**ŽILINSKÁ UNIVERZITA V ŽILINE**

FAKULTA RIADENIA A INFORMATIKY

BAKALÁRSKA

PRÁCA

Tomáš Klein

**Webové služby pre vstavané systémy**

Vedúci práce: Ing. Samuel Žák

Registračné číslo: 28360820171179

Žilina, 2017

#### [Čestné vyhlásenie](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Čestné_vyhlásenie)

Čestne prehlasujem, že som prácu vypracoval samostatne s využitím dostupnej literatúry a vlastných vedomostí. Všetky zdroje použité v bakalárskej práci som uviedol v súlade s predpismi.

Súhlasím so zverejnením práce a jej výsledkov.

...........................................

            V Žiline, dňa .......................                                         Tomáš Klein

#### [Poďakovanie](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Poďakovanie)

Chcem sa poďakovať vedúcemu práce Ing. Samuelovi Žákovi za cenné rady, pripomienky a odborné vedenie pri vypracovaní bakalárskej práce.

[ABSTRAKT](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Abstrakt)

KLEIN Tomáš: Webové služby pre vstavané systémy. [Bakalárska práca] – Žilinská Univerzita v Žiline. Fakulta riadenia a informatiky; Katedra technickej kybernetiky. – Vedúci práce: Ing. Samuel Žák – Stupeň odbornej kvalifikácie: Bakalár – Žilina: FRI UNIZA, 2017. Počet strán: 37.

Bakalárska práca sa zaoberá riešením problému komunikácie medzi režisérom a kameramanmi pri vytváraní kamerového záznamu za pomoci video strižne, ktorá spracováva obrazy z kamier v reálnom čase. V úvode práce autor opisuje spôsoby a využívanú metodiku pre spracovanie obrazových výstupov z kamier a ich záznamu. Ďalej práca opisuje samotnú internú komunikáciu a zameriava sa hlavne na dorozumievanie pomocou Tally svetelnej signalizácie, kde autor opisuje už existujúce možnosti riešenia danej problematiky a taktiež navrhuje vlastné riešenie. To pozostáva z využitia počítača Raspberry Pi a webových služieb, ktoré budú poskytovať komunikáciu pomocou Tally signalizácie a krátkych textových správ. Autor pokračuje popisovaním zvoleného riešenia a návrhom hardvérovej časti pre jeho realizáciu. Analytická časť vysvetľuje základné pojmy a technológie, ktoré sa využívajú pri realizácii webového rozhrania. Posledná časť práce sa venuje prepojeniu jednotlivých čiastkových komponentov medzi sebou a implementovaním systému do prevádzky.

[**Kľúčové slová**](http://Pokyny_pre_vypracovanie_ZP.docx)**:**  Flask, Framework, Raspberry Pi, Tally svetelný systém, WebSocket

[ABSTRACT](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Abstrakt)

KLEIN Tomáš: Web services for embedded systems. [Bachelor Thesis] – University of Žilina. Faculty of Management Science and Informatics; Department of Technical Cybernetics. – Supervisor: Ing. Samuel Žák – Degree of professional qualifications: Bachelor – Žilina: FRI UNIZA, 2017. Number of pages: 37.

Bachelor Thesis deals with the solution of communication problem between director and cameramen in creating camera recording with the help of live switcher, which processes images from cameras in real time. In the introduction, the author describes methods and methodology used for processing video outputs from cameras and their recording. Further, the Thesis describes internal communication itself and it focuses mainly on communication with the help of Tally light signalization, where the author describes existing possible solutions of given issues and he also suggests his own solution. It consists of the use of Raspberry Pi computer and web services, which will provide communication with the help of Tally light signalization and short text messages. The author continues with describing selected solution and with suggesting hardware part for its implementation. Analytical section explains basic concepts and technologies which are used for implementation of web interface. The last section of Thesis is devoted to the linking of the various sub-components among themselves and implementing the system into operation.

**Key words:**  Flask, Framework, Raspberry Pi, Tally light system, WebSocket

[Obsah](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Obsah)

[Zoznam obrázkov, tabuliek a grafov 8](#_Toc480399709)

[Zoznam skratiek 9](#_Toc480399710)

[1 Úvod 10](#_Toc480399711)

[1.1 Tally svetelný systém 12](#_Toc480399712)

[1.2 Princíp a existujúce riešenia 13](#_Toc480399713)

[1.2.1 Základné riešenia 14](#_Toc480399714)

[1.2.2 Pokročilejšie riešenia 15](#_Toc480399715)

[1.2.3 Profesionálne riešenie 16](#_Toc480399716)

[1.3 Návrh štruktúry systému 17](#_Toc480399717)

[1.3.1 Aktuálne riešenie 17](#_Toc480399718)

[1.3.2 Navrhované riešenia 17](#_Toc480399719)

[1.3.3 Výber riešenia a nastavenie cieľov práce 18](#_Toc480399720)

[2 Technické zabezpečenie 19](#_Toc480399721)

[2.1 Raspberry Pi 19](#_Toc480399722)

[2.2 Podporné dosky 21](#_Toc480399723)

[2.2.1 Tally - GPIO 21](#_Toc480399724)

[2.2.2 Doska pre bezpečné vypnutie systému 23](#_Toc480399725)

[2.3 Ochranný obal 24](#_Toc480399726)

[3 Webové služby 25](#_Toc480399727)

[3.1 Framework - pracovná sada 25](#_Toc480399728)

[3.2 Webové protokoly 27](#_Toc480399729)

[3.3 Webový server 28](#_Toc480399730)

[3.4 Informatické pojmy 29](#_Toc480399731)

[3.4.1 HTML 29](#_Toc480399732)

[3.4.2 CSS 31](#_Toc480399733)

[3.4.3 JavaScript 31](#_Toc480399734)

[3.5 Webový aplikačný framework 33](#_Toc480399735)

[3.5.1 Django 34](#_Toc480399736)

[3.5.2 Flask 34](#_Toc480399737)

[3.5.3 Výber webového aplikačného framework-u 35](#_Toc480399738)

[3.6 Komunikácia klient-server v reálnom čase 36](#_Toc480399739)

[3.6.1 WebSocket 36](#_Toc480399740)

[3.6.2 Socket.io 37](#_Toc480399741)

[3.6.3 Flask-SocketIO 38](#_Toc480399742)

[3.7 Webová stránka 39](#_Toc480399743)

[3.7.1 Bootstrap 40](#_Toc480399744)

[4 Realizácia navrhnutého systému 42](#_Toc480399745)

[4.1 DPS 42](#_Toc480399746)

[4.2 Raspberry Pi a Flask 42](#_Toc480399747)

[4.3 Webové stránky 43](#_Toc480399748)

[4.4 Grafická zmena stavu kamery 44](#_Toc480399749)

[4.5 Textová komunikácia 45](#_Toc480399750)

[5 Záver 46](#_Toc480399751)

[Zoznam použitej literatúry 47](#_Toc480399752)

[Referencie na obrázky 50](#_Toc480399753)

[Prílohy 51](#_Toc480399754)

[Príloha A: Podporná doska – GPIO-TALLY 52](#_Toc480399755)

[Príloha B: Podporná doska pre bezpečné vypnutie systému 53](#_Toc480399756)

[Príloha C: Ochranný obal 54](#_Toc480399757)

[Príloha D: Ukážka webových stránok na mobilnom telefóne 55](#_Toc480399758)

[Príloha E: Ukážka webových stránok na prenosnom počítači – nočný dizajn 56](#_Toc480399761)

[Príloha F: Ukážka webových stránok na prenosnom počítači – denný dizajn 57](#_Toc480399762)

# [Zoznam obrázkov](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx" \l "Zoznam_obrázkov)

[Obrázok 1.1 Ilustračný obrázok – prenosná video strižňa 10](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383725)

[Obrázok 1.2 Tally svetelné indikátory: doplnkový, integrovaný 12](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383726)

[Obrázok 1.3 Označenie pinov konektora DB-15HD 13](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383727)

[Obrázok 1.4 Zariadenie DataVideo TB-5: svetelný indikátor, riadiaca jednotka 14](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383728)

[Obrázok 1.5 Zariadenie DataVideo ITC-100 14](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383729)

[Obrázok 1.6 TallyTec Pro vysielač, TallyTec Pro prijímač 15](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383730)

[Obrázok 1.7 Blackmagic Design štúdiový a kamerový konvertor 16](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383731)

[Obrázok 1.8 Aktuálne riešenie: redukcia, svetelné indikátory 17](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383732)

[Obrázok 2.1 Rozmiestnenie GPIO portov na Raspberry Pi ver.B 20](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383733)

[Obrázok 2.2 Časť schémy z redukcie Tally-GPIO 21](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383734)

[Obrázok 2.3 Počítačový model ochranného obalu 24](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383735)

[Obrázok 3.1 Ilustrácia komunikácie klient-server 28](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383736)

[Obrázok 3.2 WebSocket podpora pre internetové prehliadače 37](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383737)

[Obrázok 3.3 Responzívny dizajn – porovnanie dvoch zariadení 41](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383738)

[**Zoznam tabuliek**](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx)

[Tabuľka 1.1 Rozloženie signálov pre Tally v konektore DB-15HD 13](#_Toc480383721)

[Tabuľka 2.1 Výpočet hodnôt obmedzovacích rezistorov 22](#_Toc480383722)

[Tabuľka 3.1 Príklady framework-ov pre rôzne programovacie jazyky 25](#_Toc480383723)

[Tabuľka 3.2 Zoznam najpoužívanejších protokolov a ich krátky popis 27](#_Toc480383724)

[**Zoznam grafov**](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx)

[Graf 3.1 Štatistika využívania JavaScript knižníc na webových stránkach 32](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383719)

[Graf 3.2 Využitie Bootstrap a Foundation na webových stránkach 39](https://d.docs.live.net/77491396eb51d89e/Bakalarka/ONEDRIVE.docx#_Toc480383720)

# [Zoznam skratiek](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx" \l "Zoznam_skratiek)

|  |  |  |
| --- | --- | --- |
| Skratka | Význam | |
| API | Application Programming Interface | |
| CPU | Central Processing Unit | |
| DOM | Document Object Model | |
| DPS | Doska plošného spoja |
| GPIO | General-Purpose Input/Output | |
| GUI | Graphical User Interface | |
| GPU | Graphics Processing Unit | |
| I2C | Inter-Integrated Circuit | |
| LED | Light Emitting Diode | |
| SDI | Serial Digital Interface | |
| SPI | Serial Peripheral Interface | |
| UART | Universal Asynchronous Receiver/Transmitter | |
| USB | Universal Serial Bus | |
| Wi-Fi | Wireless Fidelity | |

# [Úvod](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Úvod)

Používanie kamery sa stalo súčasťou našich životov. Týmto médiom chceme zachytiť naše spomienky, udalosti, rôzne podujatia. V modernej dobe sa rozmohlo najmä kamerovanie pomocou smartfónov, kde sa hardvérové vybavenie nepretržite zlepšuje. Avšak, stále je tu aj profesionálnejšie zaznamenávanie určené na rôzne športy, koncerty, televízne programy a mnoho ďalších, kde sa stále využíva klasická koncepcia profesionálnych kamier a kameramanov. Kamery bývajú rozmiestnené tak, aby pokryli celú scénu podujatia a niekedy aj viac, napríklad divákov, zákulisie a podobne. Pri takomto natáčaní sa výsledný produkt môže ukladať dvomi spôsobmi, poprípade ich kombináciou.

Prvý spôsob využíva systém, kde každá kamera ukladá svoj obrazový výstup na pamäťové médium, buď interný pevný disk alebo pamäťovú kartu a výsledný záznam je po ukončení akcie zozbieraný z kamier a zostrihaný do finálnej podoby dodatočne.

Druhý spôsob je technicky náročnejší na prípravu a materiálne vybavenie. Funguje princípom, že záznam z každej kamery je prenášaný do réžie - obslužné pracovisko, kde sa nachádza zariadenie video strižňa. Toto zariadenie je určené na vytváranie konečného záznamu, ktorý je v reálnom čase zostrihávaný z čiastkových záberov z kamier.

Obrázok . *Ilustračný obrázok – prenosná video strižňa*

Rozdiely medzi týmito spôsobmi zaznamenávania sú značné. Prvý uvedený spôsob sa využíva najmä na akcie, kde záznam slúži skôr ako spomienkový artefakt, napríklad stužková slávnosť alebo svadba. Akcia sa nasníma a až po jej ukončení záznamy putujú do počítača, kde sú v špecializovanom softvéri zostrihané. Režisér má pri tomto spôsobe dostatok času na výber, ktorý záber je najlepší alebo v prípade nerozhodnosti sa k nemu môže vrátiť neskôr. Záznam vytváraný video strižňou sa využíva najmä pri živých prenosoch, kedy udalosť snímaná pomocou kamier musí byť zostrihaná okamžite. Kladený je tu veľký dôraz na rýchlosť, skúsenosť a umeleckú stránku človeka obsluhujúceho toto zariadenie. Výsledný obraz je okamžite distribuovaný divákovi, čiže tu nie je miesto na omyly. Ako už bolo spomenuté, tento spôsob je náročnejší na technické zabezpečenie, pretože obrazový výstup z každej kamery musí byť privedený do réžie, čo nie je vždy jednoduché.

Pri využívaní kombinácie oboch predchádzajúcich spôsobov zaznamenávania vznikajú isté benefity. Napríklad vo výslednom produkte sa môže nepodarený prestrih réžie dodatočne upraviť zo záznamu, ktorý bol uložený na kamerách. Využíva sa najmä ak živý záznam bude neskôr ešte zužitkovávaný. Špeciálnym prípadom sú napríklad športové udalosti, kde sa všetky výstupy z kamier ukladajú priamo v réžií a nie v kamerách. Tieto zábery z kamier sa následne môžu použiť pri opakovaných záznamoch, ktoré bývajú prehrávané opakovane v hlavnom zázname aj niekoľko krát. Obrazový výstup kamier môže byť zaznamenávaný po celú dobu alebo len v niektorých okamihoch. Tu uvedieme príklad z hokeja, kde záznam zo všetkých kamier sa ukladá len v okamih, kedy je bránka ohrozená.

V prípadoch, kedy využívame služby video strižne, musíme myslieť na komunikáciu. Režisér by mal dávať informácie kameramanom, čo sa v danú chvíľu na scéne chystá, ako sa ich záznam využíva alebo na čo sa v danom okamihu majú zamerať. Existujú dva spôsoby dorozumievania sa: pomocou internej hlasovej komunikácie (interkom) alebo pomocou svetelnej signalizácie. Interná hlasová komunikácia je vhodná na rýchle a presné korigovanie kameramanov. Problémy tu môžu vznikať pri zaznamenávaní hlučných akcií, kameraman môže tieto požiadavky alebo informácie režiséra prepočuť. Toto rieši druhý spôsob dorozumievania sa pomocou svetelnej signalizácie, nazývajúci sa Tally. Súži na zobrazenie stavu využitia obrazového výstupu kamery.

