

Projekt: GreenBoard
Implementacja: Python, Java, H5+JS*, Node.js*

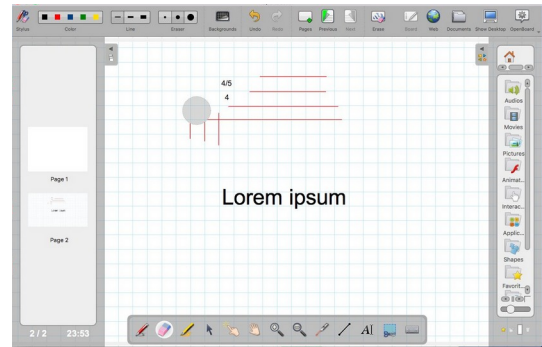
Opis:

Aplikacja do wykonywania adnotacji i zaznaczeń na ekranie oraz wykonywania szybkich graficznych prezentacji. Połączenie Notatnika (sticky notes), Painta i Powerpointa.

Wymagania:

- ◆ tryb notatek graficznych na płótnie (slajdzie)
- ◆ tryb desktop - adnotacji i zaznaczeń na Ekranie
- ◆ możliwość wykonania rzutu ekranu do pliku oraz do slajdu
- ◆ zmiana płótna slajdów, ukrywanie/pokazywanie slajdów
- ◆ rysowanie odręczne, rysowanie kształtów, pisanie tekstów/wzorów (w wybranych kolorach)
- ◆ zaznaczanie elementów na slajdach
- ◆ automatyczny zapis pracy, eksport do pliku (jpg, png, svg, pdf)
- ◆ konfiguracja skrótów klawiszowych do narzędzi programu
- ◆ przenoszenie paneli z narzędziami
- ◆ import plików graficznych i tekstowych do slajdów
- ◆ udostępnianie ekranu w sieci LAN

**czy jest to możliwe, przy zachowaniu wszystkich warunków?*



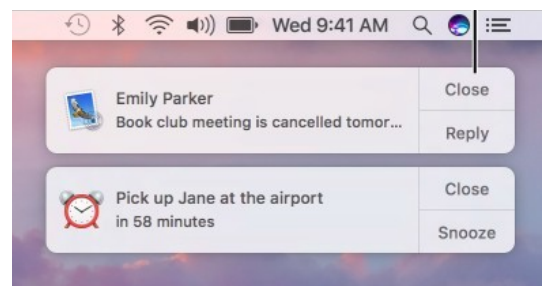
Projekt: Notifier
Implementacja: Klient: Python, Java;
Serwer: Node.js, JavaScript, PHP, Java, Python

Opis:

Aplikacja do odbierania komunikatów z serwera oraz wyświetlania powiadomień (w zasobniku powiadomień systemowych oraz w obrębie ekranu). Pozwalając powiększyć okno wiadomości, oznaczyć wiadomość jako przeczytaną przez klienta.

Wymagania:

- ◆ aplikacja startuje wraz z systemem i nie można jej wyłączyć
- ◆ integracja z zasobnikiem systemowym (ikona w SysTray) oraz z powiadomieniami systemowymi
- ◆ wyświetlanie wiadomości jako notyfikacji oraz w dedykowanym oknie (w tym fullscreen)
- ◆ wiadomości pobierane z serwera, w formacie HTML lub RTF
- ◆ wgrywanie wiadomości przez plik tekstowy, HTML (lub obrazek dla komunikatów fullscr)
- ◆ zgrywanie kopii wiadomości oraz dzienników zdarzeń (odczytów wiadomości u klientów)
- ◆ możliwość przydzielania kategorii oraz priorytetu wiadomości
- ◆ lista aktywnych w sieci LAN klientów
- ◆ możliwość komunikacji (czat) użytkowników aplikacji w obrębie sieci LAN
- ◆ możliwość przesyłania plików pomiędzy użytkownikami sieci LAN



