

# Elektroniczna przychodnia lekarska (PHP, MYSQL)

*Paweł Wilczek Łukasz Szkaradek Łukasz Mamak*

Zespołowe przedsięwzięcie inżynierskie

Informatyka

Rok. akad. 2017/2018, sem. I

Prowadzący: dr hab. Marcin Mazur

# Spis treści

<b>1</b>	<b>Opis projektu</b>	<b>2</b>
1.1	Członkowie zespołu . . . . .	2
1.2	Cel projektu (produkt) . . . . .	2
1.3	Potencjalny odbiorca produktu (klient) . . . . .	2
1.4	Metodyka . . . . .	2
<b>2</b>	<b>Wymagania użytkownika</b>	<b>2</b>
2.1	User story 1 . . . . .	2
2.2	User story 2 . . . . .	2
2.3	User story 3 . . . . .	2
2.4	User story 4 . . . . .	2
2.5	User story 5 . . . . .	3
2.6	User story 6 . . . . .	3
2.7	User story 7 . . . . .	3
2.8	User story 8 . . . . .	3
<b>3</b>	<b>Harmonogram</b>	<b>3</b>
3.1	Rejestr zadań (Product Backlog) . . . . .	3
3.2	Sprint 1 . . . . .	3
3.3	Sprint 2 . . . . .	3
3.4	Sprint 3 . . . . .	4
<b>4</b>	<b>Product Backlog</b>	<b>4</b>
4.1	Backlog Item 1 . . . . .	4
4.2	Backlog Item 2 . . . . .	5
4.3	Backlog Item 3 . . . . .	6
4.4	Backlog Item 4 . . . . .	6
4.5	Backlog Item 5 . . . . .	6
4.6	Backlog Item 6 . . . . .	8
4.7	Backlog Item 7 . . . . .	9
<b>5</b>	<b>Sprint 1</b>	<b>9</b>
5.1	Cel . . . . .	9
5.2	Sprint Planning/Backlog . . . . .	10
5.3	Realizacja . . . . .	10
5.4	Sprint Review/Demo . . . . .	13
<b>6</b>	<b>Sprint 2</b>	<b>13</b>
6.1	Cel . . . . .	13
6.2	Sprint Planning/Backlog . . . . .	14
6.3	Realizacja . . . . .	14
6.4	Sprint Review/Demo . . . . .	16
<b>7</b>	<b>Sprint 3</b>	<b>16</b>
7.1	Cel . . . . .	16
7.2	Sprint Planning/Backlog . . . . .	16
7.3	Realizacja . . . . .	16
7.4	Sprint Review/Demo . . . . .	17

# 1 Opis projektu

## 1.1 Członkowie zespołu

1. Łukasz Szkaradek (kierownik projektu).
2. Paweł Wilczek.
3. Łukasz Mamak.

## 1.2 Cel projektu (produkt)

Celem projektu jest stworzenie strony internetowej z zastosowaniem php i mysql, która ma na celu obsługę wirtualnej przychodni lekarskiej oraz zwiększenie jej wydajności.

## 1.3 Potencjalny odbiorca produktu (klient)

Klientem może być przychodnia lekarska chcąca usprawnić wydajność niskim kosztem.

## 1.4 Metodyka

Projekt będzie realizowany przy użyciu (zaadaptowanej do istniejących warunków) metodyki *Scrum*.

# 2 Wymagania użytkownika

## 2.1 User story 1

Jako pacjent, chciałbym mieć możliwość zapisu do danego specjalisty za pośrednictwem strony internetowej przychodni, dzięki czemu nie musiałbym udać się tam osobiście i mógłbym przez to lepiej zarządzać czasem jaki mam do dyspozycji danego dnia.

## 2.2 User story 2

Jako pacjent chciałbym mieć wgląd do mojej karty pacjenta poprzez stronę przychodni tak abym mógł łatwo sprawdzić aktualną listę moich wizyt z lekarzem.

## 2.3 User story 3

Jako dyrektor placówki chciałbym posiadać konto admina by zarządzać elektroniczną przychodnią tak, aby w pełni czuwać nad moimi pracownikami.