My sa v tejto práci zameriame práve na svetelnú signalizáciu Tally. Systém budeme vytvárať pre mediálny výbor Evanjelického cirkevného zboru v Hybiach. Cirkevný zbor využíva kamerovú techniku na distribuovanie živých prenosov služieb Božích z kostola pomocou YouTube stream služieb a občasne zabezpečuje nahrávanie rôznych podujatí. Využíva sa tu šesť kamier na snímanie scény a dve obrazovky pre prehrávanie záznamov a tituliek z prenosných počítačov. Pretože je to cirkevná organizácia, finančné prostriedky sú obmedzené, na čo budeme musieť myslieť pri návrhu.

## Tally svetelný systém

Tento systém je buď integrovaný priamo v kamerách alebo je k nim doplnkovo pripojený a potrebné dáta preň sú získavané z výstupu video strižne. Táto informácia slúži nie len kameramanom, ale aj účinkujúcim pred kamerou, napríklad aby snímaná osoba vedela, do ktorej kamery sa má pozerať, ak je kamerovaná z viacerých uhlov. Ako príklad uvediem televízne noviny natáčané v štúdiu, kde sa moderátor vždy pozerá do správnej kamery aj keď sa zábery priebežne menia. Tally využíva viacero farebných svetelných indikátorov na zobrazenie rôznych stavu, v ktorom sa obrazový výstup kamery nachádza.

Červená farba – LIVE signál – je určená pre situáciu, kedy obrazový výstup je v hlavnom živom výstupe. Kameramanovi dáva informáciu o tom, aby v danú chvíľu výrazne nemenil kompozíciu snímania.

Zelená farba – READY signál - signalizuje stav, kedy sa réžia pripravuje na prestrih na danú kameru, ktorej obraz je v náhľade. Kameraman by mal v danú chvíľu doladiť kompozíciu poprípade upraviť svetelné podmienky, pre vyváženosť záberu.

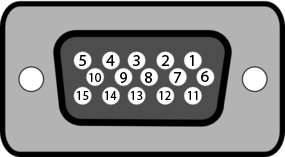
Žltá farba – ISO signál – skratka je z anglického slova „isolated“, v preklade izolovaný. Používa sa pri spôsobe dodatočného nahrávania záznamu kamery, ktorý sa ukladá špeciálne popri hlavnom zázname. Tento indikátor je menej častý, pretože technické zabezpečenie dodatočného nahrávania je náročné.

Obrázok . *Tally svetelné indikátory: doplnkový (vľavo), integrovaný (vpravo)*

## Princíp a existujúce riešenia

Signál pre Tally svetelný systém je generovaný z video strižne a privedený ku kamerám buď káblom alebo bezdrôtovo. Video strižne na to poskytujú konektory, pričom jeden obsahuje signál pre 4 kamery, zriedkavo pre 8 kamier. Odtiaľ signál putuje do rozvádzača, ktorý distribuuje signál osobitne ku každej jednej kamere. Existujú aj neštandardné strižne, ktoré nemajú priamo výstup pre Tally ale signál je distribuovaný iným rozhraním a pre prípadné využitie klasických Tally konektorov ponúkajú rôzne redukcie.

Vysvetlíme si princíp fungovania systému svetelnej indikácie Tally priamo na najčastejšie používanom konektore DB-15HD, združujúci 4 kamery bez ISO signalizácie. V konektore sa nachádza jeden pin, ktorý je spoločný pre všetky kamery a slúži ako referenčná zem. Každá kamera je zastúpená ešte ďalšími dvoma pinmy, jeden pre živý výstup (LIVE signál) a druhý pre náhľadový režim (READY signál). Zvyšné piny sú nezapojené. Po zvolení kamery na video strižni sa automaticky prepojí konektor prislúchajúci danej kamere so zemou. Vždy môže byť zvolená len jedna kamera pre živý výstup a jedna pre náhľad, pričom to môže byť aj tá istá.



Obrázok . *Označenie pinov konektora DB-15HD*

Tabuľka . *Rozloženie signálov pre Tally v konektore DB-15HD*

|  |  |
| --- | --- |
|  |  |
|  |  | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
| KAMERA | 1 | L | - | R | - | - | - | - | - | - | - | - | - | - | G | - |
| 2 | - | - | - | - | - | L | - | R | - | - | - | - | - | G | - |
| 3 | - | - | - | - | - | - | - | - | - | - | L | - | R | G | - |
| 4 | - | - | - | - | L | - | - | - | - | - | - | - | - | G | R |

*L- živý výstup, R- náhľad, G –zem*

### Základné riešenia

Pri základných riešeniach sa využíva prepojovanie pinov v konektore video strižne, čo uzatvorí elektrický obvod a to spôsobí rozsvietenie svetelného indikátora.

#### DataVideo TB-5

Firma DataVideo je výrobca video strižní a aj príslušenstva k nim. Nami vybraté Tally rozšírenie je najnákladnejšie. Poskytuje malé svetelné indikátory pripojiteľné ku kamerám prepojené s riadiacou jednotkou pripojenou ku video strižni. Kamery a riadiaca jednotka sú prepojené pomocou audio káblov s koncovkou 3.5 mm stereo jack. Riadiaca jednotka poskytuje riadiace informácie pre 4 kamery, nepodporuje prenos ISO signalizácie a cena sa pohybuje okolo 200 eur. [1]



Obrázok . *Zariadenie DataVideo TB-5:   
svetelný indikátor pre kameramana (vľavo), riadiaca jednotka (vpravo)*

#### DataVideo ITC-100

Toto zariadenie je ďalšia generácia predchádzajúceho produktu tejto značky, avšak je vylepšené o internú hlasovú komunikáciu. Hlavná konzola pre zabezpečenie komunikácie a distribúciu Tally signálu sa umiestňuje v réžii. Kameramani majú pri sebe malú riadiacu jednotku, ktorá v sebe obsahuje výstup pre Tally indikátory, vstup pre slúchadlá s mikrofónom a časť na reguláciu intenzity signálu v slúchadlách. Prenosovým médiom medzi konzolou a riadiacou jednotkou je 5 pinový XLR kábel, ktorého dĺžka môže byť až 200 metrov. Konzola poskytuje pripojenie 8 kamier bez podpory ISO signalizácie. Cena tohto zariadenia sa pohybuje okolo 1100 eur. [2]

Obrázok . *Zariadenie DataVideo ITC-100*

### Pokročilejšie riešenia

Pri pokročilejších riešeniach sa výstup na konektore Tally zosníma, vyhodnotí a odosiela ku kamerám. Svetelné indikátory tak už nefungujú na mechanizme uzavretia elektrického obvodu pomocou video strižne, ale na sofistikovanejšom princípe.

#### TallyTec Pro ****Transmitters****, TallyTec Pro Receiver

Táto firma sa priamo špecializuje na problematiku svetelnej signalizácie Tally. TallyTec Pro **Transmitters** je vysielač, ktorý sníma stav Tally konektora video strižne a následná distribúcia signálu je buď bezdrôtová alebo pomocou Ethernet káblu. Tento vysielač je v ponuke pre 4 , 8 alebo 16 kamier. TallyTec Pro Receiver je prijímač určený na umiestnenie priamo na kamere, ktorý v sebe obsahuje svetelné indikátory, bez podpory ISO signalizácie. Firma ponúka dve varianty prijímača, pričom obe obsahujú možnosť káblového prepojenia a jedna zahrňuje aj bezdrôtový prenos. Dosah bezdrôtového prenosu je udávaný približne 2 km. Cena bezdrôtového setu pre 8 kamier je približne 3500 eur. [3]





Obrázok . *TallyTec Pro vysielač (vľavo), TallyTec Pro prijímač (vpavo)*

### Profesionálne riešenie

Pri profesionálnom kamerovaní sa využíva prenos viacerých signálov jedným káblom. Medzi tieto signály patria obrazový výstup kamery, Tally signál, interná hlasová komunikácia, hlavný obrazový výstup z video strižne ku kameramanovi a rôzne audio signály. O zlučovanie signálov sa stará štúdiový konvertor, ktorý je umiestnený v réžií najčastejšie priamo v prepravnom obale s video strižňou. Dáta sa prenášajú buď po optickej linke alebo pomocou SDI káblov. Pri využití SDI káblov je množstvo prenášaných signálov obmedzené viac ako pri optickej linke. Na druhom konci kabeláže sa pripája buď priamo kamera, ktorá podporuje takýto zlúčený signál a sama si ho vyselektuje na jednotlivé zložky, alebo kamerový konvertor, ktorý to urobí namiesto kamery. Kvôli zložitosti technológie je ale toto riešenie veľmi drahé. Ceny začínajú na úrovni 2500 eur za štúdiové konvertory a ceny konvertorov pri kamerách na 500 eur za kus.

#### Blackmagic Design ATEM Talkback Converter 4K a ATEM Camera Coverter

Toto zariadenie je príkladom varianty, kde kamera nevie vyseparovať signály sama. Ako prenosové médium medzi konvertormi sa využíva optická linka, ktorá môže dosahovať dĺžku 40 km. Štúdiový konvertor je určený pre 8 kamier ale pri potrebe využitia viacerých vstupov je tu možnosť prepojenia viacerých štúdiových konvertorov navzájom. Svetelná signalizácia Tally sa nachádza priamo na kamerovom konvertore, ktorý sa tak musí pripevniť na kameru. Ten obsahuje aj reguláciu hlasitosti a ovládanie mikrofónu. [4]

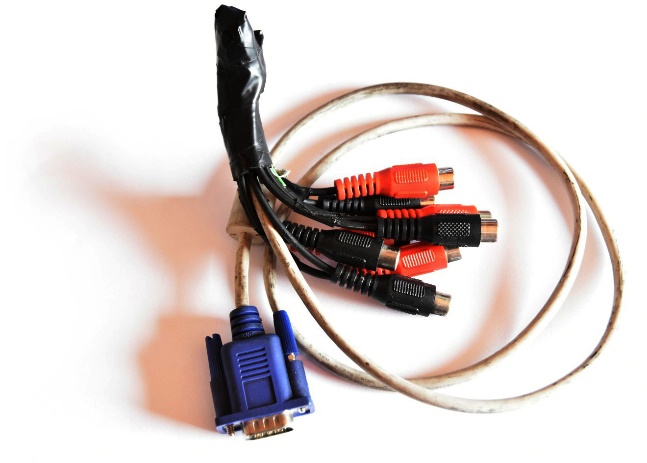


Obrázok . *Profesionálne riešenie Tally od firmy Blackmagic Design  
štúdiový konvertor(hore), kamerový konvertor(dole)*

## Návrh štruktúry systému

Riešení Tally svetelnej komunikácie je na trhu dosť, ale aj elementárne varianty sú finančne náročné. V tejto časti si popíšeme aktuálne riešenie používané v cirkevnom zbore ECAV Hybe, rozvedieme jeho nedostatky a navrhneme nové riešenia, ktoré v tejto práci aj zrealizujeme.

### Aktuálne riešenie

Daná problematika bola dočasne riešená pomocou provizórne vyrobených redukcií pre konektor DB-15HD do separátnych konektorov pre signál ku každej kamere. Pri kamerách boli upravené bezpečnostné svetielka na bicykel, ktoré sa rozsvietili pri uzavretí elektrického obvodu spôsobené zopnutí kontaktov v konektore video strižne. Prenos signálu bol realizovaný pomocou stereo audio kabeláže. Toto riešenie bolo amatérska kópia zariadenia DataVideo TB-5, ktoré sme opisovali pri základných riešeniach.

Obrázok . *Aktuálne riešenie: redukcia(vľavo), svetelné indikátory (vpravo)*

### Navrhované riešenia

Možnosti riešenia problému môžeme rozdeliť na dve hlavné skupiny. Prvá skupina obsahuje riešenia za pomoci prenosu signálu káblovým prepojením kým druhá zahŕňa bezdrôtový prenos.

#### Káblový prenos

Pri prenose pomocou káblov je vhodné využiť koncepciu základných riešení od výrobcu, tak ako je realizovaná už aj v aktuálnom riešení. Je to výhodné najmä kvôli kabeláži, ktorú daný cirkevný zbor vlastní. Pre zlepšenie vlastností aktuálneho riešenia, navrhujeme redizajnovať redukciu z video strižne, ktorá by sa nachádzala v ochrannom obale. Namiesto nekvalitného spájania káblov, t.j. stáčanie dokopy a spájkovanie,  
by separovanie pinov z konektoru video strižne bolo riešené na malom plošnom spoji ukrytom v obale. Ďalším zlepšením by bolo vytvorenie nových svetelných prvkov ku kamerám. Tie by neboli zostavené zo svetiel určených pre bicykle, ale boli by vytvorené špeciálne na tento účel. Zlepšenie oproti terajšiemu riešeniu by bolo najmä v kvalite výroby, spoľahlivosti ale aj v estetickom smere. Nepraktickou stránkou tohto riešenia je práve spomínaná kabeláž, o ktorú sa treba starať, pretože spoje okolo konektorov sa častým využívaním opotrebovávajú a neskôr môžu spôsobovať problémy. Tak isto je kabeláž nepraktická pri prevoze vzhľadom na jej objemnosť.

#### Bezdrôtový prenos

Prenos signálu bezdrôtovo je technicky náročnejší, pretože stav na Tally konektore musí byť neustále snímaný. Následne, zosnímané údaje musia byť distribuované prijímaciemu zariadeniu, kde sa dáta budú transformovať na svetelnú signalizáciu.

V tomto prípade je riešením využitie malého počítača s bezdrôtový pripojením pomocou Wi-Fi technológie a inteligentného mobilného telefónu ako prijímača signálu. Navrhnutý variant spočíva v tom, že zosnímaný stav na Tally konektore bude počítať distribuovať na webovú stránku, ktorú sám bude poskytovať na svojom web servery. Obrazovky mobilných telefónov v tomto prípade budú nahrádzať štandardné svetelné signalizačné prvky pre kameru. Po pripojení sa na webový server a následnom zvolení požadovanej kamery sa bude graficky na displeji zobrazovať jej stav.

### Výber riešenia a nastavenie cieľov práce

Pri výbere riešenia sme sa s cirkevný zborom ako zo zadávateľom zhodli,  
že navrhnutý bezdrôtový variant bude výhodná voľba.

Cieľom práce tak bude funkčný systém pre 8 zariadení, šesť kamier a dve obrazovky, ktorý bude pomocou webového rozhrania v lokálnej sieti poskytovať grafické informácie o stave kamier a pokyny z réžie pre kameramanov. Tieto pokyny budú sprostredkované pomocou krátkych textových správ. Systém bude bez podpory ISO signalizácie, pretože zadávateľ nedisponuje takouto technológiou. Webové rozhranie bude využívané mobilnými telefónmi a tak je nutná jeho optimalizácia pre použitie na nich.

# Technické zabezpečenie

Táto kapitola sa bude venovať hardvérovej časti našej práce. Budeme sa zaoberať popisom vybraného kompaktného počítača a návrhom podporných dosiek, ktoré budú potrebné pri realizácií našej práce. Posledná časť sa bude týkať ochranného obalu pre bezpečné uloženie všetkých komponentov.

