

Politechnika Poznańska
Wydział Informatyki
Instytut Informatyki

Praca dyplomowa inżynierska

TYTUŁ PRACY INŻYNIERSKIEJ

Jakub Rojek, 12345
Dawid Neumann, 12345
Wiesław Nowak, 12345
Wiktor Kierzek, 12345

Promotor
dr hab. inż. Imię Nazwisko

Poznań, 2013 r.

Tutaj przychodzi karta pracy dyplomowej;
oryginał wstawiamy do wersji dla archiwum PP, w pozostałych kopiach wstawiamy ksero.

Spis treści

1	Wprowadzenie	1
1.1	Opis problemu i koncepcja jego rozwiązania	1
1.2	Omówienie pracy	2
2	Opis procesów biznesowych	3
2.1	Aktorzy	3
2.2	Obiekty biznesowe	3
2.2.1	Obiekt 1	3
2.2.2	Obiekt 2	3
2.3	Biznesowe przypadki użycia	4
2.3.1	ID01: Nazwa przypadku użycia	4
2.3.2	ID02: Przykład przypadku użycia bez wyjątków	4
2.4	Opis procesów biznesowych z wykorzystaniem notacji graficznej	4
3	Wymagania funkcjonalne	5
3.0.1	Generowanie raportu o absolwentach	6
3.0.2	Dodanie grupy	6
3.0.3	Zmiana nazwy grupy	6
3.0.4	Usunięcie grupy	7
3.0.5	Dodanie pojęcia do grupy	7
3.0.6	Usunięcie pojęcia z grupy	7
3.0.7	Aktualizacja informacji o absolwentach	8
3.0.8	Import absolwentów	8
3.0.9	Grupowanie pojęć	8
4	Wymagania pozafunkcjonalne	9
4.1	Wstęp	9
4.2	Charakterystyki oprogramowania	9
4.3	Wymagania pozafunkcjonalne i ich weryfikacja	10
5	Architektura systemu	11
5.1	Wstęp	11
5.2	Opis ogólny architektury – Marketecture	11
5.3	Analiza SWOT	11
5.4	Perspektywy architektoniczne	11
5.4.1	Perspektywa fizyczna	11
5.4.2	Perspektywa logiczna	11
5.4.3	Perspektywa implemetacyjna	11

5.4.4	Perspektywa procesu (równoległości)	11
5.5	Decyzje projektowe	12
5.6	Wykorzystane technologie	12
5.7	Schemat bazy danych	12
6	Opis implementacji	13
6.1	Wstęp	13
6.2	Scenariusz zmiany: modyfikacja raportu	13
	Opis problemu	13
	Rozwiązanie	13
6.3	Mechanizm generowania raportu	13
6.4	Sekcja ...	13
7	Zapewnianie jakości i konserwacja systemu	14
7.1	Testy i weryfikacja jakości oprogramowania	14
7.1.1	Testy jednostkowe	14
7.1.2	Testy integracyjne	14
7.1.3	Testy akceptacyjne	14
7.1.4	Inne metody zapewniania jakości	15
7.2	Sposób uruchomienia i działania systemu	15
8	Zebrane doświadczenia	16
9	Zakończenie	17
9.1	Podsumowanie	17
9.2	Propozycja dalszych prac	17
A	Informacje uzupełniające	18
A.1	Wkład poszczególnych osób do przedsięwzięcia	18
A.2	Wykaz użytych narzędzi	19
A.3	Zawartość płyty CD	19
B	Wygląd aplikacji	20
C	Schemat bazy danych	21
	Literatura	22

Rozdział 1

Wprowadzenie

1.1 Opis problemu i koncepcja jego rozwiązania

Zgodnie z rozporządzeniem Ministerstwa Nauki i Szkolnictwa Wyższego [1,2] każda uczelnia wyższa w Polsce jest zobowiązana do monitorowania karier swoich absolwentów, w celu pozyskania informacji o ich aktualnej sytuacji zawodowej. Oprócz spełnienia wymogów formalnych, systematyczne gromadzenie i analizowanie danych o zatrudnieniu absolwentów umożliwia uczelniom weryfikację jakości i efektywności kształcenia na poszczególnych wydziałach i kierunkach. Zebrane informacje stanowią cenną wskazówkę w ciągłym procesie doskonalenia oferty dydaktycznej uczelni, pomagając w dostosowywaniu kierunków studiów i programów kształcenia do potrzeb rynku pracy. Prowadzenie rzetelnych badań na temat losów zawodowych absolwentów i prezentowanie statystyk zatrudnienia jest również istotne z punktu widzenia wizerunku uczelni.

