



**Instytut Informatyki
Kolegium Nauk Przyrodniczych
Uniwersytet Rzeszowski**

Przedmiot:

Programowanie urządzeń mobilnych

Dokumentacja techniczna projektu:

***GradeCam – aplikacja do oceniania testów
wyboru za pomocą aparatu smartfona***

Wykonał: Patryk Gwizdała, Mateusz Krajnik, Konrad Kubrak

Prowadzący: mgr inż. Adam Szczur

Rzeszów 2020

SPIS TREŚCI

1. Temat projektu	3
2. Cel projektu	3
2.1. Cele szczegółowe	3
3. Funkcjonalności aplikacji	3
4. Technologie	3
5. Harmonogram realizacji projektu	3
5.1. Wykres Gantt'a	4
6. Projekt GUI	5
7. Struktura programu	9
7.1. Dane wykorzystywane przez program	9
7.2. Opis plików zewnętrznych	9
7.3. Podział na moduły, komunikacja między modułami	9
8. Diagramy UML	10
8.1. Diagram przypadków użycia	10
8.2. Diagram czynności / aktywności	11
9. Literatura	12

1. Temat projektu

Tematem projektu było wykonanie aplikacji do oceniania testów jednokrotnego wyboru za pomocą aparatu wbudowanego w telefon komórkowy typu smartfon.

2. Cel projektu

Celem projektu było napisanie aplikacji mobilnej dla systemu operacyjnego Android (6.0+), która przy wykorzystaniu odpowiednich bibliotek będzie obrabiała wykonane przez użytkownika zdjęcie oraz rozpoznawała odpowiednio zakodowane informacje, a następnie prezentowała wynik operacji.

2.1. Cele szczegółowe

- Stworzenie systemu do skanowania, który:
 - Będzie powtarzalny,
 - Będzie uniwersalny dla wielu modeli telefonów,
 - Będzie otwarty na przyszłe aktualizacje,
- Poznanie dostępnych narzędzi do analizy wizualnej,
- Zapoznanie się z pracą przy projektach typowo mobilnych
 - Zrozumienie cyklu życia aktywności
 - Poznanie struktury projektu aplikacji mobilnej

3. Funkcjonalności aplikacji

- Operacje CRUD dla ocenianych testów – można dodać nowy test, edytować nazwę oraz klucz a także usunąć test
- Wybieranie aktywnego testu
- Skanowanie arkusza odpowiedzi za pomocą aparatu; następnie algorytm wykonuje odpowiednie operacje (poprawa perspektywy, rozpoznanie odpowiedzi) i zostaje wyświetlony rezultat

4. Technologie

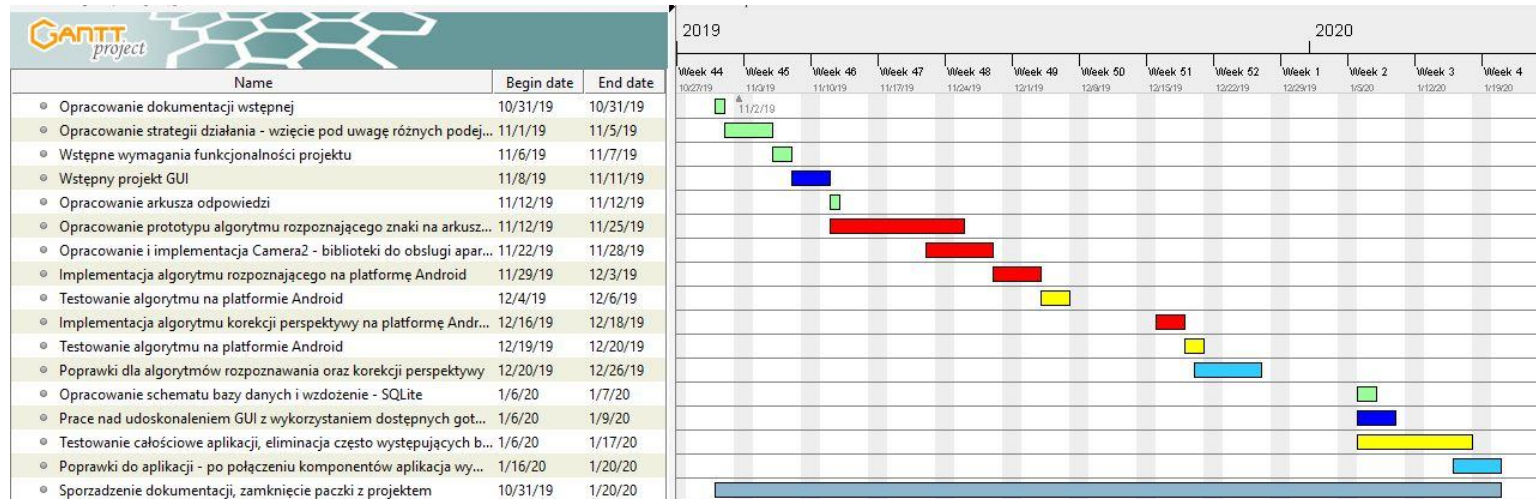
Aplikacja jako rdzeń wykorzystuje bibliotekę do wizualnej analizy obrazu – OpenCV. Jest to biblioteka udostępniona na zasadach open source. Oprócz tego do stworzenia aplikacji został wykorzystany język Java oraz środowisko Android Studio.

5. Harmonogram realizacji projektu

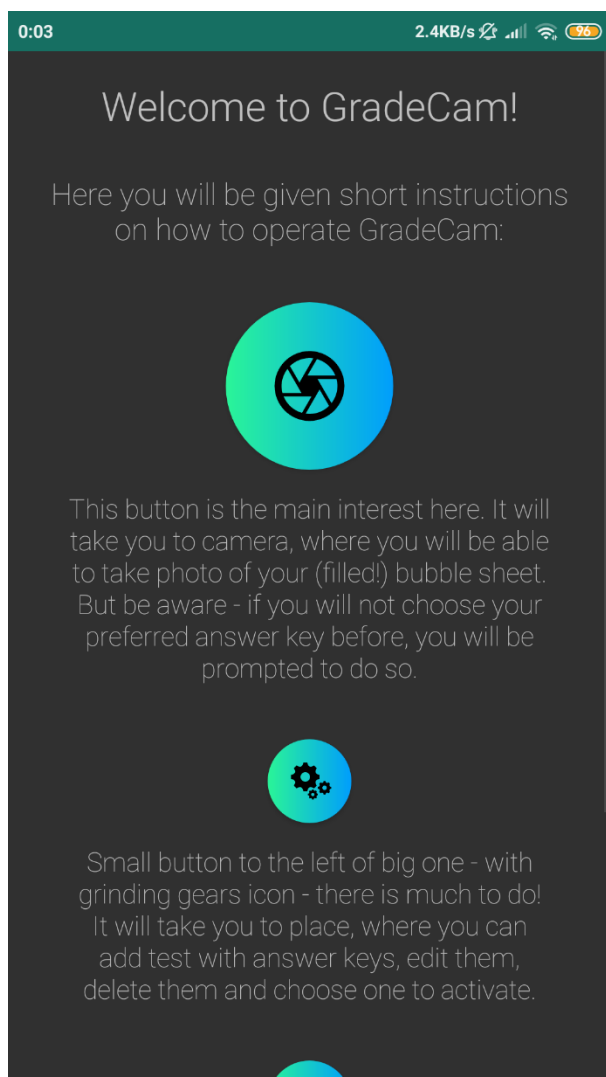
- Opracowanie strategii działania – wzięcie pod uwagę różnych podejść oraz dostępnych bibliotek
- Stworzenie wstępnych wymagań co do funkcjonalności projektu
- Opracowanie arkusza odpowiedzi, eksperymenty z różnymi fontami/formatami arkusza
- Opracowanie prototypu algorytmu rozpoznającego znaki na arkuszu odpowiedzi
- Prace nad implementacją Camera2 – biblioteki obsługi aparatów dla smartfonów

- Implementacja algorytmu na platformę Android
- Opracowanie algorytmu korekcji perspektywy
- Implementacja algorytmu na platformę Android
- Prace nad implementacją bazy danych oraz operacjami CRUD dla testów
- Prace nad udoskonaleniem interfejsu graficznego (motywy, design, kolorystyka)
- Wprowadzenie obsługi podstawowych operacji niepożądanych (nie wybranie klucza i przejście do skanowania skutkuje alertem o konieczności wyboru klucza)

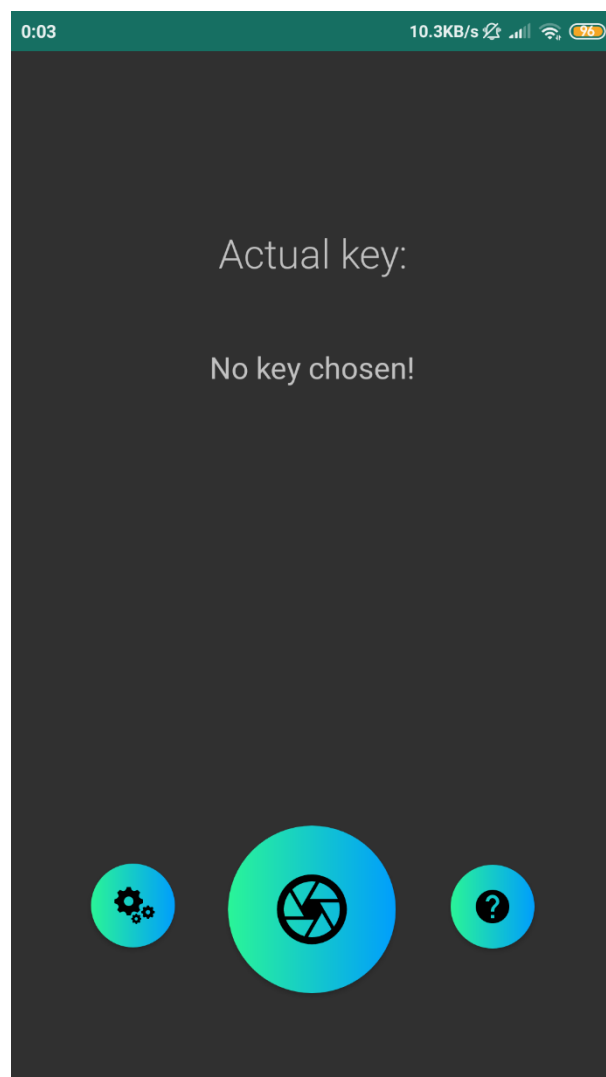
5.1. Wykres Gantta



6. Projekt GUI