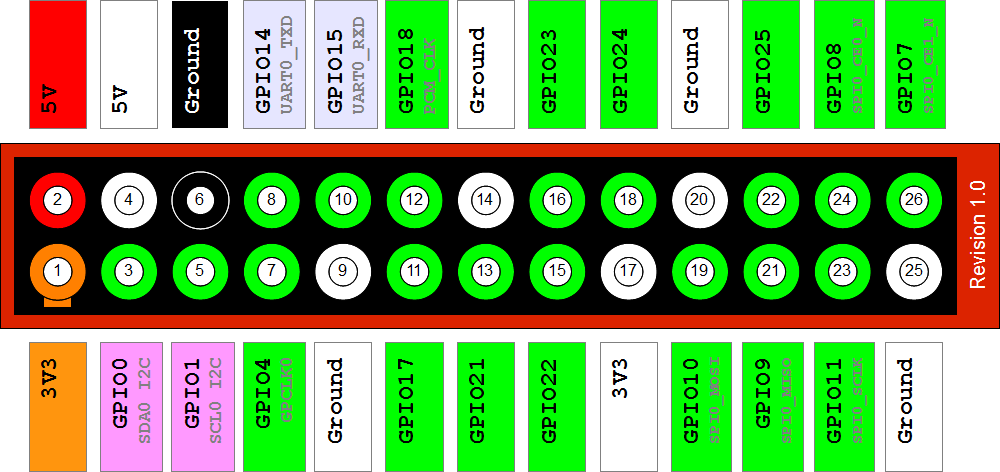
## Raspberry Pi

Pre náš projekt sme sa rozhodli využiť Raspberry Pi. Je to jednodoskový počítač, ktorý svojou veľkosťou pripomína platobnú kartu. Bol vyvinutý s cieľom podporiť informatickú výučbu na školách. Pre jeho nízku cenu a dostatočný výkon sa rozšíril aj mimo škôl a v dnešnej dobe je často využívaný na rôzne projekty. Uplatňuje sa skoro v každom odvetví informatiky, či už sa jedná o jednoduché aplikácie ako sieťový tlačový server, mediálny prehrávač alebo komplexné riešenia napríklad riadiaca jednotka dronu. Jeho široká škála využitia dosiahla vysokej úrovne, čo má za následok, že stal zaujímavým prostriedkom aj pre obyčajných ľudí zo základnými znalosťami programovania. Najpoužívanejším operačným systémov je Raspbian, čo je optimalizovaná verzia známeho linuxového systému Debian. Spolu s USB portmi a obrazovými výstupmi sa môže stať z Raspberry Pi plnohodnotný počítač s grafickým rozhraním. Operačný systém ako aj užívateľské dáta sú uložené na pamäťovej karte. Na rozšírenie funkcionality vzniklo veľa prídavných periférií, napríklad GPS modul. V ponuke nájdeme aj skupiny prvkov a senzorov zabezpečujúcich komplexnú činnosť, napríklad stavebnica meteorologickej stanice. [5]

Tento počítač sa na trhu nachádza už v niekoľkých verziách. Tie sú čoraz viac a viac výkonnejšie a pridávajú sa do nich nové integrované periférie, ako Wi-Fi alebo Bluetooth. Pre nás je postačujúci starší model, presnejšie verzia B prvej generácie.

Parametre Raspberry Pi, ver.B:

* CPU –ARM1176JZF-S, taktovaný na 700MHz
* GPU - Broadcom VideoCore IV, takt 250 MHz
* 512 MB RAM – zdieľaná s grafickým čipom
* Konektory: 2xUSB port, kompozitné video, HDMI, audio výstup- 3.5 mm jack, Ethernet - 100Mb/s, DSI Display konektor, CSI Camera konektor
* Nízko-úrovňové periférie: pole 26 vstupno-výstupných portov
* Napájanie pomocou USB typ B
* Spotreba 3.5W

Zaujímať nás budú najmä vstupno-výstupné porty čiže nízko-úrovňové periférie, ktoré budú snímať stav z Tally konektora. Tieto porty sa nazývajú GPIO – General-purpose input/output. Určitým portom môžeme pridať aj dodatočnú funkciu, a to SPI, UART alebo I2C. Operačný systém Rasbian už v sebe obsahuje knižnicu pre jednoduché ovládanie a nastavenie GPIO. [6]

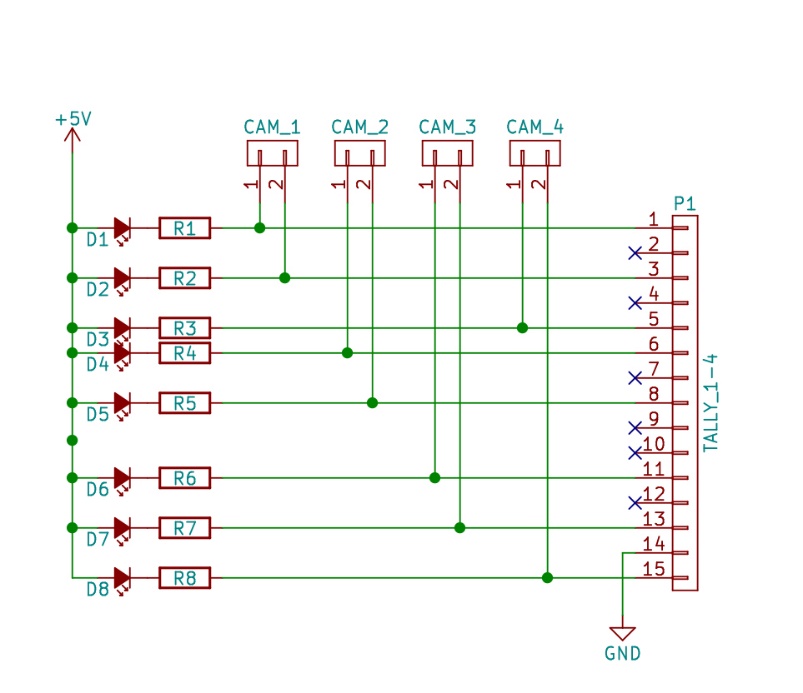
Obrázok . *Rozmiestnenie GPIO portov na Raspberry Pi ver.B*

Verzia Raspberry Pi, ktorú využívame, nemá v sebe integrovanú perifériu Wi-Fi, čiže toto rozhranie musíme doplniť pomocou adaptéru do USB portu. Prihlásenie sa do lokálnej siete je možné realizovať buď pomocou grafického rozhrania alebo pripojením Raspberri Pi pomocou ethernet konektoru a následne vzdialeným prístupom pomocou príkazového riadku.

## Podporné dosky

V tejto časti sa venujeme návrhu a výrobe dvoch dosiek plošných spojov – skratka DPS. Opíšeme si ich význam, popíšeme funkciu a navrhneme vhodné súčiastky na osadenie.

### **Tally - GPIO**

Tento plošný spoj tvorí redukciu medzi konektorom Tally a vstupno-výstupnými pinmi GPIO na Raspberry Pi. Dosku rozšírime o doplnkovú svetelnú signalizáciu, ktorá nám bude indikovať stav kamier priamo na DPS. Keďže video strižňa zadávateľa využíva konektory DB-15HD, ktoré združujú signály pre štyri kamery, doska musí obsahovať dva takéto konektory. Napájanie pre dosku bude dodávané z Raspberry Pi.

Obrázok . *Časť schémy z redukcie Tally-GPIO*

Na ukážke máme časť schémy pre jeden DB-15HD konektor, na ktorej si popíšme funkčnosť dosky pre oba prípady, ktoré môžu nastať. Rozloženie pinov v tomto konektore bolo opísané v predchádzajúcej kapitole. Analyzovať budeme možné situácie pre jednu vetvu elektrického obvodu ktorá obsahuje:

* LED diódu: D1
* Obmedzovací rezistor: R1
* Konektor GPIO na snímanie stavu kamery: CAM\_1- pin 1
* Konektor DB-15HD: P1- pin 1 a pin 14

1. Prvý stav ktorý môže nastať je, keď kamera nie je v živom výstupe, čiže pin 1 a pin 14 v konektore P1 nie sú prepojené. Tým sa elektrický obvod sa neuzavrie a napätie na CAM\_1 - pin 1 bude na vysokej úrovni.
2. Druhý stav nastane, ak sa kamera nachádza v živom výstupe a po prepojení pinov 1 a 14 v konektore P1 sa elektrický obvod uzavrie. To bude mať za následok prenesenie potenciálu zeme na CAM\_1 - pin 1, ktorý bude na nízkej úrovni. Uzavretie obvodu rozsvieti aj indikačnú LED diódu.

Pre dva rôzne signály na jednu kameru využijeme dve farby LED diód. Hodnotu obmedzovacích rezistorov dopočítame pomocou Ohmového zákonu:

V našich výpočtoch budeme používať tieto napätia:

* Napätie na dióde – UD, nachádza sa v katalógovom liste spolu s prúdom
* Vstupné napätie – UV = 5V
* Napätie na rezistore – UR

Rezistory na použitie budeme vyberať z rady E12, čiže vypočítané hodnoty budeme prispôsobovať týmto hodnotám. Vnútorný odpor vstupov GPIO je pri výpočtoch zanedbateľný - 1,2 mΩ. Celá schéma a doska plošného je umiestnená v prílohe.

Tabuľka . *Výpočet hodnôt obmedzovacích rezistorov*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Farba LED | I [mA] | UD [V] | UR [V] | R [Ω] | RE12 [Ω] |
| červená | 20 | 1.8 | 3.2 | 160 | 180 |
| zelená | 20 | 2.2 | 2.8 | 140 | 150 |

### Doska pre bezpečné vypnutie systému

Pri využívaní Raspberry Pi na úlohy, kedy počítač používame bez monitora a klávesnice musíme myslieť na problém s bezpečným vypnutím systému. Je to preto, lebo doska počítača neobsahuje integrované tlačidlo pre zapnutie alebo vypnutie ako pri bežnom osobnom počítači. Inicializácia operačného systému začína hneď po pripojení napájania. Problém s bezpečným vypnutím sa dá riešiť dvomi spôsobmi. Prvý pozostáva z vypnutia systému vzdialeným prístupom, kým druhý zahŕňa dodatočné pripojenie externého tlačidla. Pre jednoduchosť riešenia sme si vybrali pripojenie tlačidla na konektor GPIO a vytvorenie krátkeho programu bežiaceho na pozadí hneď po spustení systému. Tento program bude v pravidelných intervaloch kontrolovať či tlačidlo bolo stlačené. Ako bezpečnostný prvok proti nechcenému stlačeniu, ošetríme funkciu vypnutia o nutnosť podržania tohto tlačidla na dobu 3 sekúnd. Po tomto časovom intervale dá program pokyn operačnému systému na vypnutie a až následne je možné bezpečne odpojiť napájanie počítača. Schéma dosky plošného spoja sa nachádza v prílohe.

Kód programu je napísaný v programovacom jazyku Python:

import RPi.GPIO as GPIO

import time

import os

ACTIVE\_LEVEL\_VALUE = 1

BUTTON\_PIN = 7

CONFIRM\_TIME = 3

SLEEP = 1

GPIO.setmode(GPIO.BCM) #nastavenie označenia GPIO portov

GPIO.setwarnings(False) #zakázanie zobrazovania výstrah

GPIO.setup (BUTTON\_PIN, GPIO.IN) #nastavenie portu 7 za vstupný

while True:

if(GPIO.input(BUTTON\_PIN) == ACTIVE\_LEVEL\_VALUE): #\*

time.sleep(CONFIRM\_TIME) #kontrola nechceného stlačenia

if(GPIO.input(BUTTON\_PIN) == ACTIVE\_LEVEL\_VALUE): #\*

os.system('clear') #vyčistenie konzoly

print("OFF BUTTON- shut down Raspberry") #\*\*

os.system('shutdown -h now') #príkaz na vypnutie systému

time.sleep(SLEEP) #uspanie programu na požadovaný čas

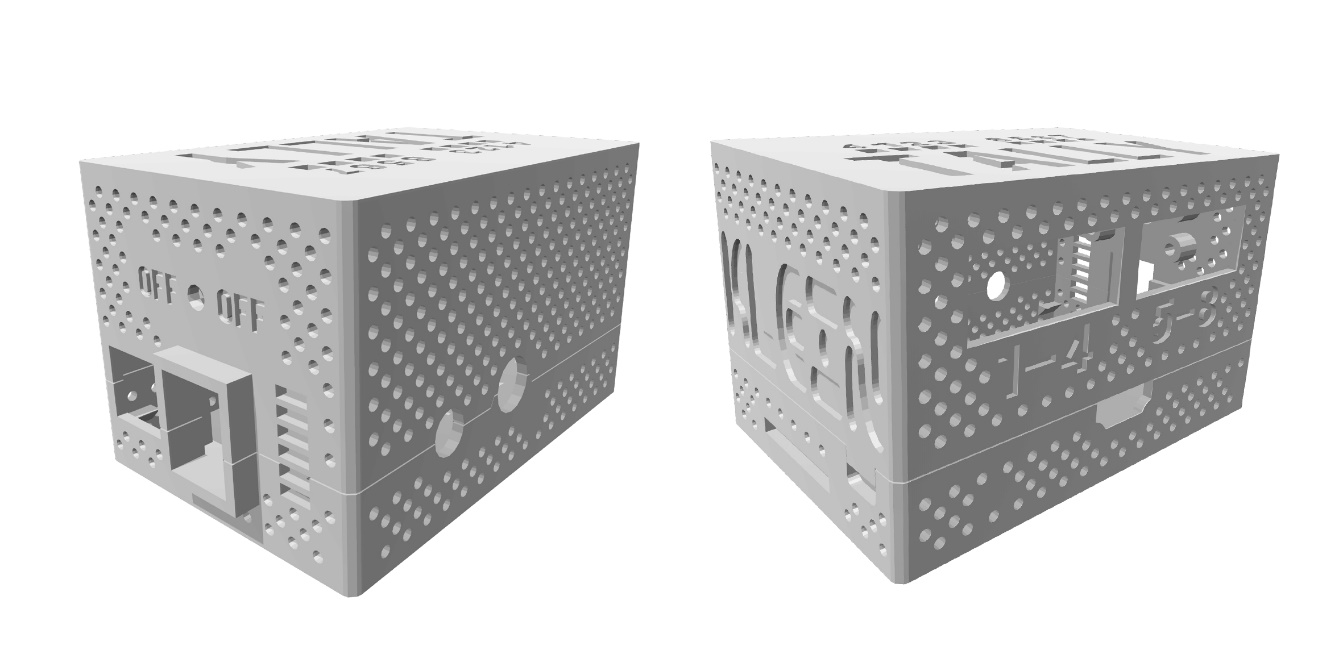
*\* kontrola stlačenia tlačidla*

*\*\* výpis na konzolu*

## Ochranný obal

Pretože náš projekt nie je len teoretický ale bude aj reálne zasadený do prevádzky, potrebuje ochranný obal, ktorý bude poskytovať ochranu pre komponenty:

* Raspberry Pi
* Redukcia Tally-GPIO
* Doska pre bezpečné vypnutie systému

Pre vytvorenie tejto krabice sme používali 3D modelovací program Thinkercad, bežiaci v internetovom prehliadači. Tento program je určený najmä na vytváranie modelov pre 3D tlačiarne, ale jeho výstupné formáty, najmä SLT, umožňujú tieto modely vytvárať aj pomocou CNC strojov. Má jednoduché ovládanie a je voľne dostupný, čo z neho robí ideálny nástroj pre občasné vytváranie malých trojrozmerných modelov.

