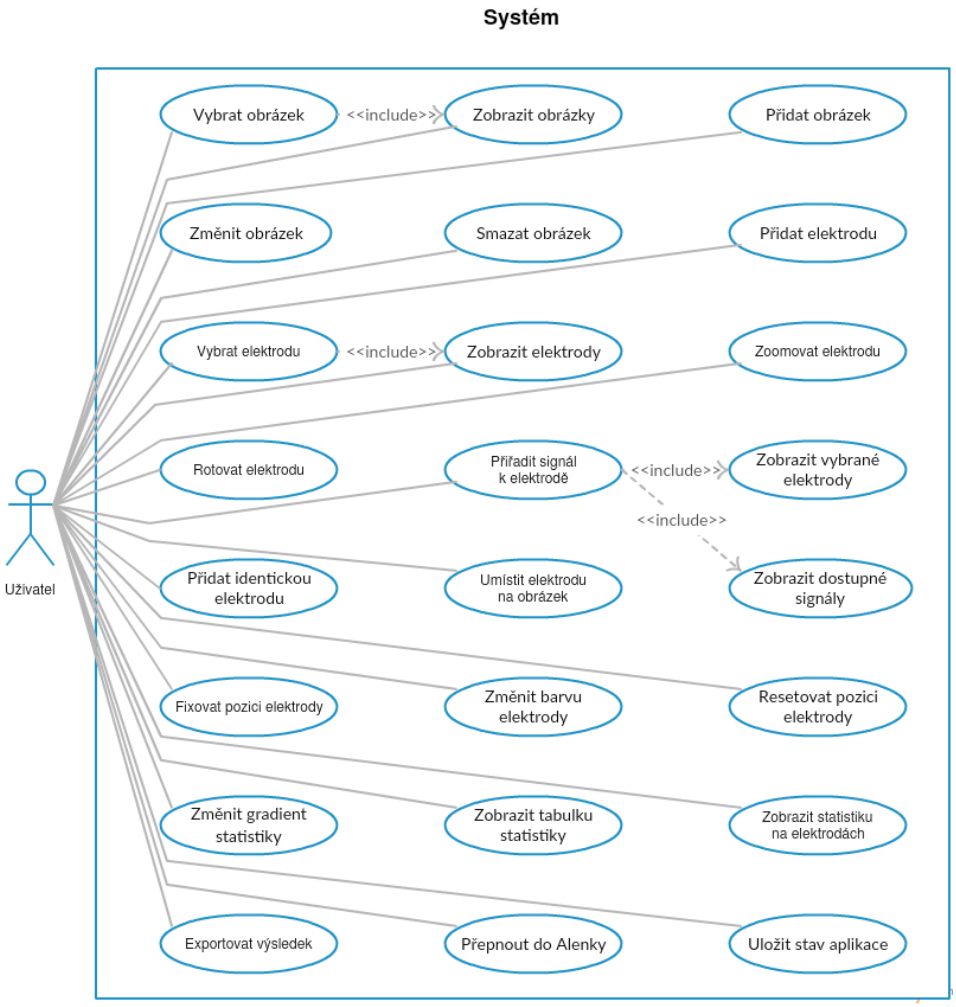
3.3 Případy užití

V této části se zaměřím na model případů užití (tzv. use case model, obr. 3.1), který definuje hranice systému. Vně systému stojí tzv. aktéři (uživatelé systému) a v systému jsou definovány případy užití, tzn. co bude systém aktérům umožňovat. Případy užití vychází z funkčních požadavků (viz kapitola 3.2.1).

Na obrázku 3.1 je znázorněn „Uživatel“, který je v případě této aplikace jediným aktérem. Dále jsou v modelu znázorněny jednotlivé případy užití a jejich vazby. Dále v této kapitole jsou popsány scénáře vybraných případů užití, případně je uveden slovní popis.

3.3.1 Změnit obrázek

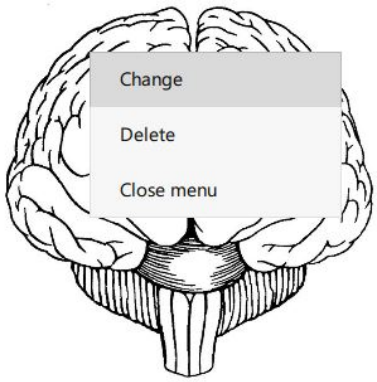
1. Use case začíná otevřením menu u obrázku a vybráním „Change“ (obr. 3.2).
2. Systém otevře dialogové okno pro výběr souboru.
3. Uživatel vybere soubor.
   a. IF soubor je obrazového formátu
   b. THEN systém zobrazí obrázek místo původního obrázku
   c. ELSE systém zobrazí dialogové okno s upozorněním, aby byl vybrán soubor s validní příponou.



Obrázek 3.1: Use case model

3.3.2 Přidat obrázek

1. Use case začíná vybráním obrázku plus.
2. Systém otevře dialogové okno pro výběr souboru.
3. Uživatel vybere soubor.
   a. IF soubor je obrazového formátu
   b. THEN systém zobrazí obrázek místo obrázku plus, zobrazí checkbox pro výběr a přidá nový obrázek plus pro přidání dalšího obrázku
   c. ELSE systém zobrazí dialogové okno s upozorněním, aby byl vybrán soubor s validní příponou.



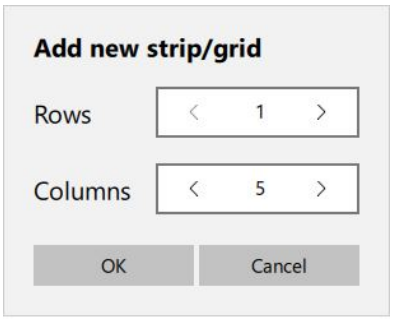
Obrázek 3.2: Ukázka kontextového menu u obrázku

3.3.3 Smazat obrázek

1. Use case začíná otevřením kontextového menu u obrázku a vybráním „Delete“ (viz obr. 3.2).
2. Systém smaže vybraný obrázek.
   a. Všechny následující obrázky systém posune o jednu pozici dopředu.
   b. IF na poslední stránce nezůstane žádný obrázek
   c. THEN systém odstraní poslední stránku.

3.3.4 Přidat elektrodu

1. Systém zobrazí dialogové okno pro určení počtu řádků a sloupců nové elektrody (obr. 3.3).



Obrázek 3.3: Ukázka dialogu pro přidání nové elektrody

2. Uživatel pomocí šipek určí počet řádků a sloupců nové elektrody a potvrdí svůj výběr.
   a. IF jeden rozměr elektrody se rovná 1 (elektroda je typu strip)
   b. THEN systém na konci seznamu se stripy přidá nový řádek se spinboxem pro určení počtu elektrody, popisek rozměrů elektrody a novou elektrodu a přepne na záložku „Strips“
   c. ELSE systém na konci seznamu s gridy přidá nový řádek se spinboxem pro určení počtu elektrody, popisek rozměrů elektrody a novou elektrodu a přepne na záložku „Grids“.

3.3.5 Přiřadit signál k elektrodě

1. Uživatel pomocí metody drag and drop umístí signál na elektrodu.
   a. IF k elektrodě už je přiřazený jiný signál
   b. THEN systém vrátí signál na výchozí pozici odkud byla vzatá
   c. ELSE systém k příslušné elektrodě přiřadí vybraný signál, změní název elektrody na název signálu a elektrodu zvýrazní.

3.3.6 Změnit barvu elektrody

1. Use case začíná otevřením kontextového menu u elektrody a vybraním „Change color..“
2. Systém zobrazí menu s výběrem barev.
3. Uživatel vybere ze zobrazeného menu barvu.
4. Systém změní barvu koncové části elektrody u všech elektrod z daného řádku (identické elektrody) a ohraničení elektrody v řádku na danou barvu. Všechny dále přidané elektrody budou mít stejnou barvu jako ostatní elektrody v řádku.

