

Inteligentní domovní systém

Autor

Bc. Tomáš Moravec

Vedoucí

doc. Ing. Josef Chaloupka, Ph.D.

Dobrý den, jmenuji se Tomáš Moravec a rád bych Vám představil svůj diplomový projekt, zabývající se komplexním řešením zabezpečovacího zařízení pro domácnost, pod vedením pana Chaloupky.

Ústředna



Ústředna je „mozek“ celého systému, který je propojen s ostatními prvky systému kabely nebo bezdrátově. Obstarává komunikaci mezi jednotlivými komponenty systému, má v integrované paměti uložené nejdůležitější nastavení. V závislosti na připojených komponentech pak může různě reagovat na splnění sledovaných podmínek. Často bývá vybavena jenom tím nejnutnějším pro vyvolání poplachu, to ať už akustického (siréna), nebo tichého (informování majitele)

Ústředna



Takto například může vypadat vnitřek typické ústředny.

Ovladač



Ovladač je prvek, který slouží k ovládání a případně též k programování ústředny. Dnešní alarmy je možné ovládat několika způsoby.

Ovladač



Jako ovladač se nejčastěji používá klávesnice vybavená tlačítky, případně čtečkou (čipovými kartami a přívěšky) nebo též dálkové ovládání. U některých systémů se dá přes klávesnici provést nastavení celého systému. Klávesnice slouží k zastřežení i odstřežení systému.

Ovladač



sms

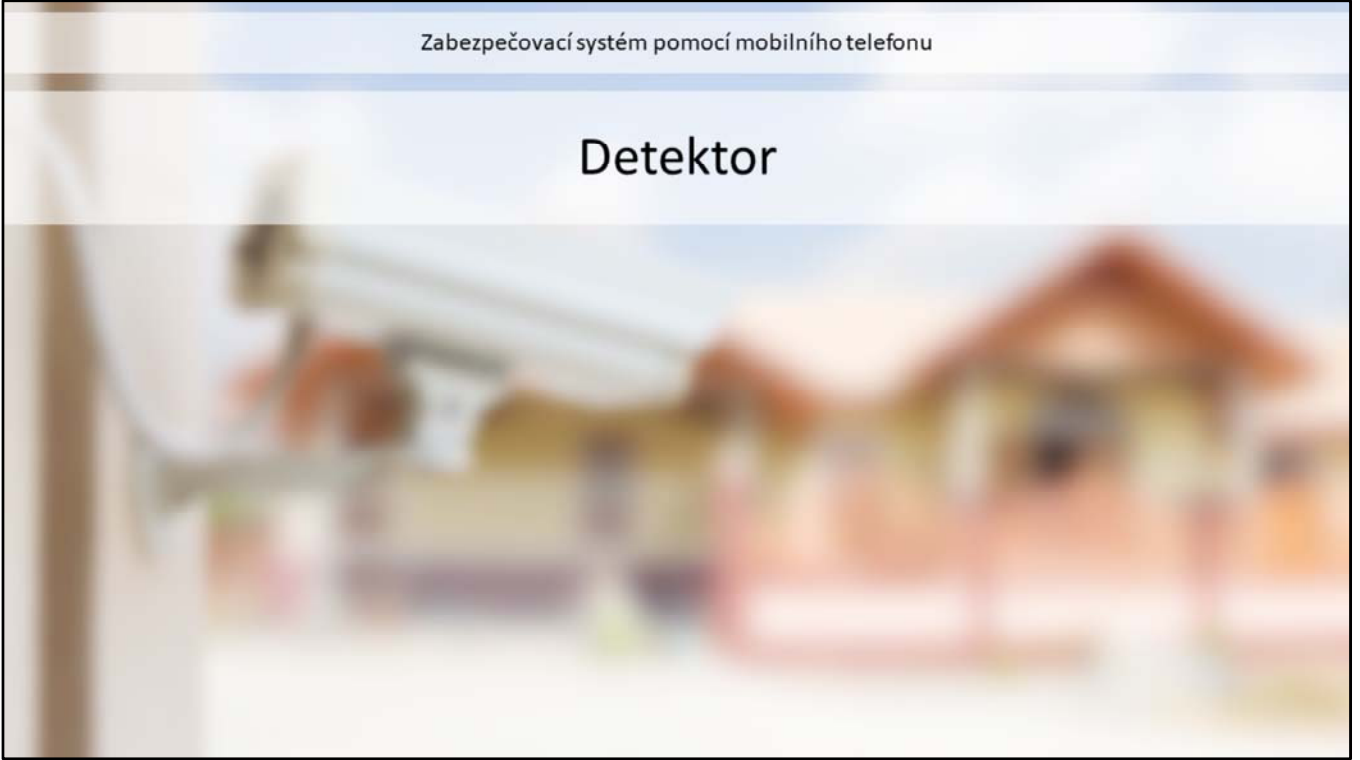
Dalšími způsoby je ovládání přes mobil (SMS příkazy).

Ovladač



Nebo například ovládání přes internet, kdy se většinou používá integrované webové rozhraní, ke kterému se uživatel může připojit po zadání hesla.

Detektor



Detektor je prvek systému, který je rozmístěn v hlídaném objektu a má za úkol reagovat aktivací při narušení (otevření, pohyb, rozbití atd.) a to tak, že tuto informaci předají ústředně, která ji následně zpracuje. Nejčastěji používané detektorové prvky jsou:

Detektor



Detektor pohybu (PIR detektor)

Detektor



Kamera

Detektor



Magnetický kontakt (dveřní čidlo)

Komunikátor



Komunikátor je zařízení, které odesílá informaci o narušení objektu, případně o odchylce od normálního provozního stavu zabezpečovacího.

Komunikátor



Komunikátor nemusí být vůbec přítomen, zařízení potom není schopné komunikovat s vnějškem a většinou pouze spustí poplach v domě, bez varování majitele, nebo bezpečnostní agentury.

