	Rozwiązanie chmurowe dla wystawy na wydziale ETI Politechniki Gdańskiej		
	<i>Scrum: Backlog sprintu</i>		
Autorzy:	Tomasz Piwowski		
Data:	02.05.2023	Wersja:	1.0

Spis treści

O projekcie i produkcji	1
Oszacowanie rozmiaru backlogu produktu	1
Założenia i dobór zakresu sprintu	2
Cel sprintu	2
Backlog sprintu	3
Kryteria akceptacji	3
Definicja ukończenia	7

O projekcie i produkcji

Aplikacja do prezentowania dodatkowych danych dot. wystawy na nowym ETI. Wykorzystania zostanie baza danych w chmurze do łatwego ich tam umieszczania przez pracowników pg oraz wydobywania dla całej reszty. Do pobrania danych służyć będą kody QR, a w przyszłości możliwe że oprócz standardowego interfejsu do ich wyświetlania, pojawi się wspierający je moduł AR (modele elementów wystawy, przyjaźniejszy sposób prezentowania treści).

Oszacowanie rozmiaru backlogu produktu

Przeprowadzona była 1 runda planning poker, na podstawie której wyliczona została średnia po znormalizowaniu. Wartości zostały zaokrąglone do typowych wartości SP występujących w planning poker, czyli 1, 2, 3, 5, 8, 13, 20, itd.

Łączna ilość SP: 82

Story points (Tomek) ...	Story Points (Wiktor) ...	Story points (średnia po normalizacji) ...
[1] 13 SP	[1] 20 SP	[1] 8 SP
[2] 5 SP	[2] 10 SP	[2] 3 SP
[3] 3 SP	[3] 10 SP	[3] 3 SP
[4] 8 SP	[4] 20 SP	[4] 5 SP
[5] 3 SP	[5] 10 SP	[5] 3 SP
[6] 2 SP	[6] 10 SP	[6] 2 SP
[7] 8 SP	[7] 20 SP	[7] 5 SP
[8] 40 SP	[8] 20 SP	[8] 20 SP
[9] 13 Sp	[9] 30 SP	[9] 13 Sp
[10] 8 SP	[10] 20 SP	[10] 5 SP
[11] 3 SP	[11] 20 SP	[11] 3 SP
[12] 5 SP	[12] 15 SP	[12] 5 SP
+ Dodaj kartę	+ Dodaj kartę	+ Dodaj kartę

Założenia i dobór zakresu sprintu

Pojemność zespołu: trudna do wyliczenia, 2-8h w tygodniu na osobę; 56-224h na całość; **średnio 140h**; **z tego 14h rezerwy** (spotkania scrumowe, spotkania z klientem, itp.)

Prędkość zespołu (zakładana): 12 SP/sprint

Elementy z BP:

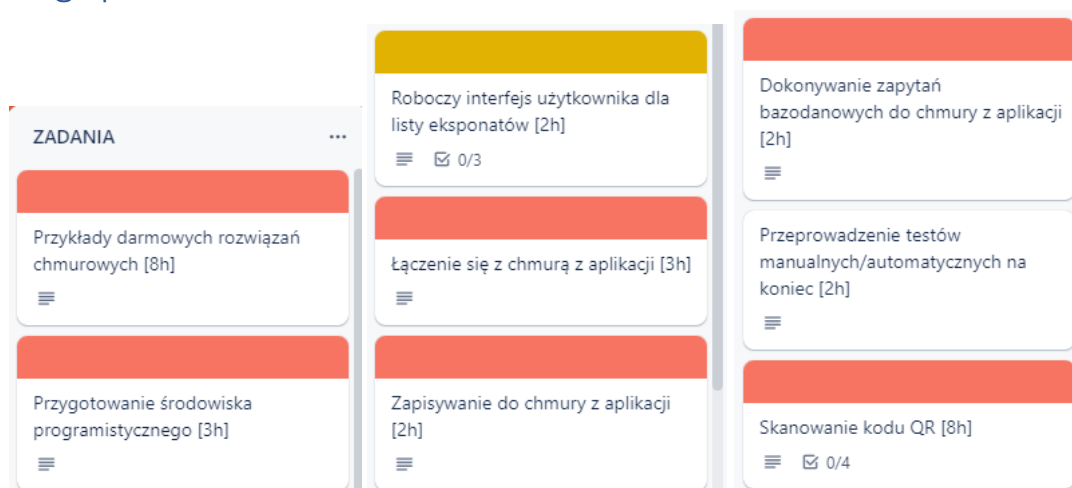
- ➔ [1] Lista eksponatów zsynchronizowana z bazą w chmurze
- ➔ [2] Skanowanie kodu qr

Uzasadnienie: [1] sama lista nie jest kluczowa, ale jej wykonanie pozwoli łatwo wdrożyć się w mechanizmy obsługi chmury i da solidne podwaliny pod resztę aplikacji. [2] Prosta funkcjonalność, która będzie kluczowa w późniejszej fazie rozwoju aplikacji. Napotkane problemy w [1] i [2] pozwolą na szybkie przeciwdziałanie dla obsługi najbardziej kluczowych komponentów aplikacji (łącność z chmurą, skanowanie qr) bez ryzyka w postaci utraty kompatybilności z pozostałą częścią aplikacji.