Ministerstwo Nauki i Szkolnictwa Wyższego nie narzuca uczelniom sposobu realizacji procesu monitorowania karier absolwentów. Większość uczelni wyższych, w tym Politechnika Poznańska, wywiązuje się z tego obowiązku za pomocą badania ankietowego. Opracowane ankiety są rozsyłane do absolwentów w formie elektronicznych lub drukowanych formularzy bądź przeprowadzane za pośrednictwem rozmów telefonicznych. Rozwiązania te są nie tylko kosztowne i czasochłonne, lecz charakteryzują się także niewielką efektywnością. Z szacunków Centrum Praktyk i Karier Politechniki Poznańskiej wynika, że z możliwości dobrowolnego wypełnienia ankiety absolwenckiej korzysta poniżej 10

W związku z wymienionymi wadami dotychczasowych metod zaproponowano stworzenie systemu informatycznego, w postaci aplikacji internetowej, który monitorowałby kariery zawodowe absolwentów wykorzystując dane udostępniane przez nich w serwisie LinkedIn. Serwis LinkedIn stanowi jedną z największych sieci zawodowych, łącząc ponad 250 mln. użytkowników w 200 krajach i terytoriach na całym świecie. Stworzony system powinien w założeniach zautomatyzować i uskutecznić proces monitorowania, pobierać dane o zatrudnieniu absolwentów z serwisu LinkedIn oraz prezentować je w formie raportów o z góry określonej strukturze. Należy przewidzieć również możliwość integracji z innymi serwisami mogącymi posłużyć jako źródło danych o sytuacji zawodowej absolwentów.

System został zrealizowany na Wydziale Informatyki Politechniki Poznańskiej w ramach zajęć Studia Rozwoju Oprogramowania. Wykonanie systemu zostało zlecone przez rzeczywistego klienta, w postaci przedstawicieli władz wydziału i uczelni. Prace były prowadzone według przyjętej metodyki i harmonogramu.

1.2 Omówienie pracy

Niniejsza praca opisuje otwarty system monitorowania karier zawodowych absolwentów LinkedInGrads (ang. LinkedInGrads: graduate career tracking system), zwany dalej Systemem, realizujący koncepcję przedstawioną w punkcie 1.1. Praca stanowi dokumentację techniczną systemu, a także wyjaśnia idee stojące za poszczególnymi decyzjami projektowymi. Powinna być przydatna zarówno dla użytkowników końcowych systemu, jak i dla osób, które zamierzają go wdrożyć, utrzymywać bądź rozwijać. Jako praca dyplomowa inżynierska jest również skierowana do członków komisji egzaminacyjnej.

W rozdziale 2. przedstawiono aktorów, obiekty biznesowe oraz przypadki użycia występujące w systemie. W rozdziale 3. opisano wymagania funkcjonalne, a w rozdziale 4. wymagania pozafunkcjonalne, wraz z informacją, które z nich zostały zrealizowane. W rozdziale 5. omówiono ogólną architekturę systemu. Rozdział 6. zawiera szczegóły implementacji systemu oraz opis wykorzystanych koncepcji i technologii. W rozdziale 7. przedstawiono metody i narzędzia wspomagające zapewnienie jakości systemu. W rozdziale 8. opisano zebrane wnioski i doświadczenia. Rozdział 9. zawiera podsumowanie całości projektu oraz propozycje dalszego rozwoju systemu. W skład dokumentu wchodzi również bibliografia pracy oraz dodatki, obejmujące informacje uzupełniające, prezentację wyglądu aplikacji, instrukcję instalacji oraz scenariusze manualnych testów akceptacyjnych.

Rozdział 2

Opis procesów biznesowych

2.1 Aktorzy

Wymienienie aktorów zdefiniowanych dla systemu (np. jako wyliczenia).

2.2 Obiekty biznesowe

2.2.1 Obiekt 1

Opis obiektu pierwszego, jego zastosowania. Poniżej wymieniamy jego atrybuty (jeśli istnieją):

- Atrybut 1
- Atrybut 2
- ...

