

***Wrocławska Wyższa Szkoła
Informatyki Stosowanej***

Wydział Informatyki

Łukasz Sidorowicz

Nr albumu: 4492

Projektowanie i implementacja aplikacji mobilnej przygotowującej do brązowej
odznaki jeździeckiej.

Praca magisterska

Kierunek: Informatyka

Specjalność/specjalizacja: Projektowanie urządzeń mobilnych

Praca wykonana pod kierunkiem:
dr inż. Katarzyna Pentoś

Wrocław 2016

Spis treści:

1. Cel i założenia pracy.....	4
2. Wstęp	
2.1 System szkolenia jeździeckiego w Polsce.....	5
2.2 Systemy operacyjne dla urządzeń mobilnych.....	6
2.3 Systemy kontroli wersji.....	8
2.4 Narzędzia Business Intelligence.....	13
3. Projektowanie i implementacja programu przygotowującego do egzaminu teoretycznego na brązową odznakę jeździecką	
3.1 Projektowanie Aplikacji.....	16
3.2 Pobieranie i instalacja Android Studio IDE, utworzenie pustego projektu.....	23
3.3 Utworzenie bazy danych SQLite.....	27
3.4 Wykonanie aplikacji mobilnej.....	32
4. Badania z użyciem aplikacji.....	51
5. Podsumowanie.....	60
6. Wykaz użytych skrótów w pracy.....	61
7. Literatura.....	62

1. Cel i założenia pracy.

Obecnie w każdym uprawianym sporcie szczególny nacisk kładzie się na bezpieczeństwo. Jeździectwo jest sportem szczególnym, gdyż wymagana jest poprawna współpraca z dużym i silnym zwierzęciem- koniem. Niestety jest to sport urazowy, a olbrzymia ilość wypadków bierze się z niewiedzy jeźdźców. Dążąc do poprawy bezpieczeństwa i poziomu wyszkolenia Polski Związek Jeździecki wprowadził system szkolenia i odznaki jeździeckie. Brązowa odznaka jeździecka jest niezbędna do udziału w zawodach, zaś jej zakres teoretyczny pozwala na znaczne zwiększenie bezpieczeństwa podczas codziennej pracy z koniem.

Jak dotąd nie powstała jednak żadna aplikacja, czy program wspomagający przygotowanie do części teoretycznej egzaminu. Jedynymi materiałami jest wykaz pytań z odpowiedziami i zalecanej literatury.

Dzisiejsza młodzież nie rozstaje się obecnie ze swoimi smartfonami. Tkwi tu duży potencjał na uzupełnienie luki poprzez stworzenie prostego programu, który można by w każdej chwili otworzyć i wrócić do przerabianego materiału.

Celem pracy będzie zaprojektowanie i zaimplementowanie aplikacji mobilnej, w której pytania zostaną podzielone na odpowiednie działy, ułatwiając naukę materiału. Całość zostanie zaimplementowana do bazy danych, a dane z bazy będą pobierane i wyświetlane w programie.

Efektywność działania aplikacji zostanie porównana w badaniach grup osób korzystających z aplikacji z grupami osób korzystających z otrzymanych materiałów drukowanych. W ramach badań zostanie przeprowadzone szkolenie na terenie stajni w Magnicach. Wśród jeźdźców chcących przygotować się do egzaminu zostaną wyłonione grupy o podobnym poziomie wiedzy początkowej i podzielone na takie, które otrzymają materiały drukowane lub aplikację mobilną. Tam zostaną też przeprowadzone ankiety weryfikujące.

Badania będą miały za zadanie pokazać, jakie metody nauczania są najefektywniejsze, czy użycie aplikacji poprawi wyniki uzyskiwane przez zdających.

Wyniki badań zostaną zaprezentowane za pomocą narzędzia Tableau.

2. Wstęp

2.1. System szkolenia jeździeckiego w Polsce.

W ostatnich latach wzrasta liczba jeźdźców, jak i posiadaczy koni. Niestety nie idzie to w parze ze znajomością zagadnień teoretycznych, takich jak psychologia konia i jego potrzeby wynikające z natury. Prowadzi to do licznych wypadków w stajni i w drodze ze stajni do ujeżdżalni, jak również przykrymi obrazami w postaci okładanych bezmyślnie batami koni przez jeźdźców. Mając to na uwadze od kilku lat wdraża się systematycznie system szkoleniowy mający poprawić obecną sytuację. Podstawą tego systemu jest zmuszenie początkujących adeptów jeździectwa do zapoznania się z podstawowymi zasadami bezpieczeństwa i postępowania z końmi. Chcąc wystartować w zawodach trzeba posiadać minimum brązową odznakę jeździecką¹. Aby startować w konkursach wyższych klas trzeba posiadać odznakę srebrną, zaś, aby być szkoleniowcem- złotą.

Rysunek 1 Odznaki jeździeckie²



Egzamin na brązową odznakę składa się z części teoretycznej i praktycznej. Zakres części teoretycznej jest dość obszerny i obejmuje praktycznie wszystkie aspekty jeździectwa. Pytania i odpowiedzi można pobrać bezpłatnie ze strony związku. Egzamin praktyczny składa się z części ujeżdżeniowej i skokowej. Jednak dla jeźdźcy, który regularnie jeździ konno to tylko formalność- przeszkody są niskie, figury na ujeżdżalni proste. Co innego część teoretyczna? Nie ma możliwości nauczyć tak szerokiego zakresu materiału w trakcie jazdy. Jeźdźcom zaleca się odpowiednie

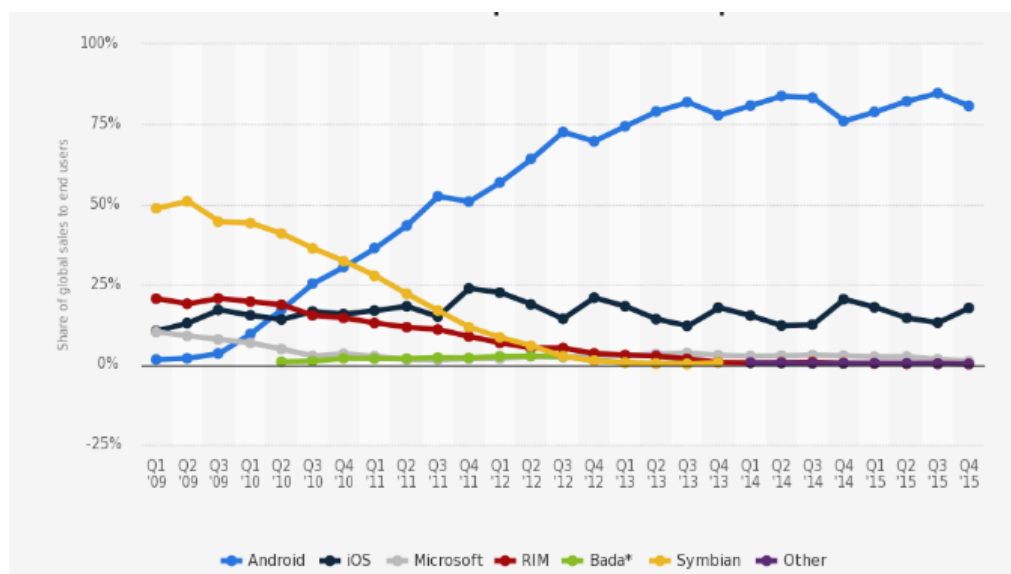
pozycje literaturowe, takie jak „Akademia Jeździecka³”, czy też „Zasady Jazdy Konnej⁴”. Jednakże instruktorzy w wielu „ośrodkach” często pomijają aspekty teoretyczne, „instruktorami” są rolnicy niemający żadnej wiedzy o koniach popołniający skandaliczne ilości błędów w podejściu do młodych adeptów jeździectwa. Odznaki jeździeckie mają za zadanie otworzyć oczy jeźdźcom takich „ośrodków”.

2.2. Systemy operacyjne dla urządzeń mobilnych

Obecnie na rynku systemów operacyjnych dla urządzeń mobilnych liczą się tylko dwaj gracze Apple IOS i Alphabet (wcześniej Google) Android. Do niedawna mówiło się o trzecim poważnym graczu – systemie Microsoftu- Windows Phone.

Jednakże jak przedstawiam na poniższym rysunku praktycznie nie ma on już udziału w rynku nowo sprzedawanych urządzeń:

Rysunek 2 Udział procentowy poszczególnych platform mobilnych⁵

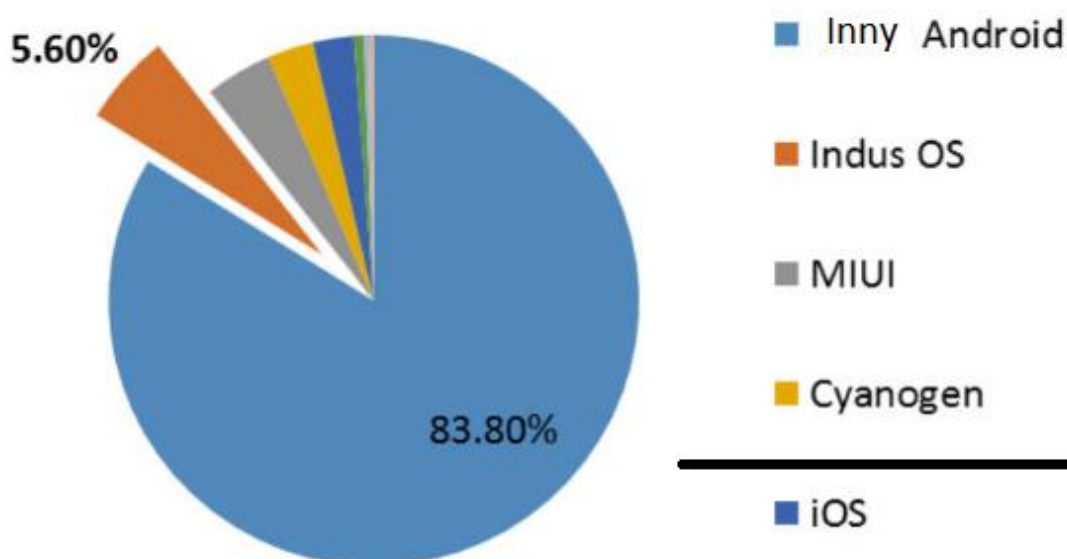


Na rysunku przedstawiono procentowy udział sprzedawanych urządzeń wykorzystujących dany system operacyjny. Android wygrywa z kilku względów- wydajny smartfon czy tablet z tym systemem można kupić w cenie poniżej 1000 zł, np. Lenovo Moto G 3 Gen już za 799 zł⁶. Aby pozwolić sobie na zakup nowego

iPhona z IOsem trzeba liczyć się z wydatkiem, co najmniej 3349zł⁷.

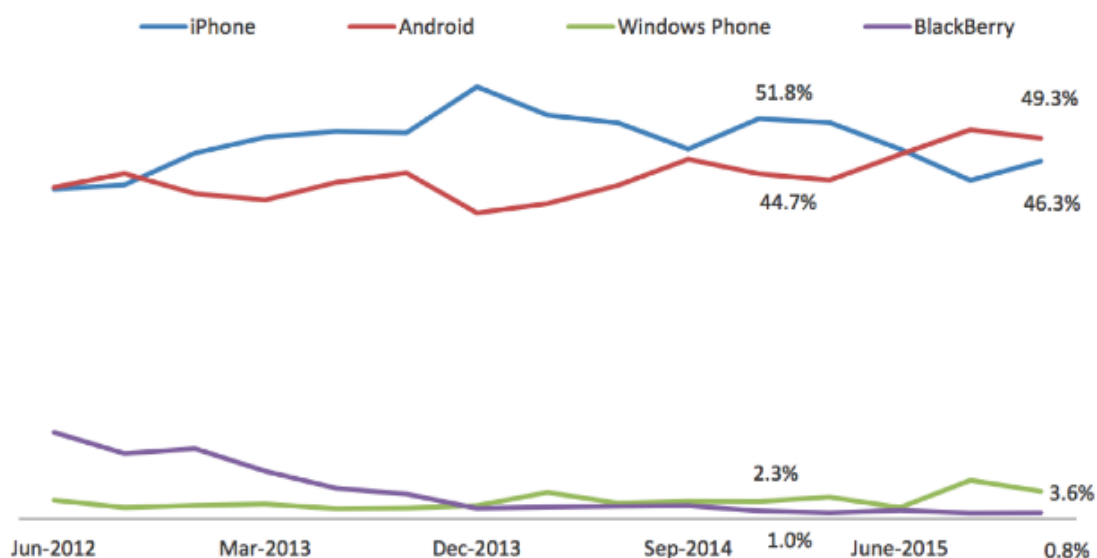
Dysproporcja udziału w rynku pomiędzy Androidem a IOsem jest największa w krajach rozwijających się takich np. jak Indie.

Rysunek 3 Udział Platform Mobilnych w Indiach⁸



W krajach rozwiniętych, takich jak np. Wielka Brytania udziały obu systemów są równe:

Rysunek 4 Udział Platform Mobilnych Wielka Brytania⁹

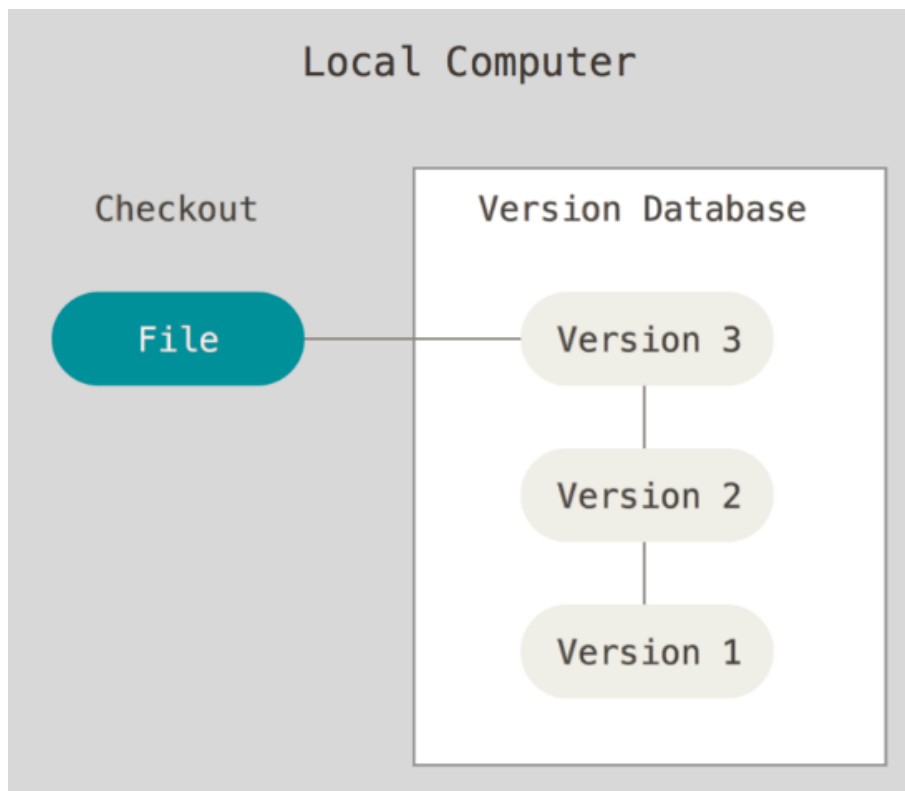


Kolejną ważną zaletą Androida jest darmowy dostęp do zintegrowanego środowiska programistycznego- Android Studio¹⁰ jak i niska cena konta deweloperskiego, która jest jednorazowa i wynosi 25 dolarów¹¹. W przypadku Apple IOS za wszystko w najtańszej wersji trzeba płacić co roku 99 dolarów¹².

2.3. Systemy kontroli wersji.

System kontroli wersji śledzi zmiany dokonywane na plikach i w folderach. Umożliwia przywrócenie dowolnej wcześniejszej wersji kodu, czy też rysunku jak i projektu. Używanie takiego systemu jest niezbędne w pracy nad każdym większym projektem. Podczas tworzenia aplikacji jedna zła linijka kodu może zniweczyć długą pracę. Dzięki systemowi kontroli wersji można wrócić do wcześniejszego działającego kodu, porównać zmiany i znaleźć dosyć łatwo błąd. Dla wielu osób metodą kontroli wersji jest kopiowanie danego pliku do innego katalogu i nadawanie mu kolejnych numerów, np. draft1, draft2 itd. Nie jest to dobry sposób, po pierwsze można zapomnieć gdzie się kopiowało, ponadto porównanie takich plików jest czasochłonne.

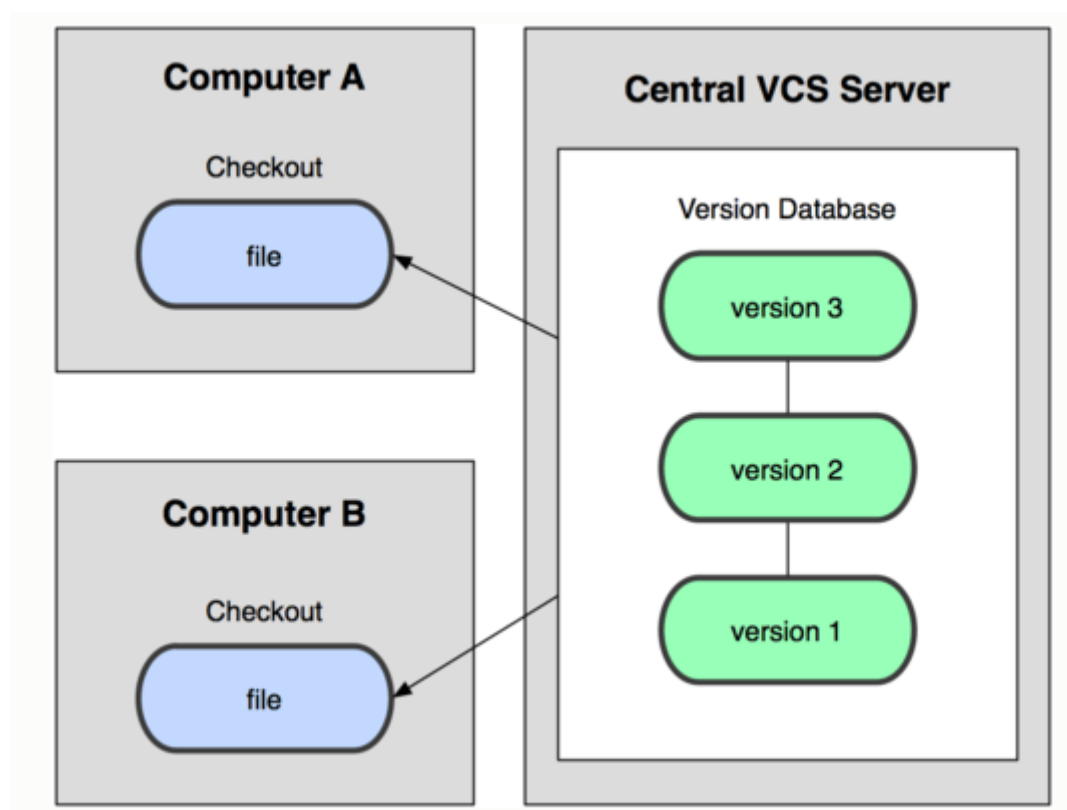
Rysunek 5 Diagram lokalnego systemu kontroli wersji¹³



Aby sobie poradzić z takimi problemami programiści stworzyli lokalne systemy kontroli wersji, który tak naprawdę był lokalną bazą danych, w której przechowywano zmiany dokonane na plikach projektowych. Jednym z pierwszych takich narzędzi był RCS- Revision Control System¹⁴.