Obrázok . *Počítačový model ochranného obalu*

Model pre náš projekt bol tlačený na 3D tlačiarni. Kvalita tlačiarne nám dovolila vytvoriť použiteľný obal, ale nutnosťou bolo dodatočné ručné opracovanie. Podrobnejšie obrázky taktiež nájdete v prílohe.

# Webové služby

Táto kapitola sa bude venovať webovým službám, ktoré majú za úlohu graficky zobrazovať stav systému na webovej stránke a poskytovať textovú komunikáciu pomocou krátkych správ. Pri vývoji si budeme pomáhať framework-ami.

## Framework - pracovná sada

Framework - anglické slovo, ktoré sa využíva bez prekladu v slovenskej literatúre a vo voľnom preklade má význam ako pracovná sada, to znamená sada prvkov, ktoré sú využívané pri práci

Je to platforma určená na urýchlenie a uľahčenie vývoja aplikácie. Môže obsahovať rôzne knižnice, podporné programy, API rozhranie, návrhové vzory ale aj doporučené postupy pre korektný vývoj aplikácie. Vhodne zvolený framework uľahčuje riešenia určitých úloh alebo môže tvoriť kostru celej aplikácie na ktorej vývojár bude stavať. Jeho cieľom je prevziať typické problémy danej oblasti a urýchliť vývoj tak, aby sa vývojár mohol zamerať najmä na svoje zadanie a základné operácie zaobstará framework. Existuje množstvo framework-ov určených na rôzne účely pre mnohé programovanie jazyky.

Tabuľka . *Príklady framework-ov pre rôzne programovacie jazyky*

|  |  |  |
| --- | --- | --- |
| Názov pracovnej sady | Programovací jazyk | Krátky popis |
| Symphony | PHP | Vývoj webových aplikácií |
| [Xamarin](http://xamarin.com/) | C# | Vývoj mobilných aplikácií |
| BioJava | Java | Spracovanie biologických dát |
| Socket.io | JavaScript | Komunikácia klient-server v reálnom čase |
| Flask | Python | Vývoj webových serverov |
| Play | Java | Vývoj webových aplikácií |

Pre lepšie pochopenie prirovnajme programovanie ku staviteľstvu a framework si pomenujme ako skúseného murára. Ak potrebujeme postaviť dom, vieme, že budeme potrebovať steny, okná, dvere, strechu a mnoho ďalšieho. V prípade, že budem potrebovať postaviť stenu využijeme murára, ktorému zadáme naše pokyny. Murár nám okamžite túto stenu postaví bez toho, aby sme sami vedeli niečo o ukladaní tehiel a malte. My si už len nakoniec stenu namaľujeme, aby bola podľa našich potrieb. Napriek tomu každý murár má svoje pravidlá a obmedzenia, a preto ho musíme vyberať dôkladne. Môže sa stať, že narazíme na murára, ktorý vie všetko o stenách ale nevie nič o oknách v nich.

Výhodou framework-ov je najmä ušetrenie času a energie pri vývoji softvéru. Samotný kód je efektívnejší a prehľadnejší a to preto, že sa využívajú správne návrhové vzory, efektívne algoritmy a rôzne štandardizované konvencie. Ďalšou výhodou je aj technická podpora, ktorá sa stará o pridávanie vždy aktuálnych technológií a taktiež sa zaoberá riešením bezpečnostných chýb. Najpoužívanejšie framework-y sa poskytujú bezplatne, čo automaticky vplýva na veľkosť ich užívateľskej základne.

Zápornou stránkou je to, že programátor sa učí o framework-och a nie samotnom programovacom jazyku. Častým sporným bodom je ušetrený čas, ktorý programátor stráca pri nutnosti študovania manuálov. Ďalším faktorom je nemožnosť modifikácie framework-u programátorom, ktorý musí rešpektovať jeho hranice a obmedzenia. Tu sa kladie dôraz pri výbere, aby sme si vždy podrobne naštudovali rôzne alternatívy a vybrali tú, ktorá vyhovie všetkým našim potrebám. Kódy framework-ov sú publikované verejne, čo taktiež môže viesť ku spornému bodu bezpečnosti. Tu sa otvára priestor pre útočníka, ktorý môže študovať kód za účelom nájdenia bezpečnostnej chyby.

Každý framework je navrhnutý na riešenie určitej skupiny úloh a problémov.   
V prípade, kedy nemôžeme nájsť vhodný framework, zriedkavé ale napriek tomu možné, je to malým signálom pre vývojára k písaniu celého kódu samostatne. [7, 8]

## Webové protokoly

**Protokol** je súbor istých pravidiel, štandardov, ktoré musí zariadenie alebo program spĺňať, aby bolo schopné komunikovať s iným programom alebo zariadením. Protokoly sa neustále vyvíjajú a dokumentácia k nim je verejne dostupná. Ich znalosť je nutná pre programátorov, ktorý píšu programy pre sieťovú komunikáciu. Bežnému užívateľovi stačí poznať názvy a význam tých najdôležitejších, s ktorými sa môže stretnúť pri nastavovaní programov.

Tabuľka . *Zoznam najpoužívanejších protokolov a ich krátky popis*

|  |  |  |
| --- | --- | --- |
| Skratka | Názov | Význam |
| IP | Internet Protocol | Zabezpečuje správne doručovanie dát jednotlivým počítačom v sieti,  zavádza pojem IP adresa. |
| TCP | Transmission Control Protocol | Realizuje spojenie medzi počítačmi v sieti, používa sa spoločné s IP – TCP/IP. |
| DNS | Domain Name System | Zaisťuje predovšetkým preklad znakových doménových adries na IP adresy.  [*www.fri.uniza.sk*](http://www.fri.uniza.sk) = 158.193.138.57 |
| DHCP | Dynamic Host Configuration Protocol | Zaisťuje automatické pridelenie IP adries,  adries DNS serverov a ďalšie nastavenia  zo sieťového serveru. |
| HTTP | HyperText Tranfer Protocol | Zaisťuje prenos webových stránok medzi serverom a internetovým prehliadačom. |
| FTP | File Transfer Protocol | Zaisťuje prenos súborov medzi počítačmi. |
| Telnet | Telnet | Zaisťuje diaľkovú komunikáciu  medzi počítačmi. |
| SSH | Secure Shell | Zabezpečuje vzdialený prístup  k príkazovému riadku počítača |
| SMTP | Simple Mail Tranfer Protocol | Zaisťuje doručenie elektronickej pošty  do schránky adresáta. |
| POP3 | Post Office Protocol | Zaisťuje výver elektronickej pošty uloženej schránke zo vzdialeného počítača. |
| IMAP | Internet Mail Access Protocol | Zaisťuje ovládanie schránky elektronickej pošty zo vzdialeného počítača. |

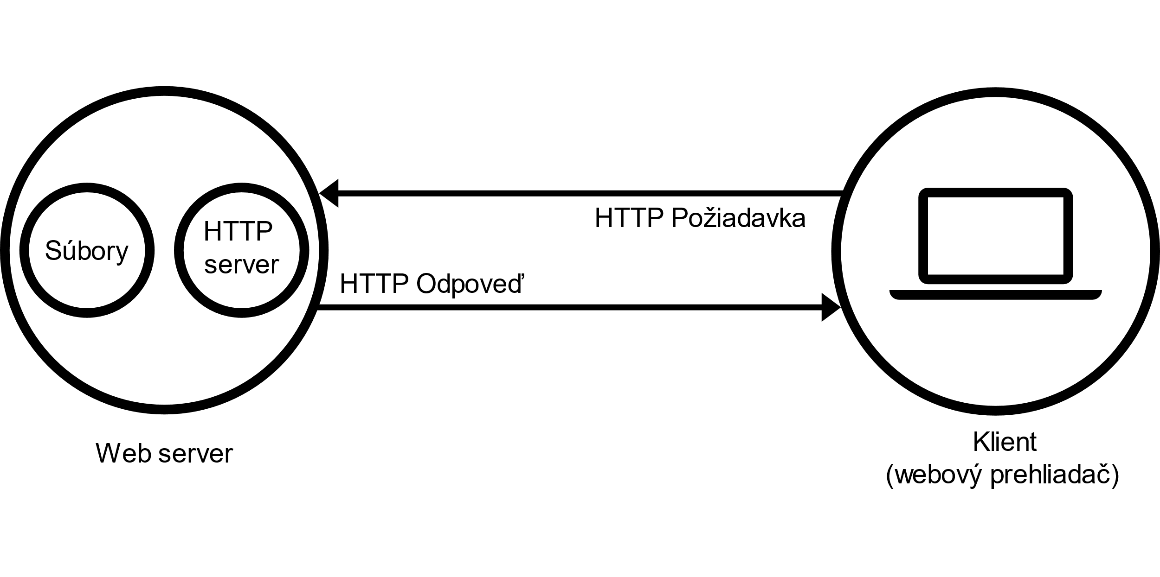
*Zdroj:*[*http://www.gvp.cz/local/new/ucebnice/Vyptech/internet/protokoly.htm*](http://www.gvp.cz/local/new/ucebnice/Vyptech/internet/protokoly.htm)*, marec 2017*

## Webový server

Webový server je program, ktorý ponúka svoje služby iným programom, nazývaným klienti. Pretože klient aj server sú počítačové programy, nemôžu medzi sebou komunikovať ľudskou rečou, ale nejakým iným, presne definovaným spôsobom. Na prenos webových stránok sa používa protokol HTTP alebo HTTPS.

Webový server má ale aj druhý význam. Je to počítač, na ktorom beží vyššie popísaný program. V tomto počítači sú uložené na pevnom disku jednotlivé webové stránky vo forme súborov. Tieto súbory môžu obsahovať rôzne HTML dokumenty, obrázky, CSS súbory alebo rôzne databázy. [9]

**HTTP**- Hypertext Transfer Protocol je internetový protokol určený na výmenu hypertextových dokumentov vo forme HTML medzi serverom a klientom. Spoločne s elektronickou poštou je to najpoužívanejší protokol. HTTP je aplikačný protokol, ktorý beží nad TCP/IP sadou.

****Protokol funguje spôsobom požiadavka - odpoveď. Užívateľ, obvykle pomocou internetového prehliadača, pošle webovému serveru požiadavku vo forme textu, obsahujúcu označenie požadovaného dokumentu, informácie o prehliadači a jeho schopnostiach. Server následne vygeneruje odpoveď pomocou textu, ktorý popisuje výsledok požiadavky, či sa požadovaný dokument našiel, akého je typu a podobne. Ak sa dokument našiel, nasledujú po tomto texte samotné dáta z dokumentu. Tento text, ktorý slúži na nadviazanie spojenia sa nazýva hlavička.

Obrázok . *Ilustrácia komunikácie klient-server*

Ak používateľ bude mať po chvíli ďalšiu požiadavku na ten istý webový server, napríklad kliknutie na hypertextový odkaz, bude sa jednať o ďalšiu, nezávislú požiadavku a odpoveď. Z hľadiska serveru nie je možné rozoznať, či mala táto druhá požiadavka súvis s predchádzajúcou. Kvôli spomínanej vlastnosti sa tento protokol nazýva bezstavový, pretože si nevie uchovať stav komunikácie. Táto vlastnosť je nepríjemná pri implementácii zložitejších webových stránok, akým je internetový obchod, kde potrebujeme uchovávať informácie ohľadom zákazníka, ako napríklad nákupný košík. Preto bol tento protokol rozšírený o HTTP cookies, ktoré umožňujú serveru uchovávať informácie o stave spojenia v počítači užívateľa. [10, 11, 12]

**Cookies** - označuje malé množstvá dát, ktoré internetový prehliadač pošle webovému serveru spolu s požiadavkou. Tieto dáta uchovávajú informácie ohľadom užívateľov, navštevovaných stránkach a podobné údaje zozbierané internetovým prehliadačom. Ich využívanie sa v poslednej dobe rozmohlo najmä na reklamné účely.

HTTPs – *HTTP secured - zabezpečený*, je zabezpečená verzia protokolu HTTP. Namiesto použitia jednoduchej textovej komunikácie, protokol HTTPs šifruje dáta použitím SSL alebo TLS protokolu a tým zabezpečuje ochranu pred odpočúvaním komunikácie. Najslabším miestom tohto protokolu je závislosť bezpečnosti na digitálne podpísaných certifikátoch. V praxi sa namiesto vysoko bezpečných certifikátov využívajú nepodpísané certifikáty, čím sa stráca bezpečnosť, ktorú HTTPs ponúka. V súčasnosti prevláda trend prechodu z HTTP práve na HTTPs. [13]

## Informatické pojmy

V tejto časti popíšeme pojmy, s ktorými sa pri vytváraní webových stránok stretneme. Jedná sa o základné pojmy, ktoré by mali byť známe aj obyčajnému programátorovi nezameranému na vývoj webového obsahu.

### **HTML**

HyperText Markup Language, v preklade hypertextový značkovací jazyk, sa používa na vytváranie webových stránok. Označenie "hypertextový" nesie pre možnosť využívania hypertextových odkazov na prepájanie stránok medzi sebou, prípadne odkazy na iné webové stránky. Značkovací jazyk je prostriedok na obohatenie textu o dodatočné informácie, napríklad štruktúra, význam alebo spôsob zobrazenia textu.   
Tieto dodatočné informácie sa vkladajú priamo do textu v podobe značiek (*angl. tags*), príkazov alebo direktív. Takéto značky sa využívajú oddávna, pretože už za čias kníhtlače autor dával tlačiarňam vedieť, kde použiť iné písmo alebo vynechať miesto pre obrázok. Výsledný zdrojový text je stále obyčajný textový súbor, a tak je možné otvoriť ho v akomkoľvek textovom editore.