2.2.2 Obiekt 2

Ponownie opis, tym razem dla obiektu drugiego. Oczywiście, inne obiekty opisujemy tak samo:

- Atrybut 1
- Atrybut 2
- ...

2.3 Biznesowe przypadki użycia

Poniżej przedstawicie biznesowe przypadki użycia. Proponuję tutaj wykorzystać szablon, który również jest przerobionym szablonem LaTeXowym i dostosowanym do naszych potrzeb, specjalnie dla opisywania przypadków użycia.

2.3.1 ID01: Nazwa przypadku użycia

Przypadek użycia: ID01: Nazwa przypadku użycia
Aktorzy: Aktor 1, Aktor 2
Scenariusz Główny
1. Pierwszy punkt głównego scenariusza
2. Drugi punkt głównego scenariusza
3. Trzeci punkt głównego scenariusza

Roz-

szerzenia

-
- 3.A Opis sytuacji wyjątkowej 1
 - 3.A.1 Pierwszy krok sytuacji wyjątkowej 1
 - 3.B Opis sytuacji wyjątkowej 2
 - 3.B.1 Pierwszy krok sytuacji wyjątkowej 2
 - 3.B.2 Drugi krok sytuacji wyjątkowej 2

2.3.2 ID02: Przykład przypadku użycia bez wyjątków

Przypadek użycia: ID02: Przykład przypadku użycia bez wyjątków
Aktorzy: Aktor 1
Scenariusz Główny
1. Pierwszy punkt głównego scenariusza
2. Drugi punkt głównego scenariusza
3. Trzeci punkt głównego scenariusza

2.4 Opis procesów biznesowych z wykorzystaniem notacji graficznej

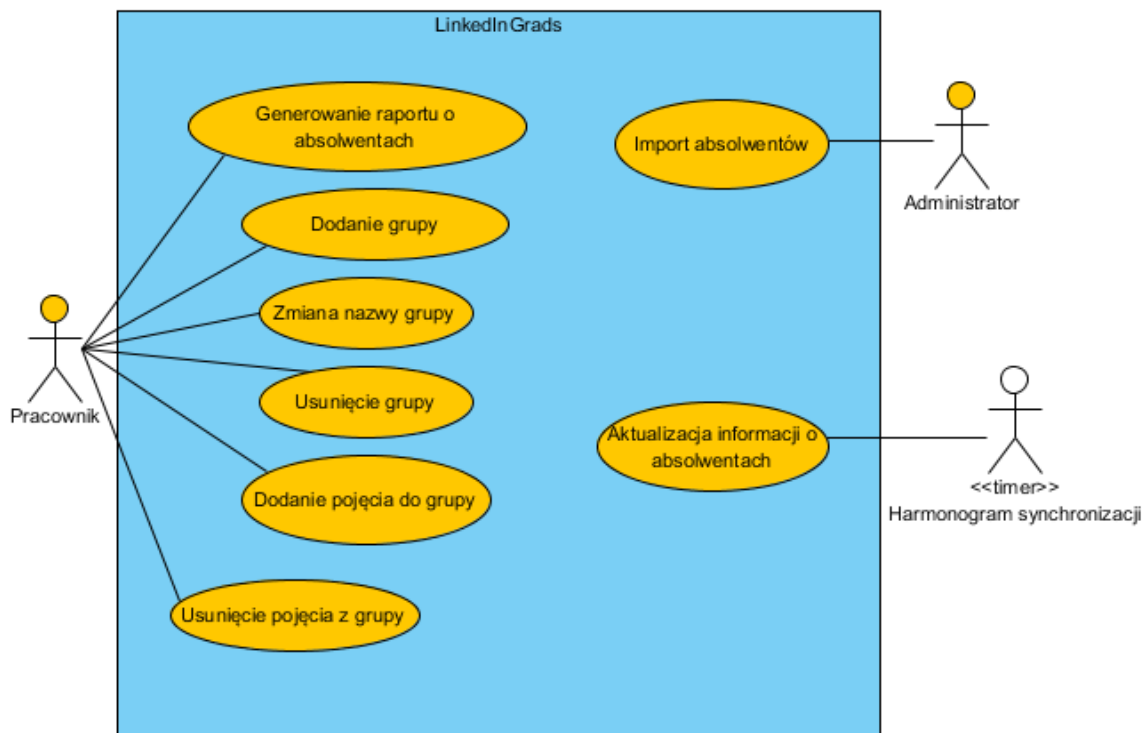
Ten podrozdział jest opcjonalny (a może nawet można to przedstawić jako podsekcję poprzedniego podrozdziału). Tutaj możemy pochwalić się diagramem (np. BPMN) dla naszych procesów biznesowych. Oczywiście, nie tylko w formie graficznej – jeśli macie inną formę, to też się nadaje (o ile jest ona przyzwoita).

