05.04.2020

Projekt wstępny:
System komunikacji dla aplikacji kontroli dostępu do pomieszczeń

Realizują:
- **Zuzanna Żak**
- **Kacper Kipa - Lider**
- **Maciej Burda**
- **Marcin Kiliański**

1. Treść zadania:

System komunikacji pomiędzy serwerem, a zbiorem kart i czujników otwierających drzwi w danej firmie.
Każdy pracownik posiada swoją własną kartę, która ma zapisane określone prawa dostępu do danych części budynku. Sterowniki uwierzytelniają każdorazowe użycie karty i sposób autoryzacji użytkownika, a także przesyłają te dane do serwera, który kontroluje oraz obsługuje takie rzeczy jak: liczba kart, liczba czujników czy prawa dostępu dla każdego użytkownika. Administrator ma możliwość dodawania kart i czujników, a także ustawiania praw dostępu dla konkretnych pracowników. System ponadto obsługuje takie funkcje jak:

- wykrywanie linii papilarnych użytkowników
- wykrywanie prób potencjalnego włamania:
  - zbyt krótkie odstępy pomiędzy użyciami danej karty
  - niezgodność użyć karty względem pomieszczeń
  - próba użycia karty przy czujniku, do którego użytkownik nie ma praw dostępu
  - wielokrotne porażka w uwierzytelnieniu użytkownika
- monitorowanie zagęszczenia pracowników w danych sektorach budynku
- wykrywanie awarii czujników i przekazywanie tej informacji za pomocą ogólnodostępnego systemu głosowego zamontowanego w budynku wszystkim pracownikom bądź poinformowanie jedynie pracowników ochrony o zaistniałym wydarzeniu, w zależności od tego co było przyczyną wystąpienia awarii

2. Nazwa własna projektowanego systemu:

StuSec - Student Security

3. Przyjęte założenia funkcjonalne i niefunkcjonalne:

Wymagania funkcjonalne:

- Każda karta i czujnik będą posiadały unikalne identyfikatory.
- Czujnik będzie autoryzował użytkownika i w rezultacie pozwalał albo nie pozwalał mu przejść przez dane drzwi.
- System udostępni odpowiednie narzędzie do wglądu w aktualną liczbę pracowników przebywających na terenie firmy oraz ich rozmieszczenie.
- Użytkownik o prawach administratora będzie miał możliwość dodawania i usuwania kart oraz czujników.

- System będzie generował różnorakie ostrzeżenia skierowane do różnych grup osób, w zależności od problemu który wystąpi i jego wagi.

Wymagania niefunkcjonalne:

- czujnik będzie weryfikował użytkowników w czasie nie dłuższym niż 3 sekundy
- system oraz czujniki będą odporne na standardowe ataki i próby podszycia się pod któreś z urządzeń
- system nie będzie udostępniał prywatnych, wewnętrznych danych nikomu kto nie posiada właściwych uprawnień

4. Podstawowe przypadki użycia:

Dodanie karty:

1. Administrator loguje się do systemu.
2. Administrator wprowadza do systemu identyfikator nowej karty.
3. System weryfikuje czy dana karta już istnieje, jeśli nie, zostaje ona dodana do bazy kart. Gdyby jednak administrator kilka razy wprowadzał błędny identyfikator karty, system wysłałby informacje do ochrony o potencjalnej próbie włamania.
4. Karta po pierwszym użyciu przez użytkownika (wprowadzeniu jego hasła, zeskanowaniu linii papilarnych) zapamiętuje wprowadzone dane, które później będą używane do weryfikacji właściciela.

Wykrycie awarii:

1. System zauważa, że karty pewnego pracownika użyto w części budynku, w której owego pracownika nie ma (dedukowane przez system na podstawie “śluz zliczających”).
2. System wysyła ostrzeżenie do strażników, a w przypadku wysokiego ryzyka (drzwi o dużym stopniu ważności) pomieszczenia są odcinane – blokowany jest system przepuszczania osób.

Przepuszczenie użytkownika przez drzwi:

1. Pracownik podchodzi do czujnika i przykłada do niego kartę.
2. Czujnik prosi o wpisanie kodu i/lub podanie odcisku palca.
3. Sterownik weryfikuje dane otrzymane przez czujnik i jeśli są one poprawne to otwiera drzwi.