## 2.4 User story 4

Jako lekarz chce żeby oprogramowanie przychodni miało łatwy i szybki dostęp do wszystkich moich aktualnych wizyt.

## **2.5 User story 5**

Jako recepcjonista w przychodni chciałbym mieć dostęp do elektronicznej bazy przychodni co w przeciwieństwie do tradycyjnej formy skróciło by mój czas reakcji i zmniejszyło ilość mojej pracy.

## **2.6 User story 6**

Jako pacjent chciałbym mieć dostęp do aktualności w przychodni takie jak opóźnienia w przyjmowaniu przez lekarzy, tak aby w takim przypadku inaczej zagospodarować czas.

## **2.7 User story 7**

Jako dyrektor chciałbym, aby platforma internetowa zwiększyła wydajność naszej placówki poprzez łatwość w rejestracji i katalogowaniu danych.

## **2.8 User story 8**

Jako lekarz chciałbym mieć dostęp do elektronicznej wersji karty pacjenta co umożliwiło by mi poznanie historii choroby pacjenta i jej edycje w razie konieczności.

# **3 Harmonogram**

## **3.1 Rejestr zadań (Product Backlog)**

- Data rozpoczęcia: 11.10.2017.
- Data zakończenia: 15.11.2017.

## **3.2 Sprint 1**

- Data rozpoczęcia: 15.11.2017.
- Data zakończenia: 29.11.2017.
- Scrum Master: Łukasz Szkaradek.
- Product Owner: Łukasz Mamak.
- Development Team: Łukasz Mamak, Paweł Wilczek, Łukasz Szkaradek.

## **3.3 Sprint 2**

- Data rozpoczęcia: 29.11.2017.
- Data zakończenia: 20.12.2017.
- Scrum Master: Łukasz Mamak.
- Product Owner: Paweł Wilczek.
- Development Team: Paweł Wilczek, Łukasz Szkaradek, Łukasz Mamak.

### 3.4 Sprint 3

- Data rozpoczęcia: 20.12.2017.
- Data zakończenia: 10.01.2018.
- Scrum Master: Paweł Wilczek.
- Product Owner: Łukasz Szkaradek.
- Development Team: Łukasz Szkaradek, Łukasz Mamak, Paweł Wilczek.

## 4 Product Backlog

### 4.1 Backlog Item 1

**Tytuł zadania.** Stworzenie bazy danych.

**Opis zadania.** Stworzenie bazy danych, która służyć będzie do testowania początkowych zadań.

**Priorytet.** Średni.

**Definition of Done.** Stworzenie w MYSQL bazy danych i tablic:

#### **SPECJALIZACJE:**

- id specjalizacji (automatycznie przypisywany numer),
- nazwa specjalizacji (nazwa specjalizacji),

#### **AKTUALNOŚCI:**

- id aktualności (automatycznie przypisywany numer),
- data (data napisania aktualności) ,
- opis (treść aktualności)

#### **PACJENCI:**

- id pacjenta (automatycznie przypisywany numer),
- imię (imię pacjenta),
- nazwisko (nazwisko pacjenta),
- pesel (pesel pacjenta),
- karta pacjenta (treść karty pacjenta),
- nr telefonu
- kod (automatycznie generowany kod, który wraz z peselem będzie umożliwiał logowanie się pacjenta)

### **SPOTKANIA:**

- id spotkania (automatycznie przypisywany numer),
- id specjalizacji (numer odnoszący się do konkretnego wiersza w tablicy specjalizacje),
- id lekarza (numer odnoszący się do konkretnego wiersza w tablicy lekarze),
- id osoby (numer odnoszący się do konkretnego wiersza w tablicy pacjenci),
- data odbycia (data wpisywana przez recepcjonistę, która określa datę spotkania z lekarzem),
- data zapisu (automatycznie przypisywana data podczas tworzenia wiersza),
- stan (równa się 0 lub 1, w zależności czy spotkanie zostało zaakceptowane przez recepcjonistę)

### **LEKARZE:**

- id lekarza (automatycznie przypisywany numer),
- login (login używany przy logowaniu na stronie recepcji),
- hasło (hasło używane przy logowaniu na stronie recepcji),
- imię (imię lekarza),
- nazwisko (nazwisko lekarza),
- id specjalizacji (numer odnoszący się do konkretnego wiersza w tablicy specjalizacje, określa jaką specjalizację posiada dany lekarz),
- nr pokoju (numer pokoju w którym dany lekarz przyjmuje),