Rozdział 3

Wymagania funkcjonalne

Wymagania funkcjonalne (ang. functional requirements) określają co system powinien oferować użytkownikowi, to jest jakie operacje można na nim wykonać. Ważne jest, by utrzymać kompletny zbiór poprawnie zdefiniowanych wymagań, pomaga to w zrozumieniu w jaki sposób powinien działać system, nawet dla osób które nie mają wiedzy technicznej oraz pomaga podczas jego projektowania.

Niniejszy rozdział przedstawia wymagania funkcjonalne za pomocą dwóch najpopularniejszych sposobów ich opisu, są to: przypadki użycia (ang. use cases) oraz opowieści użytkownika (ang. user stories). Pierwsza z metod polega na określeniu listy kroków. Reprezentuje ona interakcję między aktorem a systemem. Wykonanie kroków skutkuje osiągnięciem celu, który ma zapewnić system. Druga metoda - Opowieści użytkownika przedstawia w kilku zdaniach potrzebę użytkownika, którą ma realizować system.



RYSUNEK 3.1: Diagram przypadków użycia

3.0.1 Generowanie raportu o absolwentach

Przypadek użycia: UC1: Generowanie raportu o absolwentach
Aktorzy: Pracownik dziekanatu, System
Scenariusz Główny
<ol style="list-style-type: none">1. Pracownik wybiera opcję generowania raportu.2. System prezentuje typy raportów do wyboru.3. Pracownik wybiera typ raportu.4. System prezentuje rok ukończenia studiów do wyboru.5. Pracownik wybiera rok ukończenia studiów.6. Pracownik inicjuje generowanie raportu.7. System prosi o wybranie ścieżki zapisu raportu.8. Pracownik wybiera ścieżkę zapisu.

3.0.2 Dodanie grupy

Przypadek użycia: UC2: Dodanie grupy
Aktorzy: Pracownik dziekanatu, System
Scenariusz Główny
<ol style="list-style-type: none">1. Pracownik wybiera opcję dodania grupy.2. System prosi o podanie nazwy grupy.3. Pracownik podaje nazwę grupy.4. System informuje o pomyślnym dodaniu grupy.
Rozszerzenia
<ol style="list-style-type: none">3.A. Grupa o podanej nazwie już istnieje.<ol style="list-style-type: none">3.A.1. System informuje o błędzie.3.A.2. Powrót do kroku 3.

3.0.3 Zmiana nazwy grupy

Przypadek użycia: UC3: Zmiana nazwy grupy
Aktorzy: Pracownik dziekanatu, System
Scenariusz Główny
<ol style="list-style-type: none">1. Pracownik wybiera opcję zmiany nazwy grupy.2. System prezentuje listę istniejących grup.3. Pracownik wybiera grupę.4. System prosi o podanie nowej nazwy grupy.5. Pracownik podaje nową nazwę grupy.6. System informuje o pomyślnej zmianie nazwy grupy.
Rozszerzenia
<ol style="list-style-type: none">5.A. Grupa o podanej nazwie już istnieje.<ol style="list-style-type: none">5.A.1. System informuje o błędzie.5.A.2. Powrót do kroku 3.

3.0.4 Usunięcie grupy

Przypadek użycia: UC4: Usunięcie grupy
Aktorzy: Pracownik dziekanatu, System
Scenariusz Główny
<ol style="list-style-type: none">1. Pracownik wybiera opcję usunięcia grupy.2. System prezentuje listę istniejących grup.3. Pracownik wybiera grupę do usunięcia.4. System informuje o pomyślnym usunięciu grupy.

