**GROUP SCHEDULE**

****

Dokumentacja procesowa

AUTORZY:

Ogórek Zbigniew

Sądel Bartosz

**AGH**

Wydział Informatyki, Elektroniki i Telekomunikacji

1. **Cele projektu**

Celem naszego projektu było stworzenie aplikacji umożliwiającej stworzenie kalendarza, który wspomagałby organizację w grupach, np. znajomych lub projektowych.

1. **Analiza ryzyka**

*Integracja pomiędzy technologiami* – W naszym projekcie zaszła konieczność połączenia kilku różnych technologii. Istniała więc duża szansa na problemy podczas prób integracji różnych komponentów aplikacji. Dodatkowo nie znaliśmy wcześniej większości z elementów, z których korzystaliśmy, nie wspominając o jakichkolwiek próbach połączenia ich w spójną całość.

*Integracja pomiędzy serwerem, a aplikacją natywną* – z powodu wybranej architektury naszego rozwiązania musieliśmy się zmierzyć z integracją systemu backend’owego z aplikacją natywną systemu android. Choć wykorzystaliśmy teoretycznie proste rozwiązanie usług REST’owych, wciąż istniało spore ryzyko, iż pierwotnie wyspecyfikowane usługi, będą musiały w przyszłości zmienić się, z powodu nieoczekiwanych zmian w innej części systemu.

*Komunikacja w zespole* – Znaczącym problemem, jaki mógł się pojawić w naszym zespole była komunikacja. Każdy z nas preferuje inny styl pracy oraz dysponuje innymi zasobami czasowymi. W związku z tym mieliśmy problemy z organizacją regularnych spotkań. W celu zmniejszenia ryzyka wywołanego brakiem komunikacji, korzystaliśmy z dodatkowych narzędzi, takich jak Skype, Github.

1. **Opis produktu**

Ostatecznym produktem, który powinien powstać po zakończeniu prac projektowych powinny być dwie osobne aplikacje. Pierwsza z nich stanowi backend naszego projektu. Jest to aplikacja EJB wdrążona na serwerze JBoss razem z działającą pod spodem bazą danych MySQL. Odpowiada ona za przetwarzanie wszystkich danych, ich persystencję. Kontaktuje się ze światem dzięki wystawianym przez nią usługom REST’owym. Komunikacja między systemem backend’owym, a aplikacją natywną jest dodatkowo zabezpieczona kluczem usług i kluczem sesji, który chroni nas przed nieautoryzowanymi zmianami. System frontend’owy stanowi u nas aplikacja natywna systemu android. Odpowiada ona za wyświetlanie informacji pobranych z serwera oraz udostępnia użytkownikowi funkcjonalności takie jak tworzenie użytkowników, dodawanie i organizację w grupach oraz tworzenie nowych wydarzeń.

1. **Zakres funkcjonalności**

Wymagania funkcjonalne:

* Implementacja systemu backend’owego w formie aplikacji EJB
* Aplikacja mobilna udostępnienia funkcjonalności tworzenia i modyfikacji użytkowników, grup i wydarzeń
* Komunikacja za pomocą serwisów REST’owych

Wymaganie niefunkcjonalne:

* Pełna persystencja danych
* Autoryzacja użytkownika

1. **Metodyka pracy**

Jednym z głównych problemów, a zaraz punktem wyjścia przy tworzeniu projektów takich jak ten jest odpowiednie podejście do pracy oraz przygotowanie się od strony zarówno technicznej jak i organizacyjnej. Należy wybrać pewną strategię wytwarzania oprogramowania i trzymać się jej by osiągnąć oczekiwane rezultaty. Wybór odpowiedniej strategii zależy w dużej mierze od specyfiki projektu oraz stylu pracy osób zaangażowanych w tworzenie systemu. Korzystając z doświadczenie posiadanego przez nas, a nabytego w pracy, zdecydowaliśmy się prowadzić ten projekt w metodologii Scrum. Zgodnie z wyborem podzieliliśmy naszą pracę 6 tygodniowych sprintów, pod koniec których, spodziewaliśmy się posiadać gotowy produkt.

* 1. **Role i podział pracy**

Choć wybrana przez nas metodologia sugeruje wskazanie Scrum Mastera, zdecydowaliśmy że przy naszym zespole, który liczył jednie 2 osoby, taka osoba nie będzie potrzebna. Podział zadań nasunął się sam, gdyż bardzo łatwo w naszym projekcie rozróżnić 2, mniej więcej równe części. Oczywiście mowa tu o aplikacji serwerowej razem z bazą danych oraz aplikacji natywnej. Z racji nieco większego doświadczenie przy tworzeniu aplikacji EJB jednego z członków naszego zespołu, Bartosza Sądla, zdecydowaliśmy się że on zajmie się backend’em naszej aplikacji. Drugi członek zespołu, Zbigniew Ogórek, automatycznie otrzymał więc zadanie stworzenia aplikacji na system Android. Dokumentację zdecydowaliśmy się tworzyć wspólnie.

* 1. **Komunikacja w zespole**

Ze względu na inne charaktery członków zespołu oraz inny styl pracy mieliśmy początkowo problemy ze zgraniem się. Głównym problemem który musieliśmy pokonać, było wspólne wyspecyfikowanie architektury systemu, oraz zdefiniowanie potrzebnych usług, którymi komunikowałyby się obie części systemu. Na szczęście po pokonaniu pierwszych trudności współpraca się poprawiła i nie mieliśmy już nią później większych problemów. Dodatkowo by dbać o jej dobry stan, komunikowaliśmy się ze sobą regularnie, informując o postępach w naszych częściach, tak by każdy zdawał sobie sprawę z tego co robi drugi.

