TEMAT: Aplikacja sprawdzająca wiedzę z przedmiotów Elektronika i Elektrotechnika.

Wstęp

W dobie postępującej informatyzacji, w niemal każdej dziedzinie życia informatycy usiłują usprawniać pewne procesy lub działania człowieka. Czasem pomysły informatyków są absurdalne i wielkomilionowe projekty trafiają do przysłowiowej „szuflady”. Częściej jednak są to pomysły, które rzeczywiście pomagają oszczędzać czas, a co za tym idzie także pieniądze. W przypadku aplikacji sprawdzającej wiedzę, oszczędzane są również inne zasoby, oprócz czasu i pieniędzy będą to niewątpliwie papier i miejsce. Celem mojej pracy jest stworzenie aplikacji sprawdzającej wiedzę z przedmiotów Elektronika i Elektrotechnika, wykorzystując pewien modułowy charakter tej aplikacji, wymyślony przeze mnie.

Fascynujący jest przepych rozwiązań związanych z elektronicznym sprawdzaniem wiedzy i naprawdę dużym problemem jest wymyślenie czegoś, co jeszcze nie zostało zaimplementowane. Najczęściej proste systemy sprawdzające wiedzę rozwijają się i powstają z tego olbrzymie platformy e-learningowe, np. Moodle, Piazza czy Blackboard. Takie platformy można dowolnie adaptować i sprawdzać na nich wiedzę z praktycznie wszystkich dziedzin. Często spotykane są również systemy dedykowane, sprawdzające wiedzę z wąskiego zakresu, np. system sprawdzający wiedzę na teoretycznym egzaminie na Prawo Jazdy. Bez wątpienia są to aplikacje, które bardzo ułatwiają pracę nauczycielom, profesorom i egzaminatorom, bez względu na dziedzinę ich pracy dydaktycznej. Ta potencjalna użyteczność tworzonej przeze mnie aplikacji, miała zasadniczy wpływ na wybór właśnie tego tematu pracy inżynierskiej. Myśl, że to co miało zostać przeze mnie zaimplementowane, może stać się narzędziem ułatwiającym pracę i może przez długie lata być wykorzystywane, a także rozwijane przez pracowników Akademii Morskiej w Szczecinie, dała mi impuls do wybrania właśnie tego tematu.

Zakładam, że tworzona przeze mnie aplikacja będzie się składała z dwóch modułów głównych: części serwerowej(prowadzący) i części klienckiej(studenci), a także z submodułów: interpreter- przetwarzający stworzony specjalnie na potrzeby aplikacji język skryptowy do budowania testów, submoduł sprawdzający wiedzę i generujący raport dla prowadzącego. Szerzej o modelu tworzonej aplikacji, jej architekturze i proponowanych rozwiązaniach, a także o rozwiązaniach już istniejących, będzie można przeczytać w pierwszym rozdziale zatytułowanym „Platformy wspomagające sprawdzanie wiedzy”.

Chciałbym przeprowadzić czytelnika przez najważniejsze etapy tworzenia systemu informatycznego. Na bazie własnych doświadczeń zawodowych, a także wiedzy wyniesionej z zajęć dydaktycznych, oraz oczywiście na bazie literatury naukowej, stworzę kompletny system informatyczny, którego tworzenie, oprócz części implementacyjnej, składa się również z planowania, zarządzania zasobami, zarządzania ryzykiem, inżynierii wymagań, wdrożenia i testowania. W drugim rozdziale zatytułowanym: „Inżynieria wymagań i proces przygotowawczy do rozpoczęcia projektu informatycznego” przybliżę czytelnikowi, jak powinien wyglądać proces zbierania wymagań, planowania zasobów, harmonogramowania, czy zarządzania ryzykiem. Opisane zostaną również użyteczne wzorce projektowe, oraz bardzo dzisiaj pożądane techniki zwinnego programowania, tudzież zarządzania projektem.

Kolejnym a zarazem najważniejszym rozdziałem jest rozdział trzeci, pt. „Implementacja”. W tym rozdziale zostaną zaprezentowane platformy i technologie, które mogłyby być wykorzystane do budowy tejże aplikacji. Zostanie także uzasadniony mój wybór, czyli platforma .NET i język C#. Oprócz tego zostanie poruszony bardzo ważny temat – tworzenie użytecznego interfejsu graficznego, oraz silników i powiązanych z nimi API, do tworzenia interfejsów. W następnym podrozdziale będzie można przeczytać o podstawach aplikacji sieciowych, realizowanych w architekturze klient-serwer. Została tam wyjaśniona terminologia związana z użyciem aplikacji sieciowych i ich synchronizacji za pomocą wątków. Znajdzie się tam również opis użytecznych bibliotek i klas.

Ostatni rozdział nosi tytuł „Testowanie” i będzie zawierał szereg informacji o testach przeprowadzanych na tworzonym oprogramowaniu. W tym rozdziale oprócz opisu samych technik testowania, znajdą się również przykładowe implementacje testów jednostkowych i opis framework’ów do tworzenia testów automatycznych, a także opis techniki budowania i testowania aplikacji, która pozwala zintegrować pracę programistów i testerów. Technika BDD, bo o niej mowa, to również pewien schemat komentowania kodu i przetwarzania go przez testy jednostkowe, a także sposób formułowania wymagań. BDD to zagadnienie niezwykle ciekawe, które znacząco poprawia wydajność pracy programistów i testerów. Ostatni podrozdział opisuje proces wdrożenia oprogramowania.

Literatura dotycząca tworzenia aplikacji w środowisku .NET jest niezwykle bogata. .NET jest platformą opracowaną przez Microsoft i rozwijaną bardzo intensywnie. Dzięki temu oprócz literatury w postaci książek istnieje także bardzo szczegółowa dokumentacja bibliotek, klas i metod w języku angielskim. Stroną, która zawiera całą dokumentację jest <https://msdn.microsoft.com>, tam również można pobrać samo środowisko i niezbędne do pracy komponenty. Istnieje również wiele tutoriali, które pokazują krok po kroku, jak wykorzystywać pewne elementy środowiska .NET. Jeśli chodzi o programowanie sieciowe, wykorzystujące sockety, protokoły warstwy transportowej i inne możliwe rozwiązania dla aplikacji w architekturze klient-serwer, również możemy znaleźć pomoc w literaturze lub w Internecie. Inżynieria oprogramowania, związane z nią metodyki planowania, harmonogramowania, diagramy, metodyki zwinne i inne, wszystko to jest bardzo dobrze opisane przez rodzimych autorów – Krzysztofa Sachę i Mariusza Chrapko. Gorzej wygląda sytuacja z literaturą związaną z e-learningiem. Znalazłem tylko jedną pozycję, która została częściowo wykorzystana. Reszta wiedzy wykorzystanej w pisaniu pracy została zaczerpnięta z Internetu, lub pochodzi z obserwacji i własnych analiz.

Maciej Łozowski.