

Tematyka i zakres projektu

System wspierający organizację komnatowych wydarzeń LARP

Firma zajmująca się organizacją wydarzeń interaktywnych chce poprawić jakość organizowanych przez siebie wydarzeń LARP (Live Action Role Play) w wariantcie komnatowym. W tym celu planuje stworzyć system poprawiający komunikację między organizatorami i uczestnikami wydarzeń LARP. W skład systemu wchodzi aplikacja webowa i aplikacja mobilna. System powinien posiadać następujące funkcjonalności:

- Zarządzanie wydarzeniami LARP – tworzenie wydarzeń ich opisów.
- Zarządzanie zapisami na wydarzenia LARP oraz selekcją uczestników.
- Obsługa płatności we współpracy z zewnętrznym systemem obsługi płatności.
- Tworzenie profili odgrywanych postaci, akcji, zasobów i przedmiotów oraz udostępnianie informacji o nich ich graczom.
- Monitorowanie zasobów i przedmiotów posiadanych przez graczy.
- Interakcja z otoczeniem za pomocą kodów QR lub odpowiednich czujników.
- Monitorowanie obecności uczestników na poszczególnych obszarach gry.
- Zarządzanie terenem gry (pomieszczeniami), w tym – definiowanie znaczenia czujników i kodów QR.
- Zlecanie i rozstrzyganie efektów akcji w świecie gry.
- Zarządzanie przebiegiem gry, w tym – obsługa historii wydarzeń.

Uwagi:

- Projekt należy wykonać na podstawie wzorca dokumentu zamieszczonego w pliku "IO_Projekt_wzorzec.doc".
- Zalecana notacja dla scenariuszy przypadków użycia opisana jest w pliku "Zalecana_notacja_scenariuszy.pdf". Plik zawiera fragment artykułu z opisem notacji (w języku angielskim). Uwaga: w notacji można zaproponować polskie odpowiedniki identyfikatorów zdań.
- Proszę ograniczyć zakres projektu do kilku pierwszych iteracji. Proszę uwzględnić najważniejsze oraz najbardziej złożone technicznie przypadki użycia, biorąc pod uwagę ograniczenia rozmiaru sprawozdania (patrz niżej).

Format dokumentu i zalecenia dotyczące treści

Rozmiar i format dokumentu

Dokument sprawozdania powinien mieć ok. 11-12 str. formatu A4 (czcionka 11, czytelne diagramy); nieprzekraczalny rozmiar dokumentu to 15 stron (nie licząc strony tytułowej). Dokument proszę dostarczyć w postaci jednego pliku w formacie PDF (tylko ten format). Proszę nie załączać dodatkowych plików z modelami z narzędzia CASE. Plik proszę umieścić na portalu dydaktycznym; proszę go NIE przysyłać mailowo.

UWAGA: zdecydowanie bardziej liczy się **jakość** wykonanej specyfikacji, niż **ilość** wyspecyfikowanych wymagań, chociaż **NIE** jest wystarczające wykonanie jednego diagramu/modelu danego rodzaju; proszę wybrać jedynie **najważniejszą** funkcjonalność systemu; proszę **NIE** uwzględniać funkcjonalności generycznej (np. logowanie do systemu, rejestrowanie użytkownika), a jedynie bezpośrednio związaną z tematyką zadanej powyżej dziedziny problemu.

Wskazówki

- Dla ułatwienia, w pliku "scrum_published.zip" zawarty jest opis metodyki Scrum. Ta sama treść jest dostępna na stronie <https://www.balticisc.eu/methodologies/scrum/>
- Proszę uwzględnić umieszczenie w Rejestrze produktu (Product Backlog) przypadków użycia systemu (ang. use case), a nie historii użytkownika (ang. user story).
- Rozwiązując zadania 2-5 warto przestudiować przykładowy model umieszczony na stronie <https://www.balticisc.eu/dziewkanat/> oraz zawarty w pliku „Dziewkanat.zip” (sekcja „Materiały dodatkowe”). Zadań 2 i 3 dotyczy pakiet „Model wymagań systemu”, a zadań 4 i 5 – „Model architektoniczny systemu”.
- Diagramy w dokumentach można wykonać przy pomocy dowolnego narzędzia CASE, umożliwiającego tworzenie modeli w języku UML (polecane: „Modelio” open source – modelio.org, „Enterprise Architect” komercyjny, możliwa wersja próbna – sparxsystems.com).

Treść sprawozdania

Proszę opracować dokumentację projektu o zadanej wyżej tematyce. Dokumentacja składa się z 6 punktów podzielonych na dwie części.