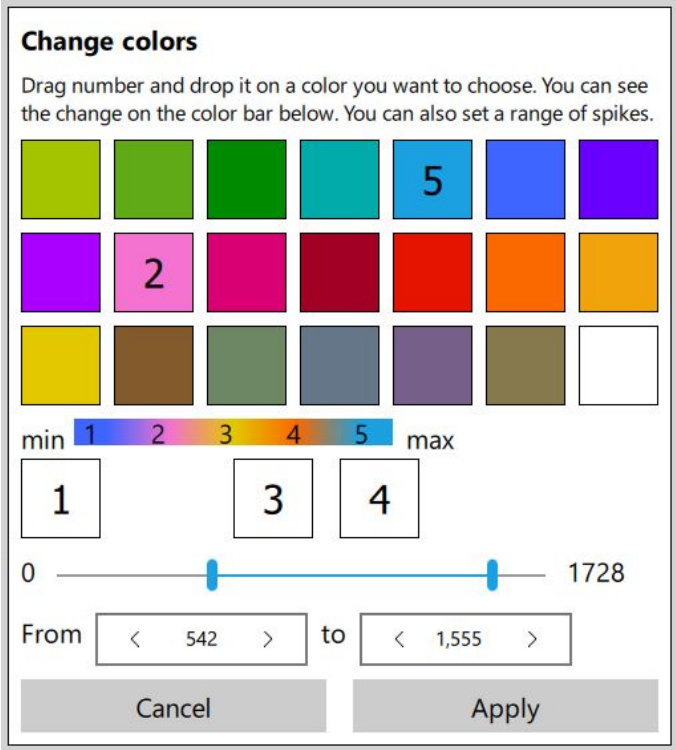
3.3.7 Přidat identickou elektrodu

Identická elektroda se přidává tlačítkem „plus“ u řádku s elektrodou, kterou chce uživatel přidat. Nová stejná elektroda slouží k tomu, aby uživatel mohl

jednu elektrodu umístit na více obrázků (schémat mozků) zároveň a měl tak možnost vidět ji z různých pohledů. K nově vytvořené elektrodě systém přiřadí stejné signály na stejná místa a elektroda je vytvořena se stejnou barvou jakou má elektroda sesterská. Obě elektrody se chovají stejně a barvy se mění zároveň na obou dvou. Je možnost přidat více stejných elektrod.

3.3.8 Změnit gradient statistiky

1. Use case začíná otevřením nastavení gradientu (obr. 3.4).



Obrázek 3.4: Ukázka nastavení barevného gradientu

2. Uživatel přetáhne čtverce s čísly na zvolenou barvu.
   a. IF uživatel číslo pustí jinde než na barevném výběru
   b. THEN systém číslo vrátí do původní pozice
   c. ELSE systém zobrazní číslo na vybrané barvě a změní barvu části barevného gradientu s odpovídajícím číslem na vybranou barvu.
3. Uživatel změní pozice ukazatelů na slideru (zhruba určí minimum a maximum).

4. Systém aktualizuje hodnoty na spinboxech.
5. Uživatel upraví hodnoty pomocí šipek na spinboxech.
6. Systém aktualizuje pozice ukazatelů na slideru.
7. Uživatel potvrdí výběr.
8. Systém zavře nastavení a změní barvy a meze barevného gradientu na stránce. Systém automaticky aktualizuje zobrazení statistiky na elektrodách podle změněného barev a mezí.

3.4 Návrh uživatelského rozhraní

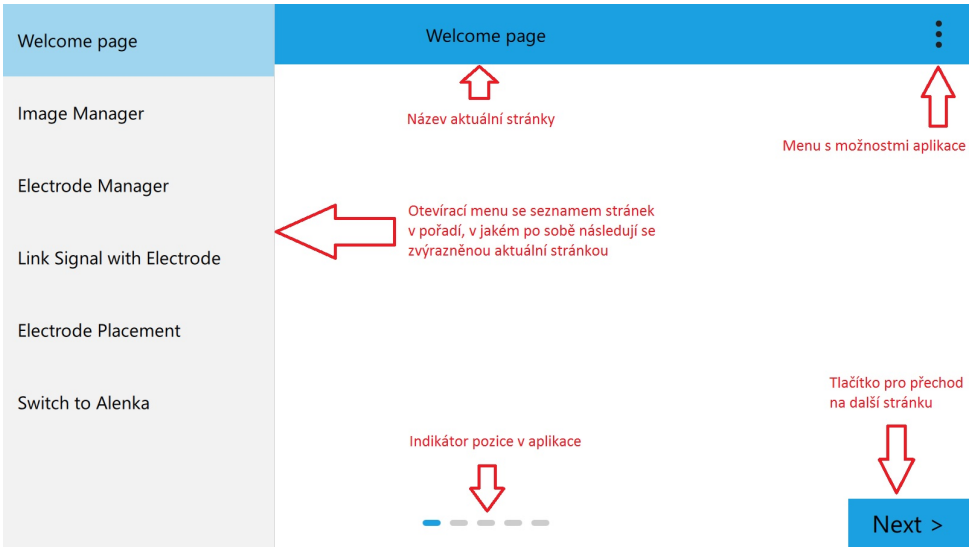
Při návrhu uživatelského rozhraní bylo klíčovým, aby ovládání bylo co nejpřehlednější a rychlé na pochopení. Čas, po který bude uživatel aplikaci na sále používat, je omezený a i kvůli tomu byla aplikace navržena tak, aby ovládání ve všech částech aplikace bylo stejné a uživatel nemusel na každé stránce hledat tlačítko pro potvrzení.

Celá aplikace je rozdělena na několik stránek, kterými uživatel musí projít, aby úspěšně dostal požadovaný výsledek (prostorově a kvantitativně přehledné informace o rozložení epileptických výbojů). Pro přepínání mezi stránkami je v aplikaci navrženo tlačítko „Next“, případně menu v levé části okna (viz dále).

Jak je naznačeno na obrázku 3.5 v dolní části okna se nachází indikátor aktuální pozice v aplikaci. Uživatel tak má lepší představu kolik stránek v aplikaci ho ještě čeká před výsledkem a lépe tak odhadne čas. Tlačítko „Next“ slouží pro potvrzení aktuální stránky a pro přejití na stránku následující. Na poslední stránce je tlačítko „Next“ nahrazeno tlačítkem „Export image“, které slouží pro exportování stránky do obrazového formátu. V levé části zápatí stránky (na obrázku pod otevřeným menu) je umístěno tlačítko „Reset“, kterým je možné vrátit aktuální stránku do výchozí pozice. Pro použití tohoto tlačítka je vyžadováno potvrzení, aby si uživatel omylem nesmazal práci na dané stránce.

V záhlaví se nachází tlačítko pro otevření menu (které je vidět otevřené na obrázku 3.5), název aktuální stránky a tlačítko pro otevření menu s informacemi o aplikaci. Otevřené menu na obrázku obsahuje seznam všech stránek v pořadí, ve kterém v aplikaci po sobě následují a ve kterém je uživatel projde pomocí tlačítka „Next“. Toto menu slouží k tomu, aby se mohl uživatel vrátit v aplikaci zpět. Menu nemusí být v průběhu používání aplikace použito, pokud uživatel nebude chtít na předchozích stránkách nic měnit nebo se nebude chtít vrátit zpět do aplikace Alenka.

Více o jednotlivých stránkách aplikace a možnostech na nich je uvedeno v příloze C této BP.



Obrázek 3.5: Návrh uživatelského rozhraní

3.5 Varianty řešení integrace s aplikací Alenka

Aplikace vyvíjená v rámci této BP (dále jen Elko) byla nejdříve navržena a programována jako samostatná aplikace, která výstupy ze spike detektoru Alenky získávala z externího XML souboru. Aplikace Alenka [4], která by zobrazovala signál, měla být přidána jako uzavřený modul. Otevírala by se z levého menu (viz obr. 3.5) a zobrazovala se na samostatné stránce. Toto řešení se nakonec ukázalo být nerealizovatelné z několika důvodů, které jsou popsány níže.

1. Alenku bylo potřeba upravit, aby byla ovladatelná na tabletu, tzn. zasáhnout do kódu. Zejména se jednalo o přidání tlačítek CTRL a SHIFT, které se jinak používají z klávesnice. Pomocí těchto kláves a kliknutím a pohybem myší (v případě tabletu dotykem) může uživatel v signálu označit události v signálu. Tuto funkcionalitu bylo potřeba zachovat. Dále bylo vhodné zvětšit některé ovládací prvky pro pohodlnější používání celé aplikace.

2. Alenka je navržená pro desktopové použití, tzn. na monitoru. Při zobrazení na tabletu jako ostatní stránky v Elku, tzn. při zachování záhlaví a zápatí by bylo zobrazení nepřehledné a nepohodlné na ovládání.

3. Implementace Alenky je rozsáhlejší a náročnější na prostředky a spouštění takového aplikace z Elka se ukázalo jako nerealizovatelné.

Z výše uvedených důvodů byla aplikace Elko navržena jako součást aplikace Alenka. Přístup k zobrazení signálů z menu byl zachován (viz obr. 3.5 - položka „Switch to Alenka“ v menu) a mezi dvěma částmi aplikace (Elkem a Alenkou) je možno přepínat. Alenka je tedy zobrazena přes celou obrazovku (bez záhlaví a zápatí Elka), aby se tak maximálně využila obrazovka tabletu (viz obr. C.1). Alenka byla mírně upravena pro potřeby použití na tabletu a bylo přidáno tlačítko „Switch to Elko“, kterým se uživatel přepne zpět do Elka. Funkcionalitu Alenky by pro potřeby použití na tabletu bylo možné zredukovat, ale po konzultacích s vedoucím BP jsme se rozhodli zachovat funkcionalitu v nezměněné podobě.

Aby se celá aplikace dala používat na počítači i na tabletu a byla optimalizovaná pro vybraný typ, je v aplikaci navržen přepínač pro přepínání mezi tablet a pc verzí.

V konečném řešení je aplikace navržena tak, že se při spuštění (ve verzi pro tablet) otevře upravená verze Alenky pro tablet, kde uživatel vybere soubor s nahraným EEG signálem a spustí nad ním spike detektor. Potom se přepne pomocí tlačítka „Switch to Elko“ do části, které byla vyvinuta v rámci této BP a je optimalizována pro tablet (aplikace Elko). Tato část se zobrazuje v režimu fullscreen (viz obr. 3.5).

Způsobu řešení integrace, tzn. implementaci, se věnuji v kapitole 4.2.

Kapitola 4

Implementace

V této kapitole vyjmenuji použité technologie a zdůvodním proč jsem si je vybrala a zaměřím se na samotnou implementaci programu. Také se podrobně věnuji integraci s aplikací Alenka [4], která byla pro tento projekt významná.

4.1 Výběr technologií

4.1.1 Qt

Pro vývoj aplikace jsem se rozhodla použít technologii Qt a s ním spojené vývojové prostředí QtCreator [18]. Qt je multiplatformní framework, který je mimo jiné využíván i pro přístroje v medicíně [17]. QtCreator využívá externí programy pro překlad a spouštění aplikací. Pro kompilaci jsem použila Microsoft Visual Studio C++ 2015 Compiler. GUI knihovnu Qt využívá aplikace Alenka, s kterou měla být vyvíjená aplikace spojena a i proto jsem se rozhodla používat stejný aplikační rámec pro snadnější integraci. Já jsem se rozhodla nepoužívat GUI knihovnu Qt, které je implementována v jazyce C++, ale použít lépe čitelný jazyk QML [15] s modulem `Qt Quick`. Jazyk C++ jsem použila pro samotné spuštění aplikace a pro změny v programu Alenka související s integrací.

4.1.2 QML

QML je deklarativní jazyk se syntaxí podobnou JSON (JavaScript Object Notation). Pro návrh UI využívám modul Qt Quick Controls 2, který obsahuje předdefinované grafické komponenty potřebné pro návrh uživatelského rozhraní. Chování aplikace lze skriptovat pomocí jazyka JavaScript, který je součástí jazyka QML.

Výhodou tohoto jazyka je především snadná čitelnost a dobrá přehlednost, která usnadní orientaci v kódu a případné další rozšiřování. Modul Qt Quick obsahuje odladěné stavební bloky, které jsou snadno použitelné a modifikovatelné pro vlastní potřeby. Další důvodem výběru jazyka QML je podpora dotykového ovládání.

4.2 Integrace s aplikací Alenka

Aplikace Alenka [4] je určená především pro desktopové použití, ale požadavek na tuto práci byl, aby ji bylo možné použít i na tabletu. Z toho důvodu bylo potřeba přidat především tlačítka CTRL a SHIFT, jejichž funkcionalita zastupuje stisknutí odpovídajících kláves na klávesnici. Stisknutím tlačítka a pohybem myší (v případě tabletu dotykem) může v uživatel v signálu označit další události. Dále bylo vhodné zvětšit některé ovládací prvky pro pohodlnější používání celé aplikace. Aby se zvětšená tlačítka nezobrazovala i při použití na desktopu, je v aplikaci implementován přepínač pro použití na tablet.

Přepínač je možné nastavit v souboru `options.ini`, který je umístěn v pracovní složce aplikace. Název přepínače je `mode` a je možné ho nastavit na hodnoty `desktop`, `tablet` nebo `tablet-full`. Pro primární použití aplikace, tedy při použití na tabletu, je určena hodnota `tablet-full`, ve které se aplikace Elko zobrazuje v režimu fullscreen, aby se maximálně využila obrazovka tabletu. Pro hodnoty `tablet` a `tablet-full` se aplikace Alenka zobrazí se zvětšenými ovládacími prvky. Rozdíl hodnoty `tablet-full` a `tablet` je v otevírání aplikace Elko v režimu fullscreen. Chování aplikace v tabletovém režimu při použití fullscreen (`tablet-full`) není zaručeno při použití na pc s dvěma monitory a mohou se vyskytnou problémy se zobrazováním dialogových oken. Aplikace pro toto použití není určena. Pokud by ale uživatel z nějakého důvodu trval na zobrazování zvětšených ovládacích prvků i při použití dvou monitorů, je k dispozici hodnota `tablet`, která obsahuje zvětšené ovládací prvky, ale nepřepíná aplikaci Elko do fullscreen. Další nastavení v konfiguračním souboru jsou pouze pro samostatnou část Alenky.

Pro realizaci výše zmíněných úprav musel být použit přímo zdrojový kód Alenky (organizace kódu viz obr. 4.1). Po hledání vhodného řešení spojení

těchto dvou aplikací se ukázalo jako nejlepší, pokud bude Elko součástí Alenky a bude se mezi nimi přepínat (viz kapitola 3.5). Aby se Elko neotvíralo jako nové okno využila jsem komponentu `QStackedWidget`, ve které je widget Alenky a widget, ve kterém je umístěno Elko. `QStackedWidget` je umístěn v centrálním widgetu `QMainWindow` (viz obr. v dokumentaci [13]), tzn. centrálním prostoru hlavního okna aplikace a mezi. Zobrazení Alenky je naznačeno na obr. C.1, při přepnutí do Elka se skryjí viditelné toolbars (na obr. nahoře) a dock widgety (na obr. vpravo), změní se aktuální widget v komponentě `QStackedWidget` a aplikace se přepne do režimu fullscreen. Při zpětném přepnutí do Alenky se toolbars a dock widgety otevřou v takovém stavu, v jakém je uživatel zanechal.

Aplikace Elko není součástí Alenky pouze vizuálně, ale je s ní propojena i na úrovni datové. Automaticky přijímá výsledky detektoru bez nutnosti uložení uživatelem. Tyto data bere ze souborů, které si Alenka průběžně ukládá (autosave). Data jsou ve formátu XML. V Elku se stav aplikace automaticky ukládá pro pozdější úpravu. Stavem aplikace se rozumí především použité elektrody, jejich umístění, rotace a rozměry. Data jsou předány do Alenky ve formátu JSON string a ukládají se do souboru `.info` spolu s dalšími informacemi z Alenky. Pro zpětné načtení je potřeba, aby se nezměnily cesty případných obrázků, které byly přidán uživatelem.

```
Alenka
├── Alenka-File
├── Alenka-Signal
├── doc
├── elko ..... aplikace vyvíjená v rámci této bakalářské práce
│   └── resources
│       ├── +universal
│       ├── images
│       ├── pages
│       ├── main.qml
│       ├── qtquickcontrols2.conf
│       └── resources.qml
├── misc
├── resources
├── src
├── unit-test
├── Alenka.pro ..... soubor pro spuštění celé aplikace v QtCreator [18]
├── build
├── download-libraries.sh
└── README.md
```

Obrázek 4.1: Organizace kódu v rámci aplikace Alenka

Při integraci bylo třeba vyřešit několik problémů spojeným zejména s tím, že se aplikace Elko neotvírá v samostatném okně, ale je zabalená do widgetu.