## **4.2 Backlog Item 2**

**Tytuł zadania.** Strona tytułowa oraz formularz logowania pacjenta.

**Opis zadania.** Stworzenie strony głównej wraz z interfejsem i logowaniem dla pacjenta.

**Priorytet.** Średni.

**Definition of Done.** Napisanie w języku PHP strony głównej w której będą zawarte:

1. Formularz logowania pacjenta (poprzez pesel i kod).
2. Odnośniki do reszty podstron takich jak:
  - formularz rejestracji pacjenta

- strony recepcji.

*Strona zawierać będzie skrypt, który po wykryciu zmiennych wysłanych przez formularz logowania pacjenta wyświetli:*

- imię pacjenta,
- nazwisko pacjenta,
- pesel pacjenta,
- listę umówionych spotkań z lekarzami,
- formularz przez, który tworzyć będzie spotkania wraz z skryptem obsługującym daną operację (dodanie do tablicy "spotkania" danych: id specjalizacji, id osoby).

### 4.3 Backlog Item 3

**Tytuł zadania.** Aktualności.

**Opis zadania.** Dodanie do strony głównej listę aktualności.

**Priorytet.** Niski.

**Definition of Done.** Edycja strony głównej tak aby w przejrzysty sposób wyświetlała dane z tablicy "aktualności": - Data aktualności (data), - Treść aktualności (opis).

### 4.4 Backlog Item 4

**Tytuł zadania.** Formularz zapisu dla pacjentów.

**Opis zadania.** Stworzenie strony dzięki której będzie można zapisać się do listy pacjentów.

**Priorytet.** Niski.

**Definition of Done.** Napisanie strony w języku PHP zawierającej formularz z danymi do wprowadzenia (Imię, Nazwisko, Pesel, Telefon kontaktowy). Po otrzymaniu danych zostaną one wprowadzone do tablicy "pacjenci". Ponadto strona po wypełnieniu formularza generować będzie 5-cyfrowy kod, który również będzie wprowadzany do tablicy.

### 4.5 Backlog Item 5

**Tytuł zadania.** Aktualizacja bazy danych.

**Opis zadania.** Dodanie do bazy danych potrzebnych tablic i ewentualna edycja starych tablic w razie potrzeby.

**Priorytet.** Wysoki.

**Definition of Done.** Dodanie do bazy danych, tablic:

**RECEPCJA:**

- id recepcjonisty (automatycznie przypisywany numer),
- imie (imię recepcjonisty),
- nazwisko (nazwisko recepcjonisty),
- login (login używany przy logowaniu na stronie recepcji),
- hasło (hasło używane przy logowaniu na stronie recepcji)

**DYREKTOR:**

- id dyrektora (automatycznie przypisywany numer),
- imie (imię dyrektora),
- nazwisko (nazwisko dyrektora),
- login (login używany przy logowaniu na stronie recepcji),
- hasło (hasło używane przy logowaniu na stronie recepcji).

*Stworzenie 2 widoków, ułatwiających obsługę strony:*

**LEKARZE:**

- imie (imię lekarza z tablicy "lekarze"),
- nazwisko (nazwisko lekarza z tablicy "lekarze"),
- specjalizacja (nazwa specjalizacji z tablicy "specjalizacje", odpowiednia dla konkretnego lekarza),
- nr pokoju (nr pokoju z tablicy "lekarze")

**LISTA:**

- id spotkania (id spotkania z tablicy "spotkania"),
- id osoby (id pacjenta z tablicy "pacjenci" odpowiednie dla konkretnego spotkania),
- id lekarza (id lekarza z tablicy "lekarze" odpowiednie dla konkretnego spotkania),
- pesel (pesel z tablicy "pacjenci"),
- imie osoby (imię osoby z tablicy "pacjenci"),
- nazwisko osoby (nazwisko osoby z tablicy "pacjenci"),
- specjalizacja (nazwa specjalizacji z tablicy "specjalizacje"),