* 1. **Wykorzystane narzędzia**

W celu usprawnienia pracy skorzystaliśmy z zestawu narzędzi które miały za cel przyspieszyć nasze prace oraz ułatwić komunikację. Środowiska w których tworzyliśmy kod to odpowiednio Intellij IDEA do pisania aplikacji serwerowej i Android Studio do pisania aplikacji natywnej. Aplikacje budowaliśmy z użyciem Gradle. Jako repozytorium kodu wykorzystaliśmy Github, oczywiście korzystając systemu kontroli wersji git. Celem polepszenia komunikacji wykorzystaliśmy Skype.

* 1. **Planowany przebieg prac**

Nasz pierwotny harmonogram zakładał podział prac na 5 sprintów, podczas których mieliśmy realizować kolejne etapy projektu. Niestety nie udało nam się w pełni ukończyć projektu. Według nas było to spowodowane zbyt dużą ilością pracy jaka przypadała na każdego z nas. Poniżej pierwotny harmonogram prac oraz opis tego co ostatecznie udało nam się w danych sprintach wykonać.

* **I Sprint**

W czasie tego sprintu zaplanowaliśmy uzgodnić architekturę naszego systemu oraz stworzyć pierwszy pierwszą specyfikację usług, którymi kontaktować się miały nasze systemy. Dodatkowo sprint ten zakładał stworzenie repozytorium, szkieletu aplikacji oraz stworzenie wymagań, jakie system ma spełniać.

Plan działań jaki zakładaliśmy na pierwszy sprint udało nam się spełnić w całości. Stworzyliśmy szkielet zarówno aplikacji serwerowej jak i androidowej. Jeden z projektów od razu wylądował w odpowiednim repozytorium, drugi został tam umieszczony nieco później. Udało nam się również stworzyć schemat aplikacji mobilnej, oraz napisać wstępną specyfikację usług wystawianą przez system backend’owy,

* **II Sprint**

Drugi sprint w naszym projekcie był zdecydowanie najcięższym i najbardziej pracochłonnym z wszystkich. Zakładał on stworzenie schematu na bazie danych, wystawienie pierwszy usług służących do tworzenie użytkowników oraz odpowiednie za to formularze po stronie mobilnej. Dodatkowo w sprincie tym planowaliśmy zająć się bezpieczeństwem komunikacji pomiędzy 2 częściami systemu.

Jak się okazało, nasz plan na ten sprint był zbyt obszerny. Zmuszenie zostaliśmy do przeniesienia części zadań do kolejnego etapu. Dotyczyły to autoryzacji kluczem sesji usług po stronie serwerowej oraz odpowiednie obsłużenie tego przez aplikację mobilną. W sprincie tym udało nam się stworzyć pierwsze usługi, które odpowiadały za tworzenie i modyfikację użytkowników, oraz logowanie do aplikacji. Po stronie mobilnej powstały formatki umożliwiające wykonanie tych właśnie działań po stronie Androida. Powstała również baza danych, w której udało się zapisać pierwsze informacje o tworzonych użytkownikach. Elementem który najbardziej wpłynął na opóźnienia w tym etapie, były problemy z konfiguracją środowisk i integracją między technologiami. Udało nam się wcześniej rozpoznać to ryzyko, jednak nie udało nam się go uniknąć, po mimo wybierania popularnych rozwiązań i połączeń.

* **III Sprint**

W tym sprincie pierwotnie planowaliśmy przeprowadzić pierwszą integrację pomiędzy oboma systemami. Dodatkowo miała się w nim zawrzeć kolejna część usług, tym razem skupiona wokół mechanizmu grup. Pod koniec sprintu użytkownik miał być w stanie poprzez aplikację mobilną stworzyć użytkownika i grupę oraz dołączyć do innej istniejącej grupy. Powinna być także możliwa modyfikacja zarówno jednego jak i drugiego.

W praktyce na początku tego etapu rzeczywiście zajęliśmy się integracją pomiędzy systemami. Jednocześnie pracowaliśmy jednak nad zabezpieczeniem usług przed nieuprawnionymi wywołaniem, za pomocą klucza sesji generowanego. Udało nam się to dokonać jednak zajęło nam to dodatkowy czas, przez co ponownie nie nadążaliśmy za planowanymi pracami. Tym razem postanowiliśmy nie przenosić zadań do kolejnego sprintu, a przedłużyć obecny, tak by nie rozdzielać jednej funkcjonalności na 2 kolejne etapy.

* **IV Sprint**

Czwarty sprint naszego projektu wstępnie zawierał jedynie zadania związane z tworzeniem i obsługą zdarzeń. W jego czasie miały powstać odpowiednie usługi, które umożliwiały by użytkownikowi zewnętrznemu stworzenie nowego wydarzenia, oraz wyświetlenie listy wydarzeń dla grup do których należy.

Sprint ten został nieco skrócony, ze względu na nie planowane przedłużenie się sprintu trzeciego. Był on jednak objętościowo mniejszy od pozostałych sprintów, tak więc udało nam się wywiązać z niego.

* **V Sprint**

Ostatni sprint zakładał intensywne testowanie aplikacji oraz poprawę znalezionych błędów.

Z powodu wcześniejszych opóźnień sprint ten trwał dużo krócej niż powinien. Choć udało nam się znaleźć w nim i poprawić sporą część błędów, nie wykonaliśmy i tak wszystkich planowanych testów. Mimo, iż staraliśmy się od początku regularnie testować nasze aplikacje, późniejsze zmiany niektórych ich części wymagały ponownego przetestowania, na co nie zawsze był czas.