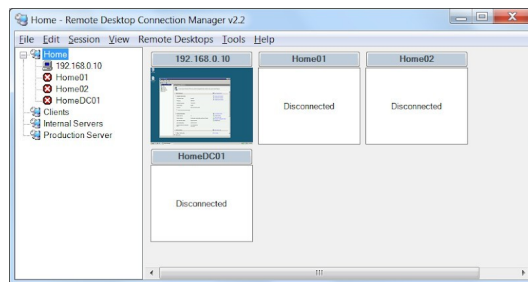
Projekt: Teacher Desk
Implementacja: Klient: Python, Java, Node.js;
Serwer: Node.js, JavaScript, PHP, Java, Python

Opis:

Aplikacja do zarządzania salą/salami komputerowymi w sieci LAN

Wymagania:

- ◆ widok układu sali komputerowej/pięter/budynku
- ◆ lista aktywnych w sieci/podsieci LAN klientów (podział na sale komputerowe)
- ◆ podgląd pulpitu klienta oraz udostępnianie widoku nauczyciela (tryb prezentacji)
- ◆ lokalna baza danych /plik zawierająca spis i konfigurację zdalnych połączeń
- ◆ przełączanie widoku pomiędzy listą, miniaturami, widokiem sali, a pełnym ekranem
- ◆ wyświetlanie statusu połączenia, adresu IP, nazwy hosta, danych użytkownika,
- ◆ ikony sterujące zdalnym pulpitem: pełny ekran, wiadomość, zdalne polecenie (wł/wył PC),
- ◆ wizualna kontrola nad umiejscowieniem miniatur
- ◆ możliwość komunikacji (czat) użytkowników aplikacji w obrębie sieci LAN/sali
- ◆ możliwość przesyłania plików pomiędzy użytkownikami sieci LAN/sali
- ◆ import/eksport ustawień i dzienników zdarzeń

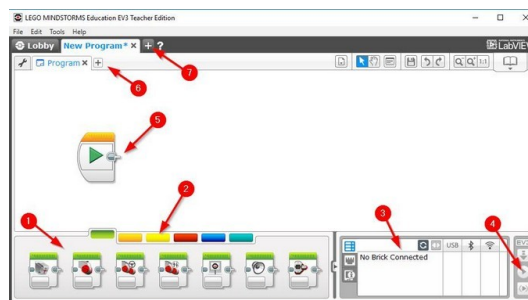


Projekt: Mindstorm Workspace
Implementacja: Klient/Serwer:
Python, Java, PHP, Node.js

Opis: klon programu LEGO EV3 Mindstorms pozwalającej na wykonywanie podobnych czynności, lecz mocno ukierunkowanej na współpracę podczas układania programu w zespołach, a zamiast wysyłania ułożonego programu do robota, uruchamianie wizualnej symulacji robota wykonującego zadania. Aplikacja pozwala stworzyć program poprzez wizualne układanie bloków.

Wymagania:

- ◆ przestrzeń do współpracy przy wizualnym układaniu puzzli / bloków programu,
- ◆ panele zawierające: bloki programowe, listę uczniów / zespołów, scenariusz zadania,
- ◆ możliwość przesłania prowadzącemu ułożonego programu do wgrania do robota,
- ◆ interpretację bloków programu poprzez uruchomienie symulacji,
- ◆ wgranie wizualnego środowiska dla symulacji



Projekt: PsSST: Problems Sharepoint Solution Tracker
Implementacja: Klient/Serwer: H5+JS+PHP+*SQL, Node.js, Python, Java

Opis: Aplikacja do zgłaszania problemów i usterek odnośnie sali i urządzeń IT w szkole/firmie.

Wymagania:

- ◆ centralny serwer udostępniania i gromadzenia danych
- ◆ możliwość dokonania zgłoszenia z każdego urządzenia, czy systemu operacyjnego
- ◆ import/eksport bazy danych, drukowanie raportów ze zgłoszeń i napraw
- ◆ podpowiadanie rozwiązań, baza wiedzy
- ◆ rozwiązania w formie kreatora-instrukcji krok po kroku z pytaniami, czy pomogło
- ◆ wysyłanie powiadomień o zgłoszeniach do administratora
- ◆ panel zarządzania zgłoszeniami, planowanie prac, oznaczanie zgłoszeń kategoriami
- ◆ generowanie kodów QR dla każdej sali