3.0.5 Dodanie pojęcia do grupy

Przypadek użycia: UC5: Dodanie pojęcia do grupy
Aktorzy: Pracownik dziekanatu, System
Scenariusz Główny
<ol style="list-style-type: none">1. Pracownik wybiera opcję grupowania pojęć.2. System prezentuje listę istniejących grup.3. Pracownik wybiera grupę.4. System prezentuje listę pojęć niezgrupowanych.5. Pracownik wybiera pojęcie do dodania.6. System informuje o pomyślnym dodaniu pojęcia do grupy.
Rozszerzenia
<ol style="list-style-type: none">3.A. Grupa została w międzyczasie usunięta.<ol style="list-style-type: none">3.A.1. System informuje o błędzie.3.A.2. Powrót do kroku 2.5.A. Grupa została w międzyczasie usunięta.<ol style="list-style-type: none">5.A.1. System informuje o błędzie.5.A.2. Powrót do kroku 2.

3.0.6 Usunięcie pojęcia z grupy

Przypadek użycia: UC6: Usunięcie pojęcia z grupy
Aktorzy: Pracownik dziekanatu, System
Scenariusz Główny
<ol style="list-style-type: none">1. Pracownik wybiera opcję grupowania pojęć.2. System prezentuje listę istniejących grup.3. Pracownik wybiera grupę.4. System prezentuje listę pojęć przypisanych do grupy.5. Pracownik wybiera pojęcie do usunięcia.6. System informuje o pomyślnym usunięciu pojęcia z grupy.
Rozszerzenia
<ol style="list-style-type: none">3.A. Grupa została w międzyczasie usunięta.<ol style="list-style-type: none">3.A.1. System informuje o błędzie.3.A.2. Powrót do kroku 2.

3.0.7 Aktualizacja informacji o absolwentach

Opowieść użytkownika: US1
Opis: Przebieg aktualizacji informacji o absolwentach. Odbywa się automatycznie, w tle działającego systemu.
Treść: Jako Pracownik chcę aby System, co ustalony czas pobierał informacje o absolwentach z portalu LinkedIn, a następnie aktualizował swoją bazę danych.

3.0.8 Import absolwentów

Przypadek użycia: UC7: Import absolwentów
Aktorzy: Administrator, System
Scenariusz Główny
1. Pracownik wywołuje skrypt importu absolwentów. 2. System prezentuje listę dodanych absolwentów.
Rozszerzenia
1.A. Niepoprawne dane wejściowe. 1.A.1. System przerywa import i informuje o błędzie. 1.A.2. Powrót do 1.

3.0.9 Grupowanie pojęć

Opowieść użytkownika: US2
Opis: Generowanie raportu z użyciem mechanizmu grupowania pojęć.
Treść: Jako Pracownik chcę aby pojęcia (umiejętności, stanowiska, organizacje) w wygenerowanym raporcie były pogrupowane (np. za pomocą metody LDA) w bardziej ogólne zbiory.

Rozdział 4

Wymagania pozafunkcjonalne

4.1 Wstęp

W niniejszym rozdziale zostaną zaprezentowane i krótko opisane charakterystyki oraz wymagania pozafunkcjonalne obowiązujące dla systemu. Ponadto zostanie podjęta próba weryfikacji, które wymagania udało się spełnić i jakie są perspektywy rozwoju

4.2 Charakterystyki oprogramowania

Poniżej są jeszcze stare charakterystyki oprogramowania (wtedy myśleliśmy, że to mądrze brzmi), kategorii wymagań pozafunkcjonalnych. Obecnie to się trochę zmieniło, zatem ta lista będzie znacznie bardziej rozbudowana.

- Dokładność (ang. *accuracy*)
- Bezpieczeństwo (ang. *security*)
- Odporność na błędy (ang. *fault tolerance*)
- Odtwarzalność (ang. *recoverability*)
- Charakterystyka czasowa (ang. *time behaviour*)
- Łatwość analizowania (ang. *analysability*)
- Łatwość zmian (ang. *changeability*)
- Adaptowalność (ang. *adaptability*)
- Instalowalność (ang. *installability*)
- Współistnienie (ang. *co-existence*)
- Zamiennność (ang. *replaceability*)

Tutaj piszemy, które podcharakterystyki były dla nas priorytetowe lub szczególnie ważne, a które mniej. Oczywiście, z uzasadnieniem. Piszemy również, jak zamierzaliśmy (lub to robiliśmy) dbać o to, aby wszystko było spełnione.

4.3 Wymagania pozafunkcjonalne i ich weryfikacja

W tablicy 4.1 przedstawiono wymagania pozafunkcjonalne związane z systemem. W kolumnach **Priorytet** oraz **Trudność** określono poziomy przy pomocy następującej notacji:

- H – wysoki priorytet lub poziom trudności
- M – średni priorytet lub poziom trudności
- L – niski priorytet lub poziom trudności
- N – wymaganie oczywiste lub bardzo proste do spełnienia

Podcharakterystyka	Wymaganie	Priorytet	Trudność
Nazwa podcharakterystyki	Wymaganie 1 (przykład procentów: 90%)	H	H
Nazwa podcharakterystyki	Wymaganie 2	M	H
Nazwa podcharakterystyki	Wymaganie 3	H	L
Nazwa podcharakterystyki	Wymaganie 4	N	L
...

TABLICA 4.1: Wymagania pozafunkcjonalne

A tutaj piszemy o wszelkich problemach, wszelkich naszych wnioskach związanych ze spełnianiem wymagań pozafunkcjonalnych – co się udało, co nie (i dlaczego). Tak, jakbyśmy opisywali nasze doświadczenia i problemy, z jakimi przyszło nam się zmagać i być może rozwiązać. Staramy się odnieść do najważniejszych wymagań (chyba że jest ich mało, wtedy do wszystkich).

Rozdział 5

Architektura systemu

5.1 Wstęp

Wprowadzenie, bardzo ogólny opis.

5.2 Opis ogólny architektury – Marketecture

Tutaj rysunek i opis marketecture.

5.3 Analiza SWOT

Analiza SWOT przyjętego podejścia architektonicznego.

5.4 Perspektywy architektoniczne

5.4.1 Perspektywa fizyczna

Rysunek wraz z opisem.

5.4.2 Perspektywa logiczna

Rysunek wraz z opisem.

This view of the architecture addresses the functional requirements of the system, in other words, what the system should do for its end users. It is an abstraction of the design model and identifies major design packages, subsystems, and classes.

5.4.3 Perspektywa implementacyjna

Rysunek wraz z opisem. Można tutaj też umieścić perspektywę kodu.

This view describes the organization of static software modules (source code, data files, components, executables, and other accompanying artifacts) in the development environment in terms of packaging and layering and in terms of configuration management (ownership, release strategy, and so on). It addresses the issues of ease of development, management of software assets, reuse, subcontracting, and off-the-shelf components.

5.4.4 Perspektywa procesu (równoległości)

Rysunek wraz z opisem.

This view addresses the concurrent aspects of the system at runtime—tasks, threads, or processes as well as their interactions. It addresses issues such as concurrency and parallelism, system start-up and shutdown, fault tolerance, and object distribution. It deals with issues such as deadlock, response time, throughput, and isolation of functions and faults. It is concerned with scalability. Examples are a flight management process, flight plan entry processes, and an airspace management process.

5.5 Decyzje projektowe

Tutaj piszemy o decyzjach projektowych, związkach pomiędzy nimi oraz innych związanych sprawach.

5.6 Wykorzystane technologie

Opis wykorzystywanych technologii (COTS).

5.7 Schemat bazy danych

Schemat bazy danych, ze względu na objętość, można umieścić w którymś dodatku.

Rozdział 6

Opis implementacji

6.1 Wstęp

Wprowadzenie. Struktura tego rozdziału nie jest z góry określona, gdyż mocno zależy to od specyfiki projektu. Generalnie w poszczególnych podrozdziałach każdy powinien opisać swoją część z takiego technicznego punktu widzenia. Piszecie, jak zrealizowaliście poszczególne wymagania, jak to wygląda „pod maską”, oczywiście też trzeba przyjąć jakiś poziom szczegółowości. W bardzo szczególnych przypadkach chyba może się zdarzyć, że trzeba będzie załączyć fragment jakiegoś kodu źródłowego czy konfiguracji – generalnie ma to być opisane w taki sposób, że jako osoba nieznająca systemu siadam i wiem, jak i co zrobiliście.

Ten rozdział ma umożliwić innym osobom modyfikację kodu. Warto zatem opisać najbardziej prawdopodobne scenariusze zmian. Np. jesteśmy świadomi, że w naszym systemie może zaistnieć konieczność zmiany wizualnej jakiegoś raportu. Wówczas należałoby wyjaśnić w jaki sposób należy zmodyfikować kod, aby tego typu zmiany wcielić w życie.