Komunikátor



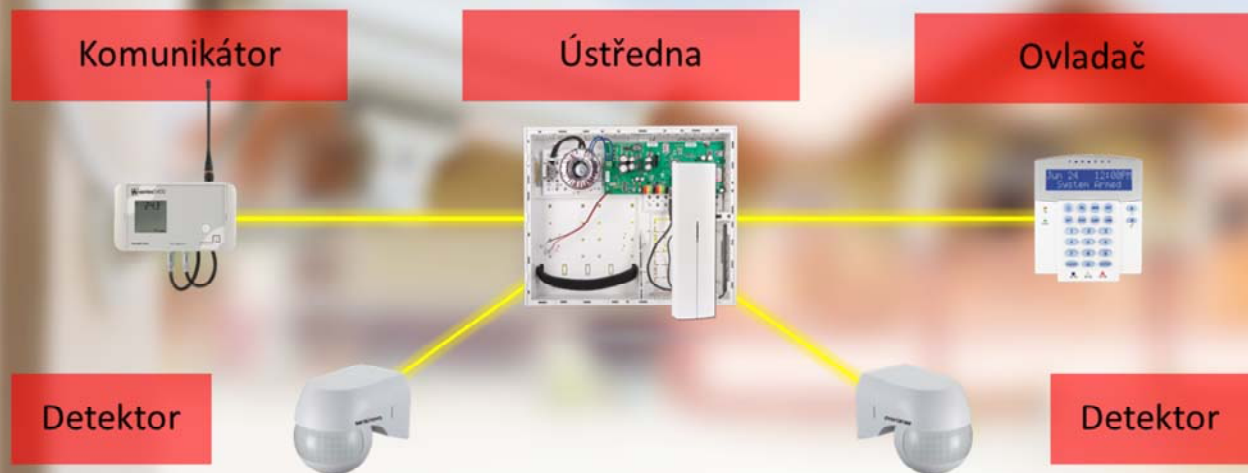
GSM komunikátor

Komunikátor



LAN komunikátor

Zabezpečovací systém



Takto může nakonec vypadat případné zapojení bezpečnostního systému. Velká část řešerše se věnuje právě volbě vhodných komponent a jejich následnému objednání.



Firmware ústředny (Arduino)

Hlavním účelem firmwaru je možnost kompletního řízení zabezpečovacího systému a umožnění komunikace, včetně poskytování API pro přístup aplikací. Software byl vyvinut pro platformu Arduino (čip ATmega) v programovacím jazyce C++ s nadstavbou Wiring (knihovna pro řízení hardwaru} za pomoci vývojového prostředí Visual Studio 2017 s rozšířením Visual Micro, které poskytuje zvýrazňování syntaxe knihovny Wiring a umožňuje komplikaci a nahrávání kódu přímo na desku Arduino.

Komunikační API

Dotaz	Parametry	Popis	Vrací
GetAllSensors()	/	Vrací list všech senzorů	Id, pin, name, state
SetSensorName()	id, name	Změna jména senzoru	OK, ERROR
AddSensor()	pin, name, type	Přidání nového senzoru	OK, ERROR
DeleteSensor()	id	Smazání senzoru	OK, ERROR
...

Firmware ústředny samozřejmě obsahuje velké množství funkcí a má velké množství schopností. Nejdůležitější je ale zřejmě komunikační API, které jsem sám vytvořil pro dotazování a odpovídání aplikacím. Detailní popis API najdete uvnitř mé práce, zde pro ukázkou mám pár příkladů. Například dotaz pro získání všech senzorů. Dotaz nemá žádné parametry a vrací list všech senzorů ve formátu JSON. Nastavovací dotaz SetSensorName má parametr ID a nové jméno zařízení. Vrací OK nebo ERROR v případě chyby při změně. Samotné ID zařízení je získáváno z GET příkazů a jejich správu si obstarává samotná ústředna, jedná se o jediný údaj který není možné měnit.

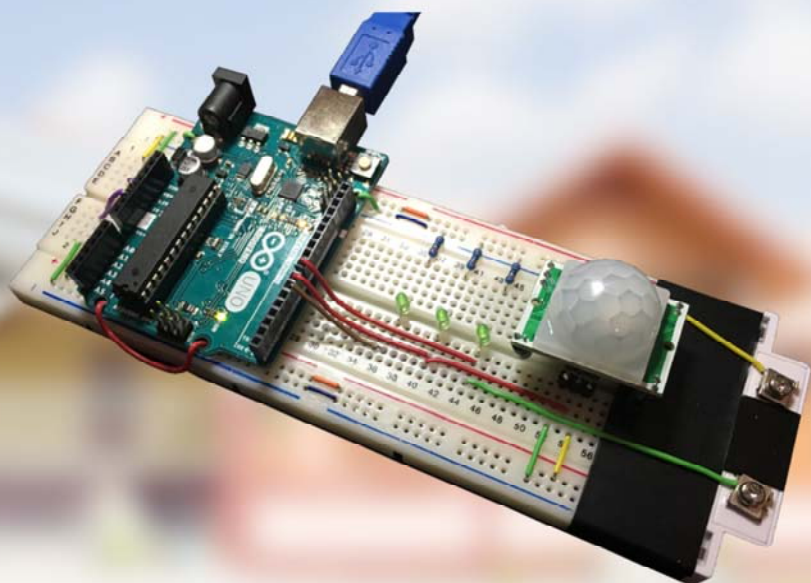


Vytvoření prototypu

Před začátkem práce na firmwaru bylo potřeba vytvořit testovací prototyp (proof of concept). Nejdříve jsem tedy navrhnul schéma zapojení.

Prototyp

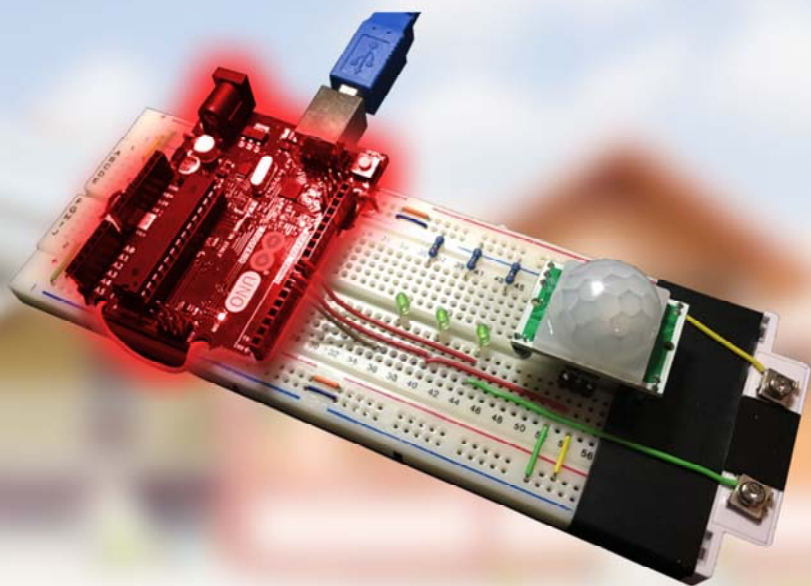
Prototyp
centrály



Na obrázku vidíte prototypové zařízení, zde bez připojeného GPRS modulu.

Prototyp

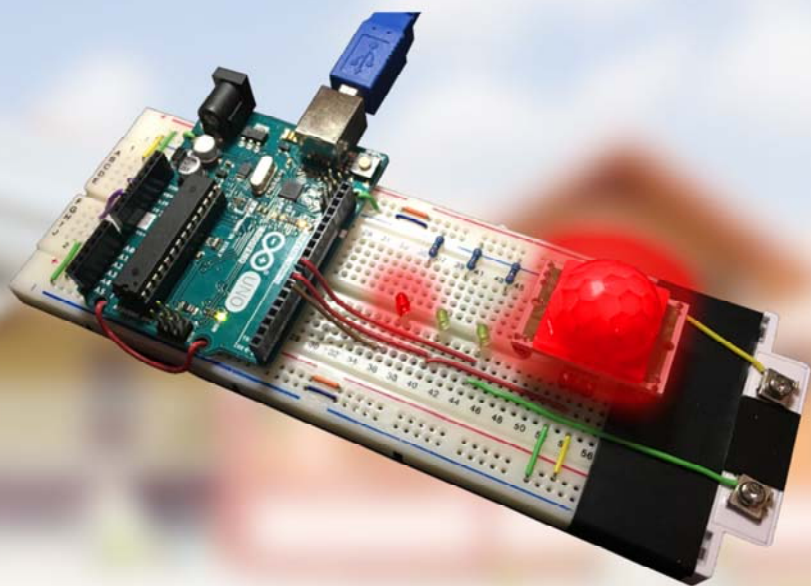
Řídící
jednotka



První se na desce vyskytuje řídící jednotka Arduino, pro kterou bylo nutné naprogramovat firmware.

Prototyp

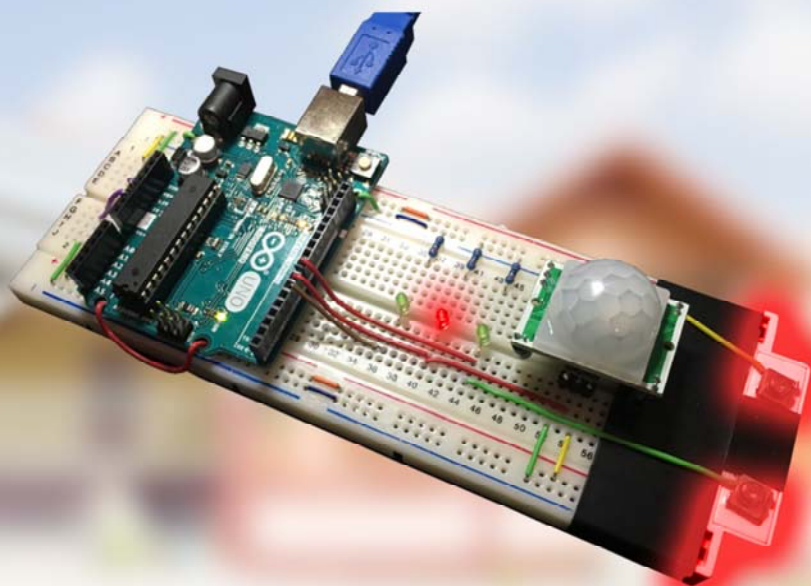
Senzor PIR
a kontrolní
dioda



Následuje senzor typu PIR a kontrolní dioda, která signalizuje že došlo k sepnutí čidla.

Prototyp

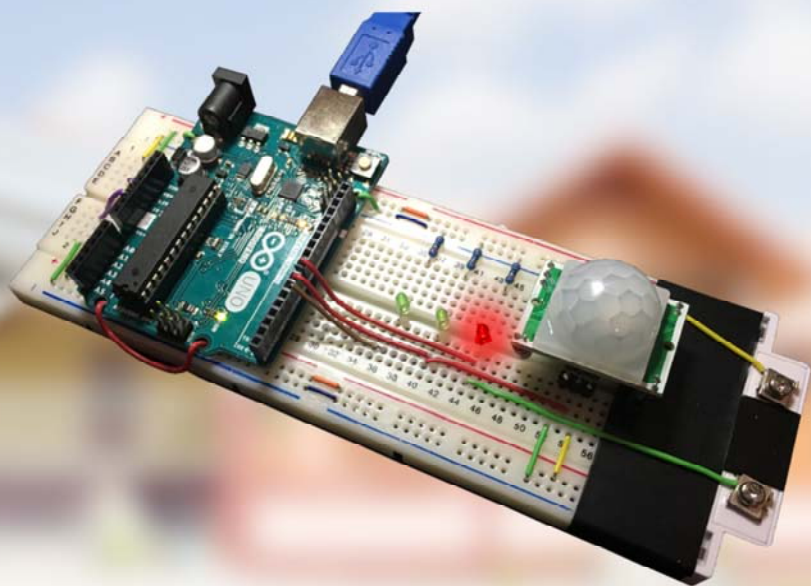
Dveřní čidlo
a kontrolní
dioda



Dále dveřní čidlo a jeho kontrolní dioda.