V značkách jazyka HTML existujú štyri druhy prvkov:

*Štruktúrové prvky* – označujú zmysel textu

<h2> Počítač </h2>

slovo Počítač je nadpisom druhej úrovne

*Prezentačné prvky* – popisujú štýl textu bez ohľadu na jeho zmysel

<b>tučné</b>

slovo tučné bude zobrazené **tučným** písmom

*Hypertextové odkazy* – vytvárajú odkazy na iné dokumenty alebo webové stránky

<a href=“http://www.priklad.sk/“>Príklad</a>

pridá ku slovu Príklad hypertextové prepojenie na stránku [www.priklad.sk](http://www.priklad.sk)

*Ovládacie prvky* – vytvárajú rôzne tlačidlá, zaškrtávacie políčka, zoznamy

<button type="button" onclick="alert(Ahoj, svet!')">Stlač ma!</button>

príklad vytvorenia tlačidla s textom „Stlač ma!“,   
ktoré po stlačení vygeneruje kontextové okno s textom „Ahoj, svet!“

HTML jazyk je aplikácia už skôr vyvinutého rozsiahleho univerzálneho značkovacieho jazyka SGML  (Standard Generalized Markup Language). Vývoj HTML jazyka bol ovplyvnený vývojom webových prehliadačov, ktoré spätne ovplyvňovali definíciu jazyka. Existujú rôzne verzie tohto jazyka, pričom najstaršie sa už nevyužívajú a aktuálna verzia v čase písania práce je HTML5. [14, 15]

### CSS

Cascading Style Sheet, v preklade kaskádové štýly, sú štýly určené najmä na oddelenie vzhľadu dokumentu (CSS) od jeho štruktúry a obsahu (HTML). Získa sa tak prehľadný a jednoduchý kód. CSS je možné presunúť do externého súboru, a tak sa dá jedným súborom zmeniť celý štýl webovej stránky. Jednoduchšia je aj zmena iba jedného elementu, pretože ten stačí zmeniť iba v jednom CSS dokumente a zmena sa prejaví vo všetkých HTML stránkach, kde sa daný súbor používa. Vlastnosti jednotlivých elementov môžeme dynamicky meniť pomocou JavaScript. Hlavnou nevýhodou CSS je častá nedostatočná podpora prehliadačov, ktoré obsahujú v implementácií CSS chyby. To má za následok zložitejšie vytváranie kódu, tak, aby sa požadovaný štýl zobrazil na všetkých prehliadačoch rovnako. [16]

### JavaScript

Je to multiplatformový skriptovací jazyk, ktorý je v dnešnej dobe využívaný hlavne ako programovací jazyk vkladaný priamo do HTML kódu. Spolu s HTML a CSS tvoria základnú časť moderných webových stránok. Pri jeho využívaní je nutná podpora zo strany internetového prehliadača, pretože po načítaní stránky z webového servera beží JavaScript na pozadí priamo v prehliadači. Môžu ním byť ovládané rôzne interaktívne prvky GUI, ako tlačidlá alebo textové políčka, ale využitie nachádza aj pri vytváraní rozličných animácií alebo efektov obrázkov. Iné uplatnenie JavaScript nachádza pri PDF dokumentoch, desktopových miniaplikáciách alebo pri vývoji mobilných aplikácií.

**jQuery** je JavaScript knižnica so širokou podporou internetových prehliadačov. Kladie dôraz najmä na interakciu medzi JavaScript a HTML. Jej účelom je uľahčenie používania JavaScript na webovej stránke. V našej práci túto knižnicu budeme využívať najmä na dynamickú grafickú zmenu prvkov, ktorá bude indikovať využívanie kamier.  
Pri využívaní tejto knižnice, ušetríme mnoho riadkov kódu, ktorý by vznikol pri písaní priamo v JavaScript. Zjednodušuje tiež nemálo zložitých vecí, ako sú napríklad AJAX volania alebo manipulácia s DOM (Document Object Model - objektovo orientovaná reprezentácia HTML). DOM sa využíva pri zvolení prvku v HTLM kóde pre následnú zmenu jeho vlastností, čiže dynamickosť stránky. Pre zvolenie prvku musí byť zadefinovaný jeho identifikátor.

Knižnica jQuery obsahuje funkcie:

* Manipulácia s HTML/DOM
* Manipulácia s CSS
* Rôzne efekty a animácie
* AJAX
* Rôzne podporné programy

Okrem toho jQuery zahŕňa obrovské množstvo prídavných modulov (plug-in) pre takmer akúkoľvek úlohu. Existuje veľa rôznych JavaScript knižníc, ale jQuery je aktuálne najpopulárnejšia a najviac rozšírenia. Veľa nadnárodných spoločností využíva práve jQuery pre svoje webové stránky a to napríklad Google, Microsoft, IBM. Licencia je voľne dostupná a otvorená (open-source). [17]

Graf . *Štatistika využívania JavaScript knižníc pre 10 000 najpoužívanejších stránok*

*Zdroj* [*https://trends.builtwith.com*](https://trends.builtwith.com/javascript/javascript-library) *– Marec 2017*

**AJAX** - [Asynchronous](https://sk.wikipedia.org/w/index.php?title=Asynchr%C3%B3nne_I/O&action=edit&redlink=1) JavaScript + XML je súhrnné označenie pre technológie vývoja interaktívnych webových aplikácií, ktoré umožňujú načítať dáta z webového servera na pozadí stránky a meniť jej obsah bez potreby opätovného obnovenia. Zahŕňa v sebe napríklad funkcionalitu na princípe pravidelného posielania požiadaviek na server – Long-Pollling. V porovnaní s klasickými webovými aplikáciami môžu AJAX aplikácie pri vhodnom návrhu poskytovať komfortnejšie prostredie pre užívateľa, vyžadujú však použitie moderných webových prehliadačov. [18]

## Webový aplikačný framework

Inak nazývaný aj web framework je kolekcia balíčkov alebo modulov, ktoré umožňujú vývojárom písať webové aplikácie alebo služby bez nutnosti podrobných znalostí protokolov, sieťovej komunikácie alebo iných nízkoúrovňových prostriedkov.

Všeobecne platí, že web framework-y poskytujú podporu pre celý rad aktivít vrátane prijímania požiadaviek od klientov (webový prehliadač), manipulácie s cookies, vytvárania odozvy na požiadavku (prezentovania dát ako HTML) alebo ukladania dát. Tým, že webové aplikácie stále vyžadujú množstvo rôznych funkcií, vznikajú framework-y poskytujúce komplexné riešenia všetkých potrebných úloh. Pre označenie takýchto framework-ov sa využíva anglický výraz - full-stack. Na druhej strane sú framework-y, ktoré obsahujú iba základné funkcie a všetko ostatné sa k nim pripája pomocou modulov. To má za následok, že si môžeme vyskladať vlastný full-stack web framework. Pre zváženie vhodného variantu pre náš projekt si porovnáme tieto dve koncepcie. Pretože Raspberry Pi má dobrú podporu programovacieho jazyka Python, budeme hodnotiť webové framework-y založené na tomto jazyku. [19]

#### Python - programovací jazyk

Je to interpretovaný, objektovo orientovaný, vysoko-úrovňový, dynamický programovací jazyk. Vďaka svojim vysoko-úrovňovým dátovým štruktúram je veľmi atraktívny pre RAD(Rapid application development - rýchly vývoj aplikácií), ale aj ako skriptovací jazyk, či jazyk spájajúci komponenty, ktoré boli vytvorené pomocou rovnakých alebo rôznych programovacích jazykov. Python má jednoduchú a ľahko zvládnuteľnú syntax, čo prispieva k lepšej čitateľnosti kódu a tým aj k ľahšej údržbe vytvorených programov. Interpreter jazyka Python a rozsiahla knižnica štandardných funkcií sú dostupné zadarmo a môžu byť voľne distribuované. [20]

### Django

Je široko používaný web framework s označením full-stack, čo vychádza aj z jeho filozofie „battery-included“. Znamená to, že výrobca už všetky potrebné funkcionality na stavbu web servera implementoval a výrobok sa dodáva ihneď pripravený na výrobu servera. Medzi tieto funkcionality môžeme zaradiť administrátorskú časť stránky, databázové rozhranie, štruktúru adresárov a podobne. Výhodou pri využití Django je kvalitne spracovaná dokumentácia, ktorá je voľne dostupná a popisuje rôzne prípady,  
ako prvky z balíčka používať. Kladným atribútom je aj veľká komunita a aj samotné časté využitie tohto web framework-u v rôznych projektoch, čo ešte viac dopomáha k ľahkému vývoju. Je to preto, že ak sa nám vyskytne určitý problém, je vysoká pravdepodobnosť, že takáto situácia bola už niekedy riešená a adekvátne  kvalitné riešenie už bolo vymyslené. To je aj jeden z argumentov vývojárov, prečo tak často nasadzujú tento framework do svojich projektov.

Z tohto všetkého nám vychádza, že Django sa hodí najmä na komplexnejšie projekty. Častými využitiami sú napríklad elektronické obchody, publicistické stránky, fóra alebo stránky rôznych produktov. Django nachádza využitie aj vo veľkých projektoch ako je napríklad Instagram, Pineterest, Spotify alebo stránka NASA, čo je dôkaz o jeho kvalite. Nevýhodou môže byť využitie pri malých projektoch, kde by systém obsahoval veľké množstvo funkcionalít, ktoré by ale neboli využité a tak by sa celý systém mohol spomaliť. [21]

### Flask

Ako druhú stranu mince sme zvolili mikro-framework Flask. Obsahuje iba základnú funkcionalitu potrebnú pre spustenie webového servera a jeho jednoduchú obsluhu. Všetky pokročilejšie funkcie musia byť pridané pomocou doplnkových modulov. Flask tak v jeho podstate poskytuje jednoduchosť, flexibilitu a presnú kontrolu nad tým, ako chceme veci do projektu implementovať, napríklad aký databázový systém budeme využívať. Prídavné moduly poskytuje výrobca, ale aj tretie strany. Výrobca ponúka väčšinou alternatívy ku full-stack framework-ovým funkcionalitám alebo pridáva ich vylepšené verzie. Toto všetko nájdeme väčšinou dobre zdokumentované a uvádzané aj s príkladmi použitia. Moduly tretích strán môžu byť malým hazardom. Na jednej strane to môžu byť kvalitné produkty, ktoré ponúkajú efektívne a inovatívne riešenia problémov s dobre spracovanou dokumentáciou. Môže sa ale stať, že natrafíme na presný opak.   
To znamená, že prídavný modul, ktorý by mal byť schopný vyriešiť presne náš problém, môže obsahovať mnoho chýb alebo produkt môže byť zle zdokumentovaný a vývojár stráca čas pri študovaní jeho štruktúry. Preto pri takýchto framework-och je veľmi dôležitý výber jednotlivých doplnkových modulov.

Flask nachádza časté využitie pri mikroprojektoch, kde vývojár implementuje iba funkcie, ktoré bude potrebovať, a tak systém ostane ľahký a výkonný. Takéto aplikácie sú veľmi rýchle, ako na chod tak aj na vývoj, pretože využívame iba to, čo naozaj potrebujeme. Samozrejmosťou je možnosť využitia pri veľkých a komplexných projektoch, tu by ale mohol nastať problém s prepojením jednotlivých modulov medzi sebou. Tých by pri Flask bolo nutné implementovať mnoho, pretože obsahuje naozaj iba základnú funkcionalitu a preto sa toto riešenie nevyužíva často. [22]

### Výber webového aplikačného framework-u

Pre potreby nášho projektu sme sa rozhodli využiť mikro-framework Flask.  
Je to najmä kvôli jeho flexibilite a jednoduchosti, pretože naša webová aplikácia bude založená na jednoduchých prvkov, ktoré Flask buď už v základe poskytuje alebo ich môžeme jednoducho doplniť. Dokumentáciu a postupy budeme využívať z oficiálnej webovej stránky. Tu nájdeme veľmi prehľadné návody ako vytvoriť základnú kostru webového serveru a  informácie o  rozširovacích modulov či už od výrobcu alebo odkazy na kvalitné moduly tretích strán. [23, 24]

## Komunikácia klient-server v reálnom čase

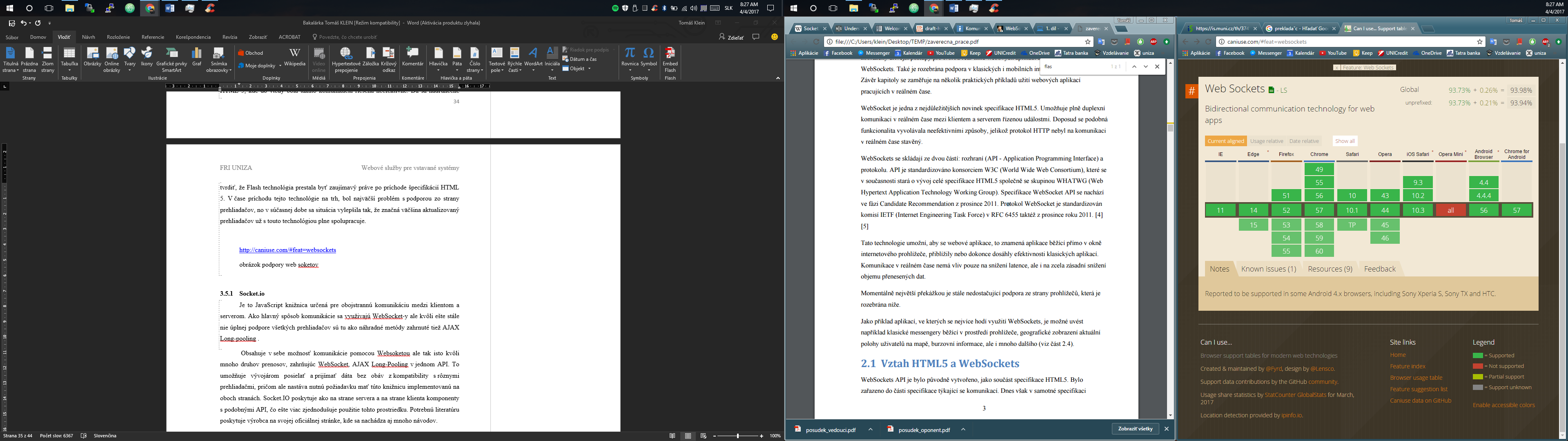
Po vytvorení nášho webového servera potrebujeme vytvoriť komunikačný kanál, ktorým budeme zasielať klientovi dáta o stave kamery a ktorý bude zabezpečovať komunikáciu pomocou krátkych správ v reálnom čase. Na zabezpečenie takéhoto prenosu dát využijeme protokol WebSocket.

### WebSocket

Protokol poskytuje obojstranný komunikačný kanál medzi webovým serverom a internetovým prehliadačom. Vytvára permanentné spojenie medzi klientom a serverom, kde jedna strana môže kedykoľvek poslať správu tej druhej. Zjednodušene sa dá povedať, že protokol sa skladá z dvoch po sebe idúcich častí, „handshake“-podanie rúk a následný prenos dát. Podanie rúk inicializuje prepojenie klienta a webového servera. Funguje na základe toho, že klient odošle HTTP požiadavku s hlavičkou upgrade, ktorá znamená že klient chce nadviazať spojenie pomocou technológie WebSocket. Ak sa nevyskytne žiadna chyba a webový server podporuje takéto prepojenie, server naspäť odošle odpoveď, tiež s hlavičkou upgrade. Po prijatí správy klientom sa zmení spojenie z HTTP na prepojenie WebSocket. Štandardne beží tento protokol na portoch určených pre HTTP – port 80 a pre šifrované pripojenie na HTTPs - port 443. Veľkou výhodou je vlastnosť, že server môže sám odosielať dáta klientovi, bez nutnosti zasielania prvotnej požiadavky klientom.  
To má dobrý vplyv na latenciu, t.j. čas od zmeny vstupných dát po zobrazenie na webovej stránke, ktorá je oveľa nižšia ako pri iných technológiách.

Jedna z takýchto technológií sa nazýva **Long-Polling.** Funguje na princípe,  
že klient odošle požiadavku na server, ktorý ju príjme ale nezareaguje hneď čím spojenie ostáva na krátko otvorené. Odpoveď odošle až v momente zmeny požadovanej informácie alebo v prípade vypršania času pre otvorené spojenie.

**Flash** technológia bola tiež alternatívou pre komunikáciu v reálnom čase,  
ale v súčasnej dobe sa prestáva používať najmä kvôli jej zraniteľnosti. Pri jeho využívaní vznikal často problém s nutnosťou pridania podporného balíčka pre prehliadač, kde ho nie vždy bolo možné nainštalovať.