Cel sprintu

Utworzenie pierwszego prototypu aplikacji i nawiązania łączności z chmurą.


Backlog sprintu





Kryteria akceptacji

 **Przykłady darmowych rozwiązań chmurowych [8h]**
na liście [ZADANIA](#)

Powiadomienia

 Obserwuj

 Opis  Edytuj


Akceptacja: Co najmniej 4 przykłady rozwiązań chmurowych wypisanych w formie listy, które zawierają:

- logo usługi
- nazwę usługi
- formę płatności (całkiem darmowa/ darmowa do pewnego stopnia wykorzystania zasobów, jakiego?)
- ilość miejsca na dane za darmo
- dokumentacja/ łatwość obsługi
- dodatkowe informacje (czym się wyróżnia na tle reszty)

Przygotowanie środowiska programistycznego [3h]

na liście [ZADANIA](#)

Powiadomienia

 Obserwuj


 Opis Edytuj

Akceptacja: Wybór środowiska programistycznego z przygotowanym uzasadnieniem. Utworzenie projektu i stworzenie programu "Hello world!". Połączenie się z gitem i wysłanie pull requesta.

Roboczy interfejs użytkownika dla listy eksponatów [2h]

na liście [ZADANIA](#)

Powiadomienia


 Obserwuj

 Opis Edytuj

Akceptacja: Przygotowanie interfejsu zdolnego wyświetlić listę przedmiotów, z którego każdy z nich zawiera:

- ikonę,
- nazwę,
- kilka dodatkowych pól z możliwością łatwego rozszerzenia w kodzie (np. datę wyprodukowania, kategorię, firmę twórcę).

Lista jest sortowalna po dowolnym rodzaju pola z wyjątkiem ikony.

 Do zrobienia:

Usuń

0%

- ☐ lista dwukolumnowa na ikonę i nazwę
- ☐ rozszerzalny model na dodatkowe pola
- ☐ sortowanie po dowolnym polu po naciśnięciu w odpowiednie pole (np. na nazwę kolumny)

Łączenie się z chmurą z aplikacji [3h]

na liście [ZADANIA](#)

Powiadomienia

 Obserwuj

 Opis

Edytuj

Akceptacja: Połączenie jest ustanowione i można je sprawdzić, np. poprzez wysłanie wiadomości z chmury do aplikacji.

Zapisywanie do chmury z aplikacji [2h]

na liście [ZADANIA](#)

Powiadomienia

 Obserwuj

 Opis


Edytuj


Akceptacja: Podczas uruchomienia aplikacji możliwe jest generowanie wybranej wiadomości (np. poprzez wpisanie na klawiaturze) i wysłanie jej do chmury. Z poziomu chmury można potwierdzić przesłanie wiadomości (np. przez interfejs webowy chmury).

Dokonywanie zapytań bazodanowych do chmury z aplika

na liście [ZADANIA](#)

Powiadomienia

 Obserwuj

 Opis

Edytuj

Akceptacja: Z poziomu kodu generowane są różne zapytania do bazy danych i wywoływane np. poprzez przycisk w aplikacji. Zwracane wyniki pojawiają się jako elementy na wcześniej utworzonej liście.

Przeprowadzenie testów manualnych/automatycznych na

na liście [ZADANIA](#)

Powiadomienia

 Obserwuj

 Opis Edytuj

Akceptacja: Przygotowanie przypadków brzegowych dla listy eksponatów skalibrowanej z chmurą i wykonanie testów na co najmniej 4 urządzeniach i różnych wersjach oprogramowania systemowego.

Skanowanie kodu QR [8h]

na liście [ZADANIA](#)

Powiadomienia

 Obserwuj

 Opis Edytuj

Akceptacja: Zeskanowanie wcześniej przygotowanego kodu QR powoduje wyświetlenie nowego widoku z unikalną wiadomością.


 **TODO:**

Usuń

0%


- ☐ 2 widoki (po zeskanowaniu QR i widok z kamerą do skanowania)
- ☐ Przygotowanie kodów QR do testów
- ☐ Zeskanowanie kodu QR powoduje przełączenie widoku i wyświetlenie wiadomości (zawsze takiej samej dla danego kodu QR)
- ☐ Naciśnięcie przycisku powrotu (ten co zawsze jest na dole w telefonu) powoduje powrót widoku kamery, gdy wyświetlony jest podgląd wiadomości.


Definicja ukończenia

 **Definicja ukończenia**

na liście [CELE](#)

Powiadomienia

 Obserwuj

 **Opis** Edytuj

1. Kod napisany, który spełnia warunki akceptacji celu.
2. Wykonane testy na co najmniej 4 różnych urządzeniach i różnym oprogramowaniu systemowym.
3. Kody, udokumentowane testy oraz wykaz dostępnych darmowych rozwiązań chmurowych umieszczone w repozytorium.
4. Akceptacja ze strony klienta.