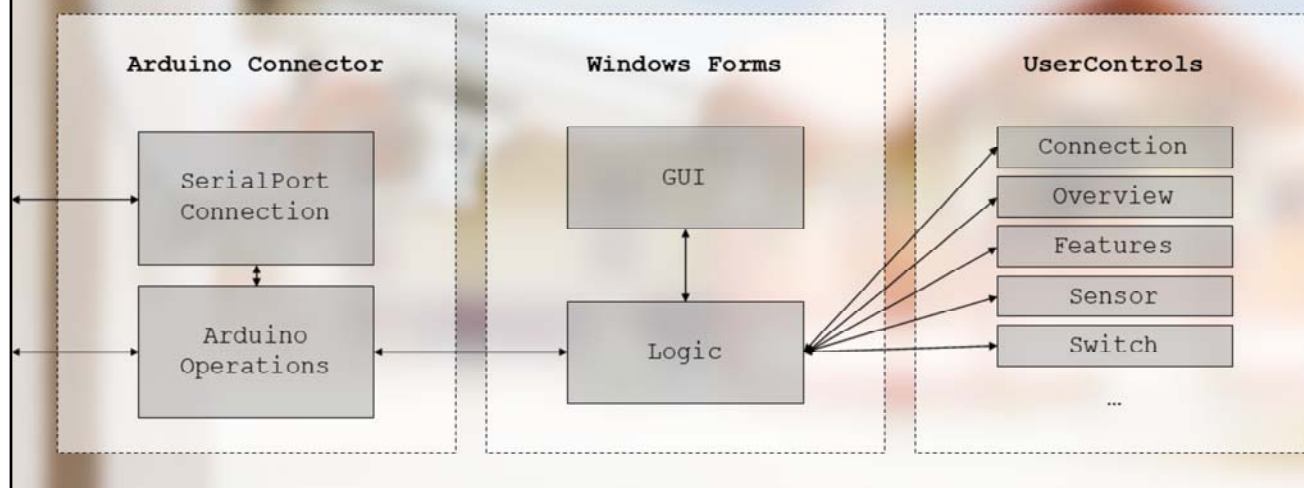
Prototyp

Spínací
prvek

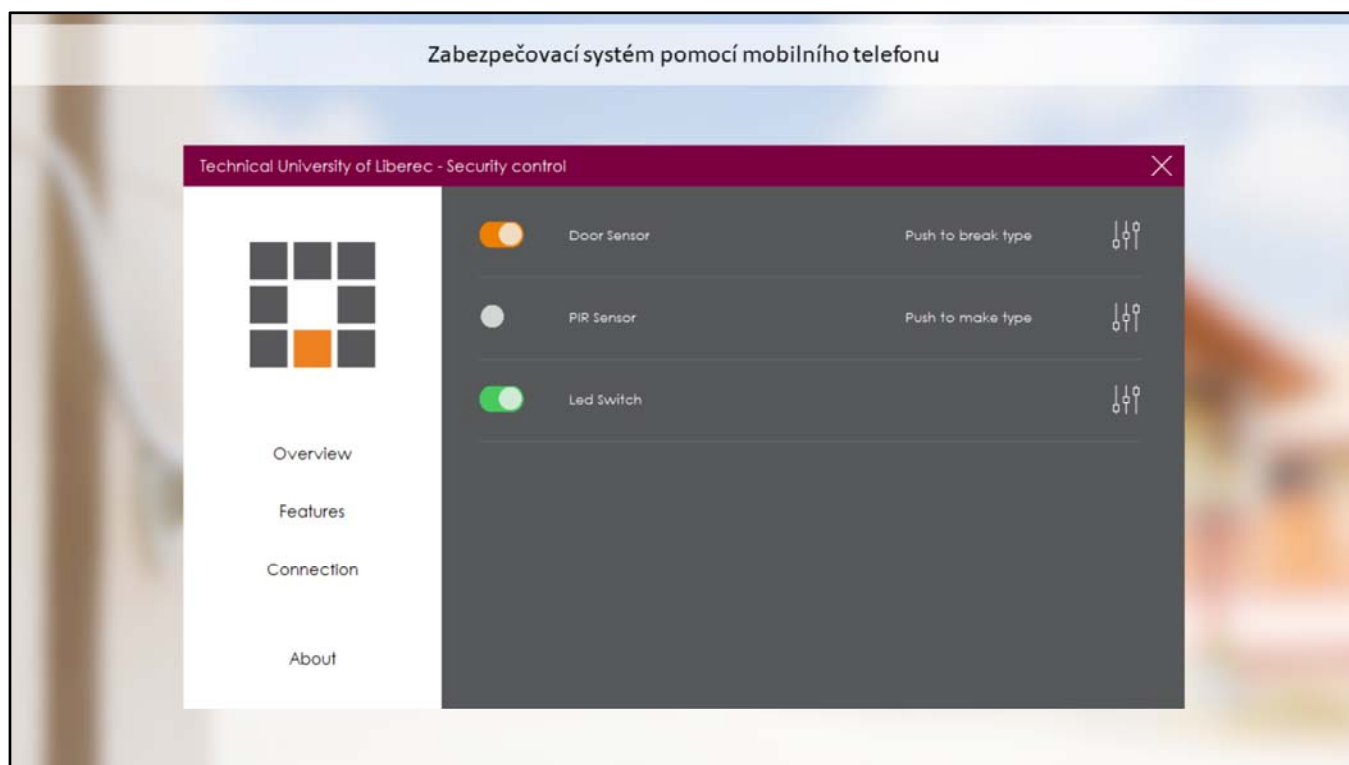


A nakonec samostatná dioda reprezentující spínací prvek.

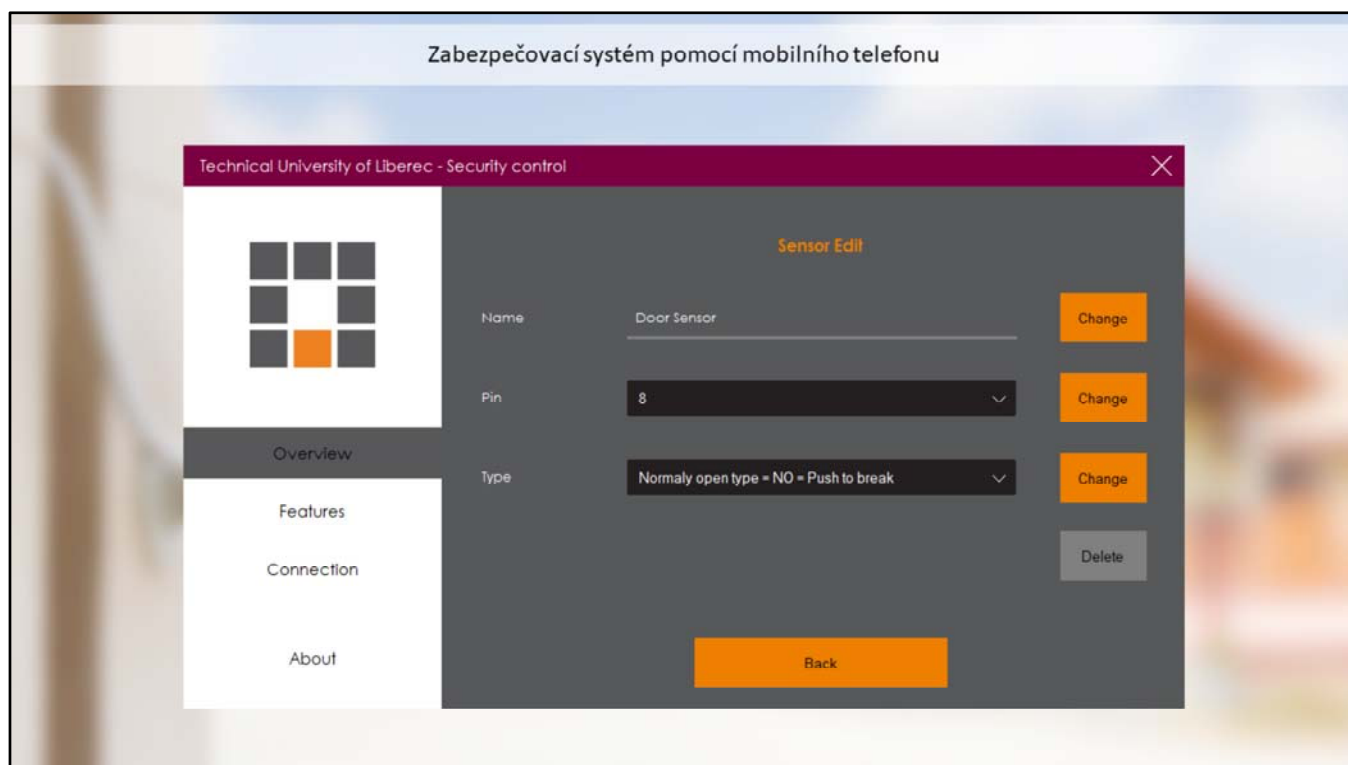
Blokové schéma desktopové aplikace



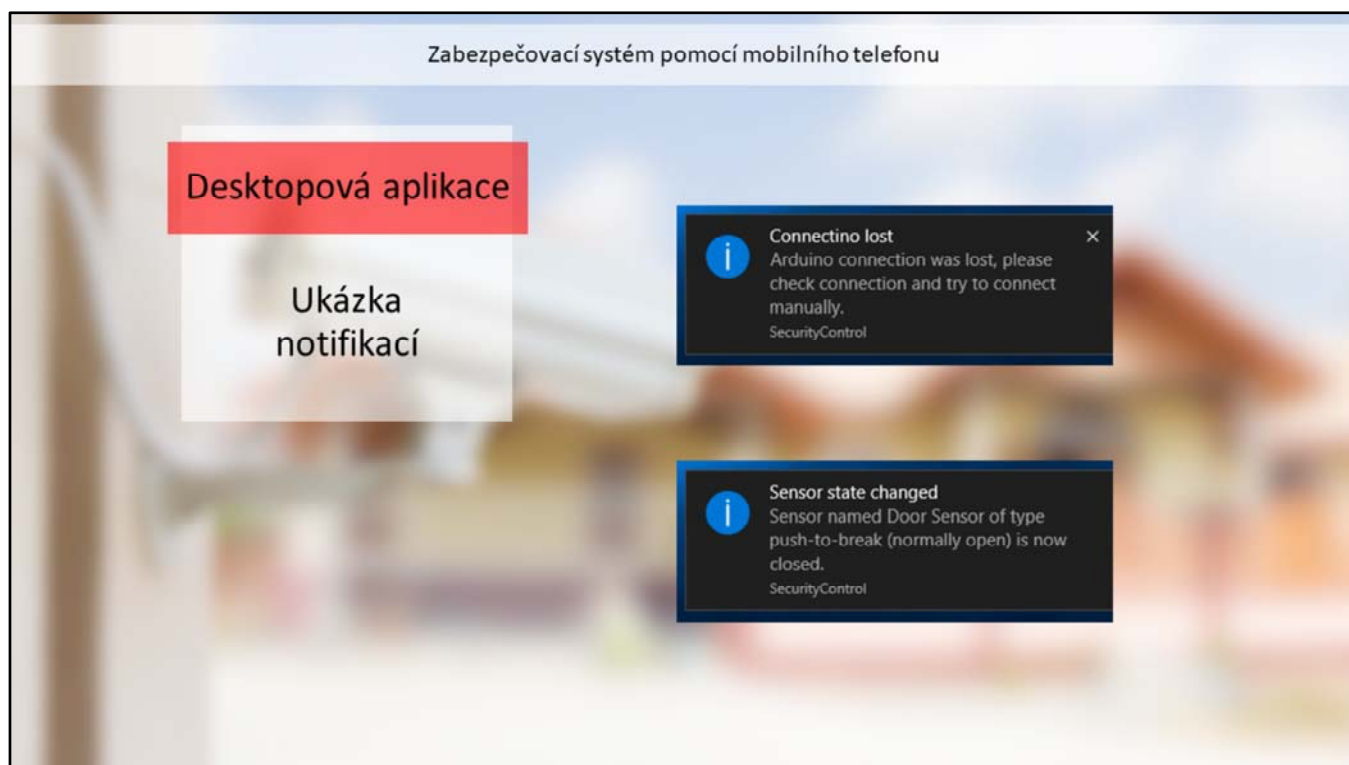
Kód aplikace je rozdělen do více bloků, které spolu vzájemně komunikují. Arduino Connector slouží k napojení zabezpečovacího zařízení a využívá API poskytované zabezpečovacím zařízením, díky čemuž poskytuje funkce pro jednoduché zapisování a čtení do Arduina. Windows Forms poskytují grafické uživatelské rozhraní, základní logiku na propojení jednotlivých funkčních celků a notifikace pro uživatele. UserControls poskytují jednotlivé funkce aplikace, z nichž každá má své vlastní GUI, které se zobrazuje ve Windows Forms. Dále budou zmíněny pouze ty UserControls, které jsou důležité pro pochopení ovládání aplikace. Design byl inspirovaný manuálem jednotného vizuálního stylu TUL



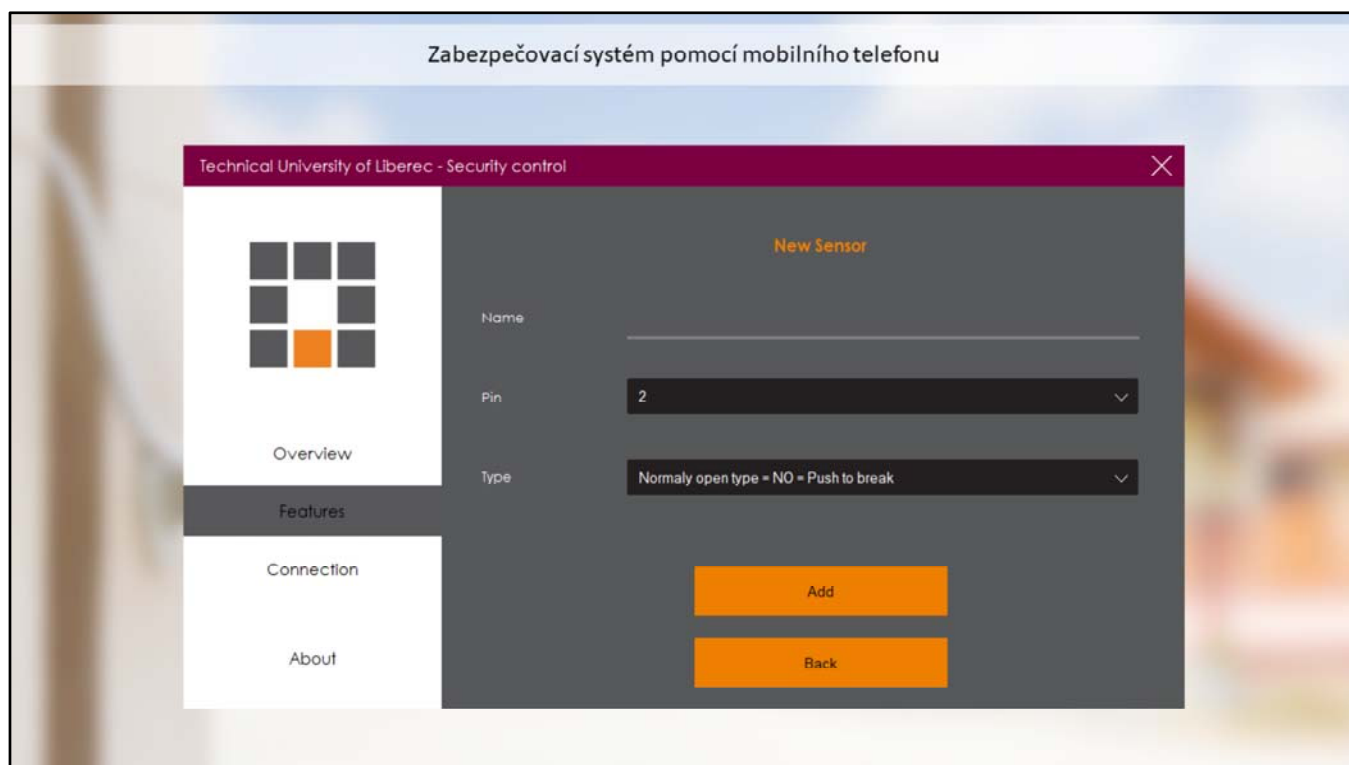
Zde vidíme již desktopovou aplikaci v hlavním okně přehledu připojených zařízení. Ty se načítají přímo z centrály a jsou vkládány pod sebe, vždy dle typu (tedy spínač nebo senzor). Řádky se generují automaticky a při přesažení okna je možné scrollovat níže. Stav spínacích prvků je možné měnit přímo hlavního okna, jednotlivá nastavení lze provádět pomocí tlačítka nastavení po pravé straně.



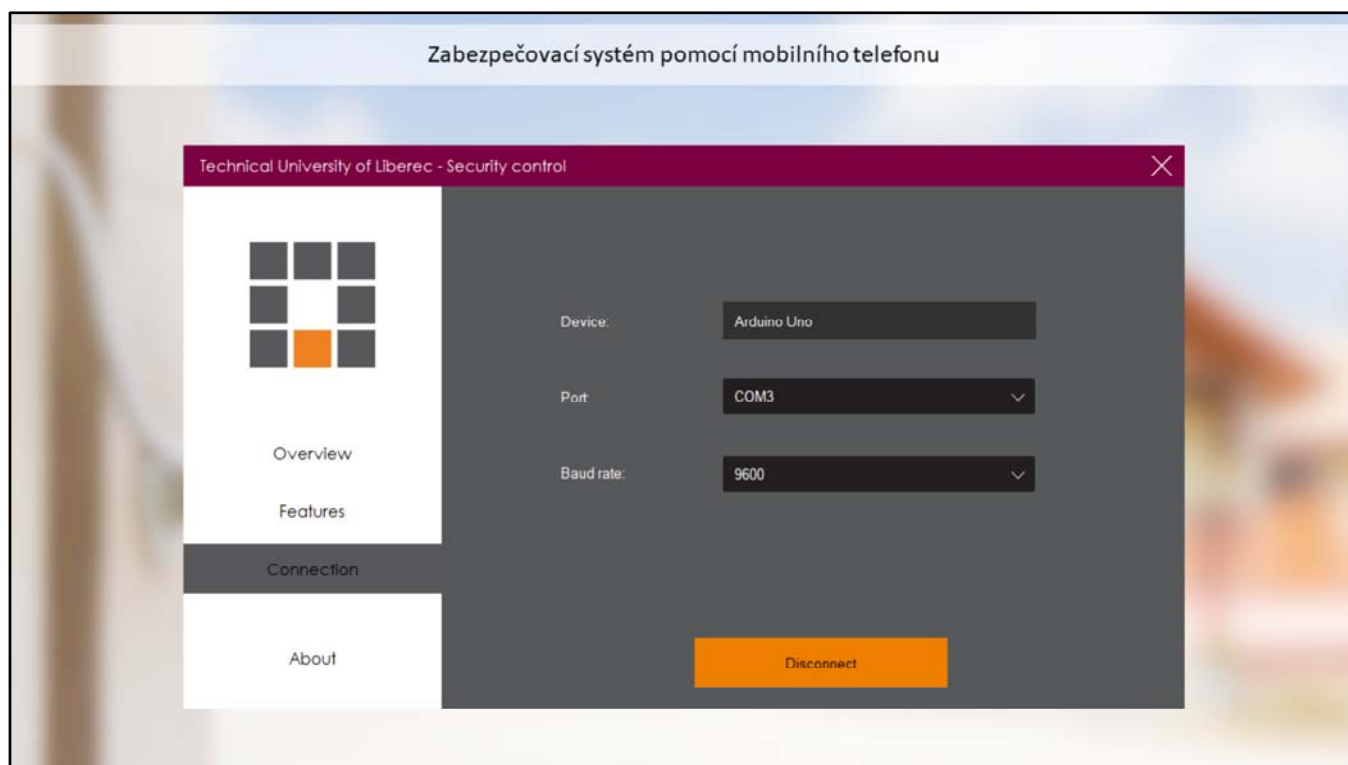
U nastavení existujících prvků lze měnit, v případě senzoru, jméno, pin a typ. Jméno má maximální délku 30 znaků, z důvodu omezené paměti Arduina. Pin lze vybrat ze seznamu, který byl vytvořen pro každé existující Arduino a obsahuje pouze piny, pro tento účel využitelné. Samotný typ zařízení je rozpoznávat při připojení. Typ senzoru může být buď spínací nebo rozpínací. Při změně jakéhokoliv nastavení se do centrály odešle požadavek, a centrála odpoví buď OK, po provedení operace, nebo chybovou hláškou.