Protokol WebSocket je v porovnaní s Long-Polling výhodnejší pre sieťovú prevádzku, ktorá sa v dôsledku obmedzenia zasielania požiadaviek odľahčuje, ale aj pre webové servery, ktoré nemusia obstarávať odozvy na požiadavky webových stránok. Oproti iným technológiám je technológia WebSocket priamo vyvíjaná pre obojstrannú komunikáciu v reálnom čase. Táto technológia prišla spolu so špecifikáciou protokolu HTML 5, kde dovtedy bola takáto komunikácia riešená neefektívne. V čase príchodu protokolu WebSocket bol najväčší problém s podporou zo strany prehliadačov, no v súčasnej dobe sa situácia vylepšila tak, že väčšina aktualizovaných prehliadačov už s touto technológiou plne spolupracuje. [25, 26]

Obrázok . *WebSocket podpora pre najpoužívanejšie internetové prehliadače - Apríl 2017*

### Socket.io

Socket.io je JavaScript knižnica určená pre obojstrannú komunikáciu medzi klientom a serverom. Ako hlavný spôsob komunikácie sa využíva technológia WebSocket. Pri využívaní prehliadača bez podpory tohto protokolu, knižnica obsahuje viaceré alternatívne možnosti komunikácie napríklad využitie Long-Polling. Toto integrovanie viacerých druhov komunikácie do jedného API je veľmi výhodné riešenie pre vývojárov, ktorí môžu písať aplikácie bez nutnosti riešenia kompatibility s rôznymi webovými prehliadačmi. Knižnica musí byť implementovaná na oboch stranách, klient-server. Socket.io poskytuje na strane klienta aj webového servera komponenty s podobnými API, čo dopomáha k jednoduchšiemu využitiu tohto prostriedku. Potrebnú literatúru poskytuje výrobca na svojej oficiálnej stránke, kde sa nachádza aj mnoho návodov a riešení problémov. [27, 28]

### Flask-SocketIO

Tento modul je implementácia knižnice Socket.io určená pre web framework Flash. Pre naše potreby si ukážeme základnú funkciu prijímania a posielania dát. Podrobné návody na využitie ďalších zaujímavých funkcií,  možností a nastavení sú kvalitne zdokumentované na oficiálnej stránke.

**Príklad odoslania správy**

Odosielanie a prijímanie dát funguje na základe udalostí. Pri odosielaní dát musíme pridať vždy názov udalosti, ku ktorej sú tieto dáta určené. Druhá strana musí mať definovaný ten istý názov udalosti, aby mohola čakať na prijatie dát. Posielané dáta môžu byť ľubovoľné JSON objekty, čo zahŕňa aj reťazce, čísla alebo polia.

**Server (Python)**

def stav\_kamera1():

socketio.emit('kamera1', {'value': data})

**Klient (jQuery)**

socket.on('kamera1', function(e) {

console.log("Kamera 1:", e.value);

}

Ukážka zobrazuje odoslanie správy webovým serverom a prijatia správy klientom. Funkcia stav\_kamera1 po zavolaní vygeneruje udalosť s dátami a tie sa následne odošlú. Klient čaká na vopred definovanú udalosť a po jej prijatí vypíše na konzolu textový výpis spolu s prijatými dátami. [29]

## Webová stránka

Poslednou časťou projektu je webová stránka, ktorá bude dynamicky zobrazovať využitie kamery video strižňou a bude poskytovať jednoduché rozhranie na krátku textovú komunikáciu.

Vývoj webovej stránky alebo aplikácie od nuly môže zabrať priveľa času.  
Pre urýchlenie vytvorenia stránok opäť použijeme služby framework-ov. Budeme využívať práve tie s prívlastkom CSS alebo front-end, keďže oba znamenajú to isté. Obsahujú už vopred preddefinované HTML, CSS a JavaScript dokumenty. Veľkou výhodou je aj fakt, že tieto framework-y sa zameriavajú už aj na optimalizáciu pre obrazovky mobilných telefónov alebo tabletových počítačov. Majú rôzne nastavenia podľa toho, aké rozlíšenie obrazovka poskytuje a obsah upravujú tak, aby bol vždy dobre čitateľný a manipulácia  
s prvkami bola pohodlná. Táto funkcia prispôsobovania obsahu obrazovke sa nazýva responzívny dizajn. Na vytvorenie celkového dizajnu stránky nám framework-y ponúkajú šablóny, ktoré si môžeme upraviť, a tak dosiahnuť požadovaný vzhľad. V prípade, že žiadna šablóna nám nevyhovuje alebo chceme vložiť viac individuality do konečného výsledku, môžeme si vzhľad stránky poskladať z komponentov, ktoré sú nám ponúknuté, napríklad postranné alebo vrchné panely.

V súčasnej dobe je populárnych viacero framework-ov, ale medzi najpoužívanejšie patria: Bootstrap a Foundation. Obidva majú svoje výhody a samozrejme aj nevýhody. Keďže naša stránka bude fungovať na jednoduchom dizajne, náš účel by splnil i jeden  
i druhý bezproblémovo. Pre väčšiu rozšírenosť sme sa rozhodli využiť služby Bootstrap CSS framework-u vo verzii 3.3.7. Dostupná je už aj verzia 4.0.0, ale tá sa nachádza ešte v testovacej fáze*.* [30, 31]

Graf . *Využitie Bootstrap a Foundation na najnavštevovanejších webových stránkach*

*Zdroj: https://trends.builtwith.com - Apríl 2017*

### Bootstrap

Front-end framework určený pre dizajn webových stránok a aplikácií s voľnou licenciou (open-source). Pôvodný názov bol Twitter Blueprint. Využíva CSS preprocesor Less, ktorý generuje CSS súbor zo zdrojového kódu napísaného JavaScript. Všetky informácie pre túto časť práce sme čerpali z oficiálnej stránky, kde je spracovaná kvalitná dokumentácia s množstvami príkladov a riešení. [32]

Jednou z najväčších výhod je už spomenutý responzívny dizajn. Mobilné telefóny a tabletové počítače sa stávajú častejšie využívanými prostriedkami na prezeranie obsahu internetu. Pre náš projekt je táto vlastnosť veľmi dôležitá, pretože kamerami pri kamerách budú sledovať stav svojej kamery práve na mobilných telefónoch alebo tabletoch. Bootstrap tento problém rieši hlavne pomocou systému mriežky, ktorá delí obrazovku vertikálne na dvanásť častí - stĺpcov. Pri horizontálnom delení používame označenie rady. Kontajneru, ktorý vkladáme na stránku tak môžeme určiť šírku, respektíve koľko stĺpcov bude zabrať. Pri zobrazení obsahu na obrazovke, ktorá nie je vhodná na zobrazenie všetkých dvanástich stĺpcov v jednom rade, by bol obsah nečitateľný a v tomto sa kontajnery rozdelia do viacerých radov pod seba. Táto funkcia zabezpečí vždy kvalitne zobrazený obsah. Nastaviť môžeme aj odlišnú šírku kontajneru pre rôzne rozlíšenia obrazoviek, tak aby sme vždy dosiahli požadovaný dizajn.

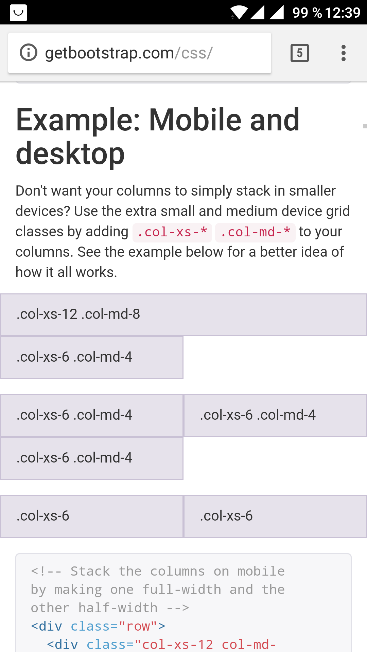
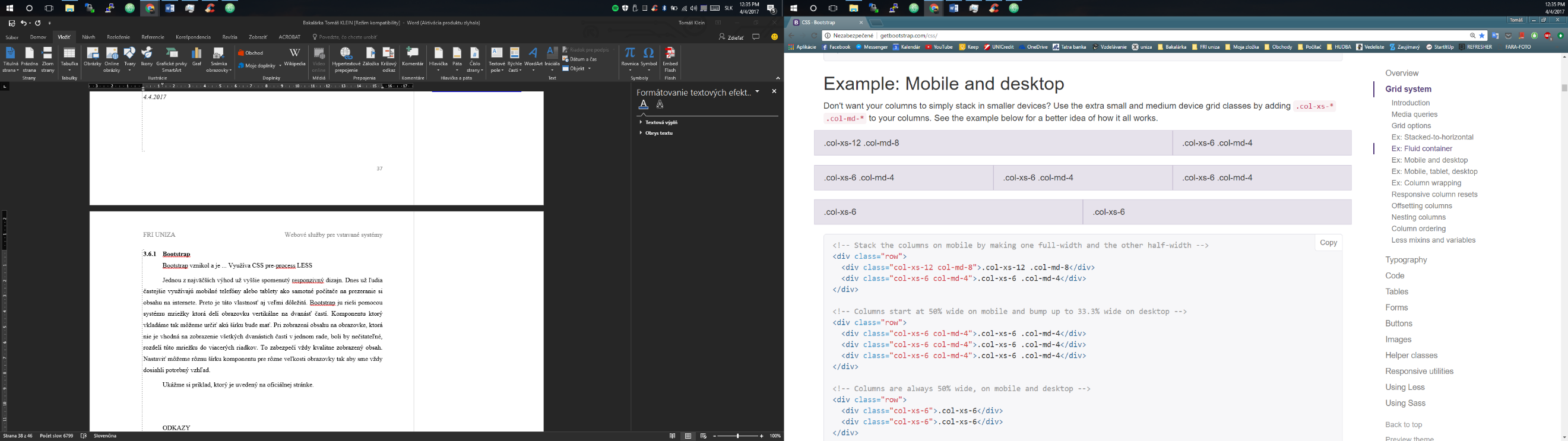
Ukážeme si príklad, ktorý je uvedený na oficiálnej stránke. Je to prípad, kedy stránka používa rôzne šírky kontajnerov pre dva typy a rozlíšenia obrazoviek. Na nastavenie šírky kontajneru využívame triedy mriežky, ktoré už v názve obsahujú pre aké zariadenia sú určené a akú šírku budú mať. Bootstrap rozoznáva 4 takéto triedy:

* Extra malé zariadenia (rozlíšenie menšia ako 768 px) - .col-xs-*z*
* Malé zariadenia (väčšie ako 768 px) - .col-sm-*z*
* Stredné zariadenia (väčšie ako 992 px) - .col-md-*z*
* Veľké zariadenia (väčšie ako 1200 px) - .col-lg-*z*

V každej triede za jej označením musíme pripísať číslo, naše písmeno z,  
ktoré určuje koľko stĺpcov daný kontajner bude zaberať. Súčet môže byť maximálne 12   
pre jeden rad. Ak sa tento počet prekročí, kontajnet, ktorý sa nezmestí do tohto súčtu sa zaradí do ďalšieho radu.

V našom ukážkovom príklade má prvý rad takéto rozdelenie:

Prvý kontajner - *.col-xs-12 .col-md-8* Druhý kontajner *- .col-xs-6 .col-md-4*

**Pri zobrazení obsahu stránky na zariadení s malou obrazovkou - xs, prvý kontajner sa zobrazí na celej šírke obrazovky a druhý prejde o riadok nižšie, kde bude zaberať polovicu obrazovky. V prípade zobrazenia na zariadení so stredne veľkou obrazovkou - md, bude prvý kontajner zaberať dve tretiny obrazovky a na zvyšnej časti bude druhý kontajner.

Obrázok . *Responzívny dizajn – porovnanie dvoch zariadení  
výrez obrazovky: prenosný počítač-md (vľavo), mobilný telefón-xs (vpravo)*

Na obrázkoch môžeme jasne vidieť, ako tento responzívny dizajn funguje v prvom rade, ktorý sme si analyzovali, ale aj v ďalších oddelených radách pod ním. Pri tomto dizajne sa Bootstrap zameriava nie len na samotnú mriežku, ale aj prispôsobuje viaceré komponenty, ako tlačidlá, názvy alebo tabuľky, ktoré sa vždy snažia zobrazovať adekvátne k obrazovke zariadenia.

Pri personalizácii stránky nám Bootstrap poskytuje možnosť meniť obsah Less komponentov a doplnkových modulov jQuery. Ďalším výrazným prispôsobením je editácia Less premenných. Tu si môžeme zmeniť preddefinované farby, štýly písma, veľkosti komponentov, nastavovať tlačidlá a mnoho ďalšieho. Medzi tie najvýznamnejšie patrí možnosť zmeniť si počet stĺpcov pri rozdelení obrazovky do mriežky a možnosť predefinovať rozlíšenie displeja pre triedu mriežky. [33]

# Realizácia navrhnutého systému

Táto kapitola bude venovaná praktickej časti našej práce. Opíšeme realizáciu použitých riešení a vysvetlíme si prepojenie medzi jednotlivými prvkami.

## DPS

Prvotná realizácia DPS začala náčrtom schémy na papieri a následne sme zapojenie otestovali na univerzálnej doske. Po potvrdení funkčnosti zapojenia sa prešlo na počítačový návrh dosiek, kde bol využitý program KiCad. Tento program je voľne dostupný a ponúka širokú škálu návrhových prostriedkov pre optimálne vytvorenie plošných spojov. Počítačový návrh sa realizoval prekreslením schémy a vytvorením jednotlivých prepojení, označením súčiastok a ciest. Následne sme jednotlivým súčiastkam zo schémy priradili ich fyzické puzdrá, ktoré ich budú zastupovať na plošnom spoji.  
Pre minimalizáciu veľkosti DPS sme sa snažili použiť čo najviac SMD súčiastok. V prípade, že program neobsahuje nami požadovanú súčiastku, je možné stiahnuť si doplnkovú knižnicu alebo si navrhnúť súčiastku priamo v KiCad. Možnosť návrhu pre konektor DB-15HD na dosku Tally-GPIO sme využili aj my. Po rozmiestnení súčiastok a vytvorení vodivých ciest na oboch podporných doskách sme pridali popisy, montážne diery a dosky boli pripravené na odoslanie do výroby. V poslednej fáze sme dosky osadili a otestovali ich funkčnosť.

