

Raport końcowy

1. Zespół projektowy

Opiekun: dr inż. Michał Wróbel
Klient: dr inż. Michał Wróbel (koło naukowe EmoRG)
Uczestnicy: inż. Natalia Niewdzięczna (KSG)
inż. Wadim Sokołowski (KSG)
inż. Przemysław Studziński (KSG)

2. Temat

Aplikacja wspierająca przeprowadzanie badań interakcji człowieka z komputerem

3. Kontekst projektu

a. Krótka charakterystyka projektu

Głównym założeniem projektu było wspomoczenie klienta w przeprowadzanych w ramach grupy naukowej EmoRG badań interakcji człowieka z komputerem. Owe badania obejmują jednoczesne działanie na różnych komputerach różnych aplikacji, których celem jest zebranie informacji na temat użytkownika. Przykładem takiego eksperymentu jest jednoczesne nagrywanie kamerą reakcji osoby poddanej badaniu i rejestracja jej sygnałów biometrycznych podczas wypełniania ankiety oceniającej stan emocjonalny.

b. Cele projektu

Celem projektu było utworzenie aplikacji ułatwiającej przeprowadzanie badań interakcji człowieka z komputerem w środowisku laboratoryjnym. Badania tego typu realizowane są według zdefiniowanego scenariusza. Otrzymany produkt powinien automatyzować odpowiednie akcje na podstawie wprowadzonego przez użytkownika scenariusza, np. uruchomienie w odpowiedniej kolejności zewnętrznych aplikacji niezbędnych do wykonania pełnego badania. Ponadto powinien umożliwiać synchronizację zegarów na różnych komputerach, wyświetlanie komunikatów, instrukcji, rysunków i animacji oraz blokowanie urządzeń. W zamierzeniu projekt powinien być intuicyjny w obsłudze oraz szeroko konfigurowalny.

c. Charakterystyka klienta

Produkt przygotowywany w ramach projektu tworzony jest pod konkretnego klienta – dr inż. Michała Wróbla. Jest on głównym źródłem wymagań w projekcie; drugim z nich jest dr inż. Agnieszka Landowska, opiekun koła naukowego EmoRG działającego w obszarze informatyki afektywnej. W kontekście tego zadania klient jest nieliczną grupą osób, które docelowo będą korzystać z utworzonego produktu. Projekt jest „szyty na miarę” pod opiekuna projektu – tak, aby jak najlepiej zaspokoić jego potrzeby.

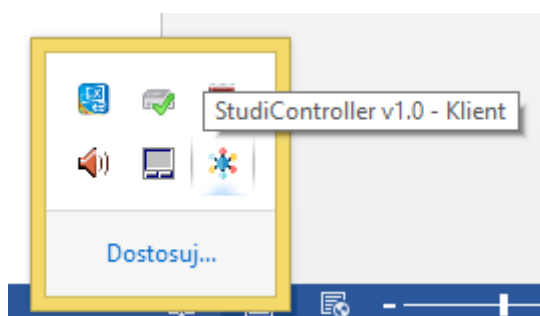
4. Osiągnięte rezultaty

Projekt został w całości wykonany w technologii Java, przy użyciu środowiska IntelliJ IDEA. W ramach zadania stworzono dwie aplikacje desktopowe, spełniające odpowiednio rolę klienta oraz serwera.

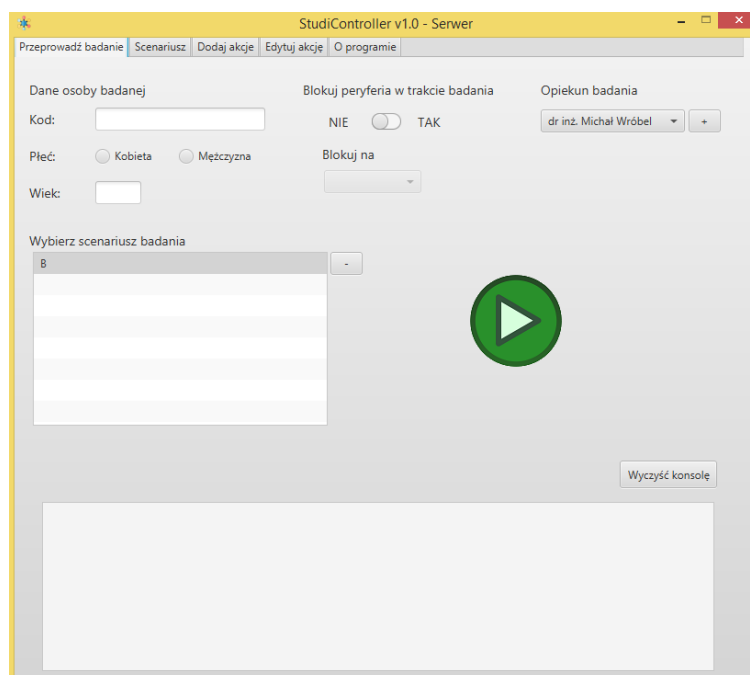
Aplikacja kliencka nie posiada żadnego interfejsu użytkownika, działa w tle. W obszarze powiadomień paska zadań (tacce systemowej, ang. *system tray*) można sprawdzić, czy program jest uruchomiony. Do klienta dołączono także niewielkie narzędzie (firmy zewnętrznej) umożliwiające blokowanie urządzeń peryferyjnych.

Aplikacja serwerowa jest zdecydowanie bardziej rozbudowana; to ona zawiera główną logikę założeń projektowych. Posiada proste w obsłudze GUI oraz foldery przechowujące zdefiniowane akcje, scenariusze oraz badania.

Zdecydowano się również na utworzenie plików wsadowych .bat uruchamiających instancje aplikacji oraz skrypty w języku VisualBasic, które posłużyły jako dodatkowa warstwa używana do odpalenia aplikacji.



Rysunek 1.1. Uruchomiona aplikacja klienta widoczna w tacce systemowej



Rysunek 1.2. Ekran główny aplikacji serwera

To, co wyróżnia nasz produkt na tle rozwiązań konkurencyjnych to perfekcyjne dopasowanie produktu do wymogów zamawiającego – zaimplementowane są dokładnie te funkcjonalności, na które istnieje zapotrzebowanie; program nie jest nadmiarowy. Co więcej, nasza aplikacja charakteryzuje się dużą konfigurowalnością, na której szczególnie klientowi zależało. Funkcjonalność tę zrealizowano poprzez stworzenie oraz dołączenie do projektu odpowiednich plików konfiguracyjnych.

5. Walidacja projektu

Podczas realizacji projektu walidacja przeprowadzana była wielokrotnie. Wiązało się to bezpośrednio z przyjętą lekką metodyką opartą na Scrum, gdzie po każdej iteracji wytworzony produkt prezentowano klientowi, a następnie notowano sugestie i wymagania względem niego.

a. Walidacja przeprowadzona przez zespół projektowy

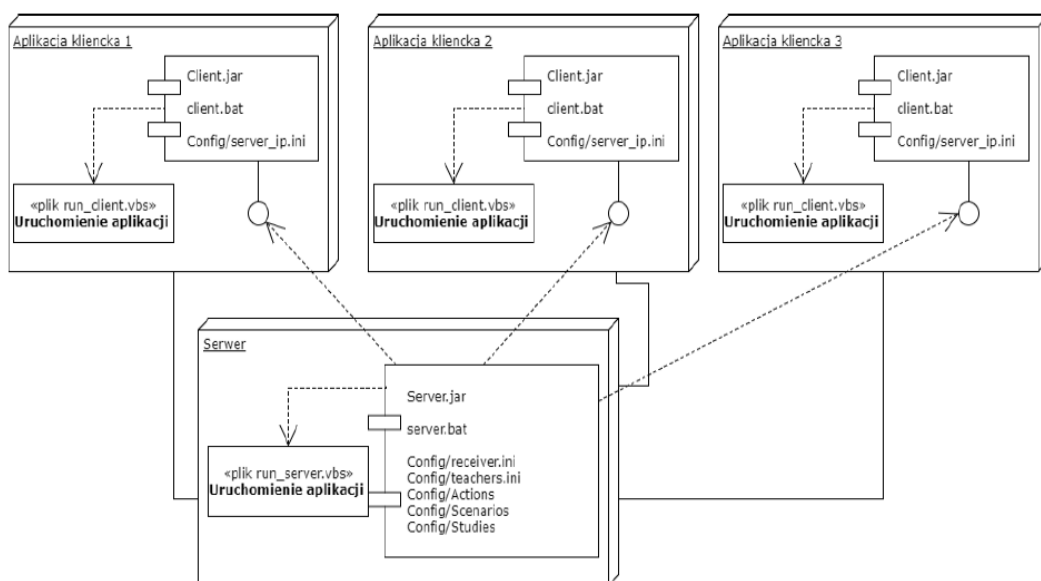
Zespół projektowy regularnie przeprowadzał walidację na lokalnych, prywatnych komputerach. Po dodaniu nowej funkcjonalności osoba ją wprowadzająca testowała manualnie, czy to, co zaimplementowano, spełnia założenia klienta. Na bieżąco walidowano poprawność działania aplikacji – czy nie występują błędy, czy zabezpieczono się przed wprowadzaniem błędnych danych itp. Starano się jak najlepiej przemyśleć działanie programu. Sprawdzano także, czy po dodaniu nowej funkcjonalności nie zostały uszkodzone istniejące elementy i logika działania (testy regresyjne).

b. Walidacja z udziałem klienta

Po zakończeniu każdego sprintu powstały produkt walidowany był w środowisku docelowym wraz z udziałem klienta. Prezentowano aktualny stan zaawansowania funkcjonalności aplikacji oraz jej działanie. Opowiadano klientowi o utworzonych w ramach iteracji możliwościach oraz dyskutowano o napotkanych problemach. Wszelkie sugestie, nowe wymagania czy rozbieżności w wizji zespołu projektowego a wizji klienta były notowane i starano się je wprowadzić w kolejnej wersji produktu. Aplikacje były udostępnione klientowi, aby ten mógł swobodnie samodzielnie przeprowadzić dodatkową walidację.

6. Wdrożenie projektu

Wdrożenie gotowego produktu nie nastręcza większych kłopotów. Wymagany jest system operacyjny Windows oraz Java w wersji przynajmniej 1.8. Utworzono skrypty uruchamiające, które dodatkowo upraszczają otwarcie aplikacji. Najpierw należy uruchomić instancję serwera; następnie można dołączyć dowolną liczbę klientów. Przedtem trzeba jednak odpowiednio skonfigurować produkt – dla programu serwera ustawić w pliku konfiguracyjnym adresy IP klientów wraz z odpowiadającymi im nazwami i analogicznie dla klienta – adres IP serwera.



Rysunek 1.3. Diagram wdrożenia produktu

7. Proces realizacji projektu

a. Organizacja projektu

Zespół projektowy składał się z trzech osób. Praca odbywała się w różnych miejscach, a więc wymagana była częsta wymiana opinii i wątpliwości. Komunikacja przeprowadzana była zarówno w bezpośredni sposób, jak i pośredni z wykorzystaniem odpowiedniego oprogramowania.

b. Metodologie

Wybrano lekką metodykę zarządzania projektem opartą na Scrum, z którego zastosowano pojęcia: sprint, przyrost, backlog (pol. rejestr) produktu, backlog sprintu, scenariusz. Wprowadzone zostały zmienne długości sprintów. Backlog produktu nie był zarządzany przez właściciela produktu, jak ma to miejsce w metodyce Scrum; pełnił pomocniczą rolę – stanowił spis zadań, które zespół miał do wykonania.

Metodykę tę wybrano ze względu na niewielki rozmiar projektu i zespołu projektowego, a także charakter pracy z klientem. Na decyzję wpłynęły także zmienność wymagań oraz doświadczenie w pracy z daną metodyką.

c. Wsparcie narzędziowe

Wśród narzędzi wspomagających komunikację znalazły się komunikator Messenger dostępny na portalu Facebook, poczta e-mail oraz program Skype. Do współdzielenia dokumentów oraz kodu używane były: serwis GitHub jako przechowalnia plików, klient Git do zarządzania plikami repozytorium a także usługa Google Docs. Dokumentacja natomiast została utworzona przy pomocy programu MS Word. Organizację projektu wspomógł narzędziem Acunote, w którym formowano backlog produktu oraz backlogi kolejnych sprintów. Użyto również programu paint.net do obróbki obrazów używanych przez stworzony produkt.

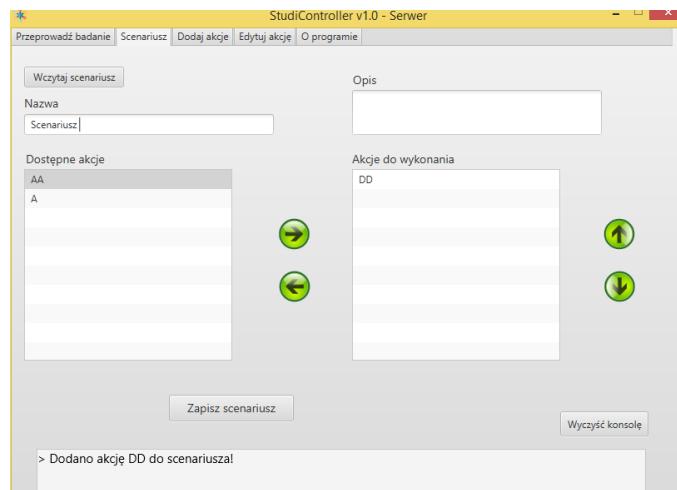
8. Dokumentacja

a. Dokumentacja techniczna

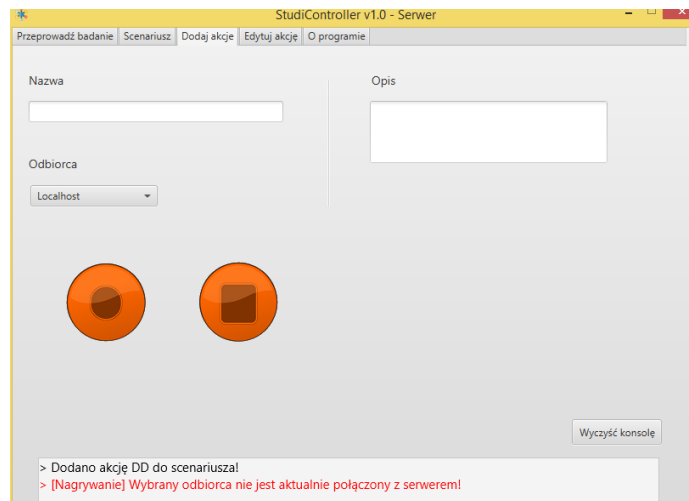
1. Aplikacja serwera

W programie spełniającym rolę serwera zaimplementowano główną logikę powierzonych w projekcie zadań. Interfejs użytkownika utworzono przy pomocy platformy JavaFX, natomiast komunikacja sieciowa została zrealizowana poprzez gniazda (ang. socket). Aplikacja jest wielowątkowa; taką infrastrukturę wymusiło między innymi jednoczesne nasłuchiwanie w oczekiwaniu na klienta i obsługa już połączonych. Postarano się o zapobieżenie wszelkim błędom – w GUI umieszczono podpowiedzi dotyczące określonych przycisków, dodano walidację podawanych przez użytkownika informacji, pomyślano o różnych scenariuszach zakończenia pracy programu i odpowiedniej obsługi wątków tak, aby żaden z nich nie pozostał. Zdecydowano się również na okno konsoli mieszczące się w dolnej części interfejsu – można w nim na bieżąco obserwować wykonywane akcje.

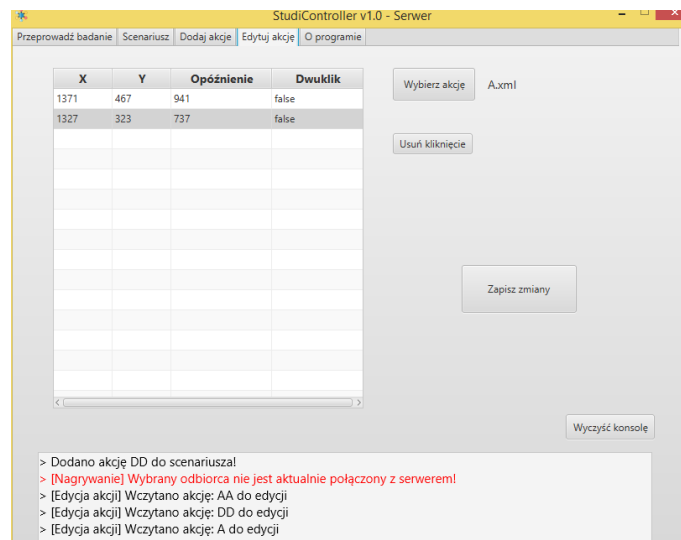
Utworzono także pliki konfiguracyjne umożliwiające ustawienie adresów IP klientów wraz z nazwami (receivers.ini) oraz deklarację dopuszczalnych opiekunów badania (teachers.ini). Zorganizowano również strukturę przechowywania nagranych akcji, stworzonych scenariuszy i wyników badań.



Rysunek 1.4. Zakładka tworzenia i edycji scenariusza



Rysunek 1.5. Zakładka nagrywania akcji



Rysunek 1.6. Zakładka edycji akcji

2. Aplikacja klienta

Program po stronie klienta jest znacznie mniejszy i mniej skomplikowany. Nie posiada interfejsu użytkownika, działa w tle. Tutaj również zaimplementowano wielowątkowość z racji wymaganego jednoczesnego nasłuchiwanie na komunikaty od serwera i wykonywania własnych akcji. Do symulowania kliknięć myszą zastosowano klasę Robot – dzięki niej możliwe było odtworzenie nagranych akcji.

Podobnie jak w przypadku programu serwera, aplikacja klienta posiada zewnętrzny plik konfiguracyjny odpowiedzialny za definicję adresu IP serwera. Dodatkowo załączono program umożliwiający blokowanie urządzeń zewnętrznych; uruchamiany jest w przypadku, gdy badanie zakłada taką opcję.

b. Dokumentacja procesowa

Backlog przedstawiony w poniższej tabeli zawiera opis poszczególnych zadań do wykonania wraz z ich oszacowaniem w story points. Story points to arbitralna miara rozmiaru trudności danego elementu. Dla zadań, które są łatwe w implementacji, oszacowano niewielką wartość story points. Elementy z wysokim wskaźnikiem tej miary zostały przewidziane jako skomplikowane we wdrożeniu.

Product Backlog	
Opis zadania	Story Points
Wybranie języka programowania i technologii	1
Określenie architektury systemu	5
Opracowanie prototypu interfejsu użytkownika	4
Utworzenie karty akcja	2
Utworzenie karty badanie	2
Wybór biblioteki służącej do rejestrowania współrzędnych kliknięć myszy	5
Zapoznanie się ze specyfikacją minimum dwóch bibliotek	2
Zarejestrowanie kilku kliknięć i ich odtworzenie na poziomie kodu Java	2
Dokonanie wyboru konkretnej biblioteki	1
Wybór biblioteki lub zewnętrznego programu służącego do blokowania urządzeń peryferyjnych (myszy/klawiatury)	5
Zapoznanie się ze specyfikacją minimum dwóch aplikacji	2
Zablokowanie/odblokowanie urządzeń peryferyjnych „na żądanie” na poziomie kodu Java	2
Dokonanie wyboru konkretnej biblioteki/programu	1
Utworzenie głównej aplikacji – Serwer	75
Utworzenie zakładki „Dodaj akcję”	18
Zestawienie komunikacji typu „wielu klientów – serwer” pomiędzy dwoma lub więcej komputerami	5
Mechanizm rejestrowania współrzędnych kliknięć na zdalnym komputerze	10
Możliwość zdefiniowania nazwy akcji i jej odbiorcy (maszynie o konkretnym adresie IP)	2
Zapis akcji do pliku po zakończeniu nagrywania	1
Utworzenie zakładki „Edytuj akcję”	6
Wczytanie akcji z pliku	1
Edycja konkretnego kliknięcia – zmiana jego współrzędnych oraz flagi podwójnego kliknięcia	3
Usunięcie konkretnego kliknięcia z listy	1
Zapis zmodyfikowanej akcji do pliku	1
Utworzenie zakładki „Scenariusz”	12
Utworzenie scenariusza składającego się z jednej lub więcej akcji	4
Możliwość nadania nazwy scenariusza	1

Zmiana kolejności odtwarzania akcji w scenariuszu	3
Zapis scenariusza do pliku	1
Wczytanie uprzednio utworzonego scenariusza z pliku i edycja jego akcji	3
Utworzenie zakładki „Przeprowadź badanie”	24
Wypełnienie informacji o badanej osobie	3
Nadanie unikalnego kodu	1
Wybór płci	1
Podanie wieku	1
Wybór opiekuna badania	1
Wybór scenariusza do odtworzenia	1
Możliwość usunięcia danego scenariusza	2
Możliwość wyboru blokady peryferii na konkretnych maszynach (odbiorcach akcji)	6
Uruchomienie scenariusza na aplikacji klienckiej	6
Uruchomienie na lokalnym komputerze	2
Uruchomienie na zdalnych komputerach	4
Implementacja licznika czasu pozostałego do końca badania	2
Automatyczne minimalizowanie/maksymalizowanie okna po rozpoczęciu/zakończeniu badania	1
Zapis krótkiego logu do pliku z podsumowaniem badania	2
Utworzenie zakładki „O programie”	3
Krótki opis aplikacji	1
Informacja o autorach i opiece projektu	1
Informacja o licencji	1
Utworzenie konsoli wewnątrz aplikacji	4
Logowanie do konsoli komunikatów o wykonywanych akcjach	3
Możliwość wyczyszczenia konsoli w dowolnym momencie	1
Przygotowanie grafik	2
Walidacja danych wprowadzanych do pól tekstowych	4
Dodanie tooltipów (dynamicznych opisów) do przycisków	2
Utworzenie aplikacji - Klient	22
Implementacja komunikacji z serwerem i przesyłania informacji o kliknięciach	9
Implementacja odtworzenia konkretnej akcji	6
Blokada lokalnych peryferii po otrzymaniu sygnału od serwera	4
Wyświetlenie informacji jeżeli serwer jest nieosiągalny	2
Dodanie ikony aplikacji klienta w trayu (schowku systemowym)	1
Utworzenie plików konfiguracyjnych	7
Konfigurowalność po stronie aplikacji serwera	6
Utworzenie struktury katalogów/plików do zapisu akcji/scenariuszy	3
Definiowanie potencjalnych odbiorców – klientów w zewnętrznym pliku	2
Definiowanie opiekunów badania w zewnętrznym pliku	1
Konfigurowalność po stronie aplikacji klienta	1

Definiowanie adresu serwera w zewnętrznym pliku	1
Obsługa błędów i sytuacji wyjątkowych	8
Przygotowanie plików/skryptów do łatwego uruchomienia aplikacji serwera/klienta	2
Przeprowadzenie testów	17
Testy funkcjonalne	9
Testy pozytywne	6
Testy negatywne	3
Testy regresyjne	4
Testy integracyjne	4
Sporządzenie pełnej dokumentacji projektu	20
Suma Story Points	171

W poniższych tabelach znajdują się backlogi czterech sprintów przeprowadzonych w trakcie realizacji projektu. Pierwszy z nich odbył się w okolicach lipca 2017 roku, w czasie jego trwania zapoznano się bliżej z technologiami, określono wstępną architekturę systemu oraz utworzono prototyp interfejsu użytkownika. Sprint drugi i trzeci odbyły się odpowiednio na przełomie sierpnia i września oraz września i października. W czasie tych sprintów realizowana była pełna implementacja aplikacji serwera oraz klienta. Czwarty, zamykający i zarazem najdłuższy ze wszystkich sprintów, trwał od końca listopada aż do połowy stycznia. W czasie jego trwania zakończono implementację aplikacji, naprawiono kilka błędów, obsłużono sytuacje wyjątkowe, przeprowadzono szereg testów, a także utworzono pełną dokumentację projektową

Sprint 1	
Opis zadania	Story Points
Wybranie języka programowania i technologii	1
Określenie architektury systemu	5
Opracowanie prototypu interfejsu użytkownika	4
Utworzenie karty akcja	2
Utworzenie karty badanie	2
Wybór biblioteki służącej do rejestrowania współrzędnych kliknięć myszy	5
Zapoznanie się ze specyfikacją minimum dwóch bibliotek	2
Zarejestrowanie kilku kliknięć i ich odtworzenie na poziomie kodu Java	2
Dokonanie wyboru konkretnej biblioteki	1
Suma Story Points	15

Sprint 2	
Opis zadania	Story Points
Wybór biblioteki lub zewnętrznego programu służącego do blokowania urządzeń peryferyjnych (myszy/klawiatury)	5
Zapoznanie się ze specyfikacją minimum dwóch aplikacji	2
Zablokowanie/odblokowanie urządzeń peryferyjnych „na żądanie” na poziomie kodu Java	2

Dokonanie wyboru konkretnej biblioteki/programu	1
Utworzenie głównej aplikacji – Serwer	18
Utworzenie zakładki „Dodaj akcję”	18
Zestawienie komunikacji typu „wielu klientów – serwer” pomiędzy dwoma lub więcej komputerami	5
Mechanizm rejestrowania współrzędnych kliknięć na zdalnym komputerze	10
Możliwość zdefiniowania nazwy akcji i jej odbiorcy (maszynie o konkretnym adresie IP)	2
Zapis akcji do pliku po zakończeniu nagrywania	1
Utworzenie aplikacji - Klient	22
Implementacja komunikacji z serwerem i przesyłania informacji o kliknięciach	9
Implementacja odtworzenia konkretnej akcji	6
Blokada lokalnych peryferii po otrzymaniu sygnału od serwera	4
Wyświetlenie informacji jeżeli serwer jest nieosiągalny	2
Utworzenie plików konfiguracyjnych	7
Konfigurowalność po stronie aplikacji serwera	6
Utworzenie struktury katalogów/plików do zapisu akcji/scenariuszy	3
Definiowanie potencjalnych odbiorców – klientów w zewnętrznym pliku	2
Definiowanie opiekunów badania w zewnętrznym pliku	1
Konfigurowalność po stronie aplikacji klienta	1
Definiowanie adresu serwera w zewnętrznym pliku	1
Suma Story Points	52

Sprint 3	
Opis zadania	Story Points
Utworzenie głównej aplikacji – Serwer	45
Utworzenie zakładki „Scenariusz”	12
Utworzenie scenariusza składającego się z jednej lub więcej akcji	4
Możliwość nadania nazwy scenariusza	1
Zmiana kolejności odtwarzania akcji w scenariuszu	3
Zapis scenariusza do pliku	1
Wczytanie uprzednio utworzonego scenariusza z pliku i edycja jego akcji	3
Utworzenie zakładki „Przeprowadź badanie”	21
Wypełnienie informacji o badanej osobie	3
Nadanie unikalnego kodu	1
Wybór płci	1
Podanie wieku	1
Wybór opiekuna badania	1
Wybór scenariusza do odtworzenia	1

Możliwość usunięcia danego scenariusza	2
Możliwość wyboru blokady peryferii na konkretnych maszynach (odbiorcach akcji)	6
Implementacja licznika czasu pozostałego do końca badania	2
Uruchomienie scenariusza na aplikacji klienckiej	6
Uruchomienie na lokalnym komputerze	2
Uruchomienie na zdalnych komputerach	4
Utworzenie konsoli wewnątrz aplikacji	4
Logowanie do konsoli komunikatów o wykonywanych akcjach	3
Możliwość wyczyszczenia konsoli w dowolnym momencie	1
Przygotowanie grafik	2
Walidacja danych wprowadzanych do pól tekstowych	4
Dodanie tooltipów (dynamicznych opisów) do przycisków	2
Suma Story Points	45

Sprint 4	
Opis zadania	Story Points
Utworzenie głównej aplikacji – Serwer	12
Utworzenie zakładki „Edytuj akcję”	6
Wczytanie akcji z pliku	1
Edycja konkretnego kliknięcia – zmiana jego współrzędnych oraz flagi podwójnego kliknięcia	3
Usunięcie konkretnego kliknięcia z listy	1
Zapis zmodyfikowanej akcji do pliku	1
Utworzenie zakładki „Przeprowadź badanie”	
Automatyczne minimalizowanie/maksymalizowanie okna po rozpoczęciu/zakończeniu badania	1
Zapis krótkiego logu do pliku z podsumowaniem badania	2
Utworzenie zakładki „O programie”	3
Krótki opis aplikacji	1
Informacja o autorach i opiece projektu	1
Informacja o licencji	1
Obsługa błędów i sytuacji wyjątkowych	8
Przygotowanie plików/skryptów do łatwego uruchomienia aplikacji serwera/klienta	2
Przeprowadzenie testów	17
Testy funkcjonalne	9
Testy pozytywne	6
Testy negatywne	3
Testy regresyjne	4
Testy integracyjne	4
Sporządzenie pełnej dokumentacji projektu	20
Suma Story Points	59

9. Zmiany w trakcie projektu

Praca w zespole przebiegała dość sprawnie i nie było znacznych zmian w obrębie organizacji projektu. Udało się utrzymać częstą komunikację i realizować zadania przypadające na dany sprint. Grupa trzymała się określonego harmonogramu, dzięki czemu udało się uniknąć pracy na ostatnią chwilę i poślizgów w oddaniu aplikacji opiekunowi. Zakres pracy nieco się poszerzył – w trakcie spotkań z klientem sprecyzowano dodatkowe wymagania, np. możliwość edycji akcji i scenariusza, które udało się z sukcesem zrealizować. Pewnych sugestii nie zdążono zaimplementować – jako przykład można podać działanie programu zarówno w języku polskim, jak i angielskim.

Zakładane nakłady pracy pokryły się mniej więcej z rzeczywistymi – jak zazwyczaj okazały się one nieco większe. Mimo wszystko zespół oszacował je wystarczająco dobrze, co przełożyło się na pełną implementację założonych funkcjonalności.

10. Podsumowanie

W ramach projektu grupowego udało się stworzyć w pełni działający i odpowiadający potrzebom klienta program wspomagający przeprowadzanie badań interakcji człowieka z komputerem. Zespół projektowy dobrze ocenił swoje możliwości i czas, jaki powinien przeznaczyć na konkretne zadania. Praca przebiegała sprawnie i bezkonfliktowo. Efektem są dwie aplikacje realizujące założenia projektowe.

Zespół nauczył się, jak ważna jest komunikacja pomiędzy poszczególnymi członkami grupy i wysłuchanie opinii osób trzecich. Z pewnością doceniono doskonały i bezproblemowy kontakt z klientem i jednocześnie opiekunem projektu. Zrozumiano, jak ważne są pierwsze etapy tworzenia produktu – analiza wymagań oraz planowanie czy wizja systemu.