***#30 Marzenie laboranta***

*Żaneta Mielczarek*

*Tomasz Kukuczka*

*Szymon Bartyzel*

*Łukasz Michalik*

1. Treść ogólna zlecenia:

system do zarządzania zgłoszeniami serwisowymi:

* baza danych o awarii,
* czas zgłoszenia,
* priorytet,
* użytkownicy,
* laboratoria i urządzenia,
* data rozwiązania problemu,
* statystyki
* wow: powiadomienia typu push w aplikacji mobilnej

1. Baza:
   1. Tabele
      1. **users**
         * id\_u
         * name
         * lastname
         * company
         * role (rola określająca prawa, np. użytkownik, administrator, serwisant)
      2. **issue**
         * id\_i
         * descr (opis usterki)
         * notif\_d (data, uzupełnia się gdy zostanie dodane zgłoszenie)
         * state (status, np. oczekuje, przyjęte, wykonano…)
         * priority (Zależny od zleceniodawcy, priorytet naprawy)
         * accept\_d (data kiedy serwisant zmienia status na przyjęto)
         * solve\_d (data rozwiązania, pusta dopóki ktoś nie zmieni ‘state’ na ukończono/wykonano)
         * solver\_id (serwisant, który naprawił)
      3. **device**
         * id\_d (numer przypisany urządzeniu przez uczelnie)
         * type (komputer, laptop, myszka, router…)
         * brand
         * model (model, np. Galaxy S5)
         * lab (nr sali)
      4. **sensitive\_data**

* id\_s
* login
* password
  1. Połączenia
     1. **users -> issue 1:n**
        + tabela relacji (id\_u|id\_i)
     2. **users -> sensitive\_data 1:1**
* tabela relacji (id\_u|id\_s)
  + 1. **issue -> device 1:1**
* tabela relacji (id\_i|id\_d)
  + 1. **users -> device 1:n**(przypisanie serwisanta do laboratorium)
* tabela relacji (id\_u|lab)

1. Funkcjonalność:
   1. Ogólne
      1. Rejestracja
      2. Logowanie
      3. Zabezpieczenia (jak zostanie czas)
      4. powiadomienia push włączone w tle( jak będzie czas to przy wyłączonej też)
   2. Użytkownik
      1. Panel dodanych zgłoszeń (tylko swoje, edycja, flaga rozwiązania problemu)
      2. Możliwość dodania zgłoszenia
   3. Administrator
      1. Panel zarządzania wszystkimi użytkownikami
      2. Panel dodanych zgłoszeń (wszystkie, usunięcie/edycja)
      3. Statystyki
         * Tydzień/miesiąc
         * Statystyki dla danego użytkownika
         * Dla danego serwisanta
         * Ile zgłoszeń, ile rozwiązanych, średni czas rozw. na podstawie accept\_date i notif\_date (powiadomiono, przyjęto, rozwiązano)
         * 7 dni wstecz, 30 wg. daty
   4. Serwisant
      1. Panel dodanych zgłoszeń (przypisane do niego, po sali + limit zgłoszeń, jeśli przekracza to trzeba przypisać do najmniej obciążonego serwisanta)
      2. Edycja tylko statusu zgłoszeń
      3. Statystyki tygodniowe/miesięczne „swoich” zgłoszeń
2. Planowane funkcje, określone na podstawie powyższych wymagań:

•Funkcja Login

Wejście: login, password, role

Działanie: Porównywanie Logina, Hasła i typu konta do danych z Bazy

Wyjście: 0, 1

•Funkcja Rejestracja

Wejście: login, password, name, lastname, company, role

Działanie: Sprawdzanie, czy użytkownik istnieje w bazie, jeżeli nie, utworzenie go

Wyjście: 0, 1 (0, błąd 1, utworzenie konta)

•Funkcja DodajZgoszenie

Wejście: id\_d, description,

Działanie: Sprawdzenie czy w bazie istnieje ta usterka jeżeli nie, dodanie jej ustawienie na status oczekujace

Wyjście: 0,1 (0, brak wypełnionych pól lub usterka jest w bazie 1, dodano do bazy)

•Funkcja wyswietl\_zgloszenia

Wejście: id\_u

Działanie: Wyświetla zgłoszenia gdzie użytkownik który dodał jest taki sam

Wyjście: Lista,która będzie wyświetlona

•Funkcja Edytuj

Wejście: id\_i

Działanie: Zmiana danych w zgłoszeniu

Wyjście: 0,1

•Funkcja ZmienStan

Wejście: id\_i

Działanie: Zmienia stan zgłoszenia (przyjęto, wykonano, w trakcie)

Wyjście : 0,1

•Funkcja WyswietlStatystyki

Wejście: id\_u, time

Działanie: Wyświetla statystyki, oblicza wszystkie przypadki serwisanta

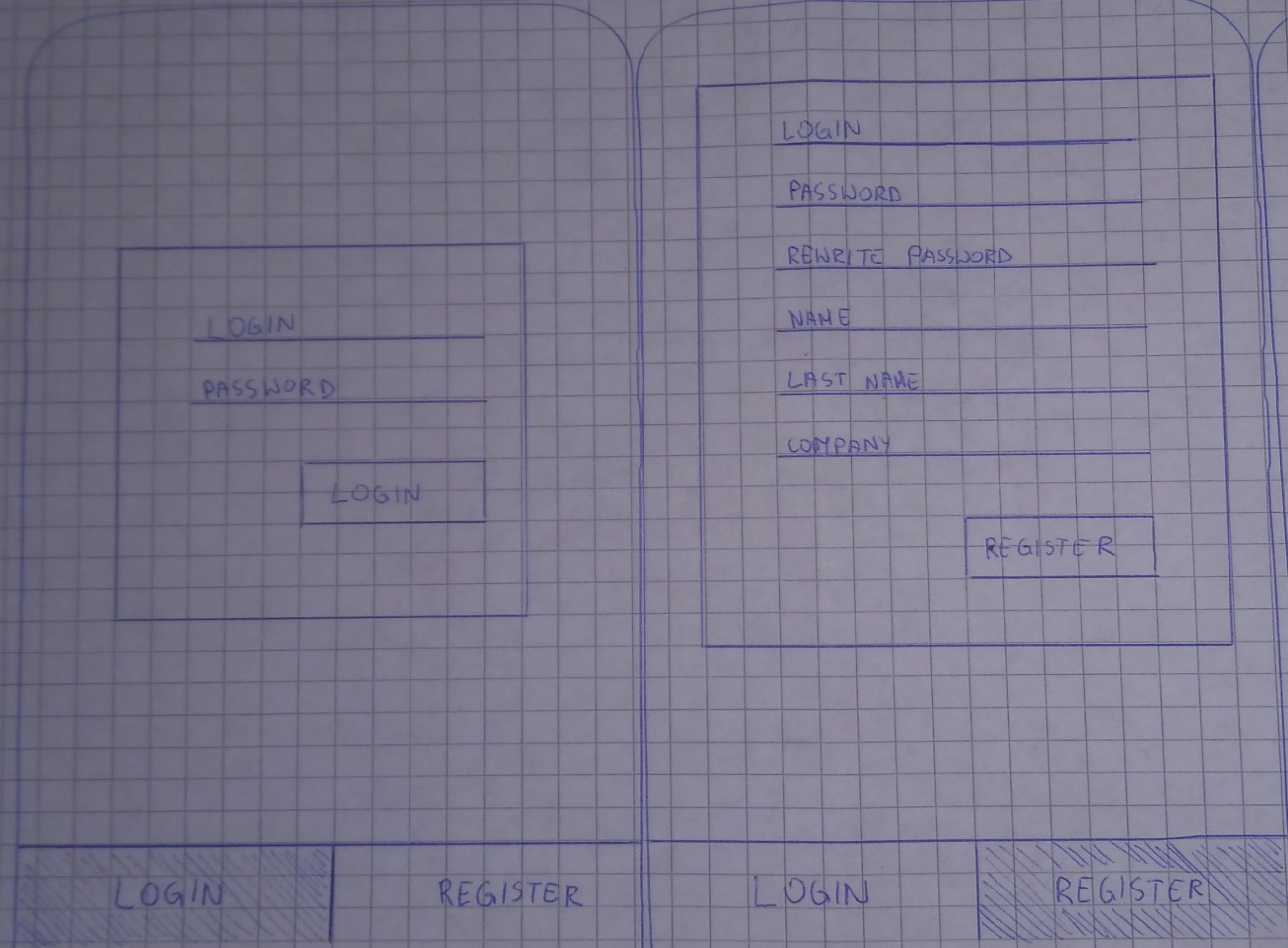
Wyjście: Statystyki

•Funkcja Usuń

Wejscie: id\_u

Działanie: Usuwanie Użytkownika z bazy

Wyjście: 0,1

1. Poglądowe schematy wyglądu poszczególnych paneli:

