

Aplikacja "Memorize!"

Damian Hadała, Dawid Sułowicz, Krzysztof Radoń, Krzysztof Jamro

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	3
2.3	User story 3	3
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
2.9	User story 9	4
2.10	User story 10	4
2.11	User story 11	4
2.12	User story 12	4
2.13	User story 13	5
3	Harmonogram	5
3.1	Rejestr zadań (Product Backlog)	5
3.2	Sprint 1	5
3.3	Sprint 2	5
3.4	Sprint 3	5
3.5	Sprint 4	6
3.6	Sprint 5	6
4	Product Backlog	6
4.1	Backlog Item 1	6
4.2	Backlog Item 2	6
4.3	Backlog Item 3	7
4.4	Backlog Item 4	7
4.5	Backlog Item 5	7
4.6	Backlog Item 6	8
4.7	Backlog Item 7	8
4.8	Backlog Item 8	8
4.9	Backlog Item 9	9
4.10	Backlog Item 10	9
4.11	Backlog Item 11	9
4.12	Backlog Item 12	10
4.13	Backlog Item 13	10
4.14	Backlog Item 14	10
4.15	Backlog Item 15	11
4.16	Backlog Item 16	11
4.17	Backlog Item 17	11
4.18	Backlog Item 18	12

4.19	Backlog Item 19	12
4.20	Backlog Item 20	12
5	Sprint 1	12
5.1	Cel	12
5.2	Sprint Planning/Backlog	13
5.3	Realizacja	14
5.4	Sprint Review/Demo	14
6	Sprint 2	14
6.1	Cel	14
6.2	Sprint Planning/Backlog	15
6.3	Realizacja	15
6.4	Sprint Review/Demo	16

1 Opis projektu

1.1 Członkowie zespołu

1. Krzysztof Radoń (kierownik projektu).
2. Damian Hadała.
3. Dawid Sułowicz.
4. Krzysztof Jamro.

1.2 Cel projektu (produkt)

Aplikacja internetowa "Memorize!" pozwalająca użytkownikom zapisywać interesujące ich informacje i tym samym dodawać wpisy do dziennika podczas poznawania danego zagadnienia. Po zalogowaniu się użytkownik będzie miał możliwość utworzenia nowego tematu, dodania nowego wpisu, a także odczytywania i edytowania istniejących wpisów.

1.3 Potencjalny odbiorca produktu (klient)

Użytkownicy poznający nowe zagadnienia prowadzą dziennik, w którym zapisują to, czego się nauczyli. Dzięki aplikacji "Memorize!" proces nauki i zapamiętywania może być znacznie efektywniejszy.

1.4 Metodyka

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

2 Wymagania użytkownika

2.1 User story 1

Jako użytkownik aplikacji "Memorize!" chcę mieć możliwość dostępu do listy tematów, aby móc je w dowolnej chwili podglądać i sprawdzać ich zawartość.

2.2 User story 2

Jako użytkownik aplikacji "Memorize!" chcę mieć możliwość tworzenia nowych tematów, abym później mógł do nich dodawać powiązane wpisy.

2.3 User story 3

Jako użytkownik aplikacji "Memorize!" chcę mieć możliwość sprawdzenia daty i godziny umieszczenia każdego wpisu, aby móc weryfikować poczynione przeze mnie postępy w nauce.

2.4 User story 4

Jako użytkownik aplikacji "Memorize!" chcę by długie wpisy nie były wyświetlane w postaci ciągłego strumienia tekstu, aby poprawić czytelność podczas ich przeglądania.

2.5 User story 5

Jako użytkownik aplikacji "Memorize!" chcę żeby moje wpisy były prywatne tak, abym tylko ja mógł nimi zarządzać.

2.6 User story 6

Jako użytkownik aplikacji "Memorize!" chcę mieć możliwość edytowania wpisów, abym mógł zmieniać ich treść w pożądanej przeze mnie chwili.

2.7 User story 7

Jako nowy użytkownik aplikacji "Memorize!" chcę mieć możliwość zarejestrowania własnego profilu, abym mógł dodawać własne treści.

2.8 User story 8

Jako użytkownik aplikacji "Memorize!" chcę żeby moje konto było chronione hasłem tak, aby inny użytkownik nie uzyskał do niego dostępu.

2.9 User story 9

Jako niezalogowany użytkownik aplikacji "Memorize!" chcę zobaczyć instrukcję wyjaśniającą przeznaczenie tej aplikacji, abym mógł zapoznać się z jej działaniem.

2.10 User story 10

Jako użytkownik aplikacji "Memorize!" chcę mieć dostęp z każdego komputera do listy selektorów i reguł, których cały czas się uczę.

2.11 User story 11

Jako użytkownik aplikacji "Memorize!" chcę mieć dostęp z każdego komputera do listy tematów prowadzonych przeze mnie zajęć, abym mógł się w nich zorientować pracując w kilku placówkach edukacyjnych.

2.12 User story 12

Jako użytkownik aplikacji "Memorize!" chcę mieć dostęp z każdego komputera do listy moich inspiracji, abym nie musiał szukać ich papierowej wersji na wypadek, gdybym ją zgubił.

2.13 User story 13

Jako użytkownik aplikacji "Memorize!" chcę mieć dostęp do aplikacji z urządzenia mobilnego, aby móc z niej swobodnie korzystać z dala od komputera.

3 Harmonogram

3.1 Rejestr zadań (Product Backlog)

- Data rozpoczęcia: 24.10.2017.
- Data zakończenia: 31.10.2017.

3.2 Sprint 1

- Data rozpoczęcia: 31.10.2017.
- Data zakończenia: 14.11.2017.
- Scrum Master: Damian Hadała.
- Product Owner: Krzysztof Radoń.
- Development Team: Krzysztof Jamro, Dawid Sułowicz.

3.3 Sprint 2

- Data rozpoczęcia: 14.11.2017.
- Data zakończenia: 28.11.2017.
- Scrum Master: Krzysztof Jamro.
- Product Owner: Damian Hadała.
- Development Team: Krzysztof Radoń, Dawid Sułowicz.

3.4 Sprint 3

- Data rozpoczęcia: 28.11.2017.
- Data zakończenia: 12.12.2017.
- Scrum Master: Dawid Sułowicz.
- Product Owner: Krzysztof Jamro.
- Development Team: Krzysztof Radoń, Damian Hadała.

3.5 Sprint 4

- Data rozpoczęcia: 12.12.2017.
- Data zakończenia: 09.01.2018.
- Scrum Master: Krzysztof Radoń.
- Product Owner: Dawid Sułowicz.
- Development Team: Damian Hadała, Krzysztof Jamro.

3.6 Sprint 5

- Data rozpoczęcia: 09.01.2018.
- Data zakończenia: 23.01.2018.
- Scrum Master: Krzysztof Radoń.
- Product Owner: Krzysztof Jamro.
- Development Team: Damian Hadała, Dawid Sułowicz.

4 Product Backlog

4.1 Backlog Item 1

Tytuł zadania. Utworzenie projektu w Django.

Opis zadania. Plik o nazwie `manage.py` jest niewielkim programem przeznaczonym do wydawania poleceń odpowiednim komponentom frameworka Django. Polecenia tego programu wykorzystamy do zarządzania pracą serwera i bazy danych. Plik `settings.py` kontroluje sposób współdziałania Django z systemem operacyjnym oraz pozwala na zarządzanie projektem. Plik `urls.py` wskazuje Django, które strony powinny zostać utworzone w odpowiedzi na żądania pochodzące z przeglądarki internetowej.

Priorytet. Wysoki.

Definition of Done. Zostanie utworzona struktura katalogów, która ułatwi nam wdrożenie aplikacji na serwerze.

4.2 Backlog Item 2

Tytuł zadania. Utworzenie bazy danych.

Opis zadania. Zostaną utworzone tabele bazy danych niezbędne do przechowywania informacji, których będziemy używać w projekcie.

Priorytet. Wysoki.

Definition of Done. Większość informacji związanych z projektem jest przez Django przechowywana w bazie danych. Konieczne jest zatem utworzenie bazy danych, z którą framework będzie mógł pracować.

4.3 Backlog Item 3

Tytuł zadania. Uruchomienie aplikacji.

Opis zadania. Utworzymy aplikację przeznaczoną do wykonywania większości zadań budowanego projektu. Później dodamy następną aplikację odpowiedzialną za zarządzanie kontami użytkowników. Pliku `models.py` będziemy używać do zdefiniowania danych, którymi będziemy chcieli zarządzać w tej aplikacji.

Priorytet. Wysoki.

Definition of Done. Na projekt Django składa się grupa poszczególnych aplikacji, które ze sobą współpracują, co pozwala na działanie projektu jako całości.

4.4 Backlog Item 4

Tytuł zadania. Zdefiniowanie klasy `Topic`.

Opis zadania. Utworzymy klasę o nazwie `Topic` dziedziczącą po klasie `Model`, czyli klasie nadrzędnej oferowanej przez Django. W klasie `Topic` znajdą się tylko dwa atrybuty: `text` i `dateAdded`.

Priorytet. Wysoki.

Definition of Done. Każdy użytkownik musi mieć możliwość utworzenia wielu różnych tematów. Poszczególne wpisy będą powiązane z tematem i zostaną wyświetlone. Konieczne jest również przechowywanie znacznika czasu dla każdego wpisu, aby móc pokazać użytkownikowi kiedy dany wpis został dodany.

4.5 Backlog Item 5

Tytuł zadania. Utworzenie i konfiguracja konta superużytkownika.

Opis zadania. Podanie nazwy oraz hasła. Django nie przechowuje wprowadzonego hasła, ale wygenerowany na jego podstawie ciąg tekstowy nazywany wartością hash. Za każdym razem, gdy użytkownik wprowadzi hasło, Django wygeneruje dla niego wartość hash i porówna ją z przechowywaną. Jeżeli obie wartości hash są identyczne, oznacza to, że użytkownik podał prawidłowe hasło i został uwierzytelniony. Jeśli atakujący uzyska dostęp do bazy danych aplikacji "Memorize!", będzie w stanie odczytać jedynie przechowywane tam wartości hash, a nie hasła.

Priorytet. Średni.

Definition of Done. Aby efektywnie móc administrować aplikacją sieciową, właściciel zwykle potrzebuje mieć bezpieczny dostęp do wszystkich przechowywanych informacji.

4.6 Backlog Item 6

Tytuł zadania. Zdefiniowanie klasy Entry.

Opis zadania. Klasa Entry dziedziczy po klasie bazowej Django, podobnie zresztą jak klasa Topic. Pierwszy atrybut (topic) to egzemplarz ForeignKey, czyli klucz zewnętrzny. Pojecie to oznacza odwołanie do innego rekordu w bazie danych. Każdemu utworzonemu tematowi zostanie przypisany klucz (identyfikator). Ponieważ cały wpis może mieć postać dość długiego tekstu, nakazujemy wyświetlenie jedynie pierwszych 50 znaków przechowywanych przez atrybut text. Dołączamy również wielokropek, aby wyraźnie wskazać, że nie został wyświetlony pełny wpis.

Priorytet. Wysoki.

Definition of Done. Aby użytkownik mógł dodawać nowe wpisy, konieczne jest zdefiniowanie modelu dla wpisów. Każdy wpis musi być powiązany z określonym tematem. Tego rodzaju powiązanie jest nazywane związkiem typu wiele do jednego, co oznacza, że z jednym tematem może być powiązanych wiele wpisów.

4.7 Backlog Item 7

Tytuł zadania. Mapowanie adresu URL strony głównej.

Opis zadania. Przygotujemy adres URL strony głównej. Będzie to bazowy adres URL podawany przez użytkownika, aby uzyskać dostęp do aplikacji.

Priorytet. Wysoki.

Definition of Done. Użytkownik żąda wyświetlenia strony internetowej przez wpisanie jej adresu URL w przeglądarce internetowej oraz przez klikanie łączy. Konieczne jest więc ustalenie, jakiego adresu URL będziemy używać.

4.8 Backlog Item 8

Tytuł zadania. Utworzenie widoku strony głównej.

Opis zadania. Funkcja widoku pobiera informacje z żądania, przygotowuje dane niezbędne do wygenerowania strony, a następnie przekazuje je do przeglądarki internetowej, używając przy tym szablonu, który definiuje wygląd strony.

Priorytet. Wysoki.

Definition of Done. Utworzymy widok dla strony głównej budowanej aplikacji.

4.9 Backlog Item 9

Tytuł zadania. Utworzenie szablonu strony głównej aplikacji "Memorize!".

Opis zadania. Po zdefiniowaniu adresu URL dla strony głównej oraz utworzeniu jej funkcji widoku utworzymy teraz prosty szablon. Po wykonaniu żądania do bazowego adresu URL strony głównej wyświetlimy stronę zbudowaną w piku index.html. Szablon określa strukturę dla strony internetowej, a także definiuje wygląd tej strony. Dzięki szablónowi możemy uzyskać dostęp do wszelkich danych dostarczanych przez widok.

Priorytet. Wysoki.

Definition of Done. Naszym celem jest zapewnienie tego, aby aplikacja "Memorize!" działała zgodnie z oczekiwaniami, w tym momencie utworzymy jedynie bardzo prostą stronę. Na obecnym etapie prac strona główna będzie wyświetlała jedynie tytuł i krótki opis.

4.10 Backlog Item 10

Tytuł zadania. Utworzenie szablonu bazowego.

Opis zadania. Zdarza się, że pewne elementy powtarzają się na wszystkich stronach. Zamiast umieszczać je bezpośrednio na każdej stronie, lepiej zrobimy, jeśli przygotujemy szablon bazowy zawierający powtarzające się elementy, a następnie zastosujemy na nich dziedziczenie po szablonie. W szablonie potomnym trzeba umieścić jedynie tę treść, która jest unikatowa dla danej strony.

Priorytet. Bardzo wysoki.

Definition of Done. Przygotujemy szablon bazowy, po którym będą dziedziczyć wszystkie szablony projektu.

4.11 Backlog Item 11

Tytuł zadania. Utworzenie strony wyświetlającej listę wszystkich tematów.

Opis zadania. Definiujemy adres URL dla strony tematów. Powszechne jest wybieranie takiego adresu URL, aby odzwierciedlał rodzaj informacji przedstawianych na tej stronie. Wykorzystamy słowo topics - dopisanie go w przeglądarce do adresu URL strony głównej spowoduje przejście na stronę tematów. Przygotowujemy zapytanie do bazy danych. Interesują nas obiekty Topic posortowane według atrybutu dateAdded.

Priorytet. Bardzo wysoki.

Definition of Done. Na treść tej strony będzie składała się wypunktowana lista wszystkich tematów. Jeśli lista nie będzie zawierała żadnych elementów zostanie wyświetlony komunikat o tym, że użytkownik nie dodał jeszcze żadnego tematu

4.12 Backlog Item 12

Tytuł zadania. Utworzenie strony wyświetlającej listę wszystkich wpisów dla poszczególnych tematów.

Opis zadania. Ponownie zdefiniujemy nowy wzorzec adresu URL, utworzymy widok oraz szablon. Pobieramy wpisy powiązane z danym tematem i układamy je według atrybutu `dateAdded`, ale w odwrotnej kolejności. To znaczy, że ostatnio dodane wpisy będą wyświetlane jako pierwsze. Szablon tematu powinien wyświetlać nazwę tematu oraz powiązane z nim wpisy. Jeśli do danego tematu nie został jeszcze dodany żaden wpis, użytkownik powinien zostać o tym poinformowany.

Priorytet. Bardzo wysoki.

Definition of Done. Ta strona pozwoli nam skoncentrować się na pojedynczym temacie. Będzie wyświetlać nazwę tematu oraz wszystkie powiązane z nim wpisy.

4.13 Backlog Item 13

Tytuł zadania. Łącza ze strony tematów.

Opis zadania. Zanim będzie można wyświetlić stronę danego tematu w przeglądarce internetowej, konieczne jest zmodyfikowanie szablonu tematów tak, aby każde łącza tematu prowadziło na odpowiednią stronę.

Priorytet. Wysoki.

Definition of Done. Od tej chwili każdy temat na liście jest łączem, którego kliknięcie powoduje przejście na odpowiednią stronę.

4.14 Backlog Item 14

Tytuł zadania. Umożliwienie użytkownikom dodawania nowych tematów.

Opis zadania. Zanim przejdziemy do budowania systemu uwierzytelniania przeznaczonego do tworzenia kont, zaczniemy od dodania pewnych stron pozwalających użytkownikom na wprowadzanie danych. Na razie tylko superużytkownik może wprowadzać dane za pomocą witryny administracyjnej. Ponieważ nie chcemy, aby użytkownicy korzystali z witryny administracyjnej, użyjemy oferowanych przez Django narzędzi do budowania formularzy, by w ten sposób przygotować strony pozwalające użytkownikom wprowadzać dane. Definiujemy

adres URL, a później tworzymy funkcję widoku i szablon. Dodajemy nowy moduł o nazwie `forms.py` zawierający formularze.

Priorytet. Wysoki.

Definition of Done. Umożliwimy użytkownikom dodawanie nowych tematów i wpisów oraz edytowanie już istniejących wpisów.

4.15 Backlog Item 15

Tytuł zadania. Utworzenie formularza dla nowego tematu.

Opis zadania. Podane przez użytkownika informacje trzeba przetworzyć i zapisać w odpowiednim miejscu w bazie danych. Framework automatyzuje większość tej pracy. Budujemy formularz na podstawie modelu `Topic`.

Priorytet. Wysoki.

Definition of Done. Kiedy użytkownik podaje informacje, konieczne jest przeprowadzenie ich weryfikacji i sprawdzenie, czy podane zostały odpowiedniego rodzaju dane, czy nie zawierają żadnego kodu o złośliwym działaniu, na przykład przeznaczonego do zakłócenia pracy serwera.

4.16 Backlog Item 16

Tytuł zadania. Adres URL dla strony z nowym tematem.

Opis zadania. Adres URL dla nowej strony powinien być krótki i jasny. W tym przypadku będzie to: `newtopic`.

Priorytet. Średni.

Definition of Done. Kiedy użytkownik będzie chciał dodać nowy temat, wówczas zostanie przekierowany na stronę `newtopic`.

4.17 Backlog Item 17

Tytuł zadania. Obsługa widoku dla strony z nowym tematem.

Opis zadania. Funkcja widoku musi się zajmować obsługą dwóch różnych akcji. Pierwsza to początkowe żądania do strony `newtopic` – w takim przypadku powinien zostać wyświetlony pusty formularz. Druga to przetworzenie wszelkich danych przekazanych w formularzu – w takim przypadku konieczne jest przekierowanie użytkownika z powrotem na stronę z tematami.

Priorytet. Wysoki.

Definition of Done. Będziemy przekierowywać użytkownika, kiedy już doda nowy temat z powrotem na stronę z tematami.

4.18 Backlog Item 18

Tytuł zadania. Utworzenie szablonu dla strony z nowym tematem.

Opis zadania. Uniemożliwiamy potencjalnym atakującym użycie formularza do uzyskania nieautoryzowanego dostępu do serwera. Django nie tworzy przycisku wysyłającego formularz, więc definiujemy go sami.

Priorytet. Wysoki.

Definition of Done. Utworzymy nowy szablon przeznaczony do wyświetlenia zbudowanego formularza dla nowego tematu. Ten szablon rozszerza `base.html`, aby strona zachowała podstawową strukturę, taką samą jak pozostałe strony w aplikacji "Memorize!".

4.19 Backlog Item 19

Tytuł zadania. Dodanie łącza prowadzącego na stronę formularza z nowym tematem.

Opis zadania. Umieszczamy łącze za listą istniejących tematów. Testujemy przygotowany formularz i dodajemy kilka nowych tematów.

Priorytet. Wysoki.

Definition of Done. Dodanie łącza prowadzącego na stronę `newtopic`.

4.20 Backlog Item 20

Tytuł zadania. Umożliwienie użytkownikom dodawania nowych wpisów c.d.n.