## Raspberry Pi a Flask

Po nainštalovaní operačného systému Rasbian na pamäťovú kartu nás čakala úvodná inicializácia. Pre možnosť pracovania vzdialeným prístupom sme si prepojili Raspberry Pi s počítačom pomocou programu Putty a protokolom SSH. To nám dalo možnosť obsluhovať konzolu počítača, cez ktorú sme si nainštalovali potrebné programy a prídavné balíčky. Pre písanie kódu sme si vybrali program Atom, ktorý zabezpečil,  
že kód písaný na počítači sa ukladal ako do počítača, tak aj na pamäťovú kartu počítača Raspberry Pi. Ako prvý sme vytvorili program pre bezpečné vypnutie systému. Jeho logika je jednoduchá a to inicializovať potrebný port z GPIO a následne kontrolovať úroveň signálu na ňom. Po stlačení tlačidla program zaznamená zmenu úrovne signálu a čaká vopred zadefinovaný čas, po ktorom kontroluje, či sa úroveň signálu nezmení.   
Tento stav čakania ošetruje nechcené stlačenie tlačidla. Ak sa úrovne zhodujú aj po tejto kontrole, program dá systému pokyn na bezpečné vypnutie. Ďalej sme pokračovali vytvorením web servera pomocou Flask framework-u, ktorý sme si importovali ako modul do python súboru. Následne sme do programu pridali balíček Flask-SocketIO, ktorý sa postará o komunikáciu s webovým prehliadačom v reálnom čase. Úvodné inicializovanie pridaných modulov pre webový server sme vykonali podľa výrobcom uvedených nastavení. Server nám bude poskytovať viacero webových stránok. Prvá hlavná stránka obsahuje všeobecný prehľad stavu kamier a obrazoviek, čiže zobrazuje všetkých 8 vstupov do video strižne na jednej obrazovke. Druhá stránka obsahuje iba textovú komunikáciu a je cielená pre používanie režisérom, ktorý bude pomocou nej komunikovať s kameramanmi. Posledný druh stránok je určený pre samotných kameramanov. Pre každú kameru a obrazovku je vytvorená osobitá stránka, ktorá obsahuje zobrazovanie stavu kamery a taktiež aj textovú komunikáciu. Webový server je poskytovaný na IP adrese počítača Raspberry Pi na porte 5000. Ako posledné sme nastavili obidva programy, webový server aj program pre bezpečné vypnutie, aby sa spúšťali na pozadí po štarte systému.

## Webové stránky

Na vytvorenie dizajnu našich webových stránok sme využívali jednotlivé komponenty, ktoré nám poskytuje framework Bootstrap. Súbory potrebné pre jeho využitie sme si uložili do jednej zložky ku webovému serveru, odkiaľ ich načítavame do HTML dokumentu. Na vrchnej časti všetkých poskytovaných stránok je umiestnený panel, ktorý prepája poskytované webové stránky. V pravej časti panela sa nachádza možnosť zvolenia si dennej alebo nočnej verzie stránky. Táto funkcia je realizovaná pomocou jQuery. Spomínané verzie stránok sa líšia iba vo farbe komponentov. Denná verzia má komponenty s výraznými kontrastnými farbami, zatiaľ čo nočná verzia má tlmené, nerušivé farby. Pre obsah stránok sme využili štandardné prvky ako sú štítky, tlačidlá alebo textové polia. Takisto sme implementovali aj knižnicu Socket.IO, ktorej funkčnosť sme si overili jednoduchým výpisom na konzole v prehliadači. Ukážky webových stránok sa nachádzajú v prílohe.

## Grafická zmena stavu kamery

Pre zisťovanie stavu kamier na Raspberry Pi konektore využívame knižnicu GPIO. Za pomoci jednoduchého ovládania sme pri inicializovaní webového serveru zvolili jednotlivé porty za vstupné. Pre kontrolu stavu signálu na portoch sme sa rozhodli využiť viaceré vlákna. Presnejšie sa jedná o osem vlákien, pričom každé vlákno kontroluje jednu kameru alebo obrazovku, čiže obstaráva dva porty. Interval kontroly sme si nastavili na pol sekundy, čo pri testovaní vyšlo ako rozumná hodnota. Vlákno funguje tak, že po zistení zmeny stavu na kontrolovaných portoch vygeneruje udalosť pomocou Socket.io, pripojí dáta a odošle ich na webovú stránku. Dáta nesú aktuálny stav kamery, ktorý môže mať štyri hodnoty pre rôzne kombinácie signálov:

* 0 – obraz kamery sa nepoužíva
* 1 – READY signál
* 2 – LIVE signál
* 3 – LIVE + READY signál

Udalosť nesie názov kamery alebo obrazovky, na ktorej sa zmena vyskytla.  
Po prijatí a spracovaní udalosti stránkou sa dáta vyhodnotia a jQuery priradí komponentu reprezentujúcemu danú kameru alebo obrazovku iné farebné prevedenie pomocou zamenenia triedy komponentu, čo sa prejaví ako grafická zmena stavu kamery na webovej stránke.

Ukážka zmeny farby komponentu pomocou zamenenia jeho triedy:

$("#live").toggleClass('alert-live', true);

$("#live").toggleClass('alert-free', false);

$("#live").toggleClass('alert-live', false);

$("#live").toggleClass('alert-free', true);

## Textová komunikácia

Textová komunikácia funguje na podobnom princípe ako zmena grafického zobrazovania stavu pomocou udalostí. Po napísaní správy v konverzácií užívateľ stlačí tlačidlo odoslať, ktoré vygeneruje udalosť s názvom *server\_message,* pripojí text správy a odošle ju na server. Ten má za úlohu túto správu rozdistribuovať do ostatných zariadení. Čaká na túto udalosť a po prijatí preposiela obsah správy pod hlavičkou novej udalosti - *message*. Webový prehliadač túto správu prijme a pridá text správy do konverzačného okna. Kontrola stlačenia tlačidla odoslať, výber správy z textového poľa, vkladanie správy do konverzačného okna a iné funkcie sa vykonávajú taktiež pomocou jQuery. Pri písaní správ sme pridali možnosť využitia prezývky, adresovanie správy pre rôzne osoby a možnosť využitia už vopred preddefinovaných správ.

# [Záver](https://d.docs.live.net/77491396eb51d89e/Bakalarka/Pokyny_pre_vypracovanie_ZP.docx#Záver)

Ciele práce sa nám podarilo splniť, keďže sme vytvorili funkčný svetelný Tally systém, splňujúci požadované vlastnosti. V úvodnej časti práce sme vysvetlili funkciu tohto systému, jeho účel a ukázali sme si už existujúce riešenia a následne sme sa venovali návrhu vlastného riešenia. Pokračovali sme vytvorením obalu a podporných dosiek využívaných v našom systém. V teoretickej časti sme si uviedli a vysvetlili všetky dôležité informácie a pojmy z danej oblasti informatiky, ktoré boli nevyhnutné pri vytváraní praktickej časti.

Systém sníma stav využitia kamier na video strižni a postupne predáva túto informáciu pomocou lokálnej siete do webového prehliadača mobilného telefónu kameramanov. Zmeny stavu sú zobrazované graficky kvôli rýchlejšiemu zaznamenaniu tejto udalosti. Tak isto sme implementovali komunikačný kanál pre krátke textové správy, dopomáhajúce k zlepšeniu dorozumievacích schopností medzi réžiou a kameramanmi. Webové stránky sú optimalizované pre mobilné telefóny. Finálny produkt našej práce je pripravený do reálneho nasadenia.

Prínosom tejto práce je možnosť využitia bezdrôtového prenosu Tally signálu aj pre amatérske organizácie, bez nutnosti investovania vysokých súm do profesionálnej techniky. Všetky framework-y a programy použité v práci sú voľne dostupné a dobre zdokumentované. Zariadenie je koncipované na jednoduchú úpravu a rozšíriteľnosť, čo dovoľuje možným záujemcom prispôsobiť si systém podľa vlastných potrieb.  
Do budúcnosti by bolo možné ďalej rozširovať funkcionalitu systému napríklad o internú hlasovú komunikáciu alebo distribúciu živého výstupu video strižne na webové stránky kameramanov, čo by ešte viac prispelo ku komfortnosti pri ich práci.

# Zoznam použitej literatúry

|  |  |
| --- | --- |
| [1] | **DataVideo Technologies,** „DataVideo TB-5,“ Marec 2017. [Online]. *http://www.datavideo.com/us/product/TB-5.* |
| [2] | **DataVideo Technologies,** „DataVideo ITC-100,“ Marec 2017. [Online]. *http://www.datavideo.com/us/product/ITC-100.* |
| [3] | **Tally Technologies,** „TallyTec Products,“ Marec 2017. [Online]. *http://www.tallytec.net/products.* |
| [4] | **Blackmagic Design,** „ATEM Studio and Camera Converters,“ Marec 2017. [Online]. *https://www.blackmagicdesign.com/products/atemconverters.* |
| [5] | **Jason Baker,** „What is a Raspberry Pi?,“ Marec 2017. [Online]. *https://opensource.com/resources/what-raspberry-pi.* |
| [6] | **Liz Upton,** „Pinout for GPIO connectors,“ Apríl 2017. [Online]. *https://www.raspberrypi.org/blog/pinout-for-gpio-connectors/.* |
| [7] | **Per Christensson,** „Framework,“ Marec 2017. [Online]. *https://techterms.com/definition/framework.* |
| [8] | **Viz Teams,** „Advantages and disadvantages of framework,“ Apríl 2017. [Online]. *http://www.vizteams.com/blog/advantages-and-disadvantages-of-frameworks/.* |
| [9] | **Jiří Kosek,** „Webový server,“ Apríl 2017. [Online].  *http://htmlguru.cz/vystaveni-webovy-server.html.* |
| [10] | **Margaret Rouse,** „HTTP (Hypertext Transfer Protocol),“ Marec 2017. [Online]. *http://searchwindevelopment.techtarget.com/definition/HTTP.* |
| [11] | **Wikipédia,** „Hypertext Transfer Protocol,“ Marec 2017. [Online]. *https://en.wikipedia.org/wiki/Hypertext\_Transfer\_Protocol.* |
| [12] | **R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, Microsoft, Xerox, W3S/MIT,** „Hypertext Transfer Protocol - RFC 2616, “ Network Working Group, Marec 2017. [Online]. *https://tools.ietf.org/html/rfc2616.* |
| [13] | **E. Rescorla,** „HTTP Over TLS - RFC 2181,“ Network Working Group, Marec 2017. [Online]. *https://tools.ietf.org/html/rfc2818.* |
| [14] | **Wikipédia,** „Hypertextový značkový jazyk,“ Apríl 2017. [Online]. *https://sk.wikipedia.org/wiki/Hypertextov%C3%BD\_zna%C4%8Dkov%C3%BD\_jazyk.* |
| [15] | **Ross Shannon,** „What is HTML?,“ Apríl 2017. [Online]. *http://www.yourhtmlsource.com/starthere/whatishtml.html.* |
| [16] | **Peter Bohuš,** „CSS – čo je to a na čo sa používa,“ Apríl 2017. [Online]. *https://cookies.sk/co-je-to-css/.* |
| [17] | **w3schools,** „jQuery Introduction,“ April 2017. [Online]. *https://www.w3schools.com/jquery/jquery\_intro.asp.* |
| [18] | **jQuery Tutorial,** „Introduction to AJAX,“ Apríl 2017. [Online].  *http://www.jquery-tutorial.net/ajax/introduction/.* |
| [19] | **Juha Koskelainen,** „Web Frameworks for Python,“ Apríl 2017. [Online]. *https://wiki.python.org/moin/WebFrameworks.* |
| [20] | **RNDr. Jozef Jirásek, PhD.,** „Python,“ Univerzita Pavla Jozefa Šafárika v Košiciach, Apríl 2017. [Online]. *http://ics.upjs.sk/~jirasek/sps/python/python.html.* |
| [21] | **Matt Makai,** „Django,“ Apríl 2017. [Online]. *https://www.fullstackpython.com/django.html.* |
| [22] | **Matt Makai,** „Flask,“ Apríl 2017. [Online]. *https://www.fullstackpython.com/flask.html.* |
| [23] | **Rayn Brown,** „Django vs Flask vs Pyramid: Choosing a Python Web Framework,“  Apríl 2017. [Online]. *https://www.airpair.com/python/posts/django-flask-pyramid.* |
| [24] | **Gareth Dwyer,** „Flask vs. Django: Why Flask Might Be Better,“ Apríl 2017. [Online]. *https://www.codementor.io/garethdwyer/flask-vs-django-why-flask-might-be-better-4xs7mdf8v.* |
| [25] | **Bc. Daniel Sák,** „Diplomová práca - Webová aplikace pro správu, “ Masarykova Univerzita - Fakulta informatiky, Apríl 2017. [Online]. *https://is.muni.cz/th/374560/fi\_m/diplomovaPraca.txt.* |
| [26] | **I. Hickson, Google Inc.,** „The WebSocket protocol,“ Network Working Group,  Apríl 2017. [Online]. *https://tools.ietf.org/html/draft-hixie-thewebsocketprotocol-76.* |
| [27] | **Wikipédia,** „ Socket.IO,“ Apríl 2017. [Online]. *https://en.wikipedia.org/wiki/Socket.IO.* |
| [28] | **Guillermo Rauch,** „Socket.IO-Engine.IO: the realtime engine,“ Apríl 2017. [Online]. *https://github.com/socketio/engine.io.* |
| [29] | **Miguel Grinberg,** „Flask-SocketIO,“ Apríl 2017. [Online]. *https://flask-socketio.readthedocs.io/en/latest/.* |
| [30] | **Wikipédia,** „Bootstrap,“ Apríl 2017. [Online]. *https://cs.wikipedia.org/wiki/Bootstrap#Znovu\_vyu.C5.BEiteln.C3.A9\_komponenty.* |
| [31] | **Brain Jackson,** „Bootstrap vs Foundation – Top 2 CSS Frameworks,“ Apríl 2017. [Online]. *https://www.keycdn.com/blog/bootstrap-vs-foundation/.* |
| [32] | **Matthew Dean,** „Less - Getting Started,“ Apríl 2017. [Online].  *http://lesscss.org/.* |
| [33] | **Bootstrap**, „Front-End Framework,“ Apríl 2017. [Online].  *http://getbootstrap.com/css/.* |

# Referencie na obrázky

**Obrázok 1.1** Ilustračný obrázok – prenosná video strižňa

[*http://www.resource.datavideo.com/images/MS-2800\_side.jpg*](http://www.resource.datavideo.com/images/MS-2800_side.jpg)

**Obrázok 1.2** Tally svetelné indikátory: doplnkový, integrovaný:  
*https://www.bhphotovideo.com/images/images750x750/o\_c\_white\_ocw\_sl3\_dvb\_ryb\_sinal\_lite\_3\_color\_column\_integrated\_1253348.jpg*

*https://images.blackmagicdesign.com/images/products/blackmagicursaminipro/liveproduction/tally-md.jpg?\_v=1488939468*

**Obrázok 1.3** Označenie pinov konektora DB-15HD *vlastný obrázok*

**Obrázok 1.4** Zariadenie DataVideo TB-5: indikátor pre kameramana riadiaca rednotka: *http://www.markertek.com/productImage/450X450/DV-TDLT.JPG*

*http://www.datavideo.com/us/product/TB-5*

**Obrázok 1.5** Zariadenie DataVideo ITC-100:

*http://www.resource.datavideo.com/images/ITC-100.jpg*

**Obrázok 1.6** TallyTec Pro vysielač,TallyTec Pro prijímač: *http://www.tallytec.net/products*

**Obrázok 1.7** Blackmagic Design: štúdiový konvertor, kamerový konvertor:

*http://www.digitaldepot.co.za/wp-content/uploads/2016/11/blackmagic-atem-camera-converter.jpg*

*https://www.tnpbroadcast.co.uk/images/blackmagic-design-swrconvrck2-atem-studio-converter-2-p7466-22408\_image.jpg*