- imię lekarza (imię lekarza z tablicy "lekarze"),
- nazwisko lekarza (nazwisko lekarza z tablicy "lekarze"),
- nr pokoju (numer pokoju z tablicy "lekarze"),
- data odbycia (data odbycia z tablicy "spotkania"),
- data zapisu (data zapisu z tablicy "spotkania"),
- stan (stan z tablicy "spotkania")

## 4.6 Backlog Item 6

**Tytuł zadania.** Strona logowania pracowników.

**Opis zadania.** Napisanie strony, która będzie dawać dostęp do logowania i wyświetlania listy opcji dla: dyrektora, recepcjonisty i lekarza.

**Priorytet.** Średni.

**Definition of Done.** Stworzenie strony na, której będzie formularz z danymi do wprowadzenia: login, hasło. Wraz z skryptem który po ich otrzymaniu przeszuka tablice: lekarze, dyrektor, recepcja w poszukiwaniu prawidłowości. Jeżeli znajdzie wykorzysta funkcję "session" i wpisze do niej dane takie jak imię, nazwisko, id oraz tablica z informacją z której tablicy dana osoba pochodzi. W przypadku wykrycia zmiennych w "session", wypisuje imię, nazwisko oraz id danej osoby, oraz w zależności od zmiennej tablica listę odnośników do strony "zarządzaj"

### DLA LEKARZE:

- odnośnik wraz z informacją w formie GET proszącą o listę spotkań.
- odnośnik wraz z informacją w formie GET proszącą o kartę pacjenta.

### DLA RECEPCJA:

- odnośnik wraz z informacją w formie GET proszącą o lista osób zapisanych.
- odnośnik wraz z informacją w formie GET proszącą o lista lekarzy .
- odnośnik wraz z informacją w formie GET proszącą o lista dostępnych specjalizacji.

### DLA DYREKTOR:

- odnośniki wraz z informacjami w formie GET proszącą o możliwość edycji, usuwania, dodawania do tablic:
- a) lekarze
- b) specjalizacje

- c) recepcja
- d) spotkania
- e) pacjenci
- f) aktualności

## 4.7 Backlog Item 7

**Tytuł zadania.** Wyświetlanie, edycja, usuwanie.

**Opis zadania.** Stworzenie strony obsługujących edycję, wyświetlanie i usuwanie danych z tablic.

**Priorytet.** Wysoki.

**Definition of Done.** Stworzenie strony "zarządzanie", która po otrzymaniu informacji formą GET i zweryfikowaniu uprawnień z funkcji "session", będzie udostępniać formę formularza wraz skryptem je obsługującym opcje dla osób z tablicy:

### LEKARZE:

- listę spotkań (wyświetlanie),
- kartę pacjenta (wyświetlanie i edycja)

### RECEPCJA:

- listę osób zapisanych (wyświetlanie i edycja),
- listę lekarzy (wyświetlanie),
- listę dostępnych specjalizacji (wyświetlanie)

### DYREKTOR:

- listę lekarzy (wyświetlanie, edycja i usuwanie),
- listę specjalizacji (wyświetlanie, edycja i usuwanie),
- listę recepcji (wyświetlanie, edycja i usuwanie),
- listę spotkań (wyświetlanie, edycja i usuwanie),
- listę pacjentów (wyświetlanie, edycja i usuwanie),
- listę aktualności (wyświetlanie, edycja i usuwanie)

## 5 Sprint 1

### 5.1 Cel

Stworzenie bazy danych dzięki której możliwe będzie przeglądanie strony tytułowej i formularza zapisu pacjentów.

## 5.2 Sprint Planning/Backlog

**Tytuł zadania.** Stworzenie początkowej bazy danych.

- Estymata: M.

**Tytuł zadania.** Strona tytułowa oraz formularz logowania pacjenta.

- Estymata: L.

**Tytuł zadania.** Aktualności.

- Estymata: S.