5. Wybrane środowisko sprzętowo-programowalne (systemy operacyjne, biblioteki programistyczne) i narzędziowe (testowanie, debugowanie):

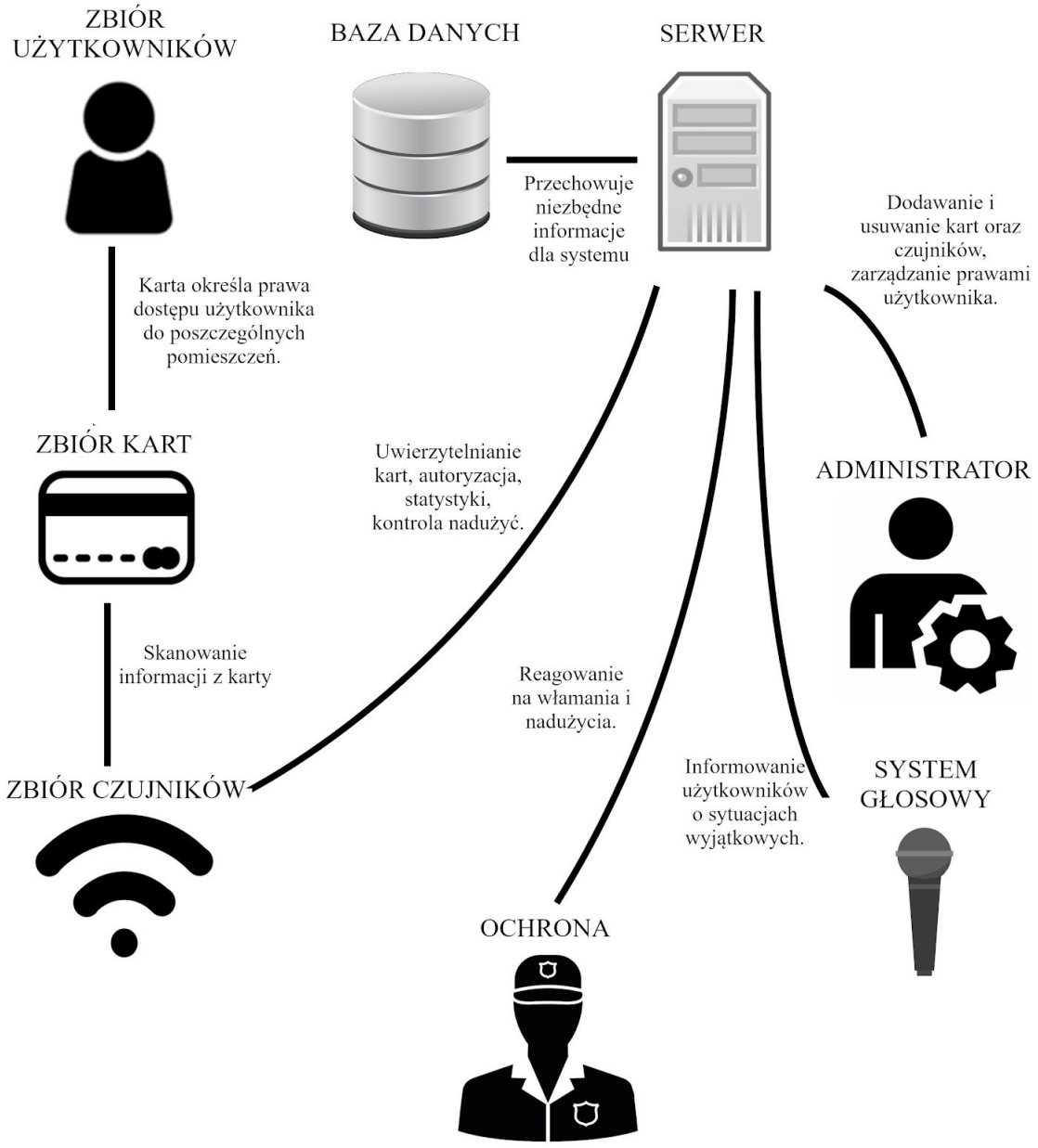
Docelowym środowiskiem użytkowym programu będzie system Linux. Program będzie napisany w języku C++. System będzie korzystał przede wszystkim ze standardowych bibliotek takich jak:

- biblioteki służącę do komunikacji (np. sys/socket.h, netinet.h, boost.asio) - będą wspomagały komunikację pomiędzy poszczególnymi modułami systemu (serwer, czujniki, itd.)
- biblioteki bazodanowe (np. OTL) - będą wspomagały bezpieczny dostęp systemu do baz danych zawierających m.in. uprawnienia, hasła, czytniki, ID kart)

- biblioteki I/O (np. iostream) - obsługują podstawowe czynności administratora
- biblioteki szyfrujące (np. libgcrypt) - odpowiadają za bezpieczne szyfrowanie danych

Testowanie odbywało się będzie za pomocą testów “boost” z wykorzystaniem podstawowej biblioteki C++. Testowana będzie przede wszystkim integralność wszystkich modułów poprawność pod względem logicznym (np. dostęp odpowiednich pracowników do odpowiednich części budynku oraz wykrycia próby nieautoryzowanego dostępu). Do debugowania wykorzystywane będzie narzędzie “GDB”.

6. Architekturę rozwiązania, tj. ilustrację i opis struktury logicznej systemu (koncepcyjnych bloków funkcjonalnych):



7. Ewentualnie API modułów stanowiących bloki funkcjonalne:

Głównymi blokami funkcjonalnymi będzie blok czujników, blok trzymający dane użytkowników (mała baza danych), a także blok będący głównym serwerem zarządzającym całością. Karty też będą stanowiły osobny blok jednak nie będzie on tak ściśle związany z resztą jak trzy bloki powyżej.

Komunikacja pomiędzy blokiem czujników a blokiem kart i serwerem:

Czujnik będzie magnetyczny i po wyczuciu zbliżenia do niego karty będzie odczytywał klucz (identyfikator) znajdujący się na karcie, a następnie analizował zapisane w nim informacje. W zależności od ważności czujnika (ważności drzwi z którymi jest powiązany), program będzie dodatkowo prosił użytkownika o podanie kodu i/lub przyłożenie palca do czytnika, w celu weryfikacji odcisku palca. Wszystkie dane, uzyskane powyżej, przez czujnik będą postaci tekstowej. Następnie sterownik będzie je szyfrował oraz uwierzytelniał, co pozwali mu zdecydować czy pracownik może czy nie może przejść. Dodatkowo zaszyfrowane dane, zostaną przesłane do modułu serwera odbierającego wszystkie zapytania o przejście przez drzwi pracowników. Osobny moduł będzie łączył się z bazą danych i analizował otrzymane dane, generował statystyki oraz dodawał informacje przesłane przez czujnik do zbioru danych używanych do weryfikacji zgodności zachowań w budynku. W przypadku gdy administrator doda albo usunie kartę z systemu, informacja ta będzie przesyłana do wszystkich czujników, które będą aktualizowały swoją pamięć.

Komunikacja pomiędzy blokiem kart a serwerem:

Jedynym innym przypadkiem komunikacji karty z serwerem, będzie dodawanie i unieważnianie karty. Operacje te wykona administrator przy pomocy swojego osobistego czujnika, za pomocą którego, po podaniu kodu i odcisku palca przez pracownika oraz danych administratora (ponownie interpretowanych jako dane tekstowe) zostanie dokonana autoryzacja karty do określonych wcześniej przez administratora zasobów. Administrator nie będzie potrzebował pracownika ani jego karty w przypadku unieważnienia karty.

Komunikacja serwera z bazą danych:

Serwer będzie jedynym komponentem struktury łączącym się z bazą danych. Będzie się to odbywało przy pomocy osobnego modułu współpracującego ze wszystkimi modułami serwera, które potrzebują dostępu do danych np. moduł odbierający zapytania z czujników czy moduł dodający kartę przez administratora.