Podsumowując można tutaj umieszczać dwa typy podrozdziałów: - tłumaczące działania określonych mechanizmów - opisujące pewne scenariusze zmiany - tego typu rozdział powinien mieć jasno określony scenariusz zmiany (warto także nawiązać do wymagań) oraz rozwiązanie:

6.2 Scenariusz zmiany: modyfikacja raportu

Opis problemu

System X umożliwia generowanie raportów o stanie finansowym (UC1). Istnieje konieczność zmiany wizualnej w układzie graficznym raportu.

Rozwiązanie

...

6.3 Mechanizm generowania raportu

Rozdział o charakterze opisowym prezentujący implementację konkretnego mechanizmu.

6.4 Sekcja ...

Dalsze opisy.

Rozdział 7

Zapewnianie jakości i konserwacja systemu

7.1 Testy i weryfikacja jakości oprogramowania

7.1.1 Testy jednostkowe

Tutaj piszemy o testach jednostkowych – jak je zrobiliśmy oraz jakie były problemy, wnioski, które się pojawiły.

7.1.2 Testy integracyjne

Podrozdział opcjonalny (bo nie wiem, czy wszyscy robią takie testy). Tutaj piszemy o testach integracyjnych – jak je zrobiliśmy oraz jakie były problemy, wnioski, które się pojawiły.

7.1.3 Testy akceptacyjne

Tutaj piszemy o testach akceptacyjnych – jak je zrobiliśmy oraz jakie były problemy, wnioski, które się pojawiły. Tutaj zwykle pojawiają się testy akceptacyjne przygotowane przez Waszych kierowników, ale także różne opisy, w tym – jeśli jest to zrobione – automatycznych testów.

Poniżej przedstawiono Manualne Testy Akceptacyjne:

MAT01 : Nazwa testu pierwszego		
Warunki początkowe		
<ul style="list-style-type: none">• Warunek początkowy 1• Warunek początkowy 2		
Krok	Akcja	Oczekiwana odpowiedź
1.	Polecenie pierwsze	Odpowiedź systemu
2.	Polecenie drugie	Odpowiedź systemu
3.	Polecenie trzecie	Odpowiedź systemu
Uwagi		
Jakaś uwaga.		

MAT02 : Nazwa testu drugiego		
Warunki początkowe		
<ul style="list-style-type: none"> • Warunek początkowy 1 • Warunek początkowy 2 		
Krok	Akcja	Oczekiwana odpowiedź
1.	Polecenie pierwsze	Odpowiedź systemu
2.	Polecenie drugie	Odpowiedź systemu
4.	Polecenie trzecie	Odpowiedź systemu
Uwagi		
Brak		

7.1.4 Inne metody zapewniania jakości

Jeśli testowaliście projekt w jakiś inny sposób, mieliście inne formy weryfikacji (np. rzucanie komputerem o ściany czy z drugiego miejsca), to tutaj to opisujecie. Prawdopodobnie dobrze będzie opisać tutaj wszelkie testy dla interfejsu, jeśli np. testowaliście użytkowników i sam system pod tym kątem.

7.2 Sposób uruchomienia i działania systemu

Tutaj można napisać jak przygotować system do działania i jak przeprowadzić konfigurację. Prawdę mówiąc, nie wiem, czy to powinno się tutaj znaleźć, ale być może warto, dlatego na wszelki wypadek to umieszczam.

Rozdział 8

Zebrane doświadczenia

Tutaj znajduje się opis wszystkich Waszych doświadczeń związanych z projektem – zarówno pozytywnych jak i negatywnych, dotyczących organizacji, środowiska czy samych już kwestii technicznych. To ma być zebranie Waszych wniosków, wraz z prawdopodobnymi nauczками dla przyszłych roczników.

To dobre miejsce na zaaplikowanie zawartości Lessons Learned Log, jeśli tak prowadziliście, ale też miejsce na własne przemyślenia.

Rozdział 9

Zakończenie

9.1 Podsumowanie

Podsumowanie powstałego systemu, czy przedsięwzięcie się udało, czy to się nadaje do czegośkolwiek. Taki ładny epilog na koniec pracy inżynierskiej.