Część 1: metodyka i analiza wymagań

1. Plan zadań w projekcie (patrz: podręcznik rozdziały 1-3, 13, Mitch Lacey – Scrum. Praktyczny przewodnik dla początkujących, Helion 2014). Plan powinien być zgodny z metodyką Scrum (patrz również plik „scrum_published.zip” lub strona <https://www.balticisc.eu/methodologies/scrum/>). Treść: a) proszę stworzyć **rejestr produktowy** (ang. *Product Backlog*) dla kilku pierwszych sprintów (iteracji) projektu; rejestr powinien zawierać nazwy **przypadków użycia** (NIE historie użytkownika!) określone w punkcie „2a)”, b) dla wybranej iteracji proszę stworzyć **rejestr sprintu** (ang. *Sprint Backlog*), proszę krótko (po 1-2 zdania) opisać **zadania** w rejestrze sprintu. WSKAZÓWKA: proszę dokładnie zapoznać się z definicją sprintu (iteracji) i zwrócić uwagę na to, czym powinna kończyć się iteracja.
2. Opis wymagań funkcjonalnych systemu (patrz: podręcznik rozdziały 4, 6 i 7). Treść: a) narysować **diagramy przypadków użycia** systemu, b) dla niektórych przypadków użycia wybranych do pierwszego sprintu w rejestrze produktowym, napisać **scenariusze**, c) dla jednego przypadku użycia scenariusze przedstawić w postaci **diagramu czynności** (aktywności).
3. Słownik dziedziny problemu. (patrz: podręcznik rozdziały 4, 5 i 7) Treść: narysować **diagram(y) klas** zawierający(e) pojęcia słownika dziedziny problemu: klasy i relacje między nimi, atrybuty klas, krotności i role (nazwy końców) relacji. UWAGA: proszę zapewnić zgodność pojęć w słowniku z pojęciami użytymi w nazwach przypadków użycia i w scenariuszach z punktu „2”.

Część 2: projektowanie i implementacja

4. Projekt struktury systemu. (patrz: podręcznik rozdziały 5 i 8) Treść: a) narysować **diagramy komponentów** (komponenty, interfejsy i relacje) opisujące strukturę systemu w modelu warstwowym, b) wykonać **krótki opis diagramów**. UWAGA: należy uwzględnić realizację wymagań funkcjonalnych i słownika z Części 1.
5. Projekt dynamiki systemu. (patrz: podręcznik rozdziały 6 i 8) Treść: a) narysować **diagramy sekwencji** opisujące realizację wybranych przypadków użycia systemu z pierwszej iteracji, b) dokonać **krótkiego opisu diagramów**. Uwaga: należy zapewnić zgodność tworzonych diagramów sekwencji ze scenariuszami przypadków użycia z punktów „2b” i/lub „2c”.
6. Opis struktury kodu systemu. (patrz: podręcznik rozdziały 8 i 12) Treść: a) narysować w języku UML **diagram klas** zawierający przykładowy interfejs z zadania „4” (wraz z kilkoma operacjami) oraz realizującą go klasę, a także odpowiadający im **kod** w języku programowania, b) opisać **sposób generacji kodu** z modelu w języku UML w wybranym narzędziu CASE.

Terminy oraz zasady oddawania projektów

1. Plik z rozwiązaniem zadania projektowego należy umieścić w portalu dydaktycznym. Pliki przysłane mailowo nie będą sprawdzane.
2. Zalecane jest odpowiednio wczesne oddanie sprawozdania w celu uzyskania uwag od prowadzącego. Dla wstępnych sprawozdań dostarczonych do dnia pierwszych konsultacji uwagi będą przekazane poprzez portal dydaktyczny i/lub podczas konsultacji. Dla wstępnych sprawozdań dostarczonych do dnia drugich konsultacji uwagi będą przekazane TYLKO podczas konsultacji.
3. Możliwe są trzy ocenione próby rozwiązania zadania projektowego. Sprawozdania wstępne, dostarczone przed drugim zjazdem konsultacyjnym nie są wliczane do powyższych prób.
4. **Termin** dostarczenia rozwiązania zadania projektowego upływa we **środę poprzedzającą pierwszy termin egzaminów** o godz. 23:55. Rozwiązanie uznaje się za dostarczone w terminie, jeśli jego treść jest znacząca, to znaczy uzyskana liczba punktów wynosi co najmniej 12.
5. Przekroczenie terminu jak wyżej skutkuje odjęciem 1 pkt. za każdy rozpoczęty dzień spóźnienia. Punkty nie są naliczane po dostarczeniu pierwszego rozwiązania spełniającego warunek z punktu 4.
6. Ostateczny, nieprzekraczalny termin dostarczenia rozwiązania zadania projektowego upływa we **środę poprzedzającą drugi termin egzaminów** o godz. 23:55. Po tym terminie projekt uznaje się za niezaliczony.
7. Projekt uznaje się za zaliczony, jeśli liczba punktów przed odjęciem punktów za spóźnienie wynosi co najmniej 20,5.