Komunikacja serwera z “światem zewnętrznym”:

Serwer będzie również posiadał moduł odpowiedzialny za nadzorowanie bezpieczeństwa w firmie, który na podstawie przekazywanych zapytań z czujników będzie dedukował położenie pracowników, czas reakcji poszczególnych komponentów systemu, potencjalnie wykryte próby włamania, a także będzie kontrolował czy nie doszło do awarii systemu. Uzyskane informacje będą wysyłane do pracowników ochrony albo do systemu głosowego zamontowanego w budynku za pomocą radiowęzła.

8. Ewentualnie listy komunikatów z określeniem nadawców i odbiorców:

Lista komunikatów (wstępna) z:

- Serwera do:
    - czujnika:
        - “Alert! Blokada drzwi” - zablokowanie w przypadku wykrycia nadużycia drzwi o wysokim poziomie
        - “Odmowa dostępu” - brak uprawnień pracownika
    - ochrony:

        - “Próba włamania przy drzwiach ###” - powiadamia o nietypowym zachowaniu
    - administratora:
        - “Pozytywnie zakończone dodawania nowego pracownika”
        - “Błąd nr ### podczas dodawania nowego pracownika”
        - “Nietypowe zachowanie przy drzwiach nr ###” - np. zbyt duża liczba użyć (może być spowodowane błędem systemu bądź czujnika)
    - systemu głosowego:
        - “Awaria drzwi nr ###”
- Czujnika do:
    - serwera:
        - “Próba nieautoryzowanego przejścia pracownika *** przez drzwi ###”
        - “Przejście pracownika *** przez drzwi ###”

9. Sposób testowania:

Do systemu zostanie na początku dodana pewna pula kart (liczba pracowników), a także plan budynku, na podstawie którego system zbuduje korelacje pomiędzy pomieszczeniami i ważnością dostępu do nich. Następnie zostaną przygotowane pliki testowe, które będą zawierały imitacje działań systemu takie jak: kolejne komunikaty wysyłane przez czujniki, dodanie kart, unieważnienie kart itd. Na podstawie tych testów będą sprawdzane reakcje systemu, w postaci wywołań komunikatów z serwera na ekran, które będą również zapisywane w pliku z rezultatami.

10. Sposób demonstracji rezultatów, tj. scenariusze testów akceptacyjnych do zaprezentowania przy odbiorze projektu:

Scenariusz 1. Poprawne działanie wszystkich pracowników:

1. Serwer nie wykrywa żadnego zagrożenia i stale informuje o użyciu danych drzwi przez pracowników.
2. Na konsoli wyświetlają się komunikaty informujące o identyfikatorze czujnika i identyfikatorze karty.

Scenariusz 2. Wykrycie potencjalnej próby włamania:

1. Serwer podczas pracy wykrywa kilkukrotną błędną weryfikację kodu i/lub odcisku palca użytkownika.
2. Serwer sprawdza poziom ważności drzwi, związanych z czujnikiem, przy którym wystąpiły nieudane próby uwierzytelnienia (sprawdzenie potrzebne do wywnioskowania odpowiedniej reakcji).
3. W przypadku wysokiego poziomu ważności karta jest unieważniana, a do pracowników ochrony zostaje wysłany komunikat ostrzegający.

Scenariusz 3. Awaria czujnika/drzwi:

1. Serwer wykrywa, że nie ma kontaktu z czujnikiem.
2. Serwer wysyła informację do systemu głosowego związanego z niesprawnością drzwi.

11. Podział prac w zespole:
Podział nie jest jeszcze precyzyjnie ustalony, natomiast dokładny przydział zadań będzie na pewno widoczny w plikach na githubie gdzie przy każdym pliku źródłowym będzie określony autor.

12. Harmonogram prac:
Wstępne założenia:
20.04 - Wstępna implementacja serwera, administratora zarządzającego kartami (dodawanie, usuwanie) oraz bazy danych przechowującej dane.
11.05 - Implementacja czujników oraz modułu odpowiedzialnego za weryfikację użycia karty.
26.05 - Dodanie pozostałych funkcji takich jak: kontrola bezpieczeństwa przez serwer, kontrola nadużyć, wykrywanie sytuacji niedopuszczalnych (fizycznie niemożliwe przechodzenie między pomieszczeniami), wysyłanie powiadomień do pracowników ochrony czy systemu głosowego.

13. Adres projektu na serwerze kontroli wersji:
https://github.com/Kicper/TIN_Cpp