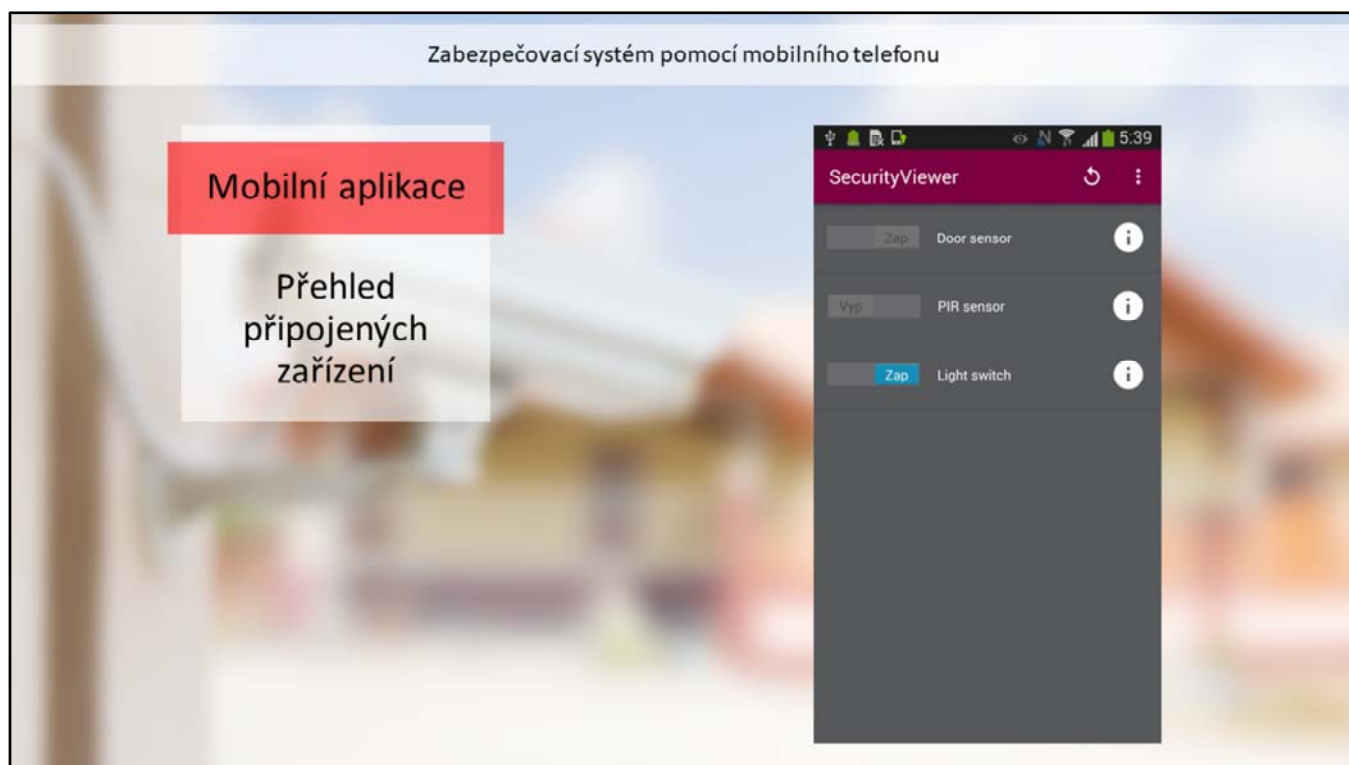
Pro informování uživatele o všemožných aktivitách aplikace byli využity notifikace Balloon. Uživatel je informován o připojení, odpojení, změnách nastavení nebo změně stavu senzoru. Seznam všech notifikací je možné nalézt uvnitř mé práce. Nejčastější notifikace se dají změnit nebo omezit v nastavení programu.