Pro funkčnost popup elementů v QML je nutné, aby QML aplikace byla samostatné okno [14]. U dialogů a menu nefungovaly především vlastnosti *modality, focus* a *closePolicy*, které jsou zodpovědné za automatické zavírání popup elementů a zabraňování uživateli interakci se zbytkem aplikace pokud je popup otevřen.

Od výše zmíněného chovaní, které by mohlo být vnímáno jako rušivé, jsem se snažila uživatele v maximální možné míře odstínit. Zejména jsem implementovala kontroly, které při používání aplikace kontrolují jestli není otevřený nepotřebný popup a případně ho automaticky zavřou. Jelikož nebylo možné zařadit tyto kontroly pro všechny menu, které je možné v aplikaci otevřít, přidala jsem do všech menu položku *Close menu*, pomocí které může uživatel menu zavřít ručně.

Pro zajištění fungování metody drag and drop v dialogu s nastavením barevného gradientu (viz obr. 3.4), konkrétně pro správné určení rodiče přesouvaného objektu, je dialog naprogramován a zobrazován jako samostatná stránka. Pro uživatele nemá tento jiný způsob implementace žádné důsledky.

4.3 Uživatelské rozhraní

Vyvíjená aplikace je cílená na platformu Windows, proto je aplikace navržena ve stylu Universal [22], který je založen na Microsoft Universal Design Guidelines [7]. Nastavení stylu je definováno v konfiguračním souboru `qtquickcontrols2.conf`. Vybrané komponenty, které jsem upravovala pro potřeby aplikace a které se v kódu často opakují, se nacházejí ve složce `+universal` (viz obr. 4.1).

V konfiguračním souboru jsou definovány i hodnoty pro styl Material [12], který je založen na Google Material Design Guidelines [6]. Tento styl není v aplikaci použit, ale definovala jsem hodnoty pro případné rozšíření aplikace, ve kterém by se přepínalo mezi styly.

4.4 Architektura

V této části se věnuji samostatně aplikaci vyvíjené v rámci této bakalářské práce. O integraci s Alenkou a vyřešení struktury kódu viz kapitoly 3.5 a 4.2.

Každá stránka aplikace a každý složitější element, který je jako samostatná třída, je rozdělen na vrstvu prezentační a aplikační. Prezentační vrstva se stará o zobrazení uživatelského rozhraní, správné uspořádání komponent

na stránce a vrstva aplikační zpracovává signály z komponent a zajišťuje komunikaci mezi stránkami.

V QML jsou soubory prezentační vrstvy pro přehlednost nazývány jménem souboru s přidaným *Form* na konci názvu. Soubory se tedy jmenují např. *ElectrodeManager*, kde je implementována logika stránky a *ElectrodeManagerForm*, kde je implementováno GUI stránky. Hlavním prvkem souboru *ElectrodeManager* je potom *ElectrodeManagerForm*. Ukázka implementace je naznačena na obr. 4.2.

/ElectrodeManagerForm.qml

```
import QtQuick 2.7
import QtQuick.Controls 2.1

Page {

    property alias addButton: addButton

    Button {
        id: addButton
        text: qsTr("Add electrode")
    }
    .
    .
    .
    .
    .
    .
}
```

/ElectrodeManager.qml

```
import QtQuick 2.7

ElectrodeManagerForm {

    addButton.onClicked: addElectrode()

    function addElectrode() {
        .
        .
        .
        .
        .
        .
    }
}
```

Obrázek 4.2: Ukázka kódu

Hlavní funkcionalita aplikace je implementována v třídách `ImageManager`, `ElectrodeManager`, `ElectrodeSignalLink` a `ElectrodePlacement`. Tyto třídy implementují funkci `confirm()` a `reset()`, které definují akci tlačítek Next a Reset v zápatí okna. Tlačítka jsou určena pro potvrzení změn na stránce a přejití na stránku následující a pro resetování veškerých změn na stránce. Uživatelské rozhraní stránek je definováno v odpovídajích `Form` třídách.

Třídy `ImageManager` a `ElectrodeManager` se starají o správu obrázků, příp. elektrod. `ElectrodeSignalLink` zajišťuje přiřazování EEG kanálů k jednotlivým kontaktům elektrod, aby na každém kontaktu byl umístěn maximálně jeden kanál tak jak to odpovídá skutečnému zapojení. V `ElectrodePlacement` probíhá samotné umisťování elektrod na schémata (obrázky) mozků a zobrazování statistiky spiků. Funkce `confirm()` implementuje export stránky do obrazového formátu.

Jednotlivé stránky aplikace a možnosti na nich jsou detailně popsány v příloze C.

4.5 Další části implementace

4.5.1 Podporované formáty a export

Pro export výsledné stránky *Electrode Placement* jsou podporovány standardní obrazové formáty `.jpeg, .bmp, .png`. Formát a cestu exportu si může uživatel zvolit v `options.ini`. Ve výchozím nastavení se export uloží ve formátu `jpg` do složky `Pictures` se stejným názvem jako je název souboru s EEG daty. Export stránky je implementován jako screenshot obrazovky, aby na výsledném obrázku bylo k dispozici i aktuální nastavení (nastavené barvy a meze). Export je možné provést i pokud je stránka přiblížená.

Pro přidání obrázku pro použití v aplikaci jsou rovněž podporovány standardní obrazové formáty. Pro výběr souborů daného formátu je v dialogovém okně k dispozici filtr.

4.5.2 Použití pro desktop

Aplikace (část Elko) je optimalizována na použití na tabletu a bude primárně používaná na operačních sálech. Aby použití nebylo takto omezené a aplikace se mohla používat i na počítačích bez dotykové obrazovky, rozhodla jsem se implementovat i ovládání pomocí myši a klávesnice. Ovládání pouze dotykově se týkalo zejména zoomování a rotace (tzn. smart zoom). Ovládání zoomu je implementováno standardním způsobem a to pomocí kolečka myši a rotace pomocí tlačítka CTRL a kolečka myši. Dále jsou pro použití na pc uzpůsobeny prvky spinbox (pro zadávání množství tlačítky „<“ a „>“), do kterých je číslo možné zadat i prostřednictvím klávesnice. Do spinboxu je možné zadat pouze čísla a to v povoleném rozsahu.

Jak bylo zmíněno v kapitole 4.2, celá aplikace má nyní dva módy - použití na desktopu a použití na tabletu. Ovládání pomocí myší a klávesnice je možné i v režimu tabletu a naopak.

Kapitola 5

Testování

Aplikaci bylo nutné vhodně otestovat. Testování bylo rozděleno na několik úrovní - manuální testování pomocí testovacích scénářů, testování v různých testovacích prostředích (systémové testování) a uživatelské akceptační testování. Jednotlivým úrovním testováním se věnuji dále v této kapitole.

Program Alenka [4], který je součástí aplikace, byl testován zvlášť v rámci diplomové práce autorem programu.

5.1 Testovací prostředí

Aplikace je určena pro použití na tabletech s operačním systémem Windows a tomu odpovídal i požadavek na tuto BP (viz kapitola 3.2.2). Díky tomu, že vybrané technologie jsou multiplatformní a aplikaci je možné používat i na pc, rozhodla jsem se aplikaci testovat i na jiném OS než Windows, konkrétně Ubuntu. Vzhledem k implementaci Alenky (výpočty jsou prováděny na grafické kartě [4]) bylo potřeba rozlišovat i různé typy grafických karet. U jednotlivých zařízení uvádím jestli je dotykové, tzn. jestli byla otestováno dotykové ovládání aplikace.

Přehled testovacích prostředí je uveden v tabulce 5.1. Aplikaci je možné spustit i ve virtual boxu, kde je nutné nastavit v `options.ini` hodnotu `glSharing` na `0`. U některých zařízeních bylo nutné změnit nastavení v `options.ini`, případně nainstalovat AMD APP SDK [1], které umožnilo provést výpočty nutné pro zobrazení signálu v Alence.