**Obrázok 1.8** Aktuálne riešenie: redukcia, svetelné indikátory *vlastný obrázok*

**Obrázok 2.1** Rozmiestnenie GPIO portov na Raspberry Pi ver.B

*http://www.raspberrypi-spy.co.uk/wp-content/uploads/2012/09/Raspberry-Pi-GPIO-Layout-Revision-2.png*

**Obrázok 2.2** Časť schémy z redukcie Tally-GPIO *vlastný obrázok*

**Obrázok 2.3** Počítačový model ochranného obalu *vlastný obrázok*

**Obrázok 3.1** Ilustrácia komunikácie klient-server *vlastný obrázok*

**Obrázok 3.2** WebSocket podpora pre najpoužívanejšie internetové prehliadače:  *http://caniuse.com/#feat=websockets*

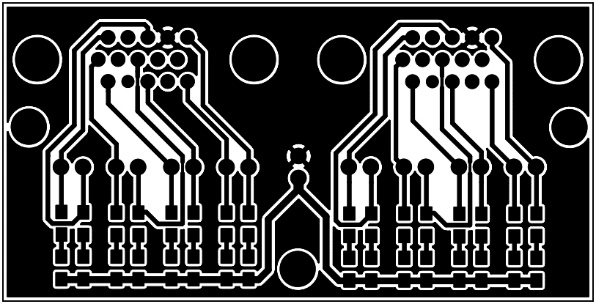
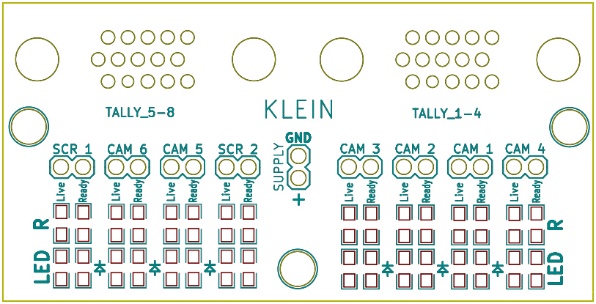
**Obrázok 3.3** Responzívny dizajn – porovnanie dvoch zariadení:

*http://getbootstrap.com/css/*

# Prílohy

## Príloha A: Podporná doska – GPIO-TALLY

*schéma*



*Vodivé spoje (vľavo), obsadzovací plán (vpravo)*

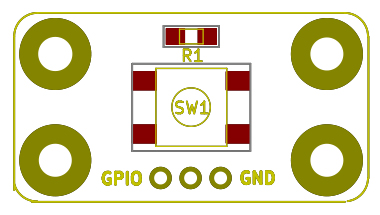
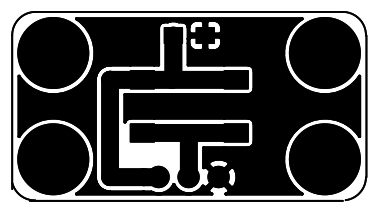
*Hodnoty obmedzovacích rezistorov a farby LED diód*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Signál | Rezistory | | LED diódy | |
| označenie | hodnota | označenie | farba |
| LIVE | R1, R3, R5, R7, R9, R11, R13, R15 | 180 Ω | D1, D3, D5, D7, D9, D11, D13, D15 | červená |
| READY | R2, R4, R6, R8, R10, R12, R14, R16 | 150 Ω | D2, D4, D6, D8, D10, D12, D14, D16 | zelená |

## Príloha B: Podporná doska pre bezpečné vypnutie systému

****

*schéma*

****

*Vodivé spoje (vľavo), obsadzovací plán (vpravo)*

Výpočet obmedzovacieho rezistora R1:

Tlačidlo SW1: IMAX = 50 mA

Rezistor R1 (0805): PMAX = 125 mW

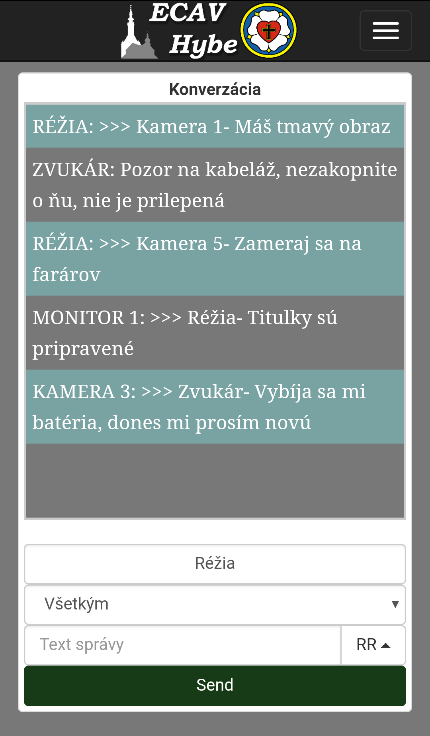
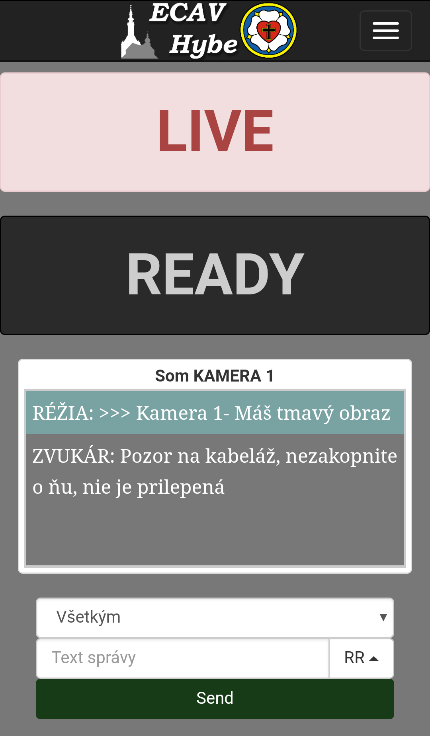
Pre rezistor R1 sme vybrali hodnotu 150Ω.

## Príloha C: Ochranný obal

*Ochranný obal pred opracovaním (hore) a po opracovaní (dole)*

*Ochranný obal vo finálnej podobe*

## Príloha D: Ukážka webových stránok na mobilnom telefóne

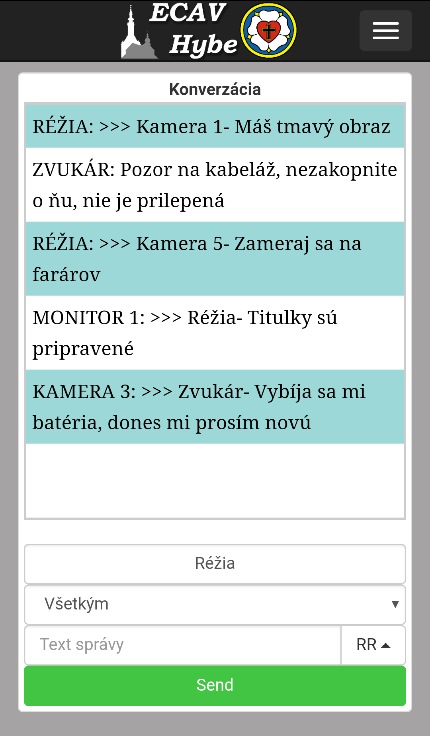
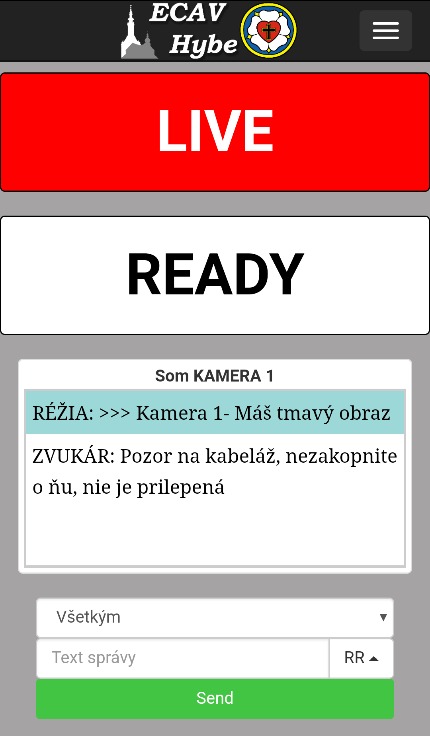
**Nočný dizajn**

## C:\Users\klein\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20170419-090855.png

*Stránka pre kameramana*

*Hlavná prehľadová stránka*

*Konverzačná stránka*

**Denný dizajn**

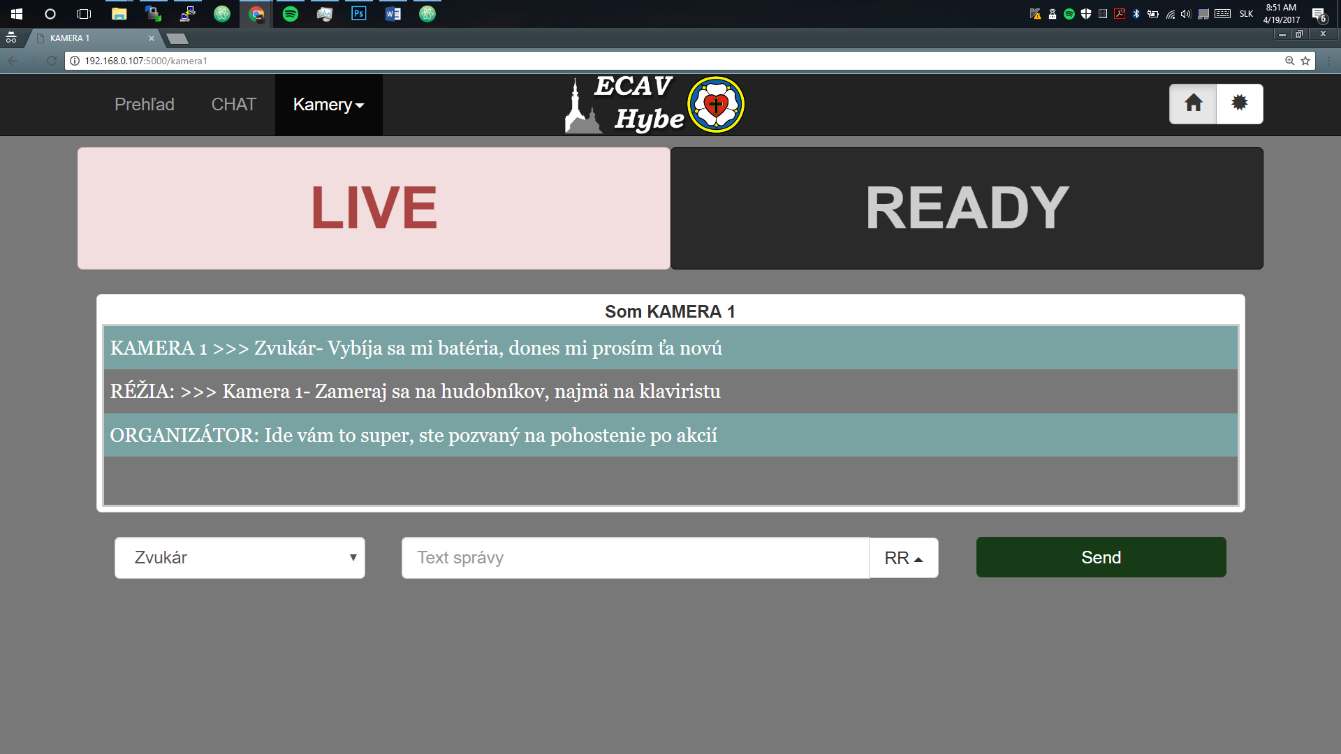
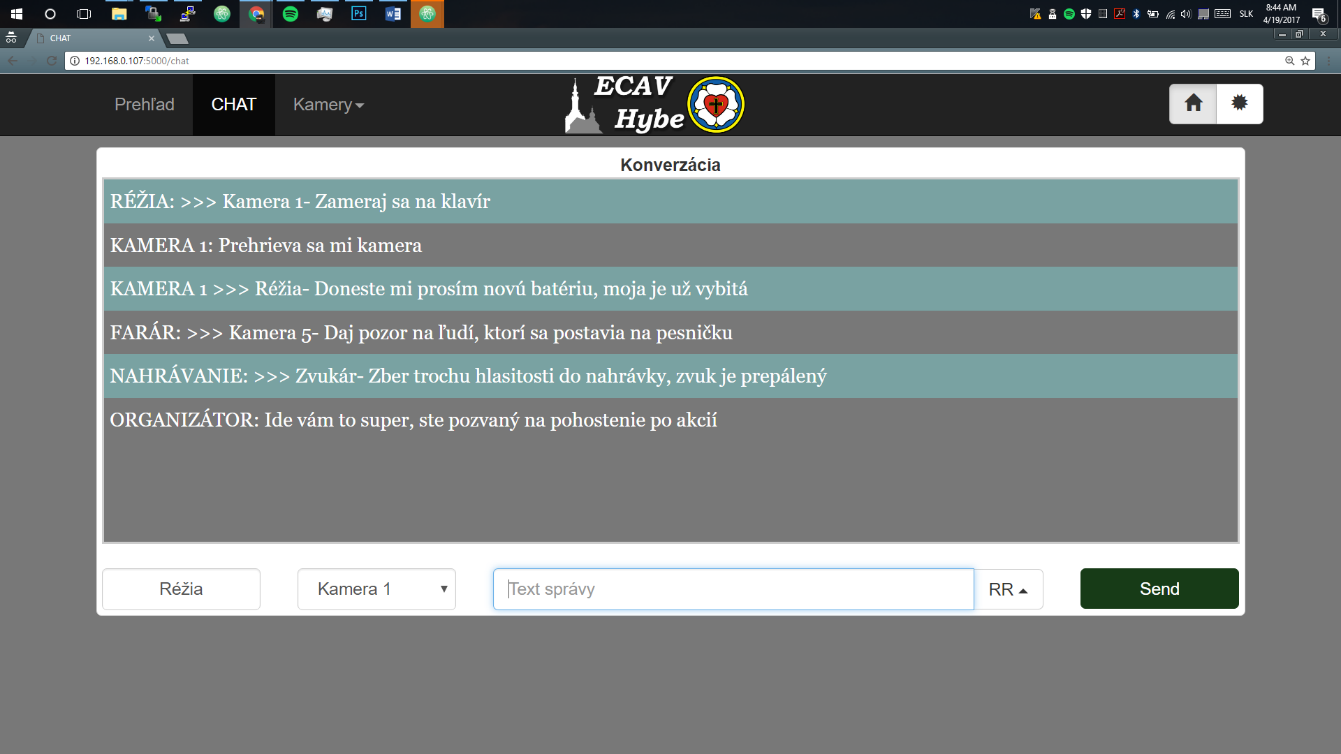
## C:\Users\klein\AppData\Local\Microsoft\Windows\INetCache\Content.Word\Screenshot_20170419-090908.png

*Stránka pre kameramana*

*Konverzačná stránka*

*Hlavná prehľadová stránka*

## Príloha E: Ukážka webových stránok na prenosnom počítači – nočný dizajn



*Konverzačná stránka*

*Stránka pre kameramana*

*Hlavná prehľadová stránka*

## Príloha F: Ukážka webových stránok na prenosnom počítači – denný dizajn

*Stránka pre kameramana*

*Konverzačná stránka*

*Hlavná prehľadová stránka*