## 5.3 Realizacja

**Tytuł zadania.** Stworzenie początkowej bazy danych.

**Wykonawca.** Łukasz Mamak

**Realizacja.** Zadaniem do wykonania było stworzenie niezbędnej do dalszych kroków bazy danych. Realizacja tegoż zadania przebiegła bez większych komplikacji. W skrypcie zostało utworzonych kilka tabel, począwszy od aktualności, poprzez tabele pacjenci, lekarze, na tabeli spotkania kończąc. W każdej tabeli zdefiniowany został klucz główny dla wartości "niepowtarzalnych" w zależności od tabeli. Zdefiniowane zostały także klucze obce, które zostały wykorzystane do utworzenia relacji między parą tabel. Podsumowując, zadanie zostało zrealizowane w całości, bez istotnych do odnotowania problemów, nie wychodzących poza ramy błędnego wpisania składni, lub też drobnych literówek.

Kod programu (środowisko *verbatim*):

```
CREATE TABLE aktualnosci (  
  id_aktualnosci INT NOT NULL auto_increment,  
  data TIMESTAMP NOT NULL,  
  opis VARCHAR(50) NOT NULL,  
  
  CONSTRAINT c_pk0 PRIMARY KEY(id_aktualnosci)  
) ENGINE = InnoDB;  
  
CREATE TABLE pacjenci (  
  id_pacjenta INT NOT NULL auto_increment,  
  imie VARCHAR(50) NOT NULL,  
  nazwisko VARCHAR(50) NOT NULL,  
  karta_pacjenta VARCHAR(255) NOT NULL,  
  pesel BIGINT NOT NULL,  
  nr_telefonu BIGINT NOT NULL,  
  kod VARCHAR(50) NOT NULL,  
  
  CONSTRAINT c_pk PRIMARY KEY(id_osoby)
```

```

) ENGINE = InnoDB;

CREATE TABLE lekarze (
id_lekarza INT NOT NULL auto_increment,
login VARCHAR(50) NOT NULL,
haslo VARCHAR(50) NOT NULL,
imie VARCHAR(50) NOT NULL,
nazwisko VARCHAR(50) NOT NULL,
id_specjalizacji INT NOT NULL,
nr_pokoju VARCHAR(50) NOT NULL,

CONSTRAINT c_pk2 PRIMARY KEY(id_lekarza)
) ENGINE = InnoDB;

CREATE TABLE spotkania (
id_spotkania INT NOT NULL auto_increment,
id_specjalizacji INT NOT NULL,
id_lekarza INT NOT NULL,
id_osoby INT NOT NULL,
data_odbycia VARCHAR(50),
data_zapisu TIMESTAMP NOT NULL,
stan enum('0','1') NOT NULL DEFAULT '0',

) ENGINE = InnoDB;

```

**Tytuł zadania.** Strona tytułowa oraz formularz logowania pacjenta.

**Wykonawca.** Łukasz Szkaradek

**Realizacja.** Celem zadania było wykonanie strony internetowej w której zawarte będą logowanie pacjenta, wyświetlanie spotkań danego pacjenta oraz ich dodawanie poprzez formularz. Wykonanie zadania odbyło się bez większych trudności, choć wykonanie niektórych algorytmów takich jak wyświetlanie wyboru godzin z pominięciem już zarezerwowanych oraz przeszłych godzin czy wyświetlanie odpowiednich dat dla wybranych dni wymagało większego wysiłku. Przydatna przy wykonaniu zadania była funkcja session, która przechowuje dane aż do ich usunięcia lub wyłączenia przeglądarki. Przy przesyłaniu danych z formularza została użyta forma POST. Zadanie zostało wykonane w całości, a wszystkie jego funkcje działają prawidłowo.

Kod programu (środowisko PHP):

```

<?php
ob_start();
date_default_timezone_set('Europe/Warsaw');

