

System elektronicznej kontroli pracowników

Mateusz Morawa (kierownik projektu)

Łukasz Mirek

Grzegorz Jasiński

Bartłomiej Kmak

Łukasz Horowski

Zespołowe przedsięwzięcie inżynierskie

Informatyka

Rok. akad. 2017/2018, sem. I

Prowadzący: dr hab. Marcin Mazur

Spis treści

1	Opis projektu	3
1.1	Członkowie zespołu	3
1.2	Cel projektu (produkt)	3
1.3	Potencjalny odbiorca produktu (klient)	3
1.4	Metodyka	3
2	Wymagania użytkownika	3
2.1	User story 1	3
2.2	User story 2	4
2.3	User story 3 - opcjonalna	4
2.4	User story 4	4
2.5	User story 5	4
2.6	User story 6	4
2.7	User story 7	4
2.8	User story 8	4
3	Harmonogram	4
3.1	Rejestr zadań (Product Backlog)	4
3.2	Sprint 1	5
3.3	Sprint 2	5
3.4	Sprint 3	5
3.5	Sprint 4	5
4	Product Backlog	5
4.1	Backlog Item 1	5
4.2	Backlog Item 2	6
4.3	Backlog Item 3	6
4.4	Backlog Item 4	6
4.5	Backlog Item 5	7
4.6	Backlog Item 6	7
4.7	Backlog Item 7	7
4.8	Backlog Item 8	8
4.9	Backlog Item 9	8
4.10	Backlog Item 10	8
4.11	Backlog Item 11	9
4.12	Backlog Item 12	9
4.13	Backlog Item 13	9
5	Sprint 1	10
5.1	Cel	10
5.2	Sprint Planning/Backlog	10
5.3	Realizacja	10
5.4	Sprint Review/Demo	13

6	Sprint 2	14
6.1	Cel	14
6.2	Sprint Planning/Backlog	14
6.3	Realizacja	14
6.4	Sprint Review/Demo	19
7	Sprint 3	19
7.1	Cel	19
7.2	Sprint Planning/Backlog	19
7.3	Realizacja	19
7.4	Sprint Review/Demo	20
8	Sprint 4	20
8.1	Cel	20
8.2	Sprint Planning/Backlog	20
8.3	Realizacja	20
8.4	Sprint Review/Demo	21

1 Opis projektu

1.1 Członkowie zespołu

1. Mateusz Morawa (kierownik projektu).
2. Łukasz Mirek.
3. Grzegorz Jasiński.
4. Bartłomiej Kmak.
5. Łukasz Horowski.

1.2 Cel projektu (produkt)

Celem projektu jest stworzenie systemu kontroli czasu pracy pracowników. Przy pomocy urządzenia o nazwie Raspberry Pi oraz modułu RFID (ang. Radio-frequency identification) w skład którego wchodzi czytnik oraz karta, pracownicy do tego zobowiązani będą identyfikowali swoje godziny przyścia i wyjścia z pracy poprzez przyłożenie karty do czytnika. Data, godzina oraz ID pracownika będą trafiały do bazy danych. Płaca każdego pracownika będzie wyliczana na podstawie ilości przepracowanych godzin zarejestrowanych w systemie. Pracodawca za pomocą strony internetowej będzie mógł sprawdzić ilu jest pracowników na zakładzie oraz ile wyniósł ich czas pracy.

1.3 Potencjalny odbiorca produktu (klient)

Potencjalnym odbiorcą oferowanego przez nas systemu jest przedsiębiorca zatrudniający grupę co najmniej kilkunastu pracowników. Przy takiej ilości osób trudno jest zarządzać ich czasem pracy bez użycia odpowiedniego systemu.

1.4 Metodyka

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

2 Wymagania użytkownika

Lista wymagań użytkownika w postaci „historyjek” (User stories). Każda historyjka opisuje jedną cechę systemu. Struktura: As a [type of user], I want [to perform some task] so that I can [achieve some goal/benefit/value] (zob. np. [5]).

2.1 User story 1

Jako kierownik wymagam od systemu, aby przy wejściu i wyjściu z zakładu pracownicy dokonywali identyfikacji za pomocą kart RFID, po to by można było ustalić ilu pracowników jest obecnie na zakładzie.

2.2 User story 2

Jako księgowy chcę, aby system na podstawie przepracowanych godzin wyliczał miesięczne wynagrodzenie dla każdego pracownika, abym mógł łatwiej obliczyć należną wypłatę.

2.3 User story 3 - opcjonalna

Jako pracownik chcę, abym mógł w zaprojektowanym systemie na bieżąco sprawdzać ilość przepracowanych godzin po to bym mógł sprawdzić szacowaną wypłatę.

2.4 User story 4

Jako księgowy chcę mieć możliwość drukowania zestawienia miesięcznego wynagrodzenia brutto pracowników, żeby mieć papierową dokumentację w razie kontroli.

2.5 User story 5

Jako kadrowy chcę mieć możliwość dodawania i usuwania nowych pracowników do bazy danych, aby w łatwy sposób uaktualnić informacje o zatrudnionych pracownikach w firmie.

2.6 User story 6

Jako pracodawca chcę, aby logowanie się na zakładowej stronie internetowej było możliwe tylko przez sieć wewnętrzną, aby zapewnić bezpieczeństwo danych w firmie.

2.7 User story 7

Jako pracodawca chciałbym, aby każdy korzystający z zakładowej strony internetowej miał możliwość przypomnienia hasła, żeby móc je przywrócić.

2.8 User story 8

Jako kierownik, chciałbym mieć dostęp do wszystkich funkcji zaimplementowanych na stronie zakładowej, ponieważ w razie nagłych wypadków chcę mieć kontrolę nad całym systemem.

3 Harmonogram

3.1 Rejestr zadań (Product Backlog)

- Data rozpoczęcia: 25.10.17 r.
- Data zakończenia: 08.11.17 r.

3.2 Sprint 1

- Data rozpoczęcia: 08.11.17 r.
- Data zakończenia: 29.11.17 r.
- Scrum Master: Łukasz Mirek.
- Product Owner: Grzegorz Jasiński.
- Development Team: Mateusz Morawa, Łukasz Horowski, Grzegorz Jasiński, Łukasz Mirek, Bartłomiej Kmak.

3.3 Sprint 2

- Data rozpoczęcia: 29.11.17 r.
- Data zakończenia: 20.12.17 r.
- Scrum Master: Łukasz Horowski.
- Product Owner: Mateusz Morawa.
- Development Team: Mateusz Morawa, Grzegorz Jasiński, Łukasz Mirek, Bartłomiej Kmak, Łukasz Horowski.

3.4 Sprint 3

- Data rozpoczęcia: 20.12.17 r.
- Data zakończenia: 10.01.18 r.
- Scrum Master: Mateusz Morawa.
- Product Owner: Kmak Bartłomiej.
- Development Team: Mateusz Morawa, Grzegorz Jasiński, Łukasz Mirek, Bartłomiej Kmak, Łukasz Horowski.

3.5 Sprint 4

- Data rozpoczęcia: 10.01.18 r.
- Data zakończenia: 24.01.18 r.
- Scrum Master: Bartłomiej Kmak.
- Product Owner: Łukasz Mirek.
- Development Team: Mateusz Morawa, Grzegorz Jasiński, Łukasz Mirek, Łukasz Horowski.

4 Product Backlog

4.1 Backlog Item 1

Tytuł zadania. Rejestrowanie wejścia i wyjścia pracownika z zakładu.

Opis zadania. Do realizacji powyższego zadania zostanie użyty Raspberry Pi i podłączony do niego czytnik kart RFID, do którego swoje karty przykładać będą pracownicy w celu rejestracji przybycia lub opuszczenia zakładu. Następnie specjalny program napisany w języku Python będzie zgromadzone dane zapisywał do pliku i okresowo wysyłał do bazy danych na firmowym serwerze.

Priorytet. 1.

Definition of Done. Dane odnośnie wejścia i wyjścia odczytane są z karty pracownika przez czytnik RFID podłączony do Raspberry Pi i wysyłane do bazy danych, znajdującą się na tym urządzeniu.

4.2 Backlog Item 2

Tytuł zadania. Utworzenie pierwotnej wersji strony www do obsługi systemu.

Opis zadania. Zostanie utworzona strona do obsługi systemu, która umożliwi przeglądanie danych pobranych z bazy danych, dotyczących zarejestrowanych przez czytnik RFID godzin przybycia lub opuszczenia terenu zakładu przez pracowników.

Priorytet. 1.

Definition of Done. Zostanie utworzona strona www przy pomocy technologii HTML, PHP którą będzie połączona z bazą danych MySQL. Umożliwi ona wyświetlanie danych odnośnie wejścia i wyjścia pracowników z zakładu.

4.3 Backlog Item 3

Tytuł zadania. Dodawanie nowych kart do bazy danych.

Opis zadania. Zostanie utworzony nowy skrypt w języku Python, który za pomocą czytnika RFID odczyta numer ID nowej karty oraz wpisze je do bazy danych, tak aby podczas rejestracji nowego użytkownika można było mu przypisać odpowiednią kartę identyfikacyjną.

Priorytet. 2.

Definition of Done. Program po przyłożeniu nowej karty do czytnika doda jej numer ID do bazy danych oraz odpowiednio wyda sygnał diodami i buzzerem o poprawnym wykonaniu tego zadania lub o tym, że dany numer ID jest już w bazie.

4.4 Backlog Item 4

Tytuł zadania. Logowanie do strony zakładowej.

Opis zadania. Strona zakładowa będzie posiadać funkcję logowania na konto użytkownika. Na początku w bazie danych zostanie ręcznie utworzone konto kierownika, na które będzie mógł się zalogować. Funkcjonalność tworzenia nowych kont zostanie zaimplementowana przy dodawaniu nowych pracowników.

Priorytet. 2.

Definition of Done. Istnieje konto kierownika z możliwością zalogowania się na stronie.

4.5 Backlog Item 5

Tytuł zadania. Podsumowanie przepracowanych godzin.

Opis zadania. System na podstawie danych zgromadzonych w bazie, będzie zliczał godziny przepracowane przez poszczególnego pracownika. Liczone będą tylko pełne godziny przepracowane na zakładzie w przedziale czasowym od ósmej do szesnastej.

Priorytet. 2.

Definition of Done. System będzie wyliczał sumę przepracowanych pełnych godzin odczytanych z rejestrów wejścia/wyjścia.

4.6 Backlog Item 6

Tytuł zadania. Obliczanie wynagrodzenia.

Opis zadania. System na podstawie danych o przepracowanych godzinach wyznaczy miesięczne wynagrodzenie zgodnie z określoną stawką godzinową zawartą w umowie. Pracownicy otrzymywać będą wynagrodzenie tylko za pełne godziny przepracowane na zakładzie w przedziale czasowym od ósmej do szesnastej. Gdy pracownik przyjdzie do zakładu w trakcie trwającej godziny pracy będzie mógł otrzymać zapłatę dopiero za następną pełną godzinę.

Priorytet. 2.

Definition of Done. System będzie wyliczał wynagrodzenie miesięczne pracowników na podstawie przepracowanych pełnych godzin. Zliczanie rozpoczyna się od godz. 8.00, kończy o godz. 16.00. Tylko przepracowanie pełnej godziny gwarantuje otrzymanie za nią wynagrodzenia. Istotna jest punktualność.

4.7 Backlog Item 7

Tytuł zadania. Funkcja umożliwiająca pracownikowi sprawdzenie wynagrodzenia (opcjonalna).

Opis zadania. System po zalogowaniu się pracownika na stronę umożliwi mu wgląd do zestawienia z poprzednich miesięcy oraz z poszczególnych dni obecnego miesiąca. Zestawienie wygeneruje pracownikowi pensję jaką otrzymał w poprzednich miesiącach a także wyświetli szczegółowe dane co do przepracowanych godzin.

Priorytet. 4.

Definition of Done. Pracownik po zalogowaniu się na stronie może zobaczyć zestawienie swoich zarobków. Zestawienie będzie wyświetlane w formie tabeli. Istnieje możliwość wyboru poszczególnych miesięcy, które interesują pracownika (maksymalnie do 12 miesięcy wstecz).

4.8 Backlog Item 8

Tytuł zadania. Drukowanie zestawienia miesięcznego wynagrodzenia brutto pracowników.

Opis zadania. W sytuacji, gdy pod koniec miesiąca dział księgowy będzie musiał wydrukować dokumenty zawierające dane o miesięcznych wynagrodzeniach brutto pracowników, system po zalogowaniu się księgowego na stronie udostępni mu odpowiednie opcje do wykonania tej czynności.

Priorytet. 4.

Definition of Done. System umożliwia księgowym drukowanie zestawienia miesięcznego wynagrodzenia brutto pracowników.

4.9 Backlog Item 9

Tytuł zadania. Dodawanie oraz usuwanie pracowników.

Opis zadania. Za pośrednictwem strony zakładowej, kadrowy będzie miał możliwość dodawania nowych pracowników oraz ich usuwania w przypadku zwolnienia. Nowi pracownicy będą dodawani do bazy danych oraz będą mieli możliwość logowania się na stronę zakładu.

Priorytet. 2.

Definition of Done. Kadrowy będzie mógł dodawać i usuwać pracowników z systemu za pomocą strony zakładowej.

4.10 Backlog Item 10

Tytuł zadania. Utworzenie lokalnego serwera HTTP i FTP dla systemu.

Opis zadania. Dostęp do serwera powinien być ograniczony do sieci wewnętrznej w celu zabezpieczenia przed zewnętrznymi atakami. Na serwerze znajdować się będzie usługa WWW z obsługą PHP.

Priorytet. 1.

Definition of Done. Istnieje w pełni działający lokalny serwer HTTP Apache, na którym zostanie zainstalowany interpreter PHP w celu uruchomienia strony internetowej zakładu, która pozwala na kontakt z bazą danych.

4.11 Backlog Item 11

Tytuł zadania. Przywracanie hasła.

Opis zadania. Strona zakładowa będzie posiadać funkcję, dzięki której kierownik, kadrowy czy pracownik będzie mógł wysłać prośbę o przypomnienie hasła. W tym celu dodatkowo uruchomiona zostanie usługa pocztowa SMTP na serwerze. Wiadomość z nowym hasłem zostanie wysłana na e-mail danej osoby. Po zalogowaniu się za pomocą wygenerowanego hasła, system wymusi jego zmianę na nowe prywatne hasło.

Priorytet. 5.

Definition of Done. Działa serwer SMTP wysyłający maile, który umożliwia kierownikowi, kadrowemu oraz pracownikom przypomnienie hasła oraz jego zmianę w systemie.

4.12 Backlog Item 12

Tytuł zadania. Uprawnienia kierownika do wszystkich funkcji.

Opis zadania. W sytuacjach wyjątkowych kierownik zakładu powinien mieć możliwość dokonywania wszelkich zmian dostępnych dla innych osób poprzez stronę internetową. Odbywać się to będzie przez odpowiednią zakładkę na stronie, która będzie rejestrować każdą zmianę wraz z powodem jej wykonania.

Priorytet. 3.

Definition of Done. Kierownik jest w stanie dokonać następujących zmian w systemie poprzez stronę internetową:

- Dodawanie nowego pracownika.
- Usuwanie istniejącego pracownika.
- Edycja danych osobowych pracownika.
- Możliwość sprawdzenia miesięcznego zestawienia wypłaty pracowników.
- Zmiany stawki wynagrodzenia pracownika.

4.13 Backlog Item 13

Tytuł zadania. Utworzenie bazy danych dla systemu wraz ze skonfigurowaniem serwera baz danych MySQL.

Opis zadania. Utworzona baza ma gromadzić informację niezbędne do funkcjonowania systemu. Na przykład po przyłożeniu karty do czytnika RFID jest wprowadzona informacja, do odpowiedniej tabeli w bazie, o godzinie wejścia lub wyjścia pracownika na teren zakładu.

Priorytet. 1.

Definition of Done. Działa serwer baz danych MySQL obsługujący bazę danych z danymi pracowników oraz godzinami ich wejścia lub wyjścia na teren zakładu.

5 Sprint 1

5.1 Cel

Celem pierwszego sprintu jest stworzenie początkowej wersji systemu pozwalającej rejestrować wejścia i wyjścia pracowników przy pomocy urządzenia Raspberry Pi wraz z dołączonym modulem RFID (karta, czytnik kart). Te dane będą przesyłane na wcześniej utworzony i skonfigurowany serwer bazy danych. Utworzona zostanie również pierwotna wersja strony na serwerze HTTP, na której wyświetlane będą takie informacje jak: imię i nazwisko pracownika, data, godzina wejścia lub wyjścia, ID karty.

5.2 Sprint Planning/Backlog

Tytuł zadania. Rejestrowanie wejścia i wyjścia pracownika z zakładu.

- Estymata: Large przy użyciu skali rozmiarów T-shirtów (13 osobogodzin).

Tytuł zadania. Utworzenie lokalnego serwera HTTP i FTP dla systemu.

- Estymata: Small przy użyciu skali rozmiarów T-shirtów (4 osobogodzin).

Tytuł zadania. Utworzenie pierwotnej wersji strony www do obsługi systemu.

- Estymata: Small przy użyciu skali rozmiarów T-shirtów (4 osobogodziny).

Tytuł zadania. Utworzenie bazy danych dla systemu wraz ze skonfigurowaniem serwera baz danych MySQL.

- Estymata: Medium przy użyciu skali rozmiarów T-shirtów (7 osobogodzin).

5.3 Realizacja

Tytuł zadania. Rejestrowanie wejścia i wyjścia pracownika z zakładu.

Wykonawca. Mateusz Morawa, Łukasz Horowski.

Realizacja. W celu rejestracji wejść i wyjść pracowników z zakładu użyty został czytnik RFID RC522, działający w standardzie Mifare, podłączony do odpowiednich pinów Raspberry Pi tak jak przedstawiono poniżej:

- SDA — Pin 24
- SCK — Pin 23
- MOSI — Pin 19
- MISO — Pin 21
- GND — Pin 6
- RST — Pin 22
- 3.3V — Pin 1

W menu konfiguracyjnym Raspberry Pi, otworzonym za pomocą komendy:

```
sudo raspi-config
```

został włączony szeregowy interfejs SPI (ang. Serial Peripheral Interface) zapewniający komunikację z wcześniej wspomnianym czytnikiem. Po wykonaniu odpowiednich podłączeń oraz włączeniu SPI zostały pobrane i zainstalowane biblioteki odpowiedzialne za obsługę modułu RFID. Wykonano niezbędne testy potwierdzające prawidłowe działanie czytnika. Dodatkowo w celach informacyjnych obok czytnika umieszczono buzzer oraz 2 diody LED sygnalizujące odczyt karty. W dalszej kolejności utworzony został skrypt w języku Python, który umożliwia odczytanie numeru ID z karty a następnie przesłanie go do bazy danych oraz steruje diodami i buzzerem.

Tytuł zadania. Utworzenie lokalnego serwera HTTP i FTP dla systemu.

Wykonawca. Mateusz Morawa, Łukasz Horowski.

Realizacja. Jako system operacyjny dla naszego serwera (znajdującego się na Raspberry Pi) został wybrany Raspbian Stretch. W celu jego instalacji użyty został program Win32DiskImager, który pozwala na przeniesienie systemu zapisanego w pliku obrazu o rozszerzeniu .iso na kartę pamięci. Na serwerze główne konta zostały zabezpieczone mocnymi hasłami w celu ograniczenia dostępu osobom niepowołanym oraz utworzono użytkownika „informatyk” mającego uprawnienia do zarządzania serwerem. W celu ułatwienia wykonywania zadań administracyjnych na serwerze uruchomiono usługę SSH oraz pulpit zdalny VNC (ang. Virtual Network Computing). Używając podstawowej komendy instalacyjnej:

```
sudo apt-get install
```

zainstalowano niezbędne pakiety serwera HTTP czyli: apache2 oraz php7. Po poprawnej realizacji tego zadania sprawdzono ich działanie oraz odpowiednio je skonfigurowano. Dodatkowo w celu ułatwienia pracy nad stroną internetową zakładu zainstalowano oraz skonfigurowano serwer FTP (ang. File Transfer Protocol) umożliwiający dwukierunkowy transfer plików z klientem FTP przy użyciu połączenia TCP.

Tytuł zadania. Utworzenie pierwotnej wersji strony www do obsługi systemu.

Wykonawca. Łukasz Mirek, Grzegorz Jasiński, Łukasz Horowski.

Realizacja. Strona została utworzona na lokalnym serwerze przy pomocy języków programowania PHP oraz HTML z CSS. Kod zapisany został przy użyciu programu Notepad++, który umożliwia kontrolę poprawności składni wybranego przez nas języka, co ułatwia pracę nad projektem. Aby zobaczyć pierwsze efekty pracy nad stroną skorzystaliśmy z oprogramowania "Xampp", dzięki czemu mogliśmy zasymulować łączenie się z bazą danych oraz obsługę skryptów PHP na naszej stronie. Po wykonaniu pierwotnej wersji strony i przetestowaniu jej działania została ona przeniesiona na urządzenie Raspberry Pi. Na obecnym etapie strona ma pozwalać na sprawdzenie danych odnośnie godzin wejścia i wyjścia pracowników na teren zakładu rejestrowanych za pomocą czytnika kart RFID. Z realizacją tej części projektu, wykonawcy nie mieli żadnych problemów. Została ona wykonana sprawnie i bez większych przeszkód.

Kod programu dotyczący łączenia się z bazą danych:

```
<?php
$config['db']['host'] = 'localhost';
$config['db']['user'] = 'user';
$config['db']['pass'] = 'zpi';
$config['db']['dbname'] = 'zpi';
$db = mysqli_connect($config['db']['host'], $config['db']['user'],
$config['db']['pass'], $config['db']['dbname'])
or die('Błąd połączenia z bazą');
mysqli_query($db,"SET NAMES 'utf8'") or die('Błąd kodowania nazwy');
?>
```

Tytuł zadania. Utworzenie bazy danych dla systemu wraz ze skonfigurowaniem serwera baz danych MySQL.

Wykonawca. Łukasz Horowski, Bartłomiej Kmak.

Realizacja. Baza danych została utworzona na lokalnym serwerze przy pomocy języka SQL. Kod zapisany został przy użyciu programu Notepad++, który umożliwia kontrolę poprawności składni języka sql, co ułatwia pracę nad projektem. Aby zobaczyć pierwsze efekty pracy nad bazą danych skorzystaliśmy z oprogramowania "Xampp", dzięki niemu mogliśmy sprawdzić czy utworzony kod bazy danych działa oraz czy są tworzone poprawnie rekordy w bazie. Po wykonaniu pierwotnej wersji bazy i przetestowaniu jej działania została ona przeniesiona na urządzenie Raspberry Pi. Na obecnym etapie baza ma pozwalać na gromadzenie danych odnośnie godzin wejścia i wyjścia pracowników na teren zakładu rejestrowanych za pomocą czytnika kart RFID. Z realizacją tej części projektu, wykonawcy nie mieli żadnych problemów została ona wykonana sprawnie i bez większych przeszkód.

Kod w języku SQL przedstawiający utworzenie konta użytkownika user do zarządzania bazą z uprawnieniami SELECT oraz możliwością wykonywania zapytań typu DML.

```
CREATE USER 'user'@'%' IDENTIFIED BY 'zpi';
GRANT SELECT, INSERT, UPDATE, DELETE ON zpi.* TO 'user'@'%';
```

Kod przedstawiający ustawienie metody kodowania znaków w bazie danych na UTF-8:

```
CREATE DATABASE zpi CHARACTER SET utf8 COLLATE utf8_general_ci;
```

Kod przedstawiający utworzenie tabeli Stanowisko:

```
CREATE TABLE Stanowisko (
  Id INT NOT NULL auto_increment,
  Nazwa VARCHAR(100) NOT NULL,
  PRIMARY KEY(Id)
)ENGINE = InnoDB;
```

5.4 Sprint Review/Demo

Dnia 29.11.17 r. odbyła się prezentacja dema produktu w ramach pierwszego Sprintu. Wszystkie założone cele zawarte w backlog itemach zaplanowanych na ten sprint zostały zrealizowane bez żadnych przeszkód. Prezentacja sprintu zawierała pokaz działającej pierwszej wersji strony zakładowej, która wyświetlała godziny wejścia/wyjścia pracowników oraz stan osobowy załogi obecnej w danym momencie na zakładzie. Zrzut ze strony poniżej:



Godziny wejścia			Godziny wyjścia			Obecni na zakładzie	
Nazwisko	Imię	Godzina wejścia	Nazwisko	Imię	Godzina wyjścia	Nazwisko	Imię
Nowak	Jan	2017-11-27 23:54:17	Nowak	Jan	2017-11-27 23:54:33	Nowak	Jan
Nowak	Jan	2017-11-27 23:54:52	Nowak	Jan	2017-11-27 23:54:55	Kowalski	Jan
Nowak	Jan	2017-11-27 23:55:14	Kozak	Michał	2017-11-27 23:55:55		
Kozak	Michał	2017-11-27 23:55:49					
Kowalski	Jan	2017-11-27 23:56:03					

Dane przedstawione na stronie pochodziły z bazy danych, która otrzymywała dane za każdym razem, gdy pracownik przykładał kartę do czytnika RFID. Buzzer oraz diody LED zaimplementowane w systemie informowały czy dane pracownika zostały poprawnie odczytane poprzez zgaszenie zielonej diody i zaświecenie się czerwonej diody, czemu towarzyszył pojedynczy dźwięk. Następnie po około 3 sekundach urządzenie sygnalizowało gotowość do kolejnego odczytu poprzez podwójny dźwięk i przywrócenie pierwotnego stanu diod. W tym momencie kolejny pracownik mógł już przyłożyć swoją kartę do czytnika a na stronie pojawiały się odpowiednie zmiany.

6 Sprint 2

6.1 Cel

Celem drugiego sprintu jest rozbudowa strony internetowej zakładu o takie funkcje jak: logowanie, funkcję dodawania i usuwania pracowników oraz możliwość obliczania przepracowanego przez nich czasu. W efekcie dostęp do strony zostanie zabezpieczony przed nieuprawnionymi osobami a kadrowy i księgowy zyskają nowe narzędzia ułatwiające ich pracę. Ten pierwszy uzyska kontrolę nad zmianami liczebności załogi zakładu, natomiast ten drugi będzie mógł w łatwy sposób dokonywać co miesięcznych zestawień przepracowanych godzin przez zatrudnione osoby. Ponadto dodana zostanie funkcja ułatwiająca dodawanie nowych kart pracowniczych do bazy danych w ten sposób, iż nie trzeba będzie ręcznie wprowadzać numeru ID karty do bazy, ale wystarczy samo jej przyłożenie do czytnika RFID a program automatycznie wykona powyższą operację.

6.2 Sprint Planning/Backlog

Tytuł zadania. Podsumowanie przepracowanych godzin.

- Estymata: Large przy użyciu skali rozmiarów T-shirtów (10 osobogodzin).

Tytuł zadania. Logowanie do strony zakładowej.

- Estymata: Medium przy użyciu skali rozmiarów T-shirtów (5 osobogodzin).

Tytuł zadania. Dodawanie oraz usuwanie nowych pracowników.

- Estymata: Medium przy użyciu skali rozmiarów T-shirtów (5 osobogodzin).

Tytuł zadania. Dodawanie nowych kart do bazy danych.

- Estymata: Medium przy użyciu skali rozmiarów T-shirtów (6 osobogodzin).

6.3 Realizacja

Tytuł zadania. Podsumowanie przepracowanych godzin.

Wykonawca. Łukasz Horowski, Mateusz Morawa.

Realizacja. W celu realizacji powyższego zadania została stworzona nowa zakładka w menu strony zakładowej zawierająca kod wykonany w języku PHP. Po jej kliknięciu następuje sprawdzenie, kiedy miało miejsce ostatnie podsumowanie przepracowanych godzin oraz wyświetlenie daty tego zestawienia. Jeśli operacja ta odbyła się w innym miesiącu niż obecny to pojawia się możliwość zainicjowania nowego podsumowania poprzez naciśnięcie przycisku "Podsumuj". Wtedy uruchomiona zostaje pętla sprawdzająca i wyświetlająca czas pracy każdego pracownika, znajdującego się w bazie danych, w poszczególnych dniach

miesiąca. Otrzymane w ten sposób wartości są zaokrąglane poprzez funkcję `floor` w celu uzyskania pełnych godzin spędzonych na zakładzie. Po zakończeniu wykonywania się pętli zostaje wyświetlona tabela podsumowująca ilość godzin przepracowanych przez każdego pracownika w danym miesiącu. Powyższe zadanie było jak dotąd najtrudniejsze ze wszystkich do wykonania - w szczególności kwestia obliczania czasu pracy pracownika, który w ciągu dnia kilkakrotnie przychodził i opuszczał zakład, sprawiła najwięcej problemów, dlatego będzie w pełni zaimplementowana dopiero w kolejnym sprincie jako osobne zadanie. Kod programu wyświetlający miesięczne podsumowanie godzin przepracowanych przez zatrudnione osoby (środowisko *verbatim*):

```
for($i=0;$i<=$liczbapracownikow-1;$i++){
    for ($j=0;$j<=3;$j++){
        switch($j){
            case 0:
                echo "Pracownik : ";
                break;
            case 1:
                echo " WE : ";
                break;
            case 2:
                echo " WY : ";
                break;
            case 3:
                echo " Godziny: ";
                break;
        }
        echo $podsum[$i][$j];
    }
    echo "</BR>";
}
```

Tytuł zadania. Logowanie do strony zakładowej.

Wykonawca. Łukasz Horowski, Bartłomiej Kmak, Grzegorz Jasiński, Łukasz Mirek.

Realizacja. W języku PHP został napisany fragment kodu uzupełniający stronę zakładową, odpowiadający za logowanie się użytkowników. Został on napisany w edytorze tekstu Notepad++, który pozwala na sprawną modyfikację kodu oraz kontrolę składni wybranego języka. Po wejściu na stronę wyświetla się formularz z miejscem na wpisanie loginu użytkownika oraz hasła. Na stronę będą mogli się zalogować kierownik, księgowy, kadrowy oraz pracownicy dodani do bazy MySQL. Pracownik, który wejdzie na stronę i będzie chciał się zalogować musi wpisać login oraz hasło. Fragment kodu przedstawionego poniżej programu sprawdzi w bazie czy istnieje pracownik z podanym loginem oraz czy podane hasło odpowiada zaszyfrowanemu hasłu w bazie danych. Jeśli login i hasło zostaną podane poprawnie, pracownik otrzyma dostęp do zawartości strony zakładowej i odpowiednich funkcji stosownych do jego stanowiska. W razie

podania błędnych danych do logowania, zostaną wyświetlone odpowiednie komunikaty: dla błędnego loginu "Nie istnieje takie konto użytkownika!" lub dla błędnego hasła "Podane hasło jest nieprawidłowe!". Kod programu (środowisko *verbatim*):

```
if(!empty($_POST)){
if (!empty($_POST['login']) && (!empty($_POST['password']))) {
$login=$_POST['login'];
$password=$_POST['password'];
//sprawdzenie poprawności
$dane = mysqli_query($db , "SELECT * FROM websiteusers
WHERE Login='$login' LIMIT 1");
if (mysqli_num_rows($dane)===1){
$dane=mysqli_fetch_assoc($dane);
if (md5($password)===$dane['Haslo']){
//logowanie i pobranie sesji
$_SESSION['id'] = $dane['Id'];
header("Location: ?a=start");
}
else {
echo '<CENTER>Podane hasło jest nieprawidłowe!</CENTER><BR>';
}
}
else {
echo '<CENTER>Nie istnieje takie konto użytkownika!</CENTER><BR>';
}
}
}
echo '<CENTER>Uzupełnij wszystkie dane!</CENTER><BR>';
}
};
```

Tytuł zadania. Dodawanie oraz usuwanie pracowników.

Wykonawca. Łukasz Horowski, Bartłomiej Kmak, Grzegorz Jasiński, Łukasz Mirek.

Realizacja. W języku PHP został napisany skrypt i umieszczony na stronie zakładowej w odpowiedniej zakładce menu, odpowiadający za dodanie do bazy danych pracownika za pomocą zamieszczonego formularza. Podczas tej operacji kierownik bądź kadrowy, po uzupełnieniu danych osobowych nowego pracownika, będzie tworzył dla niego konto umożliwiające logowanie na stronę zakładową z odpowiednim loginem oraz hasłem. Za usuwanie pracownika odpowiada osobny skrypt znajdujący się w osobnej zakładce menu. Dzięki niemu kierownik bądź kadrowy po znalezieniu na liście pracowników osoby do zwolnienia wpisuje jej ID w formularzu i poprzez kliknięcie na przycisk dokonuje jej usunięcia z bazy danych. Kod programu odpowiedzialny za dodawanie pracownika. (środowisko *verbatim*):

```
<?php
if(!empty($_POST)){
```

```

if (empty($_POST['login']) || empty($_POST['password'])
|| empty($_POST['email']))
echo '<CENTER>Uzupełnij wszystkie pola!</CENTER><BR>';
else
{
$connection = new mysqli("localhost","root", "", "zpi");
$login = ($_POST['login']);
$password = ($_POST['password']);
$email = ($_POST['email']);
$idkarty = $_POST['idkarty'];

$idstanowiska = ($_POST['idstanowiska']);
$nazwisko = $_POST['nazwisko'];
$imie = $_POST['imie'];
$datazatrudnienia = $_POST['datazatrudnienia'];
$miestowosc = $_POST['miestowosc'];
$ulica = $_POST['ulica'];
$nrdomu = $_POST['nrdomu'];
$nmieszkania = $_POST['nmieszkania'];
$kodpocztowy = $_POST['kodpocztowy'];
$stawka = $_POST['stawka'];
$connection->query("SELECT * FROM stanowisko WHERE id =$idstanowiska");
$connection->query("SELECT * FROM karty WHERE id =$idkarty");
//sprawdzanie poprawności znaków
if (! ctype_alnum($login) )
echo '<CENTER>Wprowadzono nieprawidłową nazwę
użytkownika!</CENTER><BR>';
else if (! filter_var($email,FILTER_VALIDATE_EMAIL) )
echo '<CENTER>Wprowadzono
nieprawidłowy adres email!</CENTER><BR>!';
else{

// sprawdzanie czy dane istnieją
$login2 = ($connection->query("SELECT Login FROM websiteusers
WHERE Login='$login' "));
$login2 = mysqli_fetch_assoc($login2);
$email2 = ($connection->query("SELECT Email FROM websiteusers
WHERE Email='$email' "));
$email2 = mysqli_fetch_assoc($email2);
//warunki jeżeli zajęty login lub email
if (($login2['Login'])!='') echo '<CENTER>Nazwa użytkownika
jest już zajęta!</CENTER><BR>';
elseif (($email2['Email'])!='') echo '<CENTER>Wprowadzony
adres E-mail jest już zajęty!</CENTER><BR>';
else{
// zakładanie konta
$password = md5($password);
$connection->query("INSERT INTO users (Login>Password>Email)
VALUES ('$login','$password','$email')");
$connection->query("INSERT INTO pracownicy (IdKarty, Nazwisko,

```

```

Imie, Miejscowosc, Ulica, NrDomu, NrMieszkania, KodPocztowy, Stawka)
VALUES ('$idkarty', '$nazwisko', '$imie', '$miejscowosc', '$ulica',
'$nrdomu', '$nrmieszkania', '$kodpocztowy', '$stawka');"");
echo "<CENTER>Konto zostało pomyślnie założone!</CENTER>";
}
}
}
};
?>

```

Kod programu odpowiedzialny za usuwanie pracownika. (środowisko *verbatim*):

```

if ( $id > 1){
$zapytanie="DELETE from RejestrWe WHERE IdPracownika='$id'";
$wykonaj = mysqli_query($db,$zapytanie);
$zapytanie="DELETE from RejestrWy WHERE IdPracownika='$id'";
$wykonaj = mysqli_query($db,$zapytanie);
$zapytanie="DELETE from WebSiteUsers WHERE IdPracownika='$id'";
$wykonaj = mysqli_query($db,$zapytanie);
$zapytanie="DELETE from ZestawienieDzienne WHERE IdPracownika='$id'";
$wykonaj = mysqli_query($db,$zapytanie);
$zapytanie="DELETE from ZestawienieMiesieczne WHERE IdPracownika='$id'";
$wykonaj = mysqli_query($db,$zapytanie);
$zapytanie="DELETE from Pracownicy WHERE Id='$id'";
$wykonaj = mysqli_query($db,$zapytanie);
}
else{
echo "Nie można zwolnić głównego kierownika!";
}
}

```

Tytuł zadania. Dodawanie nowych kart do bazy danych.

Wykonawca. Łukasz Horowski, Mateusz Morawa.

Realizacja. W języku Python został utworzony program, który za pomocą czytnika RFID odczytuje numery ID kart i dodaje je do bazy danych (tabela Karty). Program ten sprawdza czy odczytane ID istnieje już w tej tabeli czy też nie i wydaje odpowiednie sygnały buzzerem oraz diodami LED. Jeśli ID nie istnieje, zapala się zielona dioda i wydawany jest jeden dłuższy sygnał a po 3 sekundach 2 krótsze, wtedy numer ID jest dopisywany do bazy. W przeciwnym wypadku nadal pali się czerwona dioda a buzzer wydaje 5x krótki sygnał. Kod programu ilustrujący procedurę dodawania nowej karty do bazy danych oraz towarzyszące temu sygnały świetlne i dźwiękowe. (środowisko *verbatim*):

```

if (uid == "None"):
    db.execute("INSERT INTO Karty(IdKarty) VALUES ('+idkarty+')");
    connect.commit()
    print "Dodano nową kartę"
    GPIO.output(11, 0)

```

```
GPIO.output(12, 1)
GPIO.output(13, 1)
time.sleep(0.5)
GPIO.output(13, 0)
time.sleep(2)
GPIO.output(13, 1)
time.sleep(0.2)
GPIO.output(13, 0)
time.sleep(0.2)
GPIO.output(13, 1)
time.sleep(0.2)
GPIO.output(13, 0)
```

6.4 Sprint Review/Demo

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

7 Sprint 3

Tytuł zadania. Obliczanie wynagrodzenia.

- Estymata: Large przy użyciu skali rozmiarów T-shirtów (10 osobogodzin).

7.1 Cel

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

7.2 Sprint Planning/Backlog

Tytuł zadania. «Tytuł».

- Estymata: «szacowana czasochłonność (w „koszulkach”)».

Tytuł zadania. «Tytuł».

- Estymata: «szacowana czasochłonność (w „koszulkach”)».

«Tutaj dodawać kolejne zadania»

7.3 Realizacja

Tytuł zadania. «Tytuł».

Wykonawca. «Wykonawca».

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. «Wykonawca».

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

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

«Tutaj dodawać kolejne zadania»

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? Demonstracja przyrostu produktu».

8 Sprint 4

8.1 Cel

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

8.2 Sprint Planning/Backlog

Tytuł zadania. «Tytuł».

- Estymata: «szacowana czasochłonność (w „koszulkach”)».

Tytuł zadania. «Tytuł».

- Estymata: «szacowana czasochłonność (w „koszulkach”)».

«Tutaj dodawać kolejne zadania»

8.3 Realizacja

Tytuł zadania. «Tytuł».

Wykonawca. «Wykonawca».

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. «Wykonawca».

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

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

«Tutaj dodawać kolejne zadania»

8.4 Sprint Review/Demo

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

Literatura

- [1] S. R. Covey, *7 nawyków skutecznego działania*, Rebis, Poznań, 2007.
- [2] Tobias Oetiker i wsp., Nie za krótkie wprowadzenie do systemu L^AT_EX 2_ε, <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/notesfromtooluser/tag/scrum-by-example>
- [5] https://www.tutorialspoint.com/scrum/scrum_user_stories.htm