Operační systém	Grafická karta	Dotykové
Windows 8.1 32-bit	Intel HD Graphics in processor	ano
Windows 10 32-bit	AMD Radeon HD 5000	ne
Windows 10 64-bit	Intel HD Graphics in processor	ano
Windows 10 64-bit	AMD Radeon HD 5670	ne
Windows 7/8/10 64-bit	NVIDIA GeForce GTX 650/960	ne
Ubuntu 14/16 64-bit	NVIDIA GeForce GTX 650/960	ne

Tabulka 5.1: Přehled testovacích prostředí

5.2 Manuální testování

Testování interakce uživatele s aplikací byla provedena metodou black-box testing, při které se k aplikaci přistupuje jako k „černé skříňce“ do které se zadávají vstupy (které můžou být kromě dat i kliknutí na ikonu) a očekávají se výstupy/reakce aplikace. Aby tester věděl jak má správně aplikace reagovat po určité sekvenci vstupů, používají se tzv. testovací scénáře. Testovací scénář definuje jaká funkcionalita se testuje, jaké jsou vstupní podmínky, jak při testování postupovat a jaký očekávat výsledek. Příklady testovacích scénářů použitých při testování této aplikace ukazují tabulky 5.2 až 5.8.

ID	ADD_IMAGE_01
Název testu	Přidání vlastního obrázku
Vstupní podmínky	žádné
Popis testu	1. V levém menu klikněte na Image Manager. 2. Pohybem zprava doleva se přesuňte na další stranu. 3. Klikněte na plus obrázek. 4. Vyberte jakýkoliv soubor s příponou .jpg, .png nebo .bmp.
Očekávaný výsledek	Místo plus obrázku se zobrazí vybraný obrázek a k němu se zobrazí checkbox. Objeví se další plus obrázek pro možnost přidání dalšího souboru.

Tabulka 5.2: Testovací scénář č.1

ID	DELETE_IMAGE_02
Název testu	Smazání prázdné stránky při smazání obrázku
Vstupní podmínky	Na poslední stránce je pouze plus obrázek
Popis testu	1. V levém menu klikněte na Image Manager. 2. Na libovolném obrázku stiskněte a podržte. 3. Ze zobrazeného menu vyberte „Delete“.
Očekávaný výsledek	Obrázek se smaže a všechny následující obrázky se posunou o jednu pozici dopředu. Na konci výběru bude stále pouze jeden plus obrázek pro nové přidání. Poslední stránka se smaže.

Tabulka 5.3: Testovací scénář č.2

ID	CHANGE_COLOR_GRADIENT_05
Název testu	Změna barevného gradientu a mezí
Vstupní podmínky	Vybrané obrázky (ze stránky Image Manager), vybrané elektrody (ze stránky Electrode Manager) a přiřazené signály k elektrodám (ze stránky Electrode Signal Link), aktuální stránka Electrode Placement
Popis testu	1. Klikněte na barevný gradient. (Otevře se nastavení.) 2. Přesuňte čtverce s čísly odpovídajícími částem gradientu na vybrané barevné čtverce. Změna zkontrolujte na barevném gradientu níže. 3. Na posuvníku upravte rozsah minima a maxima pro počet spiků. Změnu zkontrolujte v textovém popisu pod posuvníkem (minimum nesmí být větší než maximum). 4. Potvrďte tlačítkem „Apply“.
Očekávaný výsledek	Na stránce „Electrode Placement“ se změní barvy barevného gradientu podle vybraných barev (barevný gradient bude stejný jako barevný gradient v dialogu nastavení). Změní se minimum a maximum a hodnoty mezi nimi se poměrově přepočítají. Při zobrazení statistiky budou použity barvy z upraveného gradientu a statistika se bude zobrazovat pouze v zadaném rozsahu (od vybraného minima do vybraného maxima).

Tabulka 5.4: Testovací scénář č.3

ID	ADD_GRID_03
Název testu	Přidání elektrody-gridu o vlastních rozměrech
Vstupní podmínky	Stránka Electrode Manager na záložce „Strips“"
Popis testu	1. V levém menu klikněte na Electrode Manager. 2. Klikněte na „Add new electode“. 3. Pomocí tlačítek „<“ a „>“ určete počet sloupců a řádků tak, aby oba údaje byly větší než 1. 4. Potvrďte tlačítkem „OK“.
Očekávaný výsledek	Stránka se přepne na záložku „Grids“ a na konec seznamu se přidá nový řádek s popisem rozměrů, spinbox pro výběr počtu elektrod nastavený na nulu a elektroda o zadaných rozměrech.

Tabulka 5.5: Testovací scénář č.4

ID	CHANGE_COLOR_04
Název testu	Přidání stejné elektrody a změna barvy elektrod
Vstupní podmínky	Vybrané obrázky, vybrané elektrody a přiřazené signály k elektrodám, aktuální stránka „Electrode Placement“
Popis testu	1. Libovolnou elektrodu umístěte přetažením na obrázek. 2. V řádku dané elektrody klikněte na „+“. (Měla by se zobrazit nová elektroda o stejných rozměrech a se stejnými přiřazenými signály.) 3. Elektrodu přetáhnutím umístěte na obrázek. 4. Na elektrodě 2x klikněte. 5. Ze zobrazeného menu vyberte „Change“ „color...“ 6. Ze zobrazeného podmenu vyberte libovolnou barvu.
Očekávaný výsledek	U stejných elektrod se změní barva koncové části na vybranou barvu. V řádku seznamu se rámeček elektrody změní na vybranou barvu. Při přidání další elektrody bude mít elektroda stejnou barvu jako ostatní.

Tabulka 5.6: Testovací scénář č.5

ID	SHOW_STATISTICS_07
Název testu	Zobrazení statistiky
Vstupní podmínky	Vybrané elektrody a přiřazené signály k elektrodám, aktuální stránka Electrode Placement
Popis testu	1. Na elektrodách zobrazte statistiky pomocí tlačítka Show spike statistics. 2. Zobrazte si tabulku se statistikou tlačítkem Show statistics table. 3. Tabulku zkontrolujte s tabulkou v aplikaci Alenka (Event Manager).
Očekávaný výsledek	Zobrazené barvy odpovídají barevné škále. Čísla větší než max., příp. menší než min. mají stejnou barvu jako maximum, popř. minimum.

Tabulka 5.7: Testovací scénář č.6

ID	CHANGE_ELECTRODES_06
Název testu	Vrácení se v aplikaci, změna vybraných elektrod
Vstupní podmínky	Vybrané obrázky (ze stránky Image Manager), vybrané elektrody (ze stránky Electrode Manager) a přiřazené signály k elektrodám (ze stránky Electrode Signal Link), aktuální stránka Electrode Placement s rozmístěnými elektrodami
Popis testu	1. Přes levé menu se vraťte na stránku „Electrode Manager“ 2. Jednu elektrodu odstraňte a další přidejte. 3. Potvrďte tlačítkem „Next“ 4. Zkontrolujte počet a rozměry elektrod odpovídají novému výběru. Signály u smazané elektrody by měly být zpátky ve sloupečku pro výběr, ostatní by měly být ve stejném stavu jako předtím. 5. Potvrďte tlačítkem „Next“.
Očekávaný výsledek	Na stránce „Electrode Placement“ zůstanou původní elektrody ve stejné pozici, smazaná elektroda se odstraní a nové elektrody se přiřadí na konec seznamu.

Tabulka 5.8: Testovací scénář č.7

5.3 Uživatelské akceptační testy

Uživatelské akceptační testy (UAT) byly prováděny průběžně během pravidelných konzultací se zadavatelem práce. Testování probíhalo na reálných datech. Aplikace zatím nebyla otestována přímo na sále během operace, protože ve FN v Motole nebyla v období dokončení této bakalářské práce prováděna žádná intraoperační elektrokortikografie, na které bych mohla aplikaci vyzkoušet.

Testování bylo simulováno v podobných časových podmínkách jako na sále. Při operacích trvá elektrokortikografie necelých 30 minut, od rozmístění elektrod až po konečné vyhodnocení. EEG se měří cca 15 minut a vyhodnocení zabere doktorům 5 až 10 minut. Operace trvají několik hodin a na dlouhé vyhodnocování není na sále čas. Během těchto 5 až 10 minut by měl být lékař schopný použít aplikaci této BP a dostat z ní požadovaný výsledek.

Aplikace byla otestována čtyřmi uživateli - studentkou VŠCHT, studentem FIT ČVUT, absolventem FEL ČVUT a absolventkou VŠE. Testeři byli vybráni tak, aby byli zastoupeni programátoři i lidé, kteří o programování znalosti nemají. Testerům byl ve stručnosti vysvětlen kontext aplikace a k čemu bude používána. K dispozici měli tablet s nainstalovanou aplikací a souborem s EEG daty a papír s obrázkem intraoperační elektrokortikografie. Rozmístění elektrod na obrázku měli za úkol nasimulovat na tabletu s použitím vhodných elektrod a schémat mozků a výsledek vyexportovat. Výsledky testování byly závislé na tom, jestli daný člověk byl zvyklý používat tablet nebo ne. Těm kteří tablet nepoužívali trval průchod aplikací v průměru o 3 minuty déle.

Kapitola 6

Závěr

6.1 Zhodnocení splnění cílů

Cílem této bakalářské práce bylo poskytnout lékařům kvantitativní a prostorově uspořádaný přehled o naměřených EEG datech. K dispozici jsem měla aplikaci Alenka [4], která EEG zpracovává. Analýza ukázala, že optimálním řešením je spojit obě aplikace do jedné a umožnit mezi nimi přepínání.

Při návrhu uživatelského rozhraní jsem zohledňovala především to, že se aplikace bude používat přímo na sále během operace a tudíž by měla být co nejjednodušší na ovládání. Uživatel proto nikde nemusí nic zadávat přes klávesnici nebo se přesně trefovat na určitý prvek (při přiřazování signálů se přetahovaný kanál automaticky umístí přesně doprostřed na kontakt elektrody, stačí aby uživatel kanál pustil, když je kontakt zelený).

Lékaři, příp. ostatní uživatelé, mají v aplikaci možnost na obrázcích, příp. fotkách nasimulovat prostorové uspořádání elektrod na mozku při intraoperační elektrokortikografii. Obrázky i elektrody mají možnost vybrat z předem definovaného výběru nebo mohou přidat vlastní (např. fotografii přímo z probíhající operace nebo elektrodu o nestandardních rozměrech). Jednotlivé kanály EEG přiřadí jednoduše přetáhnutím vybraného kanálu na kontakt, ke kterému má být kanál přiřazen. Na elektrodách s přiřazenými signály můžou uživatelé zobrazit statistiku spiků v signálu pomocí škály barev.

Aplikace byla nad rámec požadavků optimalizována i pro použití na pc (implementováno ovládání pomocí klávesnice a myši) a otestována na operačním systému Ubuntu.

6.2 Možná rozšíření a vylepšení

Ve vyvinuté aplikaci je prostor pro možná rozšíření a vylepšení. Níže uvádím seznam věcí, které mě napadly při práci, příp. při konzultacích a mohly by být pro další uživatele užitečné.

- Větší podpora dotykového ovládání aplikace Alenka - horizontální scrollování v signálu, zoomování signálu pomocí dvou prstů, checkboxy v tabulkách pro možnost označení více řádků zároveň, apod.
- Implementace dalších modulů do Alenky a díky tomu možnost zobrazení dalších statistik.
- Možnost přepínání světových jazyků (kód aplikace Elko je na lokalizace připraven, všechny textové řetězce jsou označeny pro překlad pomocí `qsTr()` [16]).
- Možnost přepínání stylů zobrazení (viz kapitola 4.3).
- Možnost vlastního očíslování elektrod.
- Možnost změnit velikost kontaktu na elektrodě.
- Export do formátu PDF.
- Rozšíření podpory obrazových formátů.

Další náměty na možná vylepšení se mohou vyskytnout po použití na operačním sále.

Literatura

[1] *APP SDK - A Complete Development Platform - AMDe* [online]. [cit. 20. 05. 2017]. Dostupné z: <http://developer.amd.com/tools-and-sdks/opencl-zone/amd-accelerated-parallel-processing-app-sdk/>.

[2] Brázdil, M. – Marusič, P. *Epilepsie temporálního laloku.* Triton, 2006. ISBN 80-7254-836-0.

[3] Brázdil, M. – Hadač, J. – Marusič, P. *Farmakorezistentní epilepsie.* Triton, 2004. ISBN 80-7254-562-0.

[4] Bárta, M. *Specializovaný systém pro zobrazování biologických signálu pacientu zařazených do epilepto-chirurgického programu – rozširovací moduly.* Bachelor thesis, České vysoké učení technické v Praze, Fakulta Informačních Technologií, May 2015.

[5] Drábek, J. *Multiplatformní implementace spike detektoru pro intraoperační elektrokortikografii.* Bachelor thesis, České vysoké učení technické v Praze, Fakulta Informačních Technologií, April 2015.

[6] *Material Design* [online]. [cit. 10. 05. 2017]. Dostupné z: <https://material.io/>.

[7] *Design Universal Windows Platform (UWP) app - UWP app developer* [online]. [cit. 2. 05. 2017]. Dostupné z: <https://developer.microsoft.com/en-us/windows/apps/design>.

[8] Hovorka, J. Farmakologická léčba epilepsie. *Česká a slovenská neurologie a neurochirurgie.* July 2010, 73/106(4), s. 351–373. Dostupné z: <http://www.csnn.eu/pdf?id=33838>.

[9] *ISARG - Intracranial Signal Analysis & Research Group* [online]. [cit. 10. 04. 2017]. Dostupné z: <http://isarg.fel.cvut.cz/index.html>.

[10] Kršek, P. *Chirurgická léčba epilepsie u dětí* [online]. [cit. 23. 04. 2017]. Dostupné z: <http://www.fnmotol.cz/kliniky-a-oddeleni/cast-pro-deti/klinika-detske-neurologie-uk-2-lf-a-fn-motol/chirurgicka-lecba-epilepsie-u-deti/>.

[11] Kwan, P. et al. Definition of drug resistant epilepsy: Consensus proposal by the ad hoc Task Force of the ILAE Commission on Therapeutic Strategies. *Epilepsia - Official Journal of the International League Against Epilepsy*. June 2010, 51, s. 1069–1077. Dostupné z: <http://dx.doi.org/10.1111/j.1528-1167.2009.02397.x>.

[12] *Material Style | Qt Quick Control 2 5.8* [online]. [cit. 10. 05. 2017]. Dostupné z: <https://doc.qt.io/qt-5/qtquickcontrols2-material.html#detailed-desc-material>.

[13] *QMainWindow Class* [online]. [cit. 1. 05. 2017]. Dostupné z: <http://doc.qt.io/qt-5/qmainwindow.html>.

[14] *Popup QML Type | Qt Quick Control 2 5.8* [online]. [cit. 2. 05. 2017]. Dostupné z: <https://doc.qt.io/qt-5/qml-qtquick-controls2-popup.html>.

[15] *QML Applications | Qt 5.8* [online]. [cit. 5. 05. 2017]. Dostupné z: <http://doc.qt.io/qt-5/qmlapplications.html>.

[16] *Internationalization and Localization with Qt Quick* [online]. [cit. 2. 05. 2017]. Dostupné z: <http://doc.qt.io/qt-5/qtquick-internationalization.html>.

[17] *Qt - Industry | Qt in Medical* [online]. [cit. 5. 05. 2017]. Dostupné z: <https://www.qt.io/qt-in-medical/>.

[18] *Qt - Product| The IDE* [online]. [cit. 1. 05. 2017]. Dostupné z: <https://www.qt.io/ide/>.

[19] Shaw, A. Classification of epilepsy and epileptic seizures. *The journal of the Royal Society for the Promotion of Health*. March 2000, 120, s. 9. Dostupné z: <https://doi.org/10.1177/146642400012000109>.

[20] Sommerville, I. *Software Engineering*. Addison-Wesley, 9th edition, 2011. ISBN 978-0-13-703515-1.

[21] Tripathi, M. et al. Intra-operative electrocorticography in lesional epilepsy. *Epilepsy Research*. 2010, 89, 1, s. 133 – 141. ISSN 0920-1211. Dostupné z: <http://doi.org/10.1016/j.eplepsyres.2009.12.007>. Special Issue on San Servolo Epilepsy Courses Alumni Meeting.

[22] *Universal Style | Qt Quick Control 2 5.8* [online]. [cit. 2. 05. 2017]. Dostupné z: <https://doc.qt.io/qt-5/qtquickcontrols2-universal.html>.

[23] *WHO | Epilepsy* [online]. [cit. 11. 04. 2017]. Dostupné z: <http://www.who.int/mediacentre/factsheets/fs999/en/>.

