

Laboratorium nr 1

Opis doświadczenia w realizacji projektów

Zarządzanie projektem informatycznym

Grupa 2

Ruslan Zhukotynskyi (nr.alb. 169408)

Projekt z rozpoznawania znaków drogowych

Podczas zajęć na studiach inżynierskich naszym zadaniem było zrealizowanie projektu dla firmy Visimind. Celem było stworzenie aplikacji mobilnej umożliwiającej rozpoznawanie znaków drogowych zarówno na zdjęciach, jak i w czasie rzeczywistym. Na wykonanie projektu przeznaczono 13 tygodni (160 godzin pracy zespołowej).

W skład zespołu wchodziło 5 osób, które podzieliły się rolami:

- 2 frontend developerów,*
- 2 backend developerów,*
- 1 osoba odpowiedzialna za Machine Learning.*

Prace rozpoczęliśmy od zapoznania się z technologiami, których wcześniej nie używaliśmy. Był to etap wymagający, gdyż nauka nowych narzędzi pochłonęła dużo czasu. Następnie otrzymaliśmy od firmy treningowy zbiór danych, który został podzielony pomiędzy grupami w celu szybszego i efektywniejszego labelowania. Każdy z członków zespołu miał przydzielone zadania wraz z harmonogramem realizacji. Pomimo przygotowanego planu, pojawiły się trudności w dotrzymywaniu terminów, głównie z powodu ograniczonego czasu (inne zajęcia na uczelni i dodatkowe projekty). Dodatkowym problemem był fakt, że lider zespołu nie egzekwował systematycznej kontroli postępów, co negatywnie wpłynęło na tempo pracy. Gdy frontend i backend osiągnęły już podstawową funkcjonalność, moim zadaniem było ich integracja. Po pierwszym udanym połączeniu zaprezentowaliśmy wersję demo firmie, która wskazała błędy oraz możliwe ulepszenia. W szczególności zwrócono uwagę na konieczność wdrożenia modelu ML bezpośrednio w aplikacji (w formacie .tflite), zamiast przesyłania zapytań do API co sekundę – rozwiązanie to było nieefektywne i generowało opóźnienia. W końcowej fazie projektu, po dopracowaniu frontendu i backendu, zająłem się powiększeniem zbioru danych w celu poprawy dokładności modelu. Współpracowałem również z członkiem innej grupy przy tworzeniu programu do automatycznego labelowania. Dzięki temu udało się znacząco usprawnić proces przygotowania danych. Finalnie backendowcy wyeksportowali wytrenowany model i wdrożyli go w aplikacji mobilnej. Na zakończenie wspólnie opracowaliśmy dokumentację. W dniu oddania projektu odbyła się prezentacja naszej aplikacji – przedstawiliśmy proces tworzenia oraz zaprezentowaliśmy działanie systemu na żywo. Firma była zadowolona z rezultatów,

choć nie udało się wdrożyć funkcji wykrywania znaków na nagraniach wideo, a cały projekt ukończyliśmy w ostatnim możliwym terminie.

Wnioski: *W procesie planowania projektu nie zostały uwzględnione czynniki ludzkie ani ryzyka związane z możliwością niedotrzymania terminów, dlatego nie zaplanowaliśmy wersji „lite” aplikacji, która spełniałaby minimalne wymagania projektowe. Zamiast tego od razu podjęliśmy się realizacji dużych zadań, z których część ostatecznie musieliśmy porzucić. Dodatkowym problemem było to, że team leader nie kontrolował regularnie postępów – nie sprawdzał cotygodniowo efektów pracy poszczególnych członków zespołu, a czasem całkowicie to odpuszczał. W końcowych etapach projektu konieczne były drobne konflikty i spory, aby zmobilizować wszystkich do intensywniejszej pracy i uzyskać satysfakcjonujący rezultat. Choć każdy miał wyznaczoną swoją część zadań, w praktyce okazało się, że niektórzy musieli poświęcać więcej czasu i przejmować obowiązki za innych, którzy nie zrealizowali swoich komponentów. W rezultacie można stwierdzić, że w przyszłości konieczne jest lepsze planowanie zadań i harmonogramu, a także uwzględnianie możliwych problemów oraz przygotowanie alternatywnych scenariuszy. Tylko w ten sposób można uniknąć sytuacji, w której projekt kończony jest w pośpiechu i wymaga improwizowanych rozwiązań.*

Projekt systemu co pozwala płacić kryptowalutą w sklepach internetowych

Realizowaliśmy projekt, którego celem było umożliwienie klientom sklepów internetowych płacenia kryptowalutą – obok tradycyjnych metod płatności, takich jak karta czy BLIK. Na etapie planowania projektu ustaliliśmy wstępne wymagania. Konieczne było zaprojektowanie bazy danych z poprawnymi relacjami, stworzenie diagramu obrazującego działanie systemu oraz określenie niezbędnych funkcjonalności. Nasz zespół składał się z koordynatora projektu oraz dwóch trainee developerów. Po ustaleniu celów i podziale zadań rozpoczęliśmy implementację pierwszych funkcjonalności. Już wtedy pojawił się problem – część kodu została wykonana w niewłaściwy sposób. Podczas rozmów z koordynatorem uświadomiliśmy sobie, że zamiast działać „po omacku”, należy zadawać pytania i

dążyć do pełnego zrozumienia tego, co realizujemy. Pierwsza działająca wersja systemu powstała w oparciu o API. Następnie otrzymaliśmy zadanie integracji z zewnętrznymi API wspierającymi tworzenie portfeli kryptowalutowych, a także zapoznania się z nową architekturą – microservices architecture. Był to duży krok naprzód, który pozwolił nam lepiej rozdzielić funkcjonalności i uniezależnić poszczególne komponenty. Kolejnym etapem było zastąpienie komunikacji w architekturze REST protokołem gRPC, co znacznie poprawiło wydajność i szybkość działania systemu. W początkowej wersji komponenty były ze sobą mocno związane, jednak po wprowadzeniu tych zmian każda część mogła być rozwijana i modyfikowana niezależnie, bez ryzyka, że wpłynie to negatywnie na pozostałe moduły. Ostatnim zrealizowanym elementem przed wypuszczeniem wersji alpha była integracja z komunikatorem Telegram. Administratorzy mogli połączyć swój dashboard z botem Telegram, który automatycznie wysyłał różnego rodzaju powiadomienia. Podczas prezentacji projektu okazało się, że nie został w pełni zrealizowany moduł weryfikacji i potwierdzania płatności, a panel administracyjny działał częściowo niepoprawnie – głównie z powodu ograniczonego czasu i braków w wiedzy zespołu. Mimo to projekt został przekazany do dalszego rozwoju i obecnie znajduje się na etapie przygotowywania wersji beta.

Wnioski: Już na samym początku realizacji projektu należy dokładnie zrozumieć cel systemu oraz jasno określić, za co odpowiada zespół, a co leży po stronie klienta. Ważne jest też zadawanie pytań i konsultowanie wątpliwości, zamiast działać w niepewności. Podczas implementacji trzeba dbać o podział funkcjonalności na mniejsze komponenty, tak aby można je było łatwo modyfikować bez ingerencji w działające elementy systemu. W sytuacji, gdy wiadomo, że danej funkcjonalności nie uda się zrealizować (ze względu na brak czasu lub wiedzy), lepiej skupić się na stworzeniu działającej wersji minimalnej i dopiero potem rozwijać projekt o kolejne rozszerzenia. Istotne jest również prowadzenie bieżącej kontroli postępów – realizowanie zadań krok po kroku, monitorowanie, co zostało wykonane i w jakim czasie, a także porównywanie tempa pracy. Dzięki temu można lepiej planować kolejne etapy i unikać sytuacji, w której projekt kończony jest w pośpiechu.