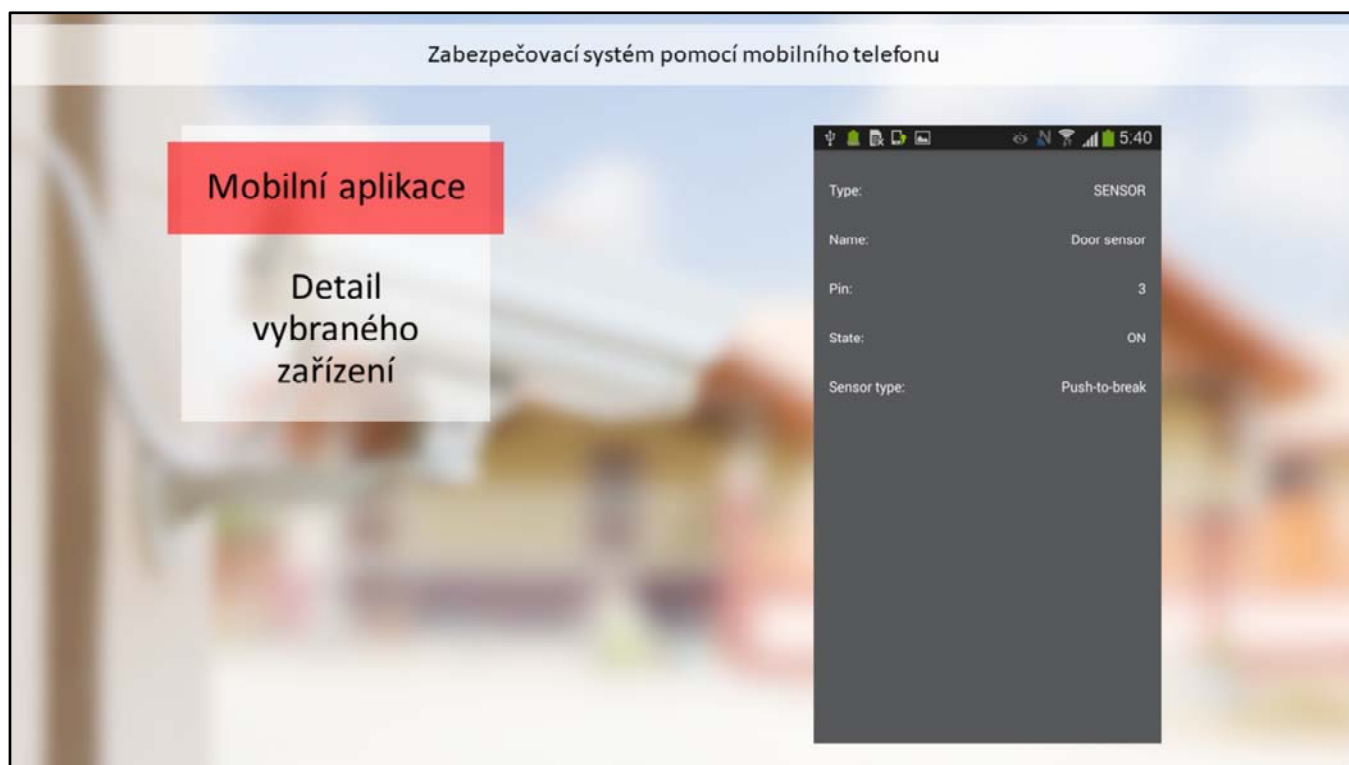
Pokud přidáváme nový senzor, je to podobné jako nastavení stávajícího. Zvolíme jméno, vybereme Pin a typ. Následně přidáme a jsem přesunuti na stránku přehledu, kde by již měl být náš nový senzor.



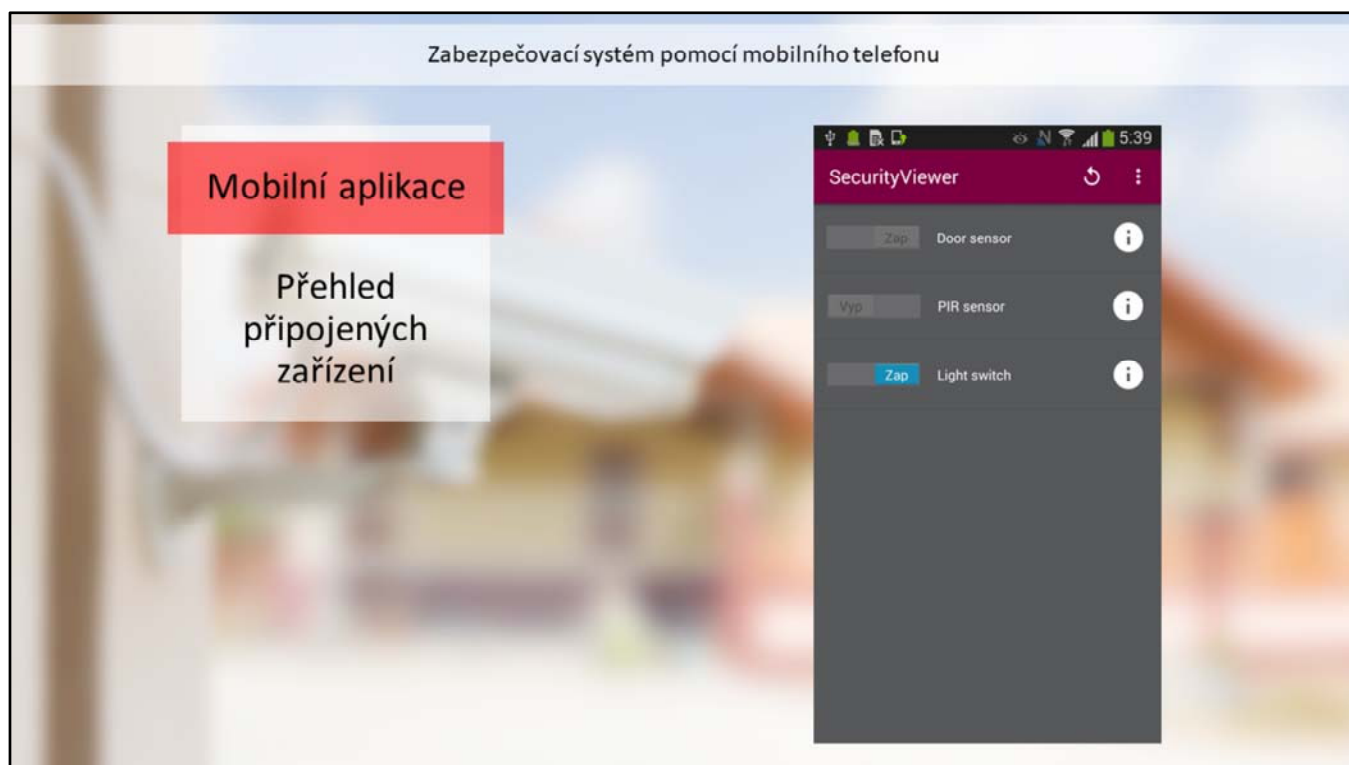
Připojování a odpojování je možné udělat ručně, nicméně aplikace si sama hlídá porty a pokud není žádné zařízení připojeno a objeví se centrála, automaticky proběhne připojení. Při ručním výběru portu se vždy pro přehlednost zobrazuje jméno zvoleného zařízení. V jednu chvíli je možné být připojený pouze k jedné centrále. Pro krajní nouzi připojení k více zařízením, je možné otevřít více oken aplikace.



Po dokončení desktopové aplikace následoval vývoj mobilní aplikace. Ta měla umožňovat pouze monitoring připojených periférií. Vývoj byl tedy značně jednodušší. Po připojení je stejně jako v desktopové aplikaci načten přehled zařízení. V případě spínače je možné měnit jeho stav. U každého zařízení lze zobrazit jeho detail, nicméně ten není možné měnit.



V detailu jsou zobrazeny všechny dostupné informace, které je možné zobrazit také v desktopové aplikaci.



V případě, že se změní stav některé z periferií, je uživatel upozorněn notifikace. Aplikace byla vyvinuta pouze pro systém Android.