[24] Winslow, T. *Definition of craniotomy - NCI Dictionary of Cancer Terms - National Cancer Institute* [online]. [cit. 15. 05. 2017]. Dostupné z: <https://www.cancer.gov/publications/dictionaries/cancer-terms?cdrid=46457>.

Příloha A

Seznam použitých zkratek

BP	Bakalářská práce
ECoG	Electrocorticography
EEG	Electroencephalography
FN	Fakultní nemocnice
GUI	Graphical User Interface
IDE	Integrated Development Environment
iEEG	Intracranial Electroencephalography
LF UK	Lékařská fakulta Univerzity Karlovy
OS	Operační systém
UAT	User Acceptance Testing
XML	eXtensible Markup Language

Příloha B

Instalační instrukce

Instalační instrukce jsou převzaté z aplikace Alenka [4], protože jsou pro tuto práci totožné.

Requirements

- Git
- Qt 5.8 (gcc/MSVC and QtCharts module)
- g++ or Microsoft Visual C++ 2015 compiler
- OpenCL 1.1
- OpenGL 2.0 and some extensions from OpenGL 3.0

On Debian-like systems you can use: sudo apt install git cmake-gui build-essential

Install Qt via the installer on their website. Select the „Qt 5.8 msvc2015 64/32-bit“ package for Windows, or „Desktop gcc 64/32-bit“ for Linux. Also select the „QtCharts“ module.

MSVC compiler can be acquired by installing Visual C++ Build Tools 2015. Choose „Custom Installation“, and uncheck all options but „Windows

8.1 SDK“. If you already have Visual Studio, you probably don’t need to install this.

If your device doesn’t support OpenCL (e.g. when running a Linux guest in VirtualBox), use AMD APP SDK for a CPU implementation of OpenCL.

Build instructions

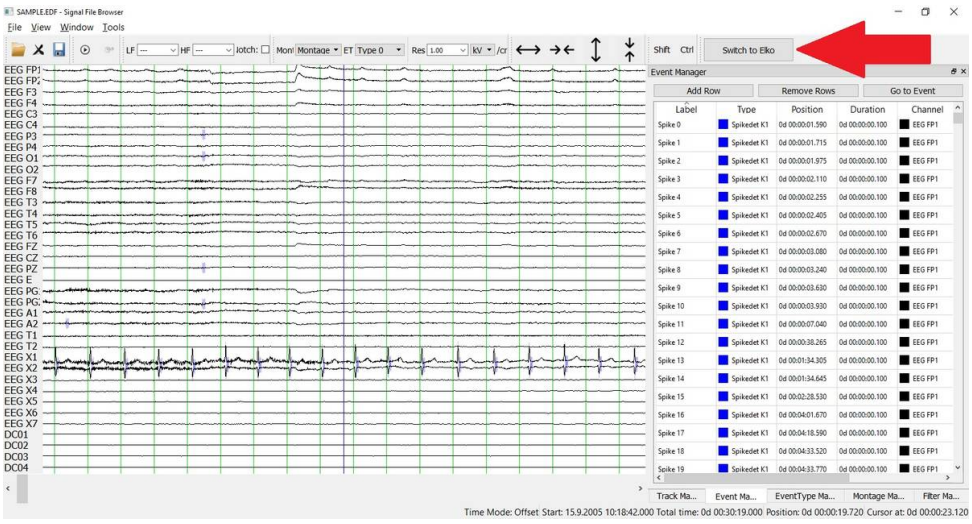
1. Run download-libraries.sh (using git-bash on Windows), and build the subprojects Alenka-File and Alenka-Signal (in subproject are README files)
2. Copy „build.template“ file to „build“
3. Modify build file to suit your system (mainly change paths to libraries)
4. Open „Alenka.pro“ in Qt Creator (an IDE that was installed alongside Qt), and setup the project (I recommend changing the working directory in Projects -> Run.)
5. Build the project using the hammer button, and run the program via the green play button

Příloha C

Návod k použití

Aplikace vyvíjená v rámci této BP (dále jen Elko) je součástí aplikace Alenka [4], ze které se rovněž spouští. Tento návod se týká aplikace Elko a nezbytně nutným průchodem aplikací Alenka. Aplikace Alenka není předmětem této BP a proto se nezabývá ani jejím ovládáním.

Alenka se spouští souborem `Alenka.exe`. Otevře se hlavní okno Alenky, ve kterém uživatel vybere soubor (prostřednictvím menu vlevo nahoře) a spustí nad ním spike detektor (také prostřednictvím menu nahoře). Tlačítkem *Switch to Elko* (obr. C.1) se uživatel přepne do Elka.



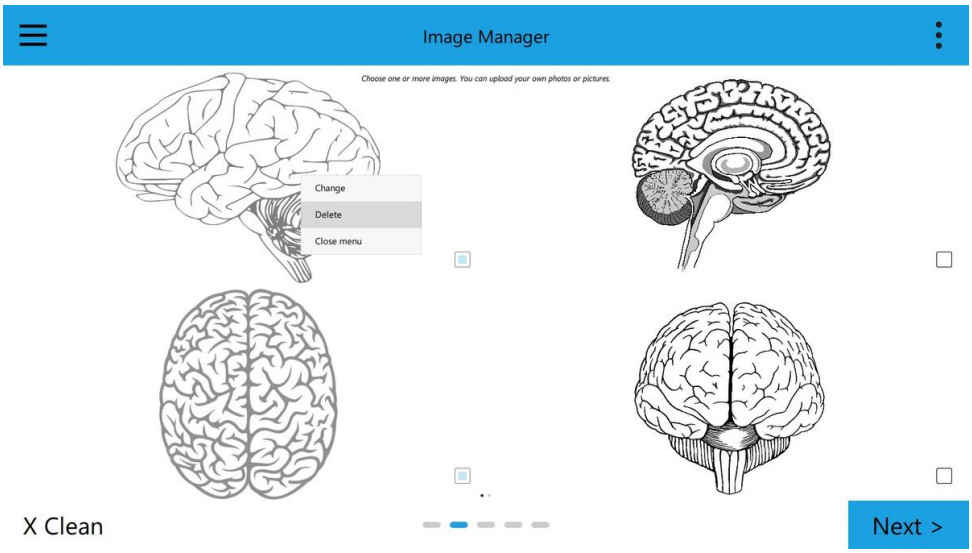
Obrázek C.1: Obrazovka aplikace Alenka [4]

V Elku je několik základních ovládání, které se používá napříč aplikací:

- Dvojklikem se otevře kontextové menu.
- Dvěma prsty se může „zoomovat“, příp. rotovat (u vybraných elementů).
- Jedním stisknutím se položka označí nebo stane aktuální.
- Výběr na stránce se potvrzuje tlačítkem *Next*, které posune uživatele na další krok.
- Tlačítko *Reset* slouží pro navrácení stránky do původního stavu, tzn. elektrody se vrátí na původní pozici nebo přiřazené signály jsou z elektrod odebrány. Tlačítko se vztahuje vždy pouze na aktuální stránku. Pro navrácení do původního stavu je vyžadováno potvrzení.
- Pomocí levého menu se lze ve stránkách vracet, pro aktualizaci změn je ale nutné změny potvrdit.

Navigační prvky jsou podrobněji popsány v kapitole 3.4.

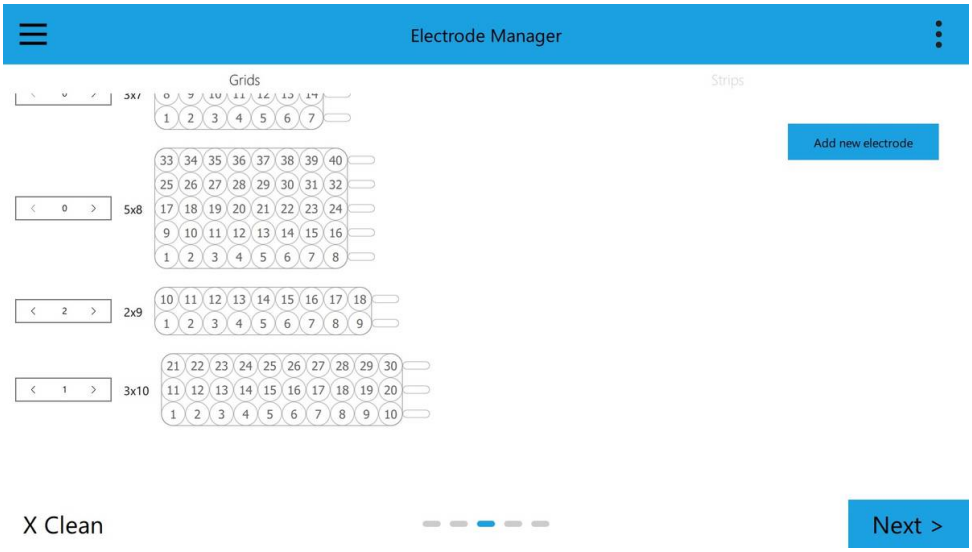
Při přepnutí se zobrazí úvodní stránka, dále se pokračuje pomocí tlačítek *Next*. Na první stránce (obr. C.2) si uživatel vybere schémata a obrázky, které chce dále použít. Posunutím doprava může přejít na další stránku výběru, kde může přidávat další obrázky. Každý obrázek je možné změnit nebo smazat prostřednictvím kontextového menu. Při přidávání obrázků se automaticky přidávají nové stránky, při mazání obrázků se prázdné stránky odstraňují.



Obrázek C.2: Obrazovka Image Manager

Další stránka (obr. C.3) se týká výběru elektrod. Uživatel může přepínat mezi stripy a gridy pomocí horních záložek. U každé elektrody je spinbox pro určení počtu elektrod daného typu. Novou elektrodu je možné přidat

pomocí tlačítka „Add new elektrode“, které otevře dialogové okno pro určení rozměrů elektrody. Nová elektroda se přidá na konec seznamu stripů nebo gridů podle toho jaké má rozměry, aktuální záložka se změní na tu, kam byla nová elektroda přidána. Uživatel pro další pokračování v aplikaci musí vybrat alespoň jednu elektrodu.

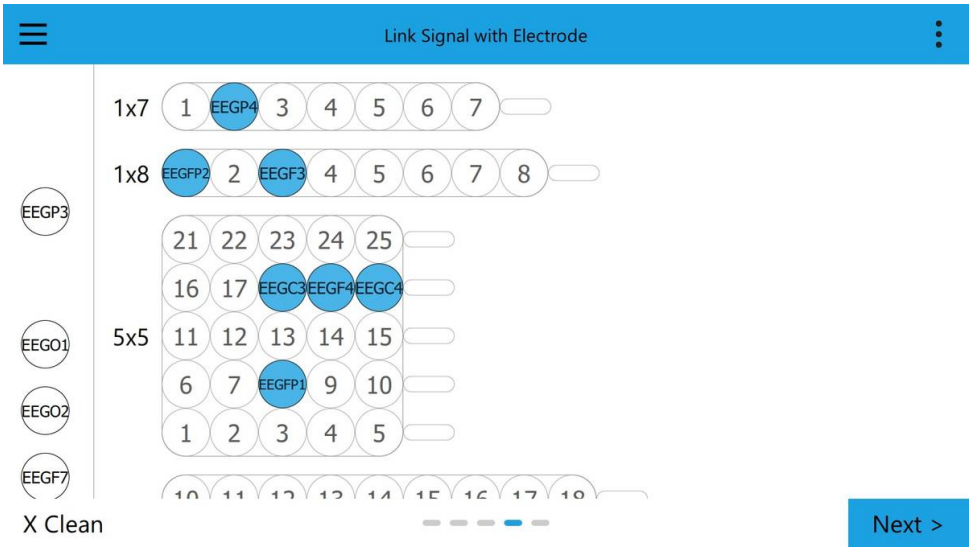


Obrázek C.3: Obrazovka Electrode Manager

V další fázi (obr. C.4) uživatel přiřadí k elektrodám signály, jejichž názvy jsou převzaté z načteného signálu. Přiřazování se provádí metodou drag and drop, ke každé elektrodě (ke každému číslu) může být přiřazen pouze jeden signál. Pokud je signál přiřazen, elektroda se zvýrazní a změní jméno podle přiřazeného signálu. Pokud se uživatel pokusí přiřadit k obsazené elektrodě nový signál, je nový signál vrácen zpět na původní místo. Odebrání signálu z elektrody se provádí kliknutím na signál.

Na poslední stránce (obr. C.5) je samotné umístění elektrod na obrázky a zobrazení statistik. Elektrody se umisťují metodou drag and drop a je možné je přiblížit/oddálit a rotovat. Každá elektroda se dá přidat víckrát (pomocí tlačítka „+“) a umístit na obrázky z různých pohledů. Pro rozlišení elektrod se stejnými rozměry se může elektrodě změnit barva přes kontextové menu. Každé elektrodě je také možno zafixovat pozici a to buď jednotlivě přes kontextové menu elektrody nebo všem najednou pomocí přepínače v pravé části stránky. U elektrod je možno zobrazit pouze názvy přiřazených signálů, defaultní čísla nebo kombinaci obojího. Výběr se provádí pomocí rozbalovacího menu.

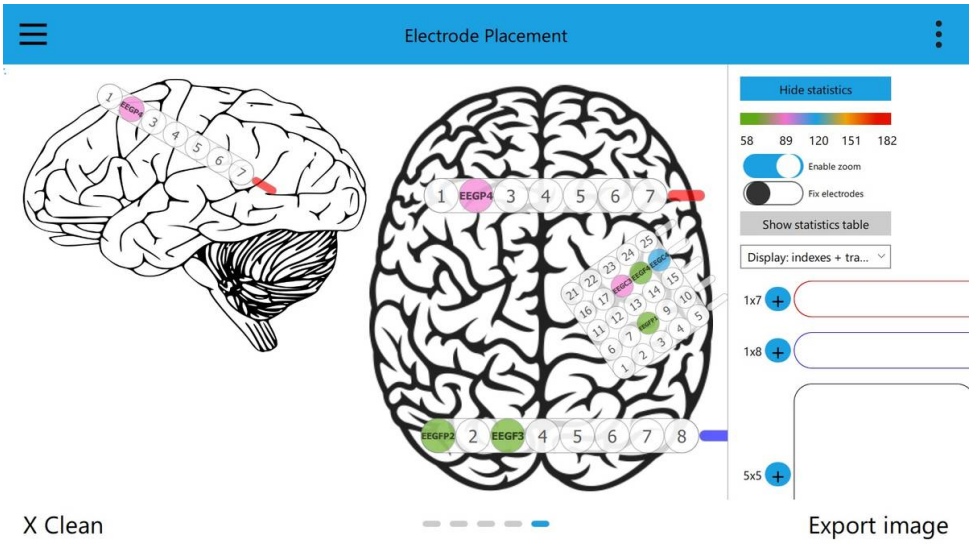
Stiskem tlačítka *Show spikes statistics* se elektrody s přiřazeným signálem zbarví do barvy odpovídající rozmezí na barevném gradientu. Pokud počet špiček v signálu není v rozmezí barevného gradientu je elektroda v barvě extrému. Barvy a rozmezí gradientu lze změnit kliknutím na barevnou škálu. Tabulka s počtem špiček u jednotlivých signálů se zobrazí tlačítkem *Show*



Obrázek C.4: Obrazovka Electrode Signal Link

statistics table.

Výsledný export stránky se provádí jako potvrzení stránky tlačítkem v pravé části zápatí.



Obrázek C.5: Obrazovka Electrode Placement

Přepnout zpět do aplikace Alenka se dá pomocí levého menu a odpovídající položky v menu. Pro aktualizaci změn z Alenky (např. přidání nebo odebrání kanálů) je potřeba potvrdit změny na stránce Link Signal with Electrode.

Příloha D

Obsah přiloženého DVD

- `samples.zip` ..... soubory pro vyzkoušení programu
- `exe` ..... předkompilované instalační balíčky
- `src`
  - `impl` ..... zdrojový kód Alenky v Qt Creator
    - `Alenka-File`
    - `Alenka-Signal`
    - `doc`
    - `elko` ..... zdrojový kód programu této BP
    - `misc`
    - `resources`
    - `src`
    - `unit-test`
    - `Alenka.pro` ..... soubor pro spuštění celé aplikace v QtCreator
    - `build`
    - `download-libraries.sh`
    - `README.md` ..... instalační instrukce
  - `thesis` ..... zdrojová forma práce ve formátu LaTeX
- `text`
  - `BP_Svobodová_Lenka_2017.pdf` ..... text práce ve formátu PDF