```

```

session_start();
$host="<<ip serwera>>";
$db_user="<<nazwa użytkownika>>";
$db_password="<<hasło użytkownika>>";
$db_database="<<nazwa bazy danych>>";
$link= mysqli_connect($host,$db_user,$db_password,$db_database);
$zat="call zatw()";
$zat_w=mysqli_query($link,$zat);
if (isset($_GET[powrot])) {
    session_destroy();
    header("Location: index.php");
}
if (isset($_POST[stan])){
    $zapytanie="SELECT * FROM pacjenci WHERE pesel='".$_POST[pesel]' and kod='".$_POST[kod]'"';
    $wykonaj=mysqli_query($link,$zapytanie);
    if(@mysqli_num_rows($wykonaj)){
        while($wiersz=mysqli_fetch_assoc($wykonaj)) {
            $_SESSION[zalogowany] = "pacjent";
            $_SESSION[id_p] = $wiersz['id_pacjenta'];
            $_SESSION['baza'] = 'pacjent';
        }
    } else {
        $brak='1';
    }
}
?>
<html>
<head>
<title>Przychodnia lekarska</title>
<META http-equiv=Content-Language content=pl>
<META http-equiv=Content-Type content="text/html; charset=windows-1250">
<style>
a {
    color: black;
    text-decoration: none;
}
</style>
</head>
.

```

**Tytuł zadania.** Aktualności.

**Wykonawca.** Paweł Wilczek

**Realizacja.** Zadanie polegało na utworzeniu aktualności na stronie głównej przychodni lekarskiej dzięki wyłuskaniu odpowiednich informacji z tabeli "aktualności" i umieszczeniu ich na stronie głównej dzięki skryptowi PHP.

Tabela aktualności:

```
CREATE TABLE aktualnosci (
id_aktualnosci INT NOT NULL auto_increment,
data TIMESTAMP NOT NULL,
opis VARCHAR(50) NOT NULL,

CONSTRAINT c_pk0 PRIMARY KEY(id_aktualnosci)
) ENGINE = InnoDB;
```

Skrypt PHP:

```
echo '<br><br><br>
<table border="1px">
<tr><td style="text-align: center;" colspan="2">Aktualności</td></tr>';
$aktualnosci="SELECT * FROM aktualnosci";
$wykonaj=mysqli_query($link,$aktualnosci);
if(@mysqli_num_rows($wykonaj)){
while($wiersz=mysqli_fetch_assoc($wykonaj)) {
echo '<tr><td width="150">'. $wiersz['data']. '</td><td width="950">'. $wiersz['opis']. '</tr>
}
}
}
```

## 5.4 Sprint Review/Demo

Naszym celem była realizacja 3 wcześniej ustalonych i zaplanowanych zadań. Dotyczyły one stworzenia bazy danych, strony tytułowej oraz formularza logowania pacjenta a także aktualności. Podczas realizacji wcześniej przydzielonych zadań, żaden z członków zespołu nie napotkał na istotne, konieczne do odnotowania problemy. Wszystkie z przewidzianych zadań zostały wykonane terminowo, co za tym idzie przyrost został osiągnięty. Z racji tematyki naszego projektu, przyrost, postęp będzie dostępny na stronie internetowej przychodni, w adresie poniżej.

*Demonstracja przyrostu produktu:*

**Link do strony:** <http://luki9696.000webhostapp.com/projekt/index.php>

## 6 Sprint 2

### 6.1 Cel

Sprint ma na celu wykonanie dwóch zadań. Pierwszym zadaniem będzie uaktualnienie bazy danych o nowe tabele, wymagane do dalszego rozwoju strony, jak i stworzenie widoków, które ułatwią będą wyświetlanie danych z tabeli. Natomiast drugim celem sprintu jest stworzenie podstrony, na której zawarty będzie formularz przez który będą mogli rejestrować się pacjenci, strona musi zawierać również kod, który będzie umożliwiał wykonanie operacji wpisania pacjenta do bazy.

## 6.2 Sprint Planning/Backlog

**Tytuł zadania.** Formularz zapisu dla pacjentów.

- Estymata: M.

**Tytuł zadania.** Aktualizacja bazy danych.

- Estymata: XL.

## 6.3 Realizacja

**Tytuł zadania.** Aktualizacja bazy danych.

**Wykonawca.** Łukasz Szkaradek.

**Realizacja.** Zadanie zostało wykonane bez problemów zostały utworzone 3 nowe tabele oraz 2 widoki ułatwiające wyświetlanie danych z tabeli.

Kod programu (środowisko verbatim):

```
CREATE TABLE specjalizacje (  
  id_specjalizacji INT NOT NULL auto_increment,  
  nazwa_specjalizacji VARCHAR(50) NOT NULL,  
  
  CONSTRAINT c_pk1 PRIMARY KEY(id_specjalizacji)  
  
  ) ENGINE = InnoDB;  
  
CREATE TABLE recepcja (  
  id_recepcjonisty INT NOT NULL auto_increment,  
  imie VARCHAR(50) NOT NULL,  
  nazwisko VARCHAR(50) NOT NULL,  
  login VARCHAR(50) NOT NULL,  
  haslo VARCHAR(50) NOT NULL,  
  
  CONSTRAINT c_pk2 PRIMARY KEY(id_recepcjonisty)  
  
  ) ENGINE = InnoDB;  
  
CREATE TABLE dyrektor (  
  id_dyrektora INT NOT NULL auto_increment,  
  imie VARCHAR(50) NOT NULL,  
  nazwisko VARCHAR(50) NOT NULL,  
  login VARCHAR(50) NOT NULL,  
  haslo VARCHAR(50) NOT NULL,  
  
  CONSTRAINT c_pk3 PRIMARY KEY(id_dyrektora)  
  
  ) ENGINE = InnoDB;
```

».

**Wykonawca.** Łukasz Mamak, Paweł Wilczek.

15



</center>  
</html>

## 6.4 Sprint Review/Demo

Naszym celem była realizacja wcześniej ustalonych i zaplanowanych zadań. Dotyczyły one stworzenia formularza zapisu dla pacjentów oraz aktualizacja bazy danych która polegała na utworzeniu trzech nowych tabel oraz dwóch widoków ułatwiających wyświetlanie danych z tabeli. Podczas realizacji wcześniej przydzielonych zadań, żaden z członków zespołu nie napotkał na istotne, konieczne do odnotowania problemy. Wszystkie z przewidzianych zadań zostały wykonane terminowo, co za tym idzie przyrost został osiągnięty. Z racji tematyki naszego projektu, przyrost, postęp będzie dostępny na stronie internetowej przychodni, w adresie poniżej.

*Demonstracja przyrostu produktu:*

**Link do strony:** <http://luki9696.000webhostapp.com/projekt/index.php>

## 7 Sprint 3

### 7.1 Cel

«Określić, w jakim celu tworzony jest przyrost produktu».

### 7.2 Sprint Planning/Backlog

**Tytuł zadania.** Strona logowania pracowników.

- Estymata: XL.

**Tytuł zadania.** Wyświetlanie, edycja, usuwanie.

- Estymata: XXL.

«Tutaj dodawać kolejne zadania»

### 7.3 Realizacja

**Tytuł zadania.** «Tytuł».

**Wykonawca.** ił.

**Realizacja.** «Sprawozdanie z realizacji zadania (w tym ocena zgodności z estymatą). Kod programu (środowisko *verbatim*):

```
for (i=1; i<10; i++) ...
```

».

**Tytuł zadania.** «Tytuł».

**Wykonawca.** `i`.

**Realizacja.** «Sprawozdanie z realizacji zadania (w tym ocena zgodności z estymatą). Kod programu (środowisko `verbatim`):

```
for (i=1; i<10; i++) ...  
».
```

## 7.4 Sprint Review/Demo

«Sprawozdanie z przeglądu Sprint'u – czy założony cel (przyrost) został osiągnięty oraz czy wszystkie zaplanowane Backlog Item'y zostały zrealizowane? Demostracja przyrostu produktu».

«Tutaj dodawać kolejne Sprint'y»

## Literatura

- [1] S. R. Covey, *7 nawyków skutecznego działania*, Rebis, Poznań, 2007.
- [2] Tobias Oetiker i wsp., Nie za krótkie wprowadzenie do systemu L<sup>A</sup>T<sub>E</sub>X 2<sub>ε</sub>, <ftp://ftp.gust.org.pl/TeX/info/lshort/polish/lshort2e.pdf>
- [3] K. Schwaber, J. Sutherland, *Scrum Guide*, <http://www.scrumguides.org/>, 2016.
- [4] <https://agilepainrelief.com/notesfromatooluser/tag/scrum-by-example>
- [5] [https://www.tutorialspoint.com/scrum/scrum\\_user\\_stories.htm](https://www.tutorialspoint.com/scrum/scrum_user_stories.htm)