Poważnym problemem tego podejścia był brak możliwości dobrej współpracy w zespole, ponadto w przypadku uszkodzenia komputera traciło się całą pracę. Aby przezwyciężyć ten problem powstały centralne systemy kontroli wersji takie np. jak Subversion¹⁵ czy CVS (Concurrent Version System)¹⁶.

Rysunek 6 Diagram scentralizowanego systemu kontroli wersji.

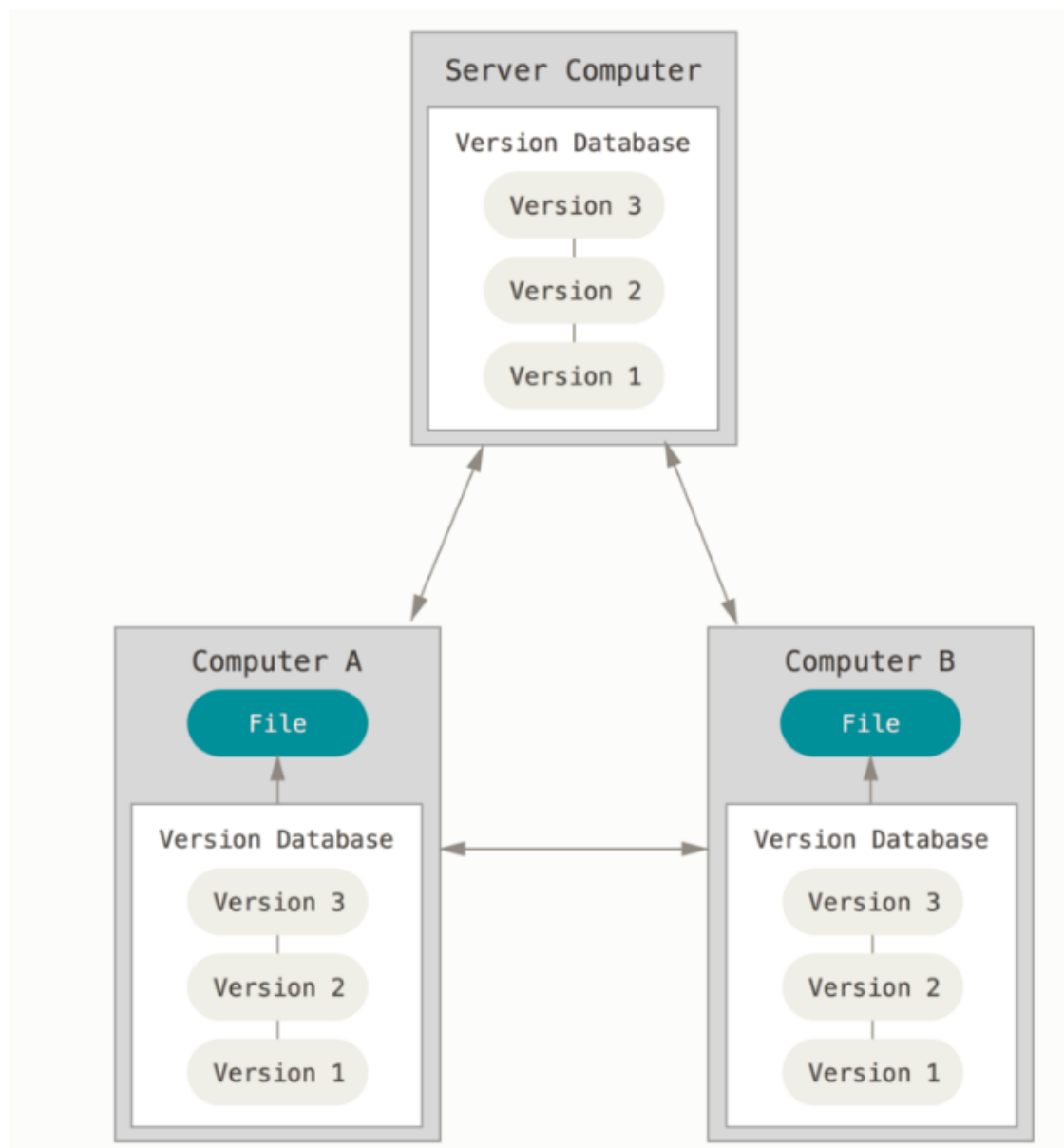


Takie systemy składają się z jednego centralnego serwera, który zawiera wszystkie pliki projektowe. Osoby pracujące w projekcie są klientami i mogą się łączyć z serwerem uzyskując dostęp do aktualnych wersji plików. Ma to olbrzymie zalety w porównaniu z lokalnym systemem- zwiększa się bezpieczeństwo danych, jak również ułatwia współpracę w dużych zespołach. Administratorzy tych systemów mają kontrolę nad uprawnieniami poszczególnych osób w projekcie.

Niestety te systemy mają swoją poważną wadę i słaby punkt- centralny serwer. W przypadku jego awarii, na czas jej trwania zespół traci możliwość współpracy, jak również nikt nie ma możliwości zapisywania zmian, których dokonywał. Co więcej, w przypadku awarii dysku na serwerze można utracić absolutnie wszystko co się robiło od ostatniego backupu.

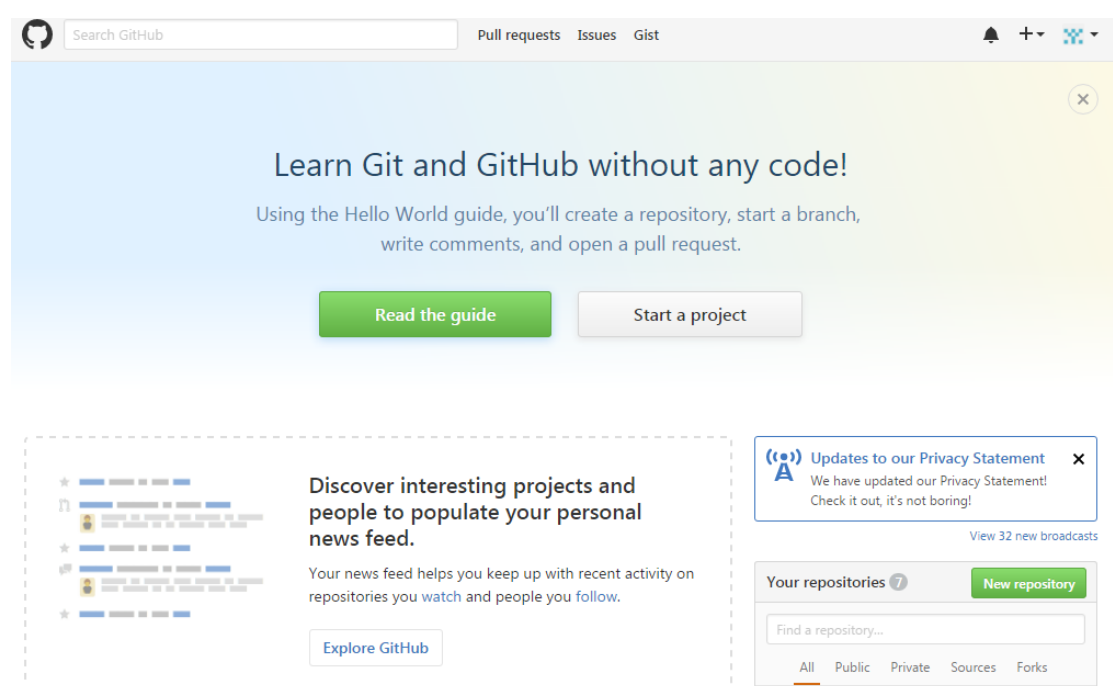
Tych wad pozbawione są najnowsze systemy kontroli wersji- rozproszone. Do takich systemów należy Git¹⁷, Mercurial¹⁸ i Bazaar¹⁹.

Rysunek 7 Diagram rozproszonego systemu kontroli wersji



W takich systemach klienci kopiują całe repozytorium, jak również mają dostęp do najnowszych wersji plików. W przypadku awarii serwera każdy członek zespołu posiada kopię na swoim komputerze. Ponadto ciągłość pracy zostaje zachowana. Szczególną popularność zdobył Git, używany od 2005 do utrzymywania wersji jądra Linuxa. Git jest prosty w obsłudze, szybki i wydajny. Z poziomu Gita bardzo łatwo jest skorzystać z internetowego hostingu GitHub²⁰ jak i Bitbucket²¹.

Rysunek 8 Serwis GitHub



Oba serwisy umożliwiają darmowe tworzenie repozytoriów open source oraz płatne prywatne repozytoria. Wybór, z którego serwisu skorzystać zależy od tego, czy pracujemy w licznych zespole, czy prowadzimy dużo małych projektów. W Bitbucket ma się nielimitowaną liczbę prywatnych repozytoriów, zaś płaci się od liczby użytkowników:

Rysunek 9 Opłaty za korzystanie z serwisu Bitbucket²²

Free	\$10 / mo	\$25 / mo	\$50 / mo	\$100 / mo	\$200 / mo
5 Users	10 Users	25 Users	50 Users	100 Users	Unlimited Users

W GitHubie płaci się od liczby prywatnych repozytoriów niezależnie od liczby użytkowników:

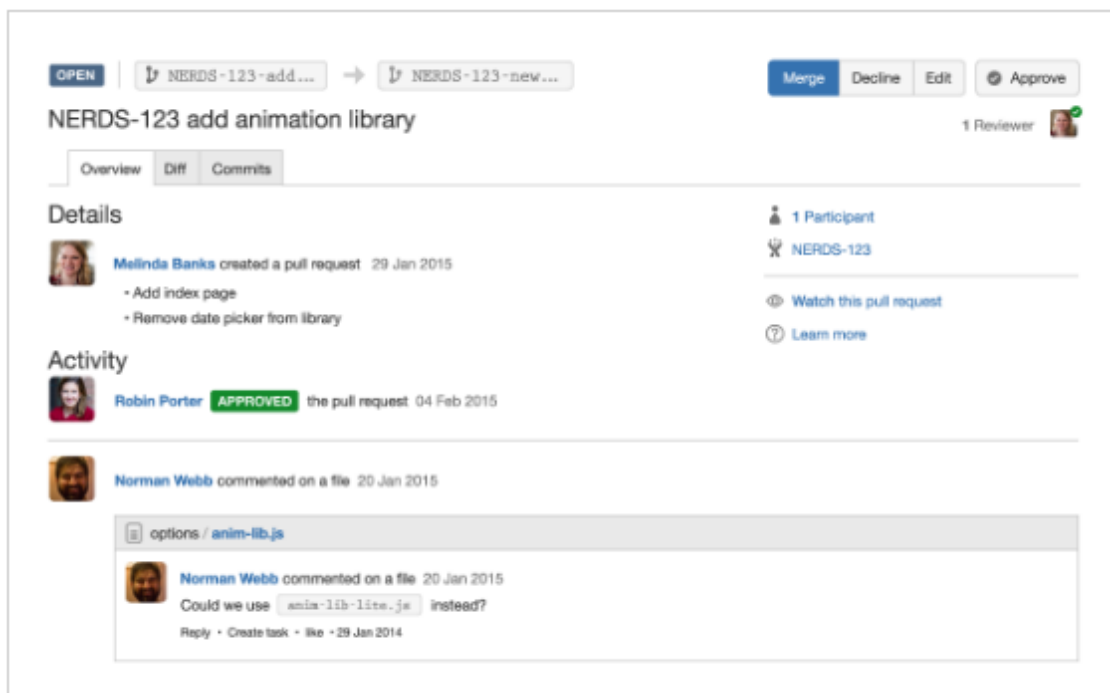
Rysunek 10 Opłaty za korzystanie z serwisu GitHub²³

Free	\$7 / mo	\$12 / mo	\$22 / mo	\$50 / mo
0 Private Repos	5 Private Repos	10 Private Repos	20 Private Repos	50 Private Repos

Ponieważ funkcjonalność obu serwisów jest podobna, małe zespoły będą decydować się na Bitbucket, ponieważ będą mogły utrzymywać dużą liczbę projektów w niewielkiej cenie, z kolei zespoły wieloosobowe będą preferować GitHub.

Na uwagę zasługuje fakt, iż do użytku na cele np. pracy dyplomowej mogą służyć otwarte repozytoria i wówczas z obu serwisów można korzystać całkowicie za darmo.

Rysunek 11 Wygląd serwisu Bitbucket



2.4 Narzędzia Business Intelligence

Business Intelligence można najprościej opisać jako proces uzyskiwania wiedzy z danych przekształcanych w informację. Ta wiedza daje przewagę nad konkurencją. Do realizacji tych celów wykorzystuje się raporty bazujące na hurtowniach danych, wydobywanie kluczowych wskaźników działalności przedsiębiorstwa jak również ich wizualizację.

W wielu korporacjach mówi się obecnie, że ten, kto ma dostęp do danych i właściwych narzędzi do ich przetwarzania, ten wygrywa. Szacuje się, iż inwestycja w narzędzia do BI zwraca się od 200-3000%²⁴. Rynek oprogramowania narzędzi BI jest bardzo dynamiczny, branża silnie się rozwija i zmienia. Siłę danej grupy narzędzi obrazuje magiczny kwadrat Gartnera.

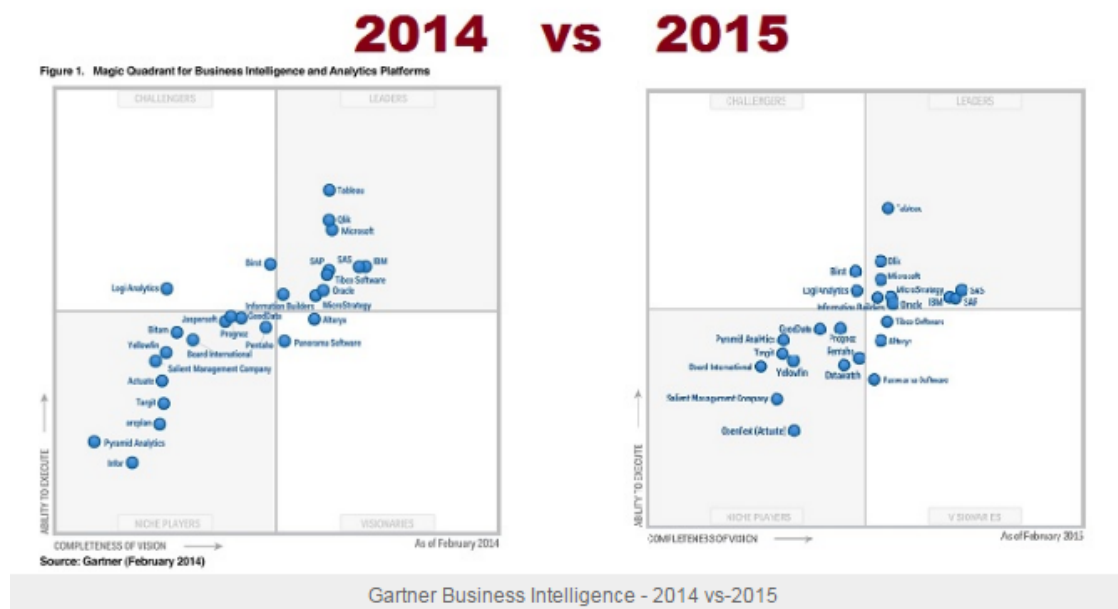
Rysunek 12 Magiczny kwadrat Gartnera 2016



Przedstawia on w zależności od opinii klientów oprogramowania podział na liderów, innowatorów, lubiących wyzwania i małych graczy. Gartner Institute²⁵ publikuje

oprócz tego szczegółowy raport o poszczególnych narzędziach prezentując ich ogólny opis oraz silne i słabe strony. Magiczny kwadrat nie do końca odzwierciedla sytuację rynkową. Niezwykle popularny jest SAP Business Objects, jednak obecnie z całego szerokiego portfela SAP, Gartner do narzędzi BI zalicza tylko SAP Lumira. Uzasadnia to tym, że pozostałe narzędzia takie jak Web Intelligence, Crystal Reports, Dashboards, czy Explorer nie spełniają norm nowoczesnych narzędzi BI²⁶. Nie tylko SAP ma problem aby w porę być wystarczająco innowacyjnym. Z zestawienia Gartnera całkowicie wypadł taki gigant jak Oracle. A jeszcze w latach 2014 i 2015 był zaliczany do liderów.

Rysunek 13 Zmiany w magicznym kwadracie Gartnera w latach 2014-2015²⁷



To pokazuje przed jakimi wyzwaniami stoją wytwórcy oprogramowania. Muszą ciągle wdrażać innowacje i uważać na błędy, gdyż wówczas pozycja ich narzędzi drastycznie się zmienia.

Tableau ostatnio umożliwił uzyskanie za darmo klucza na 6 miesięcy dla ich aplikacji desktopowej²⁸. Tableau Desktop to potężne narzędzie umożliwiające bardzo szybko wizualizację wyników wyposażony dodatkowo w liczne algorytmy ułatwiającą pracę z danymi. Utrzymuje się on ponadto na pozycji lidera już wiele lat. Do jego głównych konkurentów zalicza się SAP BO²⁹ i QlikView³⁰.

Z produktów w kategorii enterprise, zdecydowanie prym wiedzie SAP BO BI Platform. Umożliwia sprawne zarządzanie uprawnieniami, administrację serwerami,

kontami użytkowników, import autentykacji SAP, LDAP, AD.

Rysunek 14 SAP BO BI Platform CMC



Jak widać na rysunku powyżej z jednego miejsca można zarządzać połączeniami, warstwą semantyczną (uniwersami), kluczami licencyjnymi, prowadzić monitoring platformy, harmonogramować zadania.

Jest to dojrzały produkt który posiada bardzo dobrą dokumentację jak i support ze strony SAP. Platforma może być instalowana zarówno pod Linuxem jak i Windowsem. W ostatnich latach nawiązała się intensywna współpraca pomiędzy SUSE Linux i SAP, co prowadzi do dalszej optymalizacji działania. Ostatnio SUSE prowadzi intensywną kampanię swojego Linuxowego serwera oferując darmowy okres próbny 60 dni³¹.

3. Projektowanie i implementacja programu przygotowującego do egzaminu teoretycznego na brązową odznakę jeździecką.

Pracę nad tworzeniem aplikacji wykonano na komputerze Dell Latitude E5440, wyposażonym w procesor Intel(R) Core(TM) i5-4310U CPU @ 2.00GHz, 16 GB pamięci operacyjnej RAM, dysk twardy 500 GB, system operacyjny Windows 7 Enterprise Service Pack 1 64 bit. Zrzuty wykonano przy pomocy programu Microsoft One Note. Do edycji rysunków wykorzystano program Microsoft Paint.

3.1 Projektowanie Aplikacji

Cele strategiczne

Umożliwienie nauki w dowolnym miejscu i dowolnym czasie.

Ułatwienie przygotowania do egzaminu teoretycznego na brązową odznakę jeździecką.

Grupy użytkowników:

Osoby planujące zdawać egzamin na brązową odznakę jeździecką i jeźdźcy planujący poszerzyć swoją wiedzę teoretyczną:

- wybór działu do nauczania
- wyświetlanie pytań o odpowiedzi
- możliwość wyświetlenia poprzedniego pytania

Lista wymagań funkcjonalnych:

Nazwa:	Wybór działu do nauki
ID:	WF-001
Aktorzy:	Osoby planujące zdawać egzamin na brązową odznakę jeździecką i jeźdźcy planujący poszerzyć swoją wiedzę teoretyczną
Przypadek użycia:	niezbędny
Priorytet:	Wysoki

<p>Opis:</p> <p>Aplikacja ma umożliwić w prosty sposób wybór działu do nauczania.</p>
<p>Warunki początkowe:</p> <p>Posiadanie urządzenia z systemem Android</p>
<p>Kryteria akceptacji:</p> <p>Aktor klika w odpowiedni przycisk.</p>
<p>Scenariusz główny:</p> <ul style="list-style-type: none"> Aktor dokonuje wyboru działu poprzez kliknięcie w przycisk z nazwą tego działu.
<p>Scenariusze alternatywne i rozszerzenia:</p> <ul style="list-style-type: none"> Brak działu na ekranie, wywołanie kolejnego ekranu z resztą tematów.

Nazwa:	Wyświetlenie pytań i odpowiedzi
--------	---------------------------------

ID:	WF-002
Aktorzy:	Osoby planujące zdawać egzamin na brązową odznakę jeździecką i jeźdźcy planujący poszerzyć swoją wiedzę teoretyczną
Przypadek użycia:	niezbędny
Priorytet:	Wysoki
Opis:	<p>Aplikacja ma umożliwić w prosty sposób wyświetlanie pytań i odpowiedzi.</p>
Warunki początkowe:	Wybranie działu do nauczania
Kryteria akceptacji:	<p>Aktor klika w odpowiedni przycisk.</p>
Scenariusz główny:	<ul style="list-style-type: none"> Aktor wyświetla kolejne lub poprzednie pytania używając guzików poprzednie i następne.
Scenariusze alternatywne i rozszerzenia:	<ul style="list-style-type: none"> Koniec pytań w dziale- wyświetlenie komunikatu o braku większej liczby

pytań.

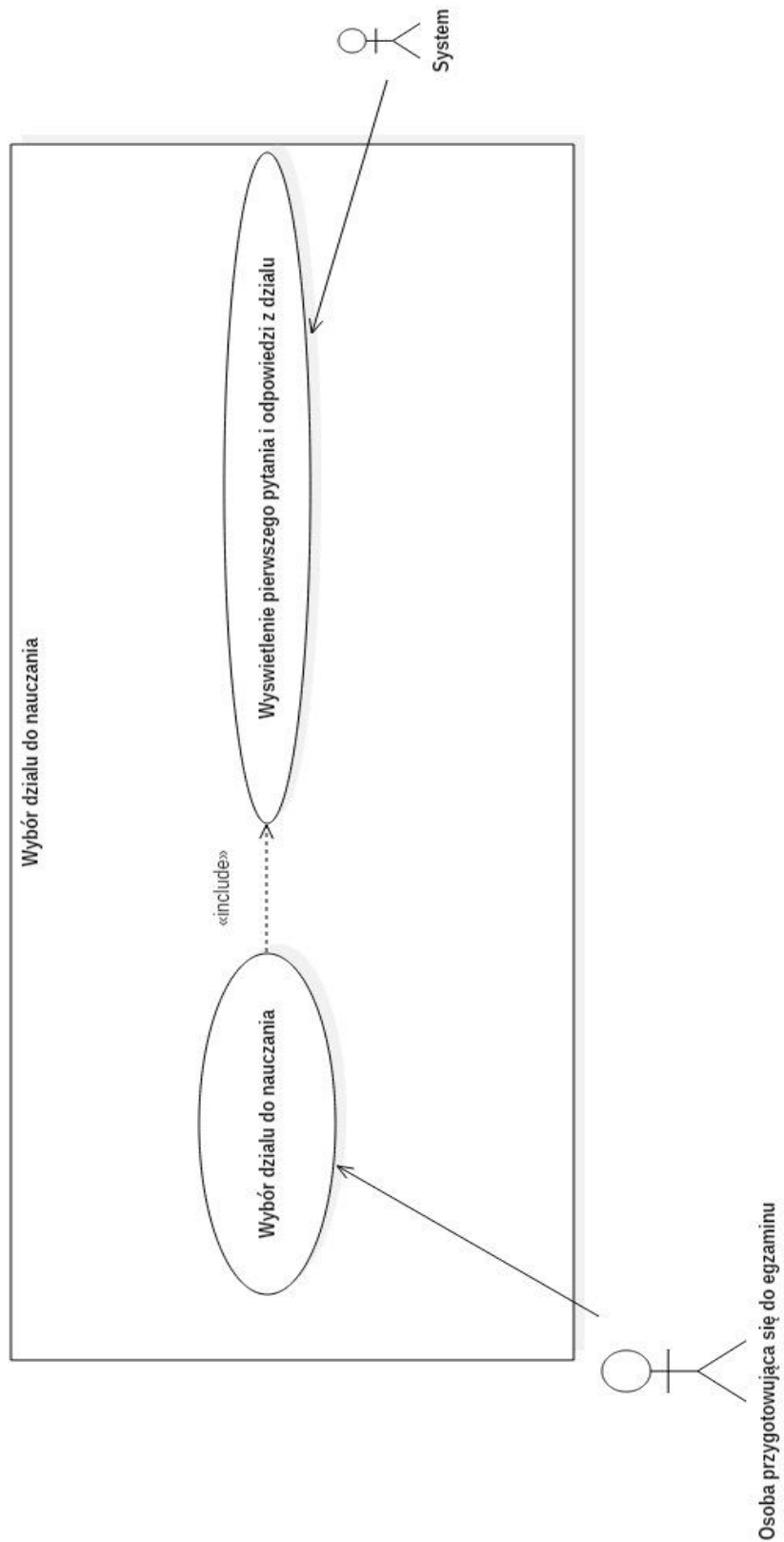
- Powrót to listy działów za pomocą przycisku

Lista wymagań niefunkcjonalnych:

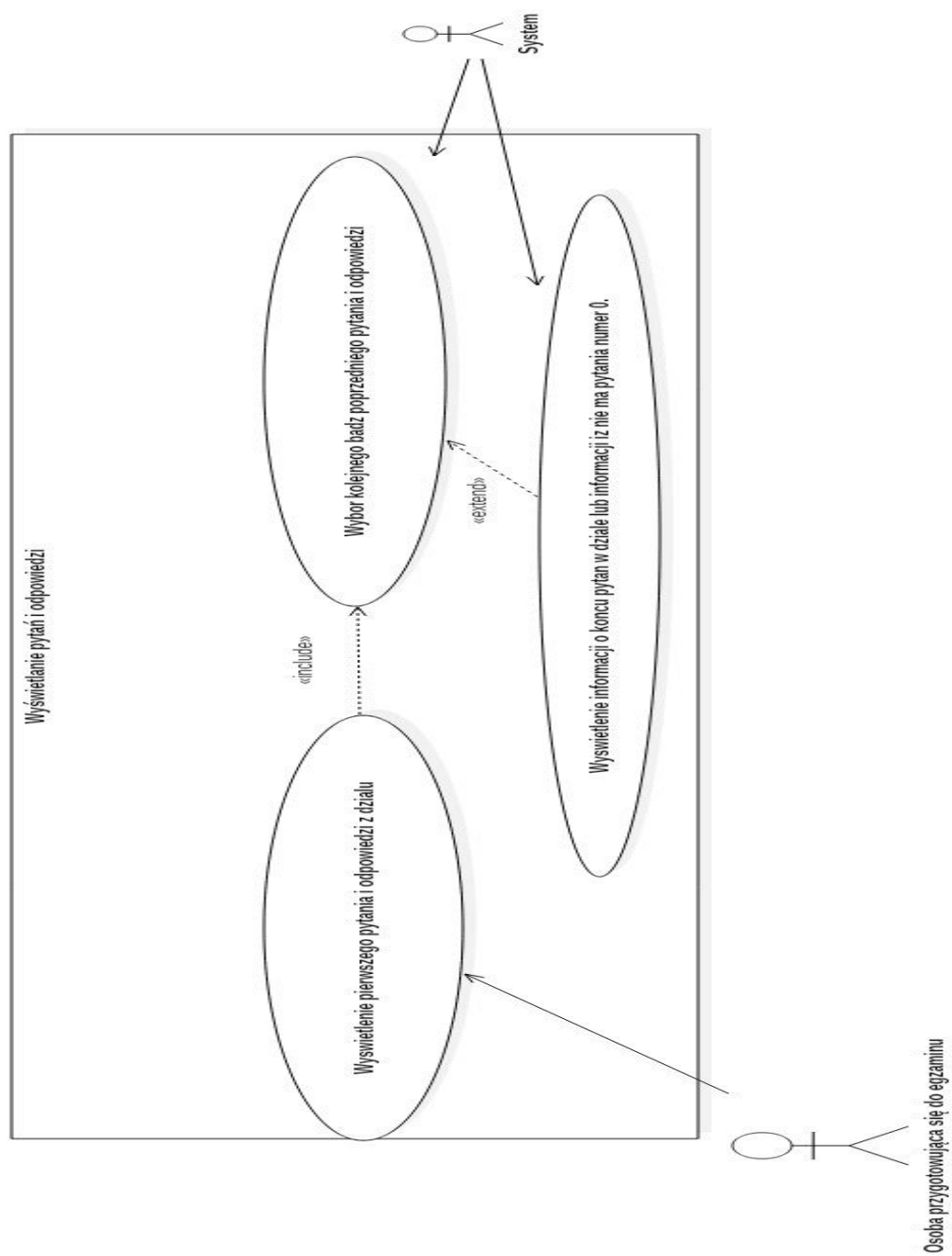
Nazwa:	Skalowalność aplikacji
ID:	WN-001
Aktorzy:	System
Przypadek użycia:	niezbędny
Priorytet:	Wysoki
Opis:	
	System ma mieć rozszerzania o kolejne elementy (nowe działy i odznaki)
Warunki początkowe:	
Kryteria akceptacji:	
Scenariusz główny:	
Scenariusze alternatywne i rozszerzenia:	

Diagramy przypadków użycia:

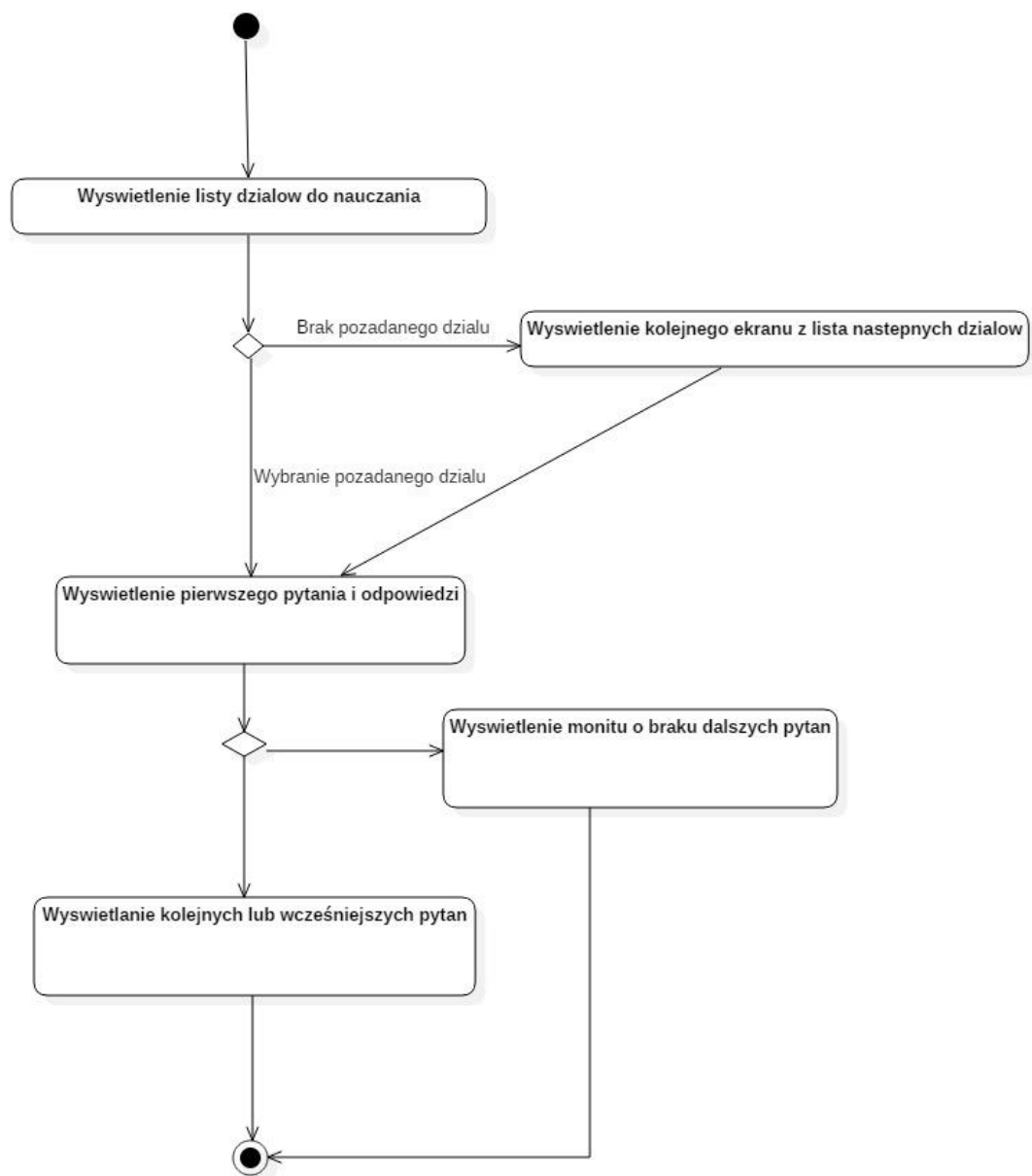
Rysunek 15 Diagram przypadków użycia- wybór działu do nauki



Rysunek 16 Diagram przypadków użycia- wyświetlanie pytań i odpowiedzi



Rysunek 17 Diagram aktywności

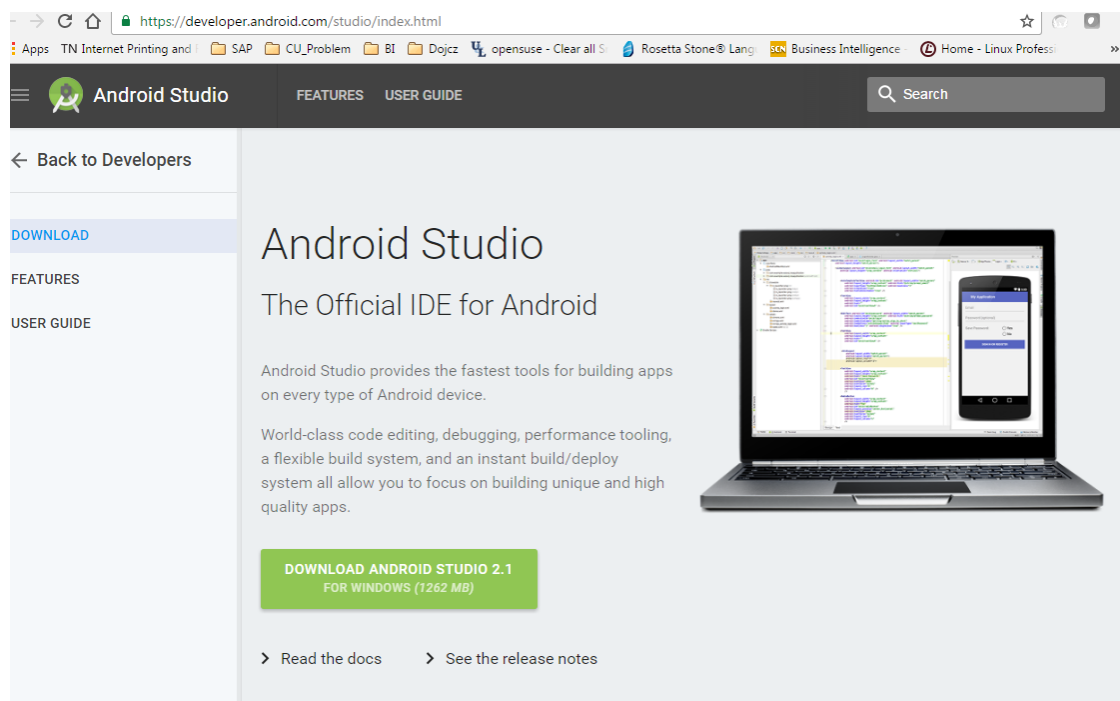


3.2 Pobieranie i instalacja Android Studio IDE, utworzenie pustego projektu.

Ze względu na to, iż do Androida zintegrowane środowisko programistyczne można pobrać i użytkować bezpłatnie, a także, iż wśród rodziny i znajomych jest to najpowszechniejszy system w smartfonach, zdecydowałem się na wybór tej platformy. Umożliwia mi to łatwą instalację aplikacji na starych „zapasowych” telefonach i rozdaniu ich uczestnikom badań.

Pracę rozpoczynam od pobrania i instalacji Android Studio³².

Rysunek 18 Pobieranie Android Studio



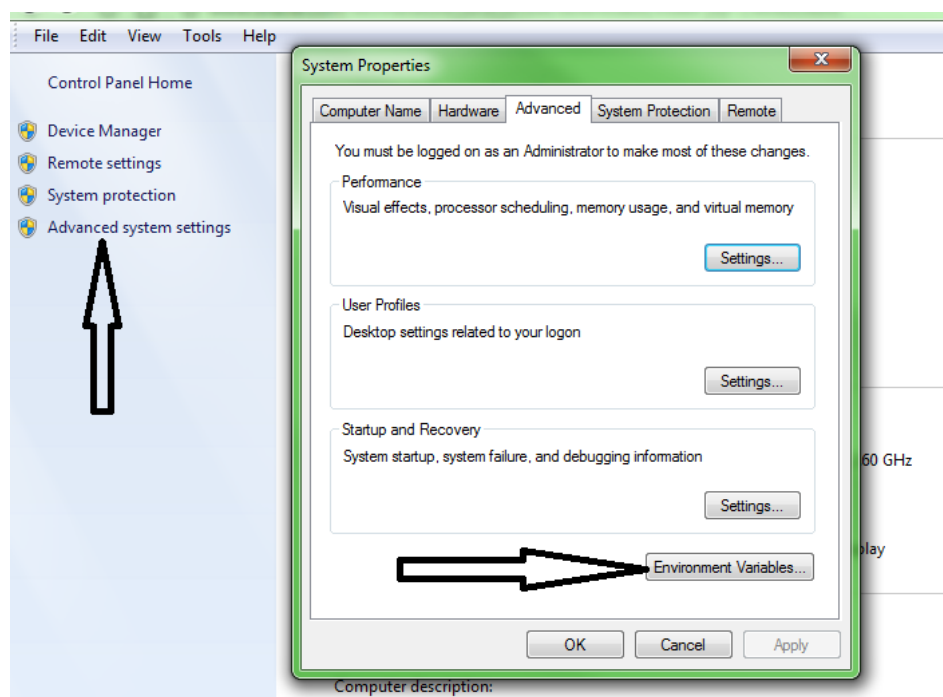
Jednym z prerekwizytów do działania Android Studio jest posiadanie Java Development Kit w wersji minimum 6, którą również pobieram bezpłatnie ze strony Oracle³³. Java Runtime Environment zainstalowany na większości komputerów standardowo nie jest wystarczający do tworzenia programów. Przy pobieraniu należy się upewnić, że ściągamy narzędzia developerskie.

Rysunek 19 Pobieranie Java Development Kit



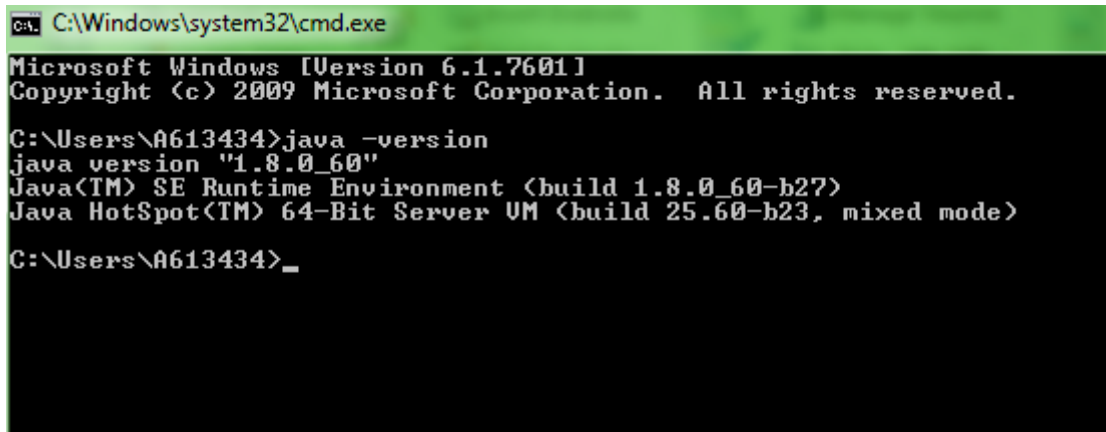
Po instalacji Java SDK należy ustawić zmienną środowiskową `JAVA_HOME`, a także dodać katalog instalacji Javy do zmiennej `PATH`. W Windowsie zmienne środowiskowe ustawia się poprzez wejście do panelu sterowania, następnie system i advanced system settings.

Rysunek 20 Ustawianie zmiennych środowiskowych



Po ustawieniu zmiennych środowiskowych upewniam się, iż zrobiłem to prawidłowo. W tym celu naciskam Super + R, wpisuję cmd i wciskam enter. Następnie w linii komend wpisuję java -version i patrzę czy system „widzi” javę.

Rysunek 21 Sprawdzanie prawidłowości konfiguracji JAVA



```
ca. C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

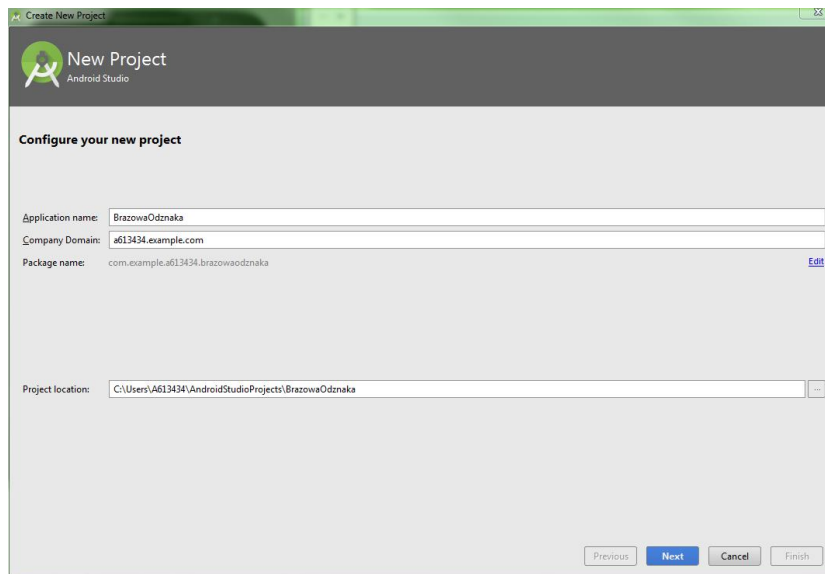
C:\Users\A613434>java -version
java version "1.8.0_60"
Java(TM) SE Runtime Environment (build 1.8.0_60-b27)
Java HotSpot(TM) 64-Bit Server VM (build 25.60-b23, mixed mode)

C:\Users\A613434>_
```

Mając zainstalowaną Javę, mogę już zainstalować i uruchomić Android Studio. Okazuje się, iż jeszcze muszę doinstalować brakujące SDK Androida jak również narzędzia developerskie. Proces ten zabiera dość dużo czasu, gdyż łącznie do pobrania jest ponad 2 GB danych.

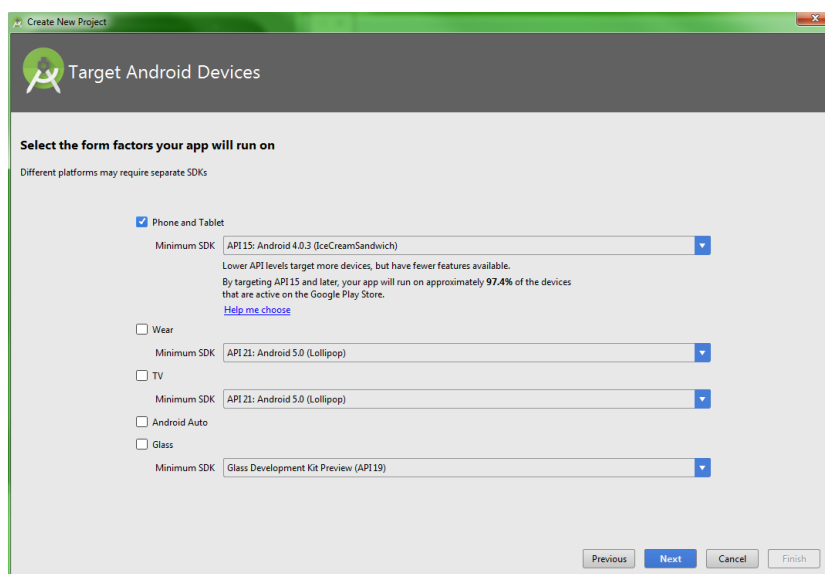
Następnie tworzę szkielet mojej aplikacji. Wybieram File -> New ->New Project. Pojawia się okno dialogowe, gdzie wpisuję nazwę aplikacji:

Rysunek 22 Tworzenie nowego projektu w Android Studio



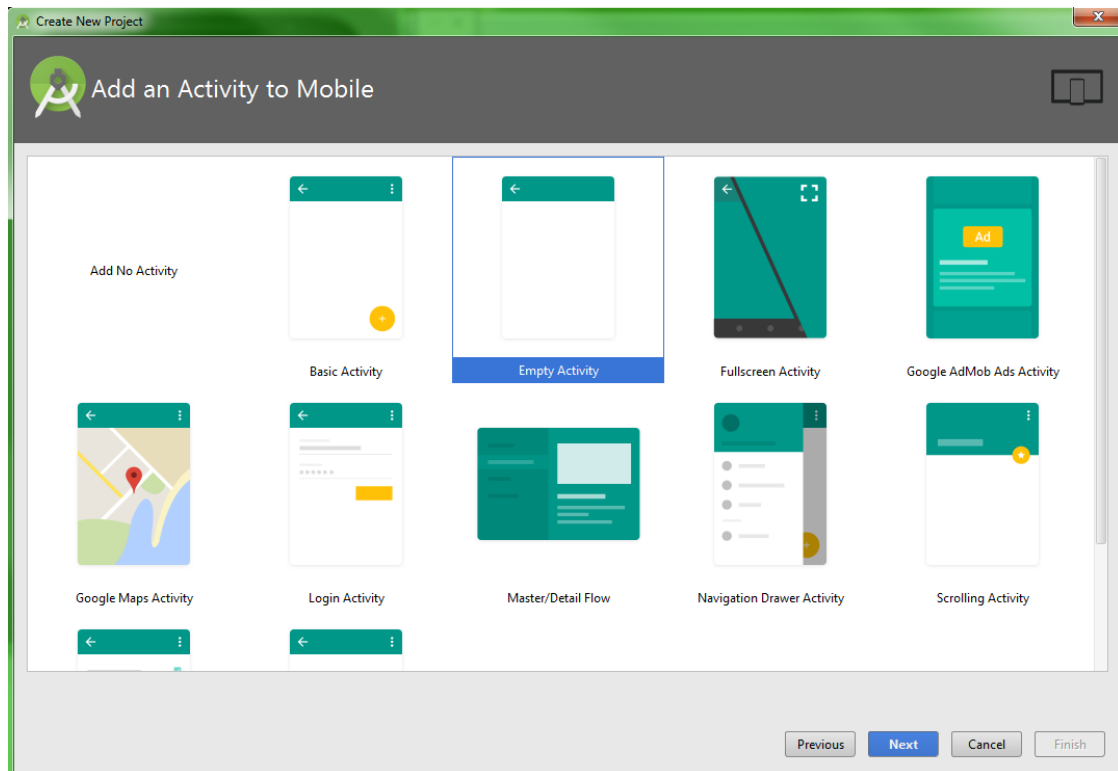
Po kliknięciu na przycisk “Next” pojawia się następne okno dialogowe. Jest ono bardzo ważne, ponieważ wybieram na nim minimalny poziom SDK, na którym będzie pracować moja aplikacja. Sprawdziłem, iż najstarszy telefon ma poziom API 16, więc na wszelki wypadek schodzę jeszcze poziom niżej. Od razu wyświetla się informacja, iż aplikacja będzie działać na 97,4% urządzeń z Androidem.

Rysunek 23 Wybór minimalnego kompatybilnego SDK



Na kolejnym ekranie pojawia się wybór aktywności jaką można wygenerować. Jako, iż na razie jestem zainteresowany utworzeniem tylko szkieletu aplikacji, wybieram Empty Activity, gdyż później sam zdecyduję co, gdzie ma się znajdować.

Rysunek 24 Wybranie jaką aktywność ma być utworzona na początku.



Następnie pojawia się okno dialogowe, gdzie można nadać nazwę aktywności, po czym zostaje utworzona gotowa aplikacja wyświetlająca na ekranie „Hallo World!”.

3.3 Utworzenie bazy danych SQLite

Egzamin teoretyczny na brązową odznakę obejmuje zakres 345pytań. Aby go zdać trzeba uzyskać 50% poprawnych odpowiedzi. Taka ilość pytań bez podziału na kategorie jest ciężka do nauczenia. Zdecydowałem podzielić się materiał na 14 kategorii:

- 1) Koń z natury
- 2) Stajnia
- 3) Podstawowe zasady bezpieczeństwa przy pracy z koniem, wypadki i pierwsza pomoc
- 4) Pielęgnacja i wyprowadzanie koni.
- 5) Cechy użytkowe konia:
- 6) Żywienie koni

- 7) Zdrowie koni
- 8) Sprzęt jeździecki
- 9) Kielznanie i siodłanie konia
- 10) Wyprowadzanie osiodłanego konia, wsiadanie i zsiadanie z konia.
- 11) Podstawowe zasady zachowania na ujeżdżalni
- 12) Podstawy jazdy konnej.
- 13) Teoria klasycznej jazdy konnej
- 14) Chody konia, skoki przez przeszkody, jazda w terenie.

Podziału dokonałem o własną wiedzę i doświadczenie- jestem instruktorem jazdy konnej od 2008 roku, jeżdżę konno od roku 1995. Przez kilkanaście lat aktywnie startowałem w zawodach z dyscypliny skoków przez przeszkody. Następnie utworzyłem tabele w Wordzie i przekopiowałem tam pytania i odpowiedzi z PDF udostępnianego przez PZJ.

To dało mi podstawę do utworzenia bazy SQLite, a jednocześnie dało materiały drukowane, które wykorzystałem później w badaniach.

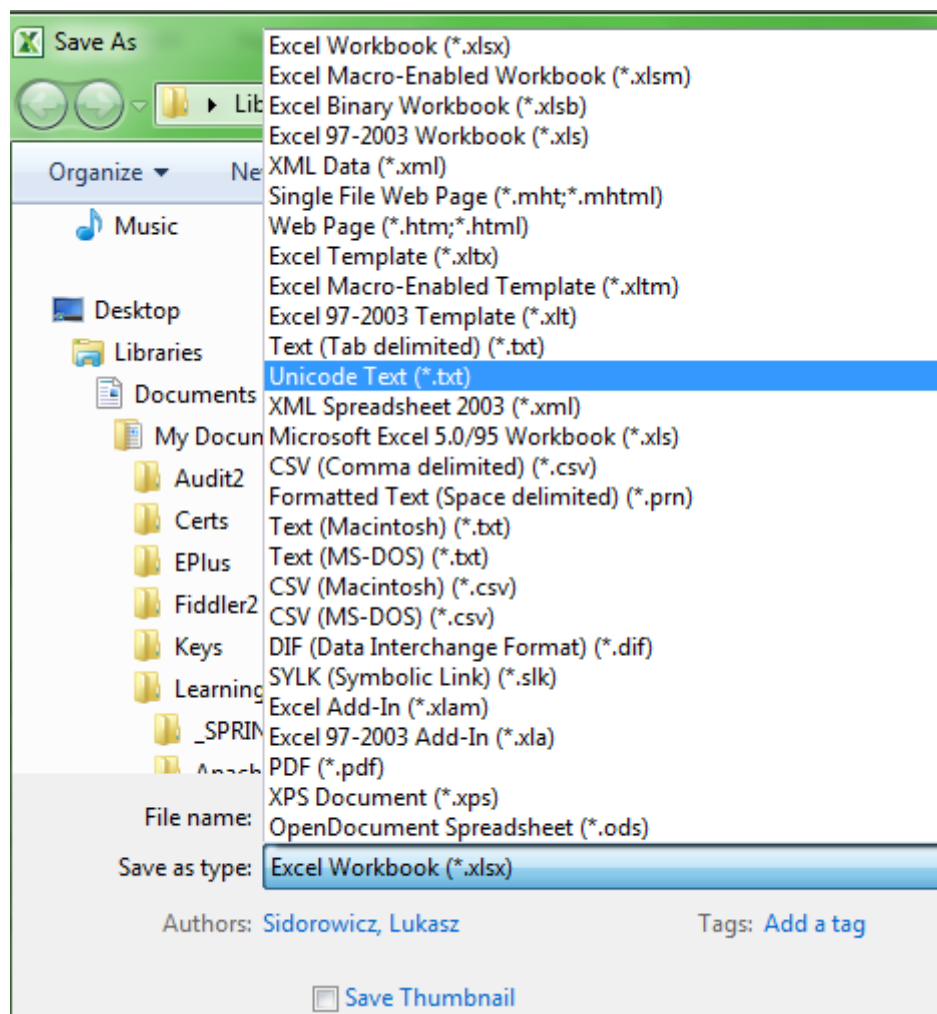
Każdą tabelę w sposób pokazany na ilustracji poniżej przenieśliem do programu Microsoft Excel 2010:

Rysunek 25 Wstawianie danych do Ms Excel 2010

	A	B	C	D	E	F	G	H
1	(Lp.	"	Pytanie	"	Odpowiedź	"	
2	(1	"	Wymień podstawowe chody konia?	"	Podstawowe chody konia to step, kłus i galop.	"	
3	(2	"	Ile taktów słychać podczas ruchu konia	"	W stepie słyszalne są cztery takty.	"	
4	(3	"	stepem na twardym podłożu?	"	np. lewa tylna, lewa przednia, prawa tylna, prawa	"	
5	(4	"	rodzaj kolepuści stawiania przez konia	"	rozróżniamy 4 rodzaje stepa: step swobodny, pośredni,	"	
6	(5	"	Jakie rozróżniamy rodzaje stepa?	"	Obustronne użycie ciężaru i rytmu pobudzają	"	
7	(6	"	Opisz pomoce stosowane do ruszania	"	stojącego konia do ruchu naprzód, ustępująca wodza na	"	
8	(7	"	stepem ze „stój”	"	jeździec podąża w rozróżnieniu za rękami konia, rytm	"	
9	(8	"	Opisz pomoce stosowane w ruchu	"	utrzymują miękki kontakt z bokami konia, ręka jeźdźcy,	"	
10	(9	"	podczas jazdy stepem.	"	konie opiera się na podłożu przekazując parę kończyn, np.	"	
11	(10	"	Opisz sposób stawiania przez konia	"	prawa tylna-lewa przednia, odbija się i na moment	"	
12	(11	"	kończyn w kłusie.	"	powstaje faza zawieszenia. Następnie stawia	"	
			"	Jakie wyróżnia się rodzaje kłusa?	"	wyróżnia się 4 rodzaje kłusa: kłus robotyczny, pośredni,	"	
			"	Jakie pomoce stosujemy przy ruszaniu	"	Obustronne użycie ciężaru i obu rądek pobudza konia	"	
			"	kłusem?	"	do ruchu naprzód, ustępująca wodza na miękki	"	
			"	Podaj cechy charakterystyczne galopu	"	Galop jest dynamicznym chodem użytkowym. Po	"	
			"	Co to jest „foule”?	"	każdym skoku galopu (tzw. „foule”) następuje faza	"	
			"		"	Foule to pojedynczy skok w galopie.	"	

Następnie z poziomu MS Excel zapisywałem plik do formatu Unicode. Umożliwia to poradzenie sobie z polskimi znakami. Próba przekonwertowania do formatu CSV powodowała wyświetlanie krzaczków zamiast polskich liter.

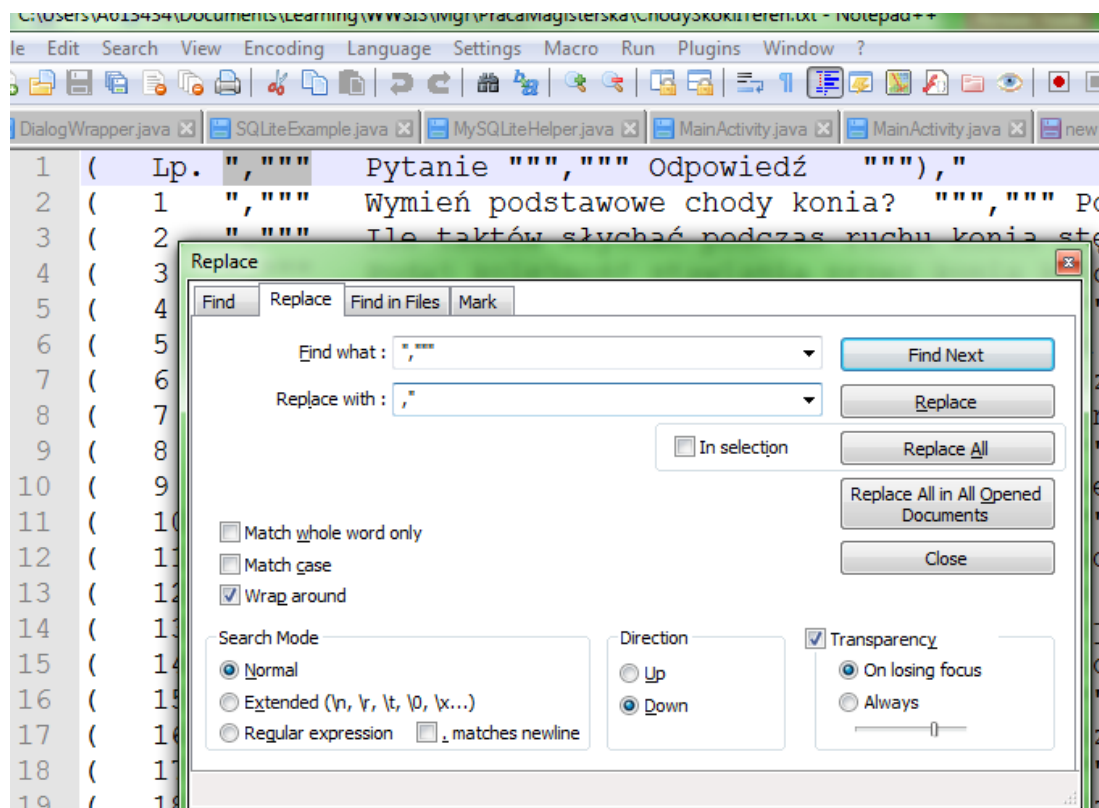
Rysunek 26 Przekonwertowanie arkusza na plik w formacie Unicode



Otrzymany plik w formacie Unicode poprawiałem w programie Notepad++³⁴ do składni SQLowej. Znając tą składnię, wiedziałem, iż każdy wiersz z excela trzeba przerobić do formatu: „(1, „Pytanie”, „Odpowiedz),”.

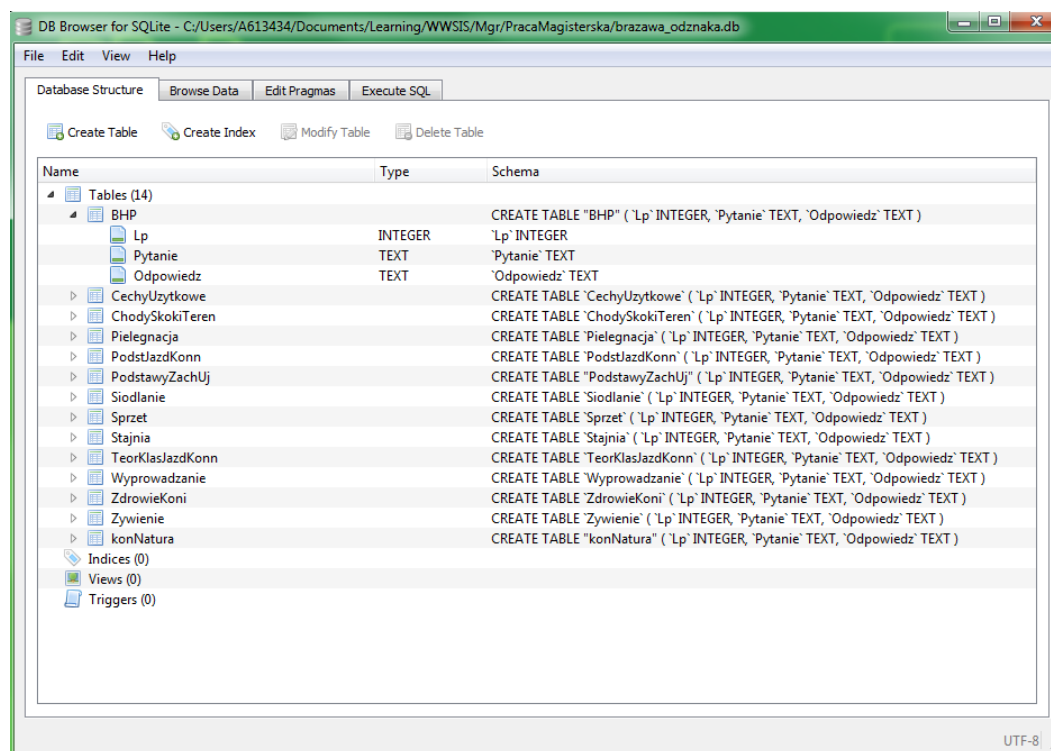
Co umożliwia później szybkie utworzenie kolejnych tabel w bazie.

Rysunek 27 Podmiana znaków w programie Notepad++



Do utworzenia bazy wykorzystałem SQLite³⁵. SQLite pomimo pewnych ograniczeń bardzo popularnym, wykorzystywanym przez takich gigantów jak Alphabet, Apple, Facebook³⁶. Pobranie i korzystanie z bazy jest bezpłatne. Silnik bazy SQLite jest wbudowany w system Android, można od razu korzystać z bazy bez konieczności instalowania czegokolwiek i problemów z kompatybilnością³⁷. Aby nie tworzyć bazy w niewygodnej linii komend, użyłem programu DB Browser for SQLite. Umożliwia on w prosty sposób utworzenie bazy, dodanie tabel i wykonywanie kodu SQL³⁸. Z pomocą tego programu utworzyłem bazę danych, utworzyłem czternaście tabel i wstawiłem do nich odpowiednie dane przygotowane wcześniej. Mając tak przygotowaną bazę mogłem wrócić do tworzenia aplikacji w Android Studio.

Rysunek 28 Tworzenie bazy danych w DB Browser for SQLite



3.4 Wykonanie aplikacji mobilnej

Dysponując szkieletem aplikacji stworzonym wcześniej i bazą danych SQLite zawierającą czternaście tabel, mogłem rozpocząć implementację aplikacji.

Pierwszą rzeczą było opracowanie ekranu startowego aby użytkownik po uruchomieniu programu widział poszczególne działy do nauczania.

Wszystkie zmiany dokonywane w projekcie dodaję do repozytorium lokalnego Git.

Na początku inicjuję śledzenie wersji poprzez otwarcie wiersza poleceń (CMD).

Przechodzę do folderu z projektem wydaję komendę:

```
git init
```

Powoduje to powstanie pustego repozytorium. Następnie trzeba dodaję wszystkie pliki i foldery do tzw. staging area poprzez komendę:

```
git add *
```

Aby pliki i foldery zostały dodane do repozytorium wykonuję polecenie

```
git commit -m „Wersja początkowa”
```

parametr „-m” powoduje możliwość dodania komentarza. W przypadku braku tego parametru otworzy się edytor tekstowy i komentarz trzeba dodać z jego poziomu. Ja preferuję wersję konsolową. Jest to indywidualna preferencja. Po każdych zmianach powtarzam komendy git add i git commit -m, żeby w razie napisania złego nowego fragmentu kodu móc wrócić do wcześniejszej działającej wersji.

Tak zwany layout projektuję w xml, podaję poniżej fragment z przykładem ustawienia etykiety (TextView) i jednego przycisku (Button):

<Button

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Koń z natury"
    android:onClick="info"
    android:id="@+id/buttonCwiczenia"
    android:layout_below="@+id/textView3"
    android:layout_toLeftOf="@+id/button"
    android:layout_alignParentLeft="true"
```



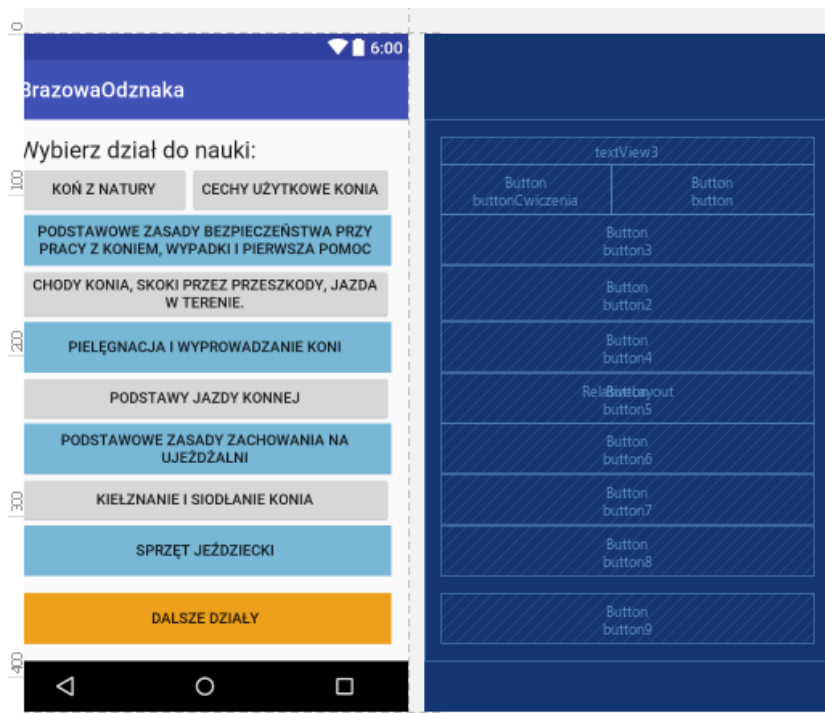
```
android:layout_alignParentStart="true" />
```

```
<TextView
```

```
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:textAppearance="?android:attr/textAppearanceLarge"
    android:text="Wybierz dział do nauki:"
    android:id="@+id/textView3"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true"
    android:layout_alignParentTop="true"
    android:layout_alignParentRight="true"
    android:layout_alignParentEnd="true" />
```

następnie korzystam z możliwości podglądu.

Rysunek 29 Tworzenie layoutu menu głównego aplikacji



Dla moich przycisków chcę używać własnych zdefiniowanych kolorów. Dodaję nowe

wartości w colors.xml:

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="colorPrimary">#3F51B5</color>
    <color name="colorPrimaryDark">#303F9F</color>
    <color name="colorAccent">#FF4081</color>
    <color name="Czerwony">#eda01b</color>
    <color name="Czarny">#ff0a0a0c</color>
    <color name="Brazowy">#ffaf3c00</color>
    <color name="blue">#78b8d6</color>
    <color name="Zielony">#ff20ff00</color>
    <color name="Morelowy">#ffffed36</color>

```

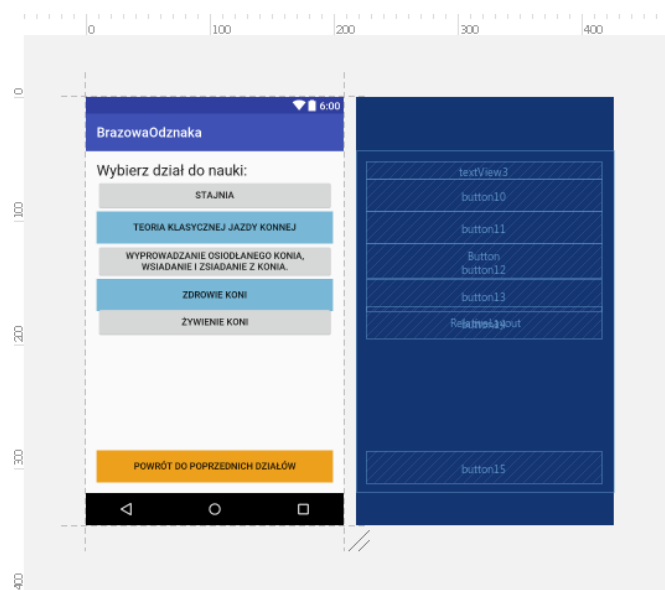
Dodaję również nazwę aplikacji do pliku strings.xml

```
<resources>
    <string name="app_name">BrazowaOdznaka</string>
    <string name="action_settings">Settings</string>
</resources>

```

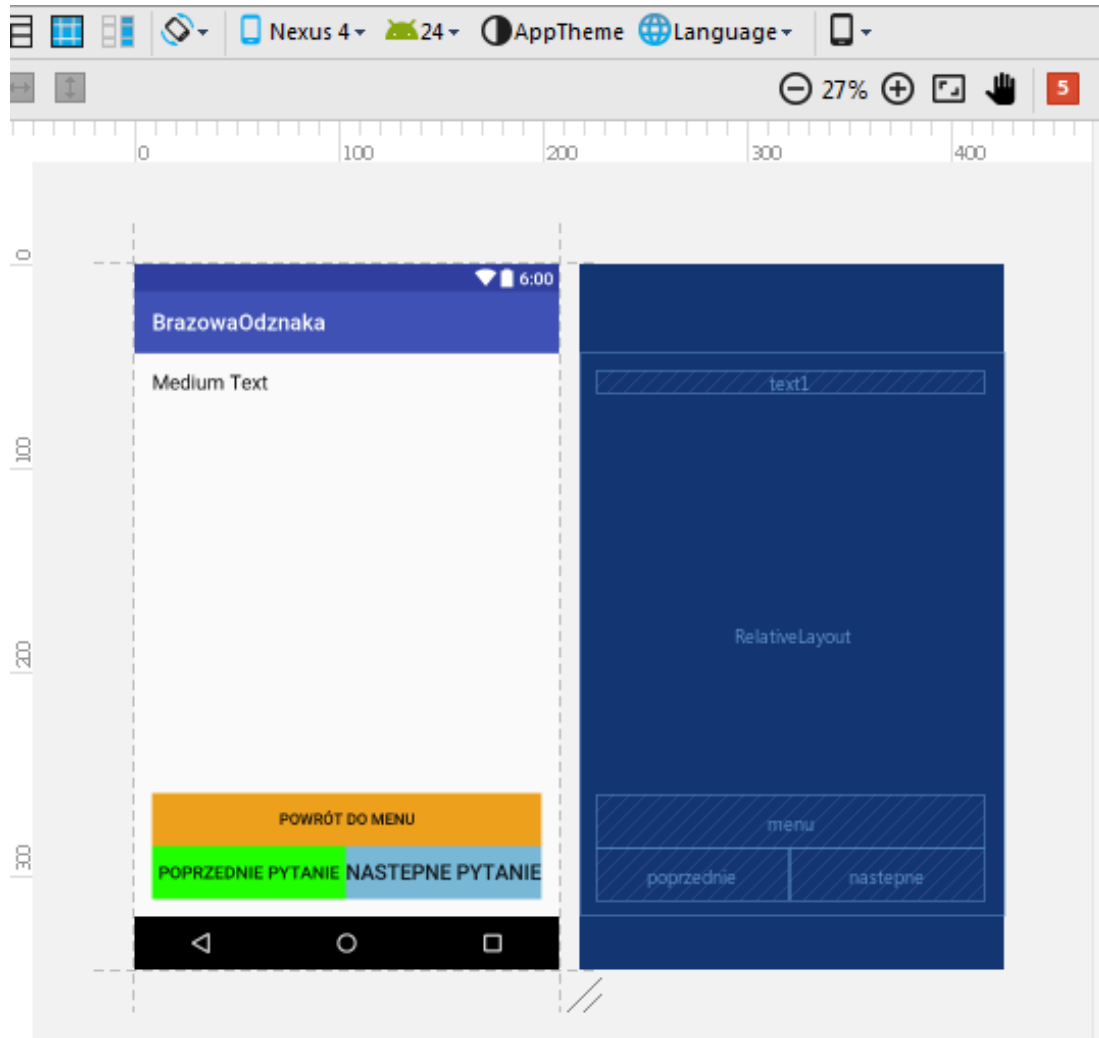
Tematy niemieszczące się na ekranie przesuwam na kolejny

Rysunek 30 Tworzenie layoutu listy pozostałych działów



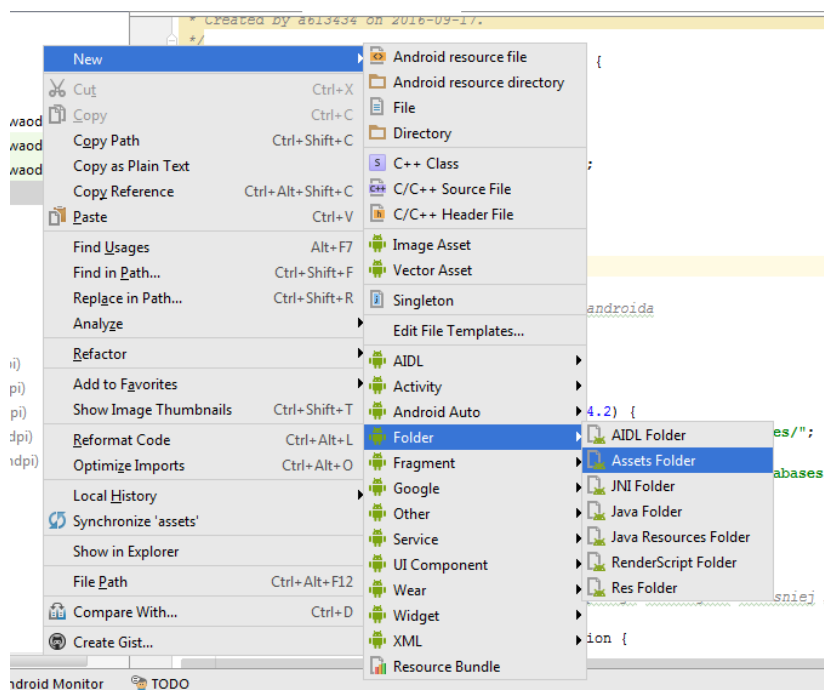
Projektuję również ekran do wyświetlania pytań i odpowiedzi:

Rysunek 31 Projektowanie layoutu do wyświetlania pytań i odpowiedzi.



Tym sposobem kończę wizualne przygotowanie aplikacji. Pozostaje oprogramować poszczególne funkcje, tak, aby spełnić założenia projektowe. Ponieważ aplikacja będzie wykorzystywać utworzoną bazę SQLite tworzę w z poziomu środowiska Android Studio folder „Assets”. Konieczne jest wykonanie tego kroku z poziomu platformy, utworzenie folderu z poziomu systemu operacyjnego powoduje, iż później nasza aplikacja nie widzi bazy.

Rysunek 32 Dodanie folderu Assets do projektu



Do utworzonego w ten sposób folderu kopiuję utworzoną wcześniej bazę `brazawa_odznaka.db`.

Dzięki temu, że SQLite jest wbudowany w Androida, mogę dostać się do bazy bez wykorzystywania jakichkolwiek sterowników takich jak JDBC.

Dodaję w pliku `AndroidManifest.xml` wymaganie, aby aplikacja miała prawo zapisu w pamięci zewnętrznej:

```
<uses-permission
```

```
android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Pracę z bazą rozpoczynam od utworzenia klasy `BazaHelper` rozszerzającej `SQLiteOpenHelper`³⁹. Na początku definiuję zmienne ze ścieżką do bazy, nazwą bazy i tabel w niej zawartych:

```
private static String DB_PATH = "";
```

```
private static String DB_NAME = "brazawa_odznaka2.db";
```

```
private static String BHP = "BHP";
```

```
private static String konNatura = "konNatura";
```

```
private static String cachaUzytkowe = "CechyUzytkowe";
```

```
private static String chodySkokiTeren = "ChodySkokiTeren";
```

```
private static String pielegnacja = "Pielegnacja";
```

```

private static String podstJazdKonnej= "PodstJazdKonn";
private static String podstZachUj="PodstawyZachUj";
private static String siodlanie = "Siodlanie";
private static String sprzet = "Sprzet";
private static String stajnia = "Stajnia";
private static String teoriaKlasJazdKonn = "TeorKlasJazdKonn";
private static String wyprowadzanie = "Wyprowadzanie";
private static String zdrowieKoni = "ZdrowieKoni";
private static String zywienie= "Zywienie";
private SQLiteDatabase myDataBase;
private final Context myContext;

```

Ponieważ system operacyjny Android zmienia się dosyć mocno z kolejną wersją, mogą występować problemy ze ścieżką do bazy. Aby rozwiązać ten problem tworzę konstruktor, który dostosowuje ścieżkę do różnych wersji Androida⁴⁰:

```

public BazaHelper(Context context) {
    super(context, DB_NAME, null, 1);
    if (android.os.Build.VERSION.SDK_INT >= 4.2) {
        DB_PATH = context.getApplicationInfo().dataDir + "/databases/";
    } else {
        DB_PATH = "/data/data/" + context.getPackageName() + "/databases/";
    }
    this.myContext = context;
}

```

Aby utworzona przeze mnie baza danych mogła być zainstalowana w aplikacji wykorzystuję metodę `createDataBase()`, która najpierw sprawdza czy jakkolwiek baza jest już utworzona, gry nie tworzy pustą bazę, a następnie kopiuje bazę z folderu Assets:

```

public void createDataBase() throws IOException {

```

```

boolean dbExist = checkDataBase();

if (dbExist) {

} else {
    this.getWritableDatabase();

    try {
        copyDataBase();
    } catch (IOException e) {
        throw new Error("Błąd kopiowania bazy");
    }
}

```

W celu sprawdzenia, czy baza istnieje implementuję metodę checkDataBase():

```

private boolean checkDataBase() {

    SQLiteDatabase checkDB = null;
    try {
        String myPath = DB_PATH + DB_NAME;
        checkDB = SQLiteDatabase.openDatabase(myPath, null,
        SQLiteDatabase.OPEN_READONLY);
    } catch (SQLException e) {

    }
    if (checkDB != null) {
        checkDB.close();
    }
}

```

```

    return checkDB != null ? true : false;
}

```

Kopiowanie bazy odbywa się z wykorzystaniem metody `copyDataBase()`, która otwiera nowoutworzoną pustą bazę, a następnie kopiuje do niej gotową bazę za pomocą strumienia bajtów:

```

private void copyDataBase() throws IOException {

```

```

    InputStream myInput = myContext.getAssets().open(DB_NAME);

```

```

    String outFileName = DB_PATH + DB_NAME;

```

```

    OutputStream myOutput = new FileOutputStream(outFileName);

```

```

    byte[] buffer = new byte[1024];

```

```

    int length;

```

```

    while ((length = myInput.read(buffer)) > 0) {

```

```

        myOutput.write(buffer, 0, length);

```

```

    }

```

```

    myOutput.flush();

```

```

    myOutput.close();

```

```

    myInput.close();

```

```

}

```

Ponieważ rozszerzałem klasę abstrakcyjną, przesłaniam metody `onCreate()`, `close()` i `onUpgrade()`:

```

@Override

```

```

public void onCreate(SQLiteDatabase db) {

```

```

}

```

@Override

```
public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {  
  
}
```

@Override

```
public synchronized void close() {  
    if (myDataBase != null)  
        myDataBase.close();  
    super.close();  
  
}
```

Jako ostatnią rzecz w klasie BazaHelper tworzę metodę selectWords(), która wykorzystując klasę Cursor odczyta dane z bazy i wstawi do ArrayListy.

```
public List<String> selectWords() {  
    List<String> list = new ArrayList<String>();  
    Cursor cursor = this.myDataBase.query(BHP, new String[] { "Lp", "Pytanie",  
"Odpowiedz" }, null, null, null, null, null);  
    if (cursor.moveToFirst()) {  
        do {  
            list.add(cursor.getString(0)+"  
.\n\nPytanie:\n" + cursor.getString(1) +  
"\n\nOdpowiedź:\n" + cursor.getString(2));  
        } while (cursor.moveToNext());  
    }  
    if (cursor != null && !cursor.isClosed()) {  
        cursor.close();  
    }  
    return list;  
}
```


Metodę tą później kopiuję dla każdej tabeli i tym sposobem mam odczytaną całą bazę danych. Pozostaje opracować wyświetlenie i przewijanie elementów ArrayListy.

Tworzę nową aktywność WyświetlaniePytan. Przy otwarciu planuję wyświetlić pytanie i odpowiedź numer jeden, będę w tym celu pobierał element ArrayListy numer 0, aby tu umożliwić tworzę następujące zmienne:

```
Integer myInt=0;  
String Pytanie;  
BazaHelper myDbHelper;  
Button poprzed;  
Button nast;  
TextView text;
```

Ustawiam następnie widok i inicjuję object z klasy BazaHelper, przyporządkowuję także pole tekstowe i przyciski do odpowiednich zmiennych:

```
setContentView(R.layout.activity_wyswietlanie_pytan);  
myDbHelper = new BazaHelper(this);  
text = (TextView) findViewById(R.id.text1);  
poprzed = (Button) findViewById(R.id.poprzednie);  
nast = (Button) findViewById(R.id.nastepne);
```

Po tym inicjuję bazę danych i ją otwieram, abym mógł pobrać i wyświetlić pierwsze pytanie i odpowiedź. Tworzę także listę ciągów znaków i ustawiam pierwszy element w polu tekstowym:

```
try {  
    myDbHelper.createDataBase();  
} catch (IOException ioe) {  
    throw new Error("Niepowodzenie tworzenia bazy");  
}
```

```

try {
    myDbHelper.openDataBase();
} catch (SQLException sqle) {
    throw sqle;
}

final List<String> names = myDbHelper.selectWords();
Pytanie = names.get(myInt);
text.setText(Pytanie);

```

Oprogramowuję pierwszy przycisk, mający służyć wyświetlaniu kolejnych pytań. Zabezpieczam program także, poza wyjściem za ostatni element listy poprzez wyświetlenie komunikatu o braku większej ilości pytań w danym dziale:

```

nast.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        myInt++;
        Integer safe = names.size()-1;
        if (myInt<=safe && myInt >=0){
            Pytanie = names.get(myInt);
            text.setText(Pytanie);}

        else {
            Toast.makeText(getApplicationContext(),"Nie ma już więcej pytań w tym
dziale.", Toast.LENGTH_LONG).show();
        }
    }
});

```

Podobnie oprogramowuję przycisk do przewijania listy do poprzednich pytań, zabezpieczając przed zejściem poniżej pierwszego elementu listy:

```

poprzed.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        myInt--;
        Integer safe = names.size()-1;
        if (myInt<=safe && myInt >=0){
            Pytanie = names.get(myInt);
            text.setText(Pytanie);}
        else {
            Toast.makeText(getApplicationContext(), "Nie ma pytania o numerze 0 i
nizej.", Toast.LENGTH_LONG).show();
        }
    }
});
}

```

Oprogramuję także przycisk pozwalający na powrót do menu głównego, za pomocą wywołania Intenta, nazwa metody została zdefiniowana wcześniej w xml:

```

public void menu (View view) {
    Intent menu = new Intent (getApplicationContext(), MainActivity.class);
    startActivity(menu);
}

```

Aby móc przetestować działanie aplikacji, czy poprawnie pobiera i wyświetla dane, muszę jeszcze oprogramować przynajmniej jeden przycisk w klasie głównej wyświetlającej działu do nauki. Ponieważ w wyświetlaniu pytań ustawiłem tabelkę „BHP”, toteż oprogramuję na razie tylko przycisk „Podstawowe zasady bezpieczeństwa przy pracy z koniem, wypadki i pierwsza pomoc”. Robię to poprzez wywołanie Intenta do zdefiniowanej wcześniej metody onClick() z widoku xml:

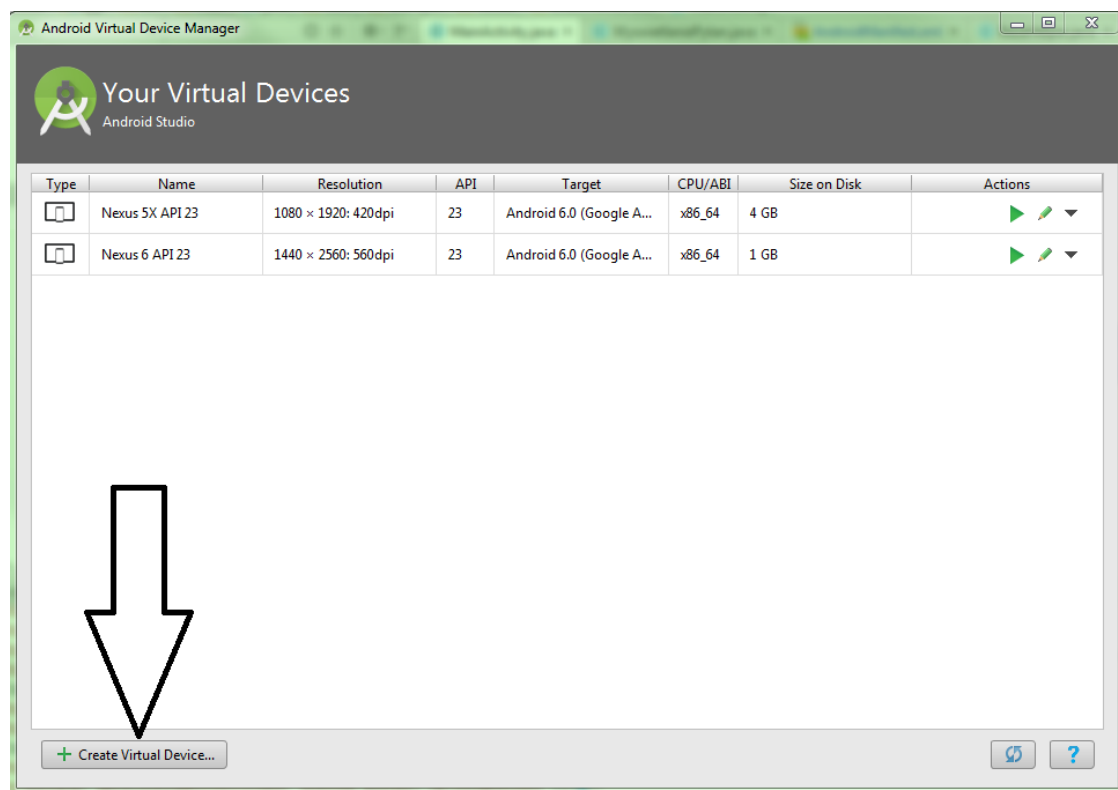
```

public void BHP (View view){
    Intent BHP = new Intent(getApplicationContext(), WyświetlaniePytan.class);
    startActivity(BHP);
}

```

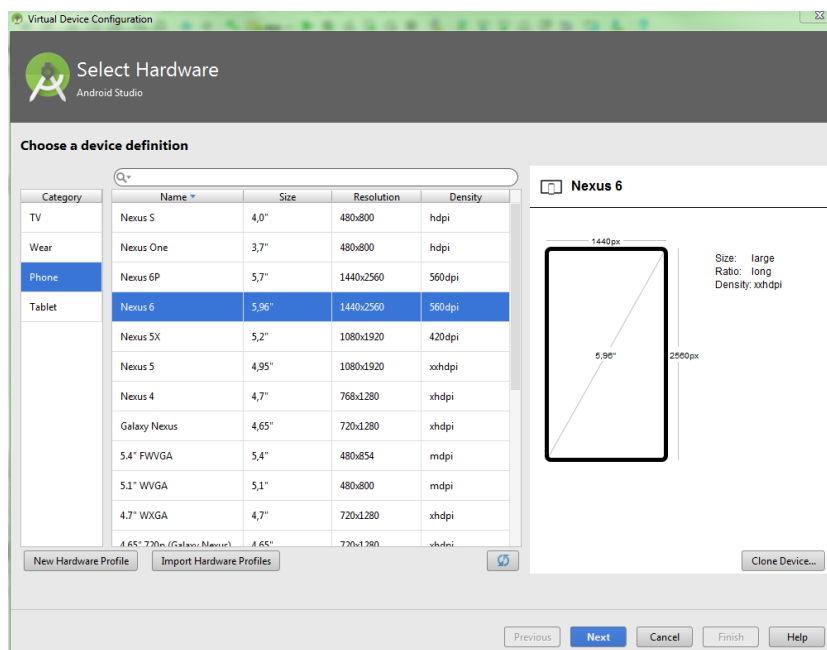
Dysponuję w tym momencie aplikacją, która powinna się uruchomić i wyświetlać pytania. Do testów najlepiej wykorzystać wirtualne urządzenie Android. Zainstalowane środowisko pozwala na utworzenie i używanie takiego urządzenia. Zaletą takiej metody testowania jest brak ingerencji w rzeczywistym urządzeniu, jak również możliwość przetestowania na różnych modelach telefonów i tabletów. Uruchamiam Android Virtual Device Manager i przyciskam Create Virtual Device:

Rysunek 33 Tworzenie wirtualnego urządzenia Android



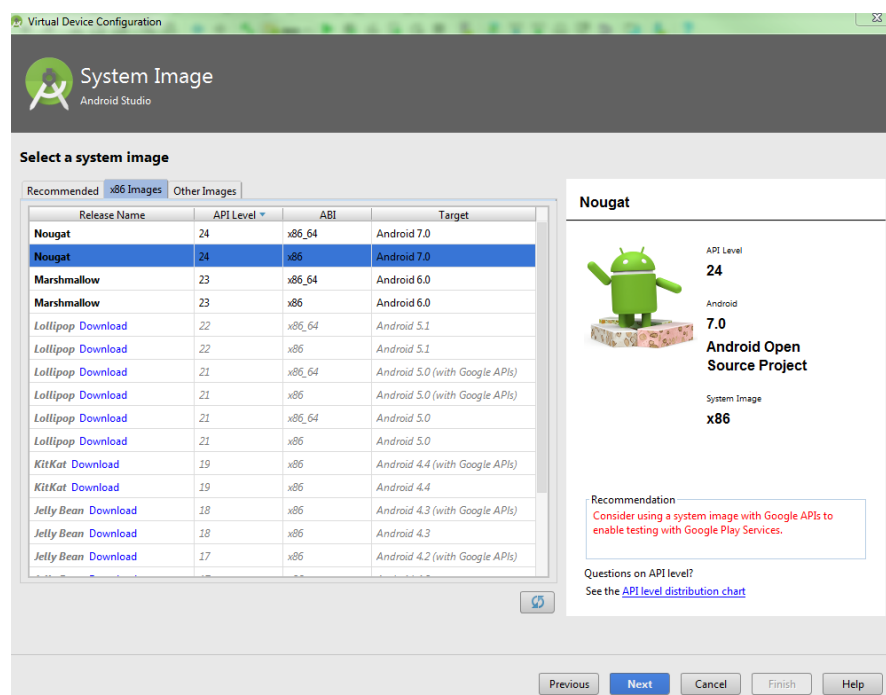
Następnie otrzymuję listę urządzeń, dostępne obecnie kategorie obejmują Phone, Tablet, Wear (Smartwatche) i nawet TV. Wybieram kategorię telefonu i model Nexus6.

Rysunek 34 Wybór tworzonego wirtualnego urządzenia



Na kolejnym ekranie wybieram, jaka wersja Androida ma być zainstalowana na urządzeniu wirtualnym. Jest to bardzo istotne w przypadku testów. Dzięki tej funkcji można przejrzeć, czy kompatybilność ze starszymi urządzeniami jest zaimplementowana prawidłowo. Jeśli jakiś poziom API jest niedostępny, można go od razu pobrać.

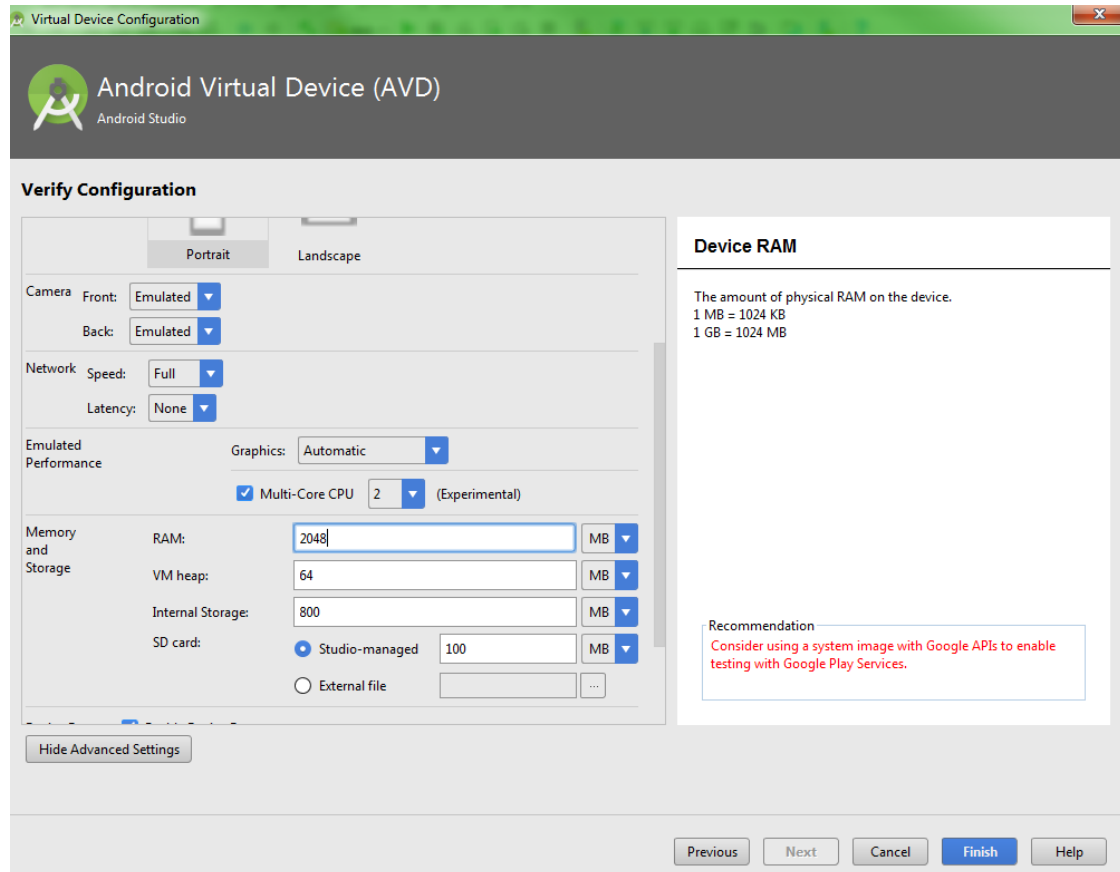
Rysunek 35 Wybór poziomu API w tworzonym urządzeniu



Na kolejnym i zarazem ostatnim ekranie mamy możliwość przejrzeć i zmienić

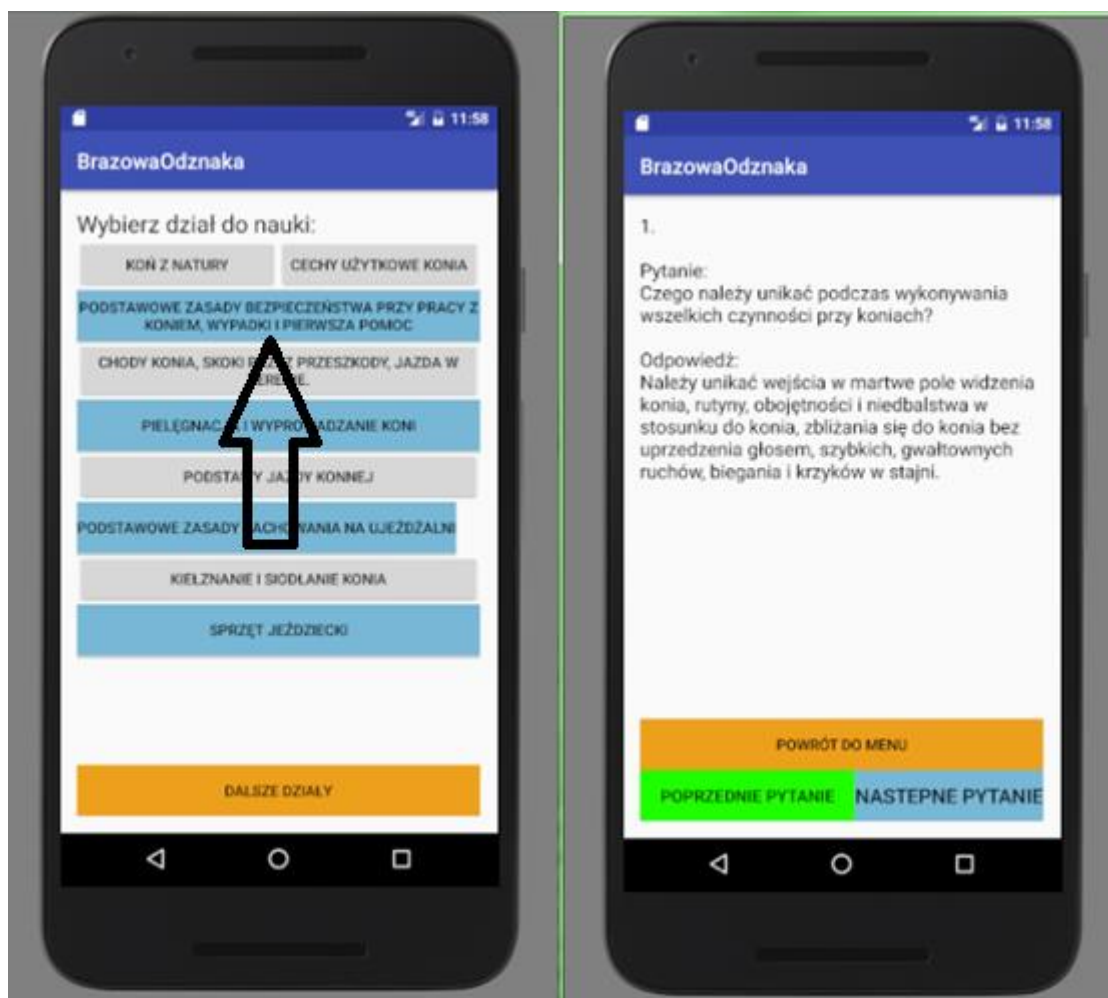
wybrane wcześniej ustawienia, jak również w opcje zaawansowane. Polecam tę opcję i zwiększenie ilości RAM. Pozwoli to na szybsze uruchamianie aplikacji i jej testowanie. Pozwala to zaoszczędzić dużo czasu w trakcie późniejszej pracy. Należy tylko pamiętać, aby dostosować ten parametr do posiadanych zasobów na komputerze.

Rysunek 36 Wybór ustawień zaawansowanych dla tworzonej maszyny wirtualnej.



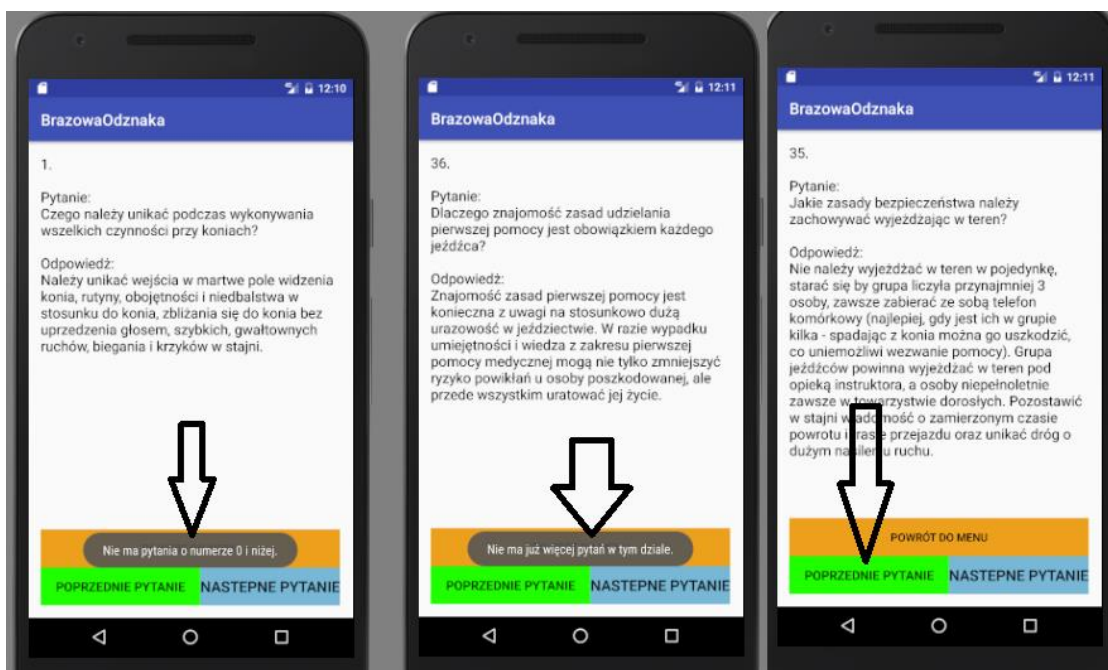
Uruchamiam nowoutworzone wirtualne urządzenie i sprawdzam działanie aplikacji. W szczególności interesuje mnie, czy prawidłowo się otwiera i po naciśnięciu w przycisk „Podstawowe zasady bezpieczeństwa przy pracy z koniem, wypadki i pierwsza pomoc” otworzy nowy ekran i wyświetli pierwsze pytanie i odpowiedź z tego działu. Będzie to oznaczało, iż prawidłowo została zaimplementowana klasa BazaHelper, jak również opcja pobrania pierwszego elementu z listy.

Rysunek 37 Otwarcie aplikacji i wyświetlenie pierwszego pytania z wybranego działu



Kolejnym elementem pozostałym do sprawdzenia pozostaje, czy działa przewijanie pytań i zabezpieczenie przed wyjściem poza tablicę działów i wyświetlenie komunikatu. W tym celu klikam najpierw przycisk „Poprzednie pytanie” i patrzę, czy wyświetla się komunikat. Następnie przewijam kolejne pytania przyciskiem „Następne pytanie” do końca listy i próbuję dalej, następnie naciskam na przycisk „Poprzednie pytanie”. Okazuje się, że wszystko działa prawidłowo:

Rysunek 38 Sprawdzenie działania aplikacji.



Dzięki sprawdzeniu, iż funkcje wyświetlania działają prawidłowo, przechodzę do implementacji kolejnych przycisków. Dla każdego tworzę nową aktywność rejestrując ją w AndroidManifest.xml:

```
<activity
    android:name="com.example.lukasz.brazowaodznaka.WyswietlaniePytan" />
<activity
    android:name="com.example.lukasz.brazowaodznaka.CechyUzytkowe" />
<activity android:name="com.example.lukasz.brazowaodznaka.Skoki" />
<activity android:name="com.example.lukasz.brazowaodznaka.Pielegnacja" />
<activity
    android:name="com.example.lukasz.brazowaodznaka.PodstawyJazdyKonne" />
<activity
    android:name="com.example.lukasz.brazowaodznaka.PodstawyZachowaniaNaUjezdzalni" />
<activity android:name="com.example.lukasz.brazowaodznaka.Siodlanie" />
<activity android:name="com.example.lukasz.brazowaodznaka.Sprzet" />
<activity android:name="com.example.lukasz.brazowaodznaka.Main2Activity"
/>
<activity android:name="com.example.lukasz.brazowaodznaka.Stajnia" />
```



```

<activity
    android:name="com.example.lukasz.brazowaodznaka.TeoriaKlasJazdKonnej"
/>

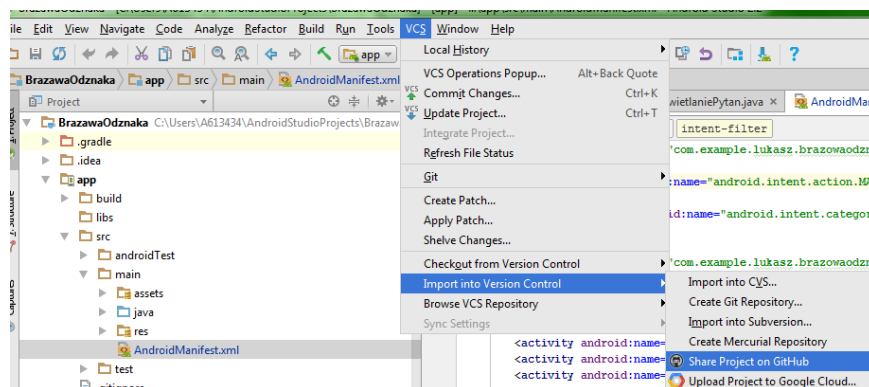
<activity
    android:name="com.example.lukasz.brazowaodznaka.Wyprowadzanie" />
<activity android:name="com.example.lukasz.brazowaodznaka.ZdrowieKoni"
/>

<activity android:name="com.example.lukasz.brazowaodznaka.Zywienie">

```

Następnie kopiuję zawartość klasy WyświetlaniePytan.java, podmieniając treść do wyświetlenia. Tym sposobem dysponuję gotową aplikacją. Dla bezpieczeństwa umieszczam ją na platformie GitHub jako repozytorium otwarte. Korzystam, z tego iż w Android Studio jest wbudowana funkcja automatycznego podzielenia się projektem na tej platformie:

Rysunek 39 Umieszczenie projektu na GitHub

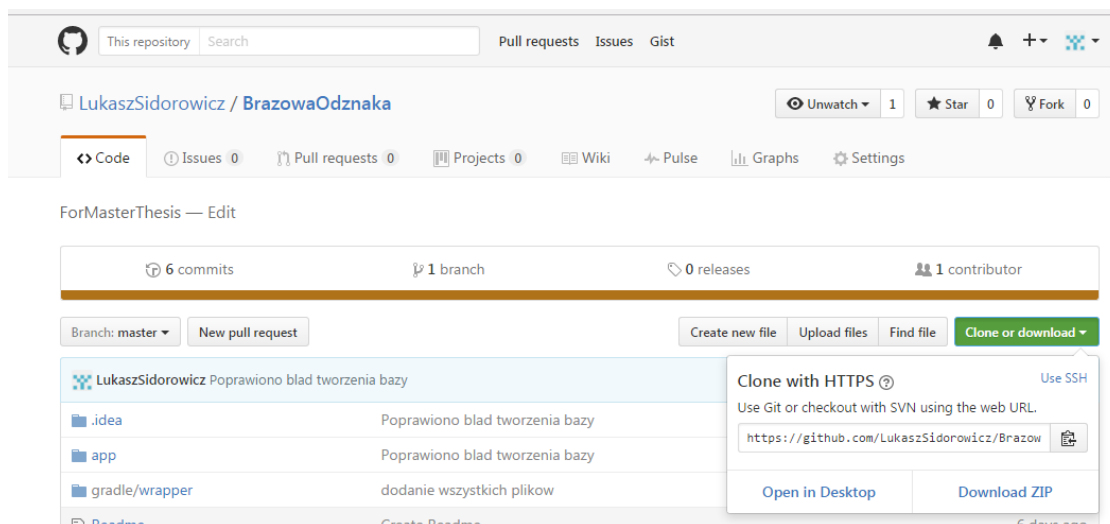


Aplikację można pobrać ze strony:

<https://github.com/LukaszSidorowicz/BrazowaOdznaka>

Aby zainstalować aplikację na telefonie należy pobrać i zainstalować Android Studio i odpowiednie SDK, a następnie wejść na podany adres repozytorium i kliknąć „Clone or Download”

Rysunek 40 Pobieranie projektu z GitHub



Następnie poprany projekt należy zaimportować do Android Studio, podłączyć telefon i zainstalować aplikację. Należy jeszcze pamiętać, aby w telefonie w ustawieniach programisty zaznaczyć opcję debugowania. W komputerach z systemem Windows mogą wystąpić problemy z wykrywaniem telefonu w środowisku programistycznym. Jeśli ktoś dysponuje komputerem z Linuxem, to zdecydowanie polecam korzystanie z tego systemu do instalacji aplikacji na telefonach i tabletach. Nigdy nie napotkałem tam problemów. Natomiast w komputerach z Windowsem należy upewnić się czy zainstalowane są sterowniki USB Googla, do pobrania za pomocą SDK menadżera w Android Studio. Czasami pomaga też zmiana trybu połączenia telefonu z komputerem np. na aparat.

4. Badania z użyciem aplikacji.

Dzięki temu, iż od ośmiu lat jestem instruktorem jazdy konnej, mam kontakt z liczną grupą jeźdźców zainteresowanych zdaniem egzaminu na odznakę. Z tej grupy wybrałem piętnaście osób, które podzieliłem na trzy grupy równe wiekowo i podobnym stanie wiedzy początkowej.

Grupy zostały tak dobrane, aby reprezentowały podobny wiek, jak również poziom wiedzy. Został przeprowadzony egzamin próbny, gdzie zostało wybrane po jednym pytaniu z każdego działu. Egzamin miał formę ustną. Poza egzaminem uczestnicy mieli do zaznaczenia kilka dodatkowych pytań, w celu lepszego zaszeregowania ich do grup. Bardzo istotne było, aby osoby, które dostaną aplikacje mobilne, miały już doświadczenie w pracy ze smartfonami, żeby nie uczyły się technologii. Formularz ankiety wstępnej wyglądał tak:

Ile lat jeździsz konno?

Ile razy w ciągu miesiąca jeździsz konno?

Czy korzystasz z treningów indywidualnych?

Czy i ile czasu dziennie korzystasz ze smartfona?

- 1) Dlaczego konie są płochliwe?
- 2) Jakie znaczenie ma regularne dbanie o czystość ściółki w boksie?
- 3) Co oznacza czerwona wstążeczka wpleciona w grzywę lub ogon konia?
- 4) W którym miejscu powinien znajdować się człowiek prowadzący konia?
- 5) Jakiego koloru grzywę i sierść ma koń gniady?
- 6) Wymień przynajmniej trzy znane Ci pasze objętościowe.
- 7) W którym miejscu występują zmiany związane z zapoprężeniem?
- 8) Wymień kilka (przynajmniej 3) znane Ci rodzaje siodeł.
- 9) Który element rzędu zakładamy najpierw koniowi, który stoi w boksie?
- 10) Co robimy z batem wsiadając na konia?
- 11) Kto może przebywać na ujeżdżalni podczas jazdy?
- 12) Co to są figury na ujeżdżalni?
- 13) Wymień naturalne pomoce jeździeckie.
- 14) Na którą nogę należy galopować jadąc po ujeżdżalni w lewo?

Jak się później okazało, wszyscy korzystają ze smartfonów i spędzają w moim odczuciu bardzo dużo czasu dziennie na jego używanie: odpowiedzi były od 2 do aż 6 godzin dziennie. Przedział wiekowy uczestników wahał się od 12 do 15 lat. Wszyscy regularnie jeżdżą konno co najmniej raz w tygodniu od 2 do 5 lat.

Wynik wstępny każdej grupy wynosił 28-29 punktów na 70 możliwych, chociaż w wielu przypadkach niewiele brakowało, abym mógł zaliczyć odpowiedź, jako prawidłową. Uczestnikami badań były wyłącznie przedstawicielki płci żeńskiej.

Zadaniem postawionym przed grupami było uczenie się przez siedem dni do egzaminu na odznakę przez dwie godziny dziennie, daje to czas po godzinę na dział. Jedna grupa otrzymała smartfony z zainstalowaną aplikacją mobilną, druga grupa otrzymała wydrukowane materiały z pytaniami i odpowiedziami podzielonymi na działy.

Materiały drukowane zostały przygotowane w oparciu o opublikowane pytania i odpowiedzi przez PZJ. Całość podzieliłem na odpowiednie działy zgodnie ze swoją wiedzą jeździecką. Ponadto podstawową zasadą dydaktyczną jest dzielenie materiału na mniejsze części. Dodatkowo podział na działy ułatwia uczącym się uzupełnienie luk tam, gdzie występują. Trzecia grupa otrzymała obie formy materiałów dydaktycznych.

Aplikacja została zainstalowana na telefonach poprzez podłączenie ich do komputera z systemem operacyjnym Ubuntu 16.04 z zainstalowanym Android Studio. Dla uczestników nieposiadających telefonów z systemem Android, urządzenia dzięki uprzejmości mojej rodziny i znajomych zostały udostępnione (na szczęście wiele osób trzyma stare telefony na zapas). Po upływie tygodnia uczestnicy ponownie podchodzili do egzaminu i wypełnili kilka pytań dodatkowych związanych z czasem poświęconym na naukę i użyteczności aplikacji. Ankieta i pytania wyglądały tym razem tak:

1. Czy poświęcałaś na naukę tyle czasu ile było założone (2 godziny dziennie)?
2. Tylko dla osób używających aplikacji mobilnej:
3. Co podobało Ci się w aplikacji?
4. Co nie podobało Ci się w aplikacji?
5. Czy uważasz, że aplikacja pomogła Ci się przygotować do egzaminu?
6. Co zmieniłabyś w aplikacji

- 1) Czy konie w naturalnych warunkach żyją samotnie czy w stadach?
- 2) Czy stajnia powinna być zimą ogrzewana?
- 3) Dlaczego należy uważać wchodząc do boksu?
- 4) Po której stronie należy stać przy nakładaniu kantara?
- 5) Co to są narowy koni?
- 6) Wymień kilka (przynajmniej 3) znane Ci pasze treściwe.
- 7) Co oznacza, że koniowi gniją strzałki?
- 8) Wymień kilka (co najmniej 4) znanych Ci rodzajów wędzideł.
- 9) Co należy sprawdzić przed nałożeniem siodła?
- 10) Czy siedząc na koniu można dociągać popręg?
- 11) Kto ma pierwszeństwo na ujeżdżalni?
- 12) Jakie wymiary ma duży czworobok?
- 13) Jakie są rodzaje dosiadu?
- 14) Jakie wyróżnia się rodzaje klusa?

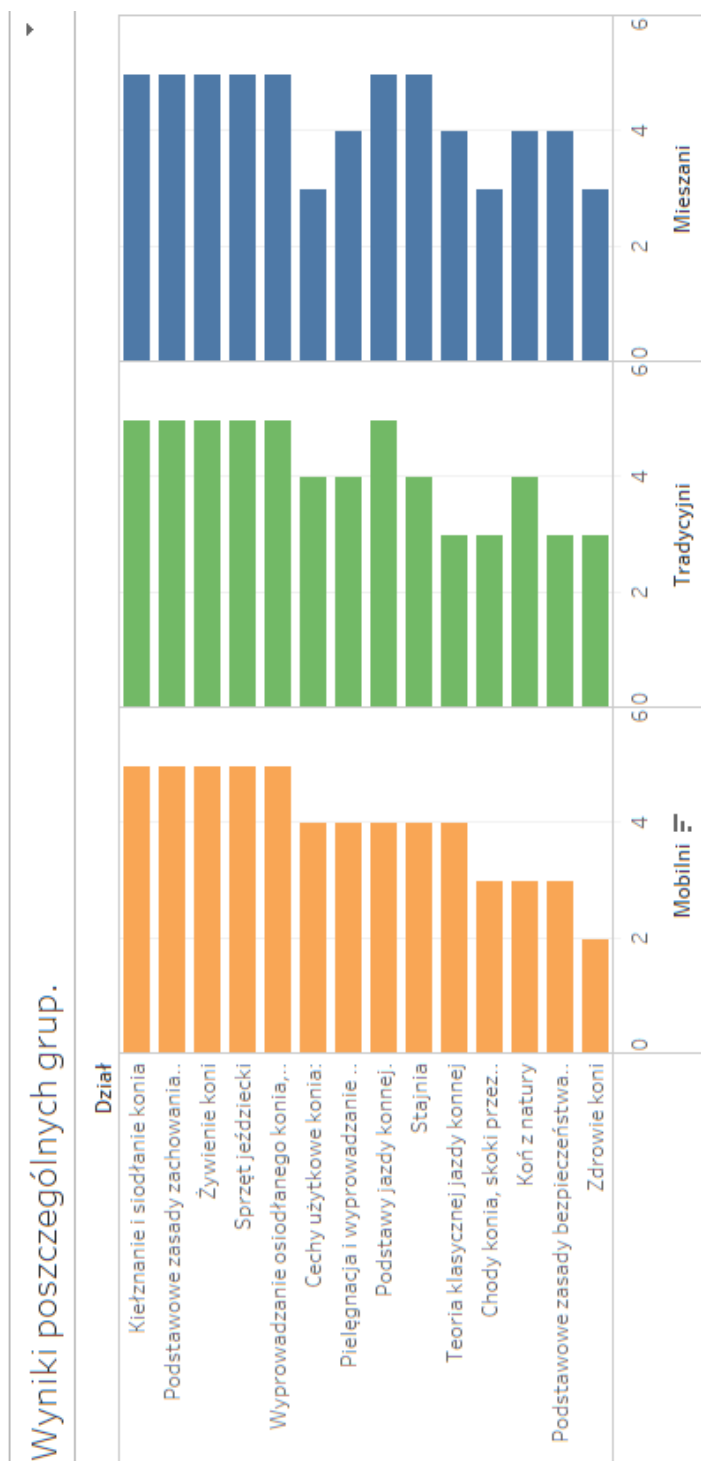
Z ankiety wynika, iż użytkownicy „Mieszani” poświęcali nieco więcej czasu, niż było założone- średnio 2.5 h, „mobilni” 1.8h, zaś „tradycyjni” 2,1h. W aplikacji podobało się to, iż łatwo i szybko można było otworzyć interesujący dział. Doceniono ponadto, iż można było w dowolnym miejscu i czasie otworzyć aplikację, bez konieczności pakowania jakichkolwiek książek czy kartek. Najgorzej w aplikacji została odebrana jej strona wizualna- brak obrazków, prostokątne przyciski, brzydkie kolory. Uczestnicy ankiet do aplikacji dodaliby chętnie ilustracje do wielu pytań, jak również chcieliby mieć możliwość rozpoczynać wyświetlanie od konkretnego pytania, a nie od początku.

Wyniki egzaminów zebrałem w formularzu programu Ms Excel 2010, jeżeli dana osoba w grupie odpowiedziała prawidłowo na pytania, wówczas dodawałem jeden punkt do grupy w dziale, w przypadku nieprawidłowej odpowiedzi wynik zostawał niezmieniony.

Dział	Mobilni	Tradycyjni	Mieszani
Koń z natury	3	4	4
Stajnia	4	4	5
Podstawowe zasady bezpieczeństwa przy pracy z koniem, wypadki i pierwsza pomoc	3	3	4
Pielęgnacja i wyprowadzanie koni.	4	4	4
Cechy użytkowe konia:	4	4	3
Żywienie koni	5	5	5
Zdrowie koni	2	3	3
Sprzęt jeździecki	5	5	5
Kielznanie i siodłanie konia	5	5	5
Wyprowadzanie osiodłanego konia, wsiadanie i zsiadanie z konia.	5	5	5
Podstawowe zasady zachowania na ujeżdżalni	5	5	5
Podstawy jazdy konnej.	4	5	5
Teoria klasycznej jazdy konnej	4	3	4
Chody konia, skoki przez przeszkody, jazda w terenie.	3	3	3

Następnie arkusz Excela podpiąłem do Tableau, aby wizualnie przedstawić wyniki. Na początku porównałem, jak ogólnie wypadły poszczególne działy dla wszystkich grup:

Rysunek 41 Wyniki egzaminu dla poszczególnych grup.

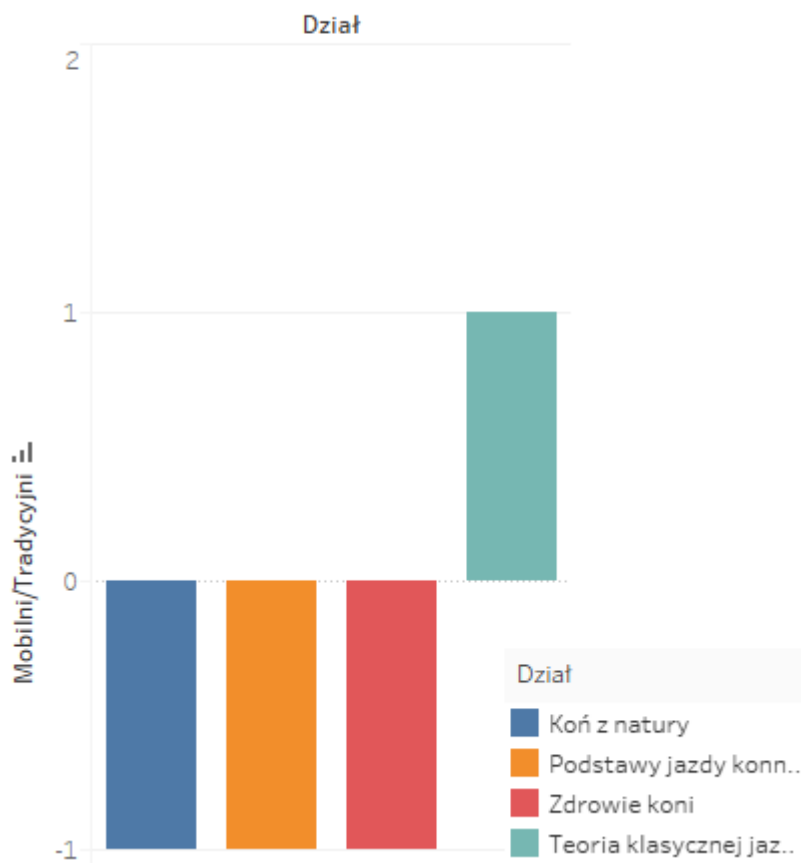


Z powyższego zestawienia wynika, iż w dalszej pracy z jeźdźcami powinienem więcej czasu poświęcić samemu koniowi, ponieważ najslabiej wypadły działy takie jak „Zdrowie koni”, „Koń z natury”, jak i bezpieczeństwu w stajni i podczas jazdy.

Następnie porównałem jak wypadła grupa mobilna w zestawieniu z tradycyjną. Utworzyłam dodatkową miarę poprzez odjęcie od wyników grupy mobilnej w poszczególnych działach, wyniki grupy tradycyjnej. Pokazuję tylko działy, gdzie wystąpiły różnice:

Rysunek 42 Porównanie wyników grupy mobilnej z tradycyjną.

Porównanie mobilnych z tradycyjnymi

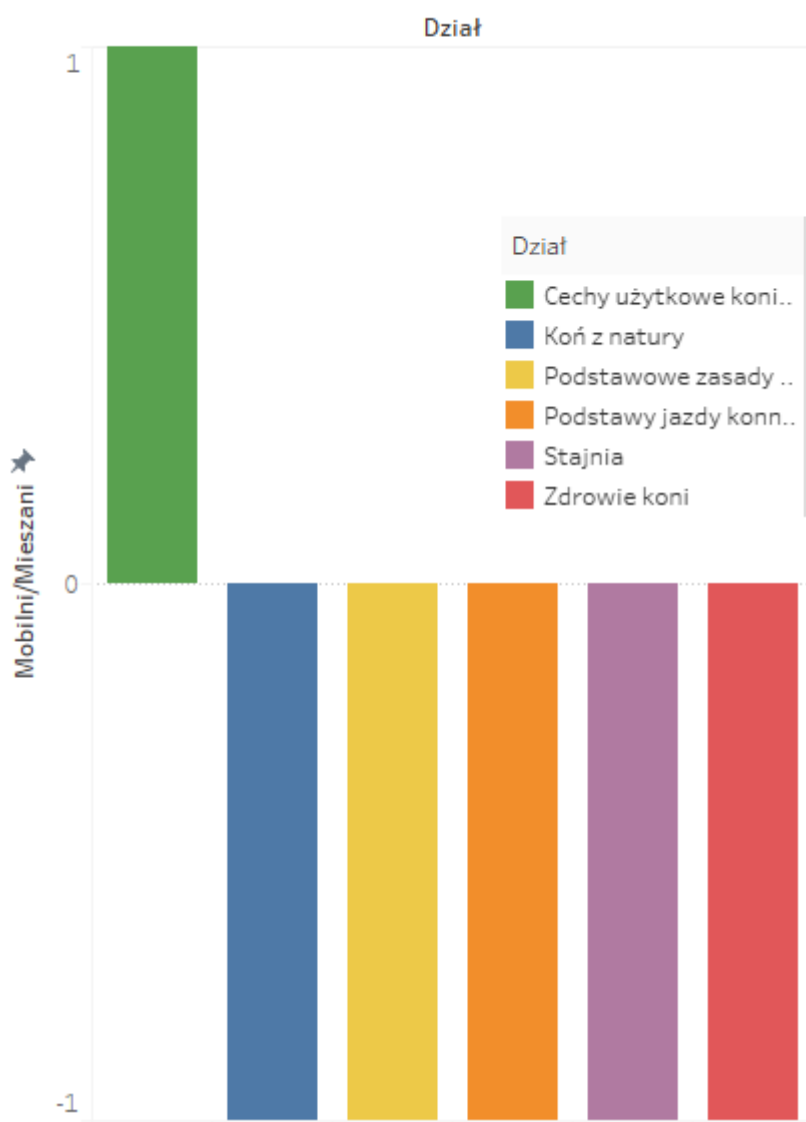


Wynika jednoznacznie z tego, iż lepsze rezultaty osiągnęły osoby korzystające z materiałów tradycyjnych. W aż trzech działach wypadły one lepiej niż mobilni. Z kolei grupa mobilna lepsze była w jednym dziale.

Kolejnym zestawieniem jest porównanie grupy mobilnej z mieszaną.

Rysunek 43 Porównanie grupy mobilnej z mieszaną.

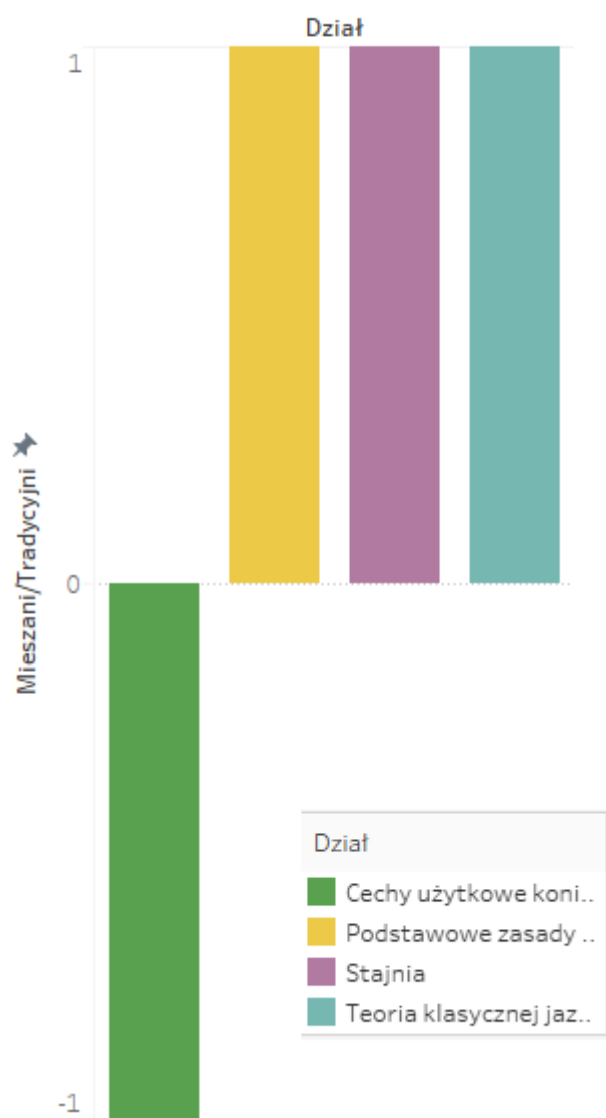
Porównanie mobilnych z mieszanymi



Tutaj różnica w wynikach okazała się już dosyć znaczna, grupa mobilna wypadła słabiej aż w pięciu działach, a była mocniejsza tylko w jednym. Moim kolejnym krokiem było zestawienie grupy mieszanej z tradycyjną.

Rysunek 44 Porównanie grupy mieszanej z tradycyjną.

Porównanie mieszanych z tradycyjnymi

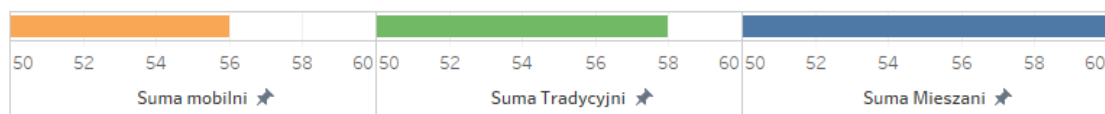


Z powyższego zestawienia, wynika, iż grupa mieszana była mocniejsza w trzech działach, zaś słabsza tylko w jednym. Wynika z tego, iż sama aplikacja mobilna jako środek dydaktyczny daje przeciętne rezultaty, jest natomiast dobrym uzupełnieniem dla klasycznych materiałów w formie skryptu.

Zestawiłem ponadto sumy punktów poszczególnych grup, ponadto utworzyłem nowe zestawienie biorąc pod uwagę wagę działów (w zależności od liczby pytań ustaliłem wagę na 1,2 lub 3), aby uzyskać takie zestawienie zmieniłem domyślną miarę z sumy na średnią:

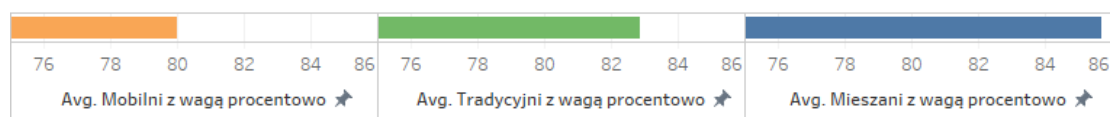
Rysunek 45 Suma punktów poszczególnych grup.

Suma punktów poszczególnych grup.



Rysunek 46 Wyniki procentowe z uwzględnieniem wagi działów.

Wyniki procentowe z uwzględnieniem wagi.



Uwzględnienie wagi działów, pokazuje całościowy obraz opanowania materiału. Różnica między grupą mobilną a mieszaną wyniosła 6%. Zaś uzupełnienie materiałów tradycyjnych o aplikację mobilną pozwoliło uzyskać o ok 3% lepszy rezultat. Z przeprowadzonych badań wynika, iż aplikację mobilną można traktować jako materiał uzupełniający i ułatwiający powtórki, nie zaś jako główne źródło wiedzy.

5. Podsumowanie:

Wykorzystując środowisko programistyczne Android zaprojektowałem i zaimplementowałem aplikację mobilną przygotowującą do brązowej odznaki jeździeckiej.

Aplikacja opiera się na bazie danych SQLite, dzięki temu można w prosty sposób ją rozszerzyć i przebudować. Bez większego wysiłku można aplikację rozszerzyć o odznakę srebrną i złotą, wystarczy dodać nowe dane do bazy i podpiąć je do programu korzystając z gotowych już klas.

Umieściłem aplikację w sieci, korzystając z bezpłatnego konta w serwisie GitHub:

<https://github.com/LukaszSidorowicz/BrazowaOdnaka>

Można ją pobrać i zainstalować na swoim smartfonie, pod warunkiem iż posiada system Android na poziomie minimum API 15 (Ice Cream Sandwich 4.04) i wyższym.

Aplikacja została przeze mnie użyta do badań na grupie 15 osobowej. Wyniki badań pokazują, iż może być doskonałym materiałem dydaktycznym uzupełniającym tradycyjne materiały książkowe i skryptowe.

Jako samodzielne źródło wiedzy niestety okazuje się nieznacznie gorsza, niż wydruk takiego samego materiału w formie skryptu.

Z kolei połączenie skryptu z aplikacją mobilną spowodowało polepszenie wyników wśród zdających.

Dzięki zastosowaniu wizualizacji przy użyciu programu Tableau uzyskałem wskazówki, na co powinienem poświęcić więcej uwagi w nauczaniu jazdy konnej swoich uczniów.

6. Wykaz użytych skrótów:

API- Application Programming Interface, interfejs programistyczny aplikacji
AVD- Android Virtual Device- wirtualne urządzenie Android
BI- Business Intelligence
BO- Business Objects
CMC- Central Management Console- centralna konsola administracyjna
CMD- Command – konsola systemu Windows
CPU- Central Processing Unit- processor
CSV- Comma Separated Value, format pliku, wartości oddzielone przecinkami
DB- DataBase- baza danych
IDE- Integrated Development Environment- zintegrowane środowisko programistyczne
iOS – iPhone Operating System- system operacyjny iPhone
JDBC- Java DataBase Connectivity- Łączy do baz danych języka Java,
PDF- Portable Document Format, format dokumentu przenośnego
PZJ- Polski Związek Jeździecki
RAM- Random Access Memory- pamięć o dostępie swobodnym
SAP- Systemanalyse und Programmentwicklung, firma wytwarzająca oprogramowanie
SDK- Software Development Kit, zestaw narzędzi dla programistów
SQL- Structured Query Language, strukturalny język zapytań
SUSE- Software- und System-Entwicklungsgesellschaft, dystrybucja Linux
XML- Extensible Markup Language, Rozszerzalny Język Znaczników

7. Literatura:

-
- ¹ www.pzj.pl
- ² Equista.pl
- ³ Pruchniewicz, Waław; Chaber PR- Akademia Jeździecka; 2003
- ⁴ Praca zbiorowa; Akademia Jeździecka; 2008
- ⁵ <http://digitalfractal.com/2016/05/mobile-operating-systems/>
- ⁶ <http://www.ceneo.pl/43008127>
- ⁷ <http://www.apple.com/pl/shop/buy-iphone/iphone-7>
- ⁸ <http://indianexpress.com/article/technology/tech-news-technology/indus-os-the-indian-operating-system-with-more-installs-that-ios-2876725/>
- ⁹ <http://www.businessinsider.com/apple-ios-v-android-market-share-2016-1?IR=T>
- ¹⁰ <https://developer.android.com/studio/index.html>
- ¹¹ <https://developer.android.com/distribute/googleplay/start.html>
- ¹² <https://developer.apple.com/support/compare-memberships/>
- ¹³ <https://git-scm.com/book/en/v2/Getting-Started-About-Version-Control>
- ¹⁴ <https://www.gnu.org/software/rcs/>
- ¹⁵ <https://subversion.apache.org/>
- ¹⁶ <http://www.nongnu.org/cvs/>
- ¹⁷ <https://git-scm.com/>
- ¹⁸ <https://www.mercurial-scm.org/>
- ¹⁹ <http://bazaar.canonical.com/en/>
- ²⁰ <https://github.com/>
- ²¹ <https://bitbucket.org/>
- ²² <https://bitbucket.org/account/user//plans/>
- ²³ <https://github.com/pricing>
- ²⁴ Howson, C.; Newbould, E.; SAP Business Objects BI 4.0: The Complete Reference, Third Edition; The McGraw-Hill Publishing; 2013
- ²⁵ <https://www.gartner.com>
- ²⁶ <https://www.gartner.com/doc/reprints?id=1-2XXET8P&ct=160204>
- ²⁷ <http://busitelce.com/bi-analytic-tools-trends/20-gainers-losers-gartner-bi-analytics-2014-vs-2015>
- ²⁸ <https://www.coursera.org/learn/analytics-tableau>
- ²⁹ <http://go.sap.com/product/analytics/bi-platform.html>

-
- ³⁰ <http://www.qlik.com/us/>
- ³¹ <https://www.suse.com/products/sles-for-sap>
- ³² <https://developer.android.com/studio/index.html>
- ³³ <http://www.oracle.com/technetwork/java/javase/downloads/index-jsp-138363.html>
- ³⁴ <https://notepad-plus-plus.org/>
- ³⁵ <https://www.sqlite.org/>
- ³⁶ Owens, Mike; *The Definitive Guide to SQLite*; APRESS; 2006
- ³⁷ Grant, Allen; *Beginning Android*; APRESS; 2015
- ³⁸ <http://sqlitebrowser.org/>
- ³⁹ Allen,G.; Owens,M.; *The Definitive Guide to SQLite*, second edition, Apress, 2010
- ⁴⁰ MacLean,D.; Komatineni, S.; Allen, G.; *Pro Android 5*, Apress 2015