Opis zadania. «Opis».

Priorytet. «Priorytet».

Definition of Done. «Określić (w języku zrozumiałym dla wszystkich członków zespołu), co oznacza ukończenie danego zadania».

5 Sprint 1

5.1 Cel

Do opracowania projektu o nazwie "Memorize!" wykorzystamy framework Django. Zaczniemy od przygotowania środowiska dla projektu, a następnie zdefiniujemy klasy dla danych, z którymi będzie działała aplikacja. Witrynę administracyjną Django wykorzystamy do wstawienia pewnych danych początkowych, a następnie przejdziemy do tworzenia widoków i szablonów, za pomocą których Django będzie budować strukturę naszej aplikacji internetowej.

5.2 Sprint Planning/Backlog

Tytuł zadania. Utworzenie projektu w Django.

- Estymata: "XS".

Tytuł zadania. Utworzenie bazy danych.

- Estymata: "S".

Tytuł zadania. Uruchomienie aplikacji.

- Estymata: "M".

Tytuł zadania. Zdefiniowanie klasy Topic.

- Estymata: "L".

Tytuł zadania. Utworzenie i konfiguracja superużytkownika.

- Estymata: "S".

Tytuł zadania. Zdefiniowanie klasy Entry.

- Estymata: "L".

Tytuł zadania. Mapowanie adresu URL strony głównej.

- Estymata: "M".

Tytuł zadania. Utworzenie widoku strony głównej.

- Estymata: "M".

Tytuł zadania. Utworzenie szablonu strony głównej aplikacji "Memorize!" .

- Estymata: "S".

Tytuł zadania. Utworzenie szablonu bazowego.

- Estymata: "M".

Tytuł zadania. Utworzenie strony wyświetlającej listę wszystkich tematów .

- Estymata: "M".

Tytuł zadania. Utworzenie strony wyświetlającej listę wszystkich wpisów dla poszczególnych tematów.

- Estymata: "L".

Tytuł zadania. Łącza ze strony tematów.

- Estymata: "XS".

«Tutaj dodawać kolejne zadania»

5.3 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»

5.4 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».

6 Sprint 2

6.1 Cel

Utworzymy intuicyjne, przyjazne użytkownikom strony pozwalające im na dodawanie nowych tematów oraz wpisów, a także na edycję istniejących wpisów. Zajmiemy się budowaniem systemu rejestracji użytkowników, który pozwoli użytkownikom na tworzenie kont i wykorzystywanie naszej aplikacji internetowej. Umożliwimy każdemu użytkownikowi znajdującemu się w dowolnym miejscu na świecie przeprowadzenie operacji rejestracji konta w aplikacji "Memorize!" i rozpoczęcie korzystania z niej. Przygotujemy formularze pozwalające użytkownikom zakładać nowe tematy, dodawać nowe wpisy oraz edytować istniejące. Następnie przystąpimy do implementacji systemu uwierzytelniania użytkownika. Zbudujemy stronę rejestracji, która umożliwi użytkownikom założenie konta,

oraz ograniczy dostęp do wybranych stron, które będą mogli wyświetlić tylko zalogowani użytkownicy. Zmodyfikujemy niektóre funkcje widoku, aby użytkownicy mogli widzieć jedynie własne dane.

6.2 Sprint Planning/Backlog

Tytuł zadania. Umożliwienie użytkownikom dodawania nowych tematów.

- Estymata: "L".

Tytuł zadania. Utworzenie formularza dla nowego tematu.

- Estymata: "M".

Tytuł zadania. Adres URL dla strony z nowym tematem.

- Estymata: "S".

Tytuł zadania. Obsługa widoku dla strony z nowym tematem.

- Estymata: "M".

Tytuł zadania. Utworzenie szablonu dla strony z nowym tematem.

- Estymata: "M".

Tytuł zadania. Dodanie łącza prowadzącego na stronę formularza z nowym tematem.

- Estymata: "S".

6.3 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»

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? 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^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/notesfromatooluser/tag/scrum-by-example>
- [5] https://www.tutorialspoint.com/scrum/scrum_user_stories.htm