9.2 Propozycja dalszych prac

Być może system wymaga jakichś prac w przyszłości lub są jakieś propozycje rozszerzenia funkcjonalności. Dotyczy to zarówno tego, co być może sami będziecie dalej robić (jeśli Wam oczywiście zapłacą) lub mają po Was przejąć inne osoby (z następnymi rocznikami włącznie).

Dodatek A

Informacje uzupełniające

A.1 Wkład poszczególnych osób do przedsięwzięcia

Skład zespołu pracującego nad projektem został przedstawiony w tablicy A.1.

Stanowisko	Osoba
Założyciel projektu, klient	Tytuł Imię Nazwisko
Główny użytkownik	Tytuł Imię Nazwisko
Główny dostawca	Tytuł Imię Nazwisko
Dostawca od strony DRO	Tytuł Imię Nazwisko
Starszy konsultant	Tytuł Imię Nazwisko
Konsultant	Tytuł Imię Nazwisko
Kierownik projektu	inż. Imię Nazwisko
Analitik/Architekt	inż. Imię Nazwisko
Programiści	Imię Nazwisko Imię Nazwisko Imię Nazwisko Imię Nazwisko

TABLICA A.1: Osoby związane z przedsięwzięciem

Teraz bardzo ważna rzecz – w tym miejscu piszecie, co kto przygotowywał w tekście pracy inżynierskiej. Prawdopodobnie tutaj będziecie musieli wymienić, które rozdziały zostały dla Was przygotowane przez kierownika projektu, analityka, architekta lub inną osobę. Oczywiście, piszecie też, za które części dokumentu Wy jesteście odpowiedzialni. To jest ważne, aby ta część była tutaj precyzyjnie przygotowana – takie są wymogi uczelni oraz też uwzględnienia pracy innych osób.

Odpowiedzialność za część implementacyjną systemu została przedstawiona poniżej:

Imię i nazwisko pierwszego programisty

- Odpowiedzialność 1
- Odpowiedzialność 2
- Odpowiedzialność 3
- ...

Imię i nazwisko drugiego programisty

- Odpowiedzialność 1

- Odpowiedzialność 2
- Odpowiedzialność 3
- ...

Imię i nazwisko trzeciego programisty

- Odpowiedzialność 1
- Odpowiedzialność 2
- Odpowiedzialność 3
- ...

Imię i nazwisko czwartego programisty

- Odpowiedzialność 1
- Odpowiedzialność 2
- Odpowiedzialność 3
- ...

Ewentualne podziękowania dla innych osób, które Wam pomagały, mowy dziękczynne, itd.

A.2 Wykaz użytych narzędzi

Wprawdzie jest odpowiedni podrozdział w rozdziale 5, ale tutaj można wymienić nawet małe narzędzia i biblioteki, które wykorzystywaliście (np. narzędzie do robienia makiet interfejsu) i które warto wymienić, także dla przyszłych roczników (można też dać linki).

A.3 Zawartość płyty CD

Do dokumentu załączono płytę CD o następującej zawartości:

- Zawartość 1
- Zawartość 2
- Zawartość 3
- ...

Dodatek B

Wygląd aplikacji

Dodatek opcjonalny, tutaj można zamieścić jakieś zrzuty ekranu czy inne materiały dotyczące interfejsu. Jeżeli nie chcecie tego dodatku, zakomentujecie załączenie tego pliku w `thesis-bachelor-polski.tex`.

Dodatek C

Schemat bazy danych

Dodatek opcjonalny, tutaj można zamieścić schematy bazy danych, jeśli nie zmieścił się w rozdziale o architekturze. Może być też tak, że będziecie mieli legen-czekaj-darny schemat o formacie A3 i będziecie go osobno drukować i wklejać w tym miejscu. Jeżeli nie chcecie tego dodatku, zakomentujecie załączenie tego pliku w `thesis-bachelor-polski.tex`.

Literatura



© 2013 Jakub Rojek, Dawid Neumann, Wiesław Nowak, Wiktor Kierzek

Instytut Informatyki, Wydział Informatyki
Politechnika Poznańska

Skład przy użyciu systemu L^AT_EX.

BibT_EX:

```
@mastersthesis{ key,  
  author = "Jakub Rojek \and Dawid Neumann \and Wiesław Nowak \and Wiktor Kierzek",  
  title = "{Tytuł pracy inżynierskiej}",  
  school = "Poznan University of Technology",  
  address = "Pozna{\n}, Poland",  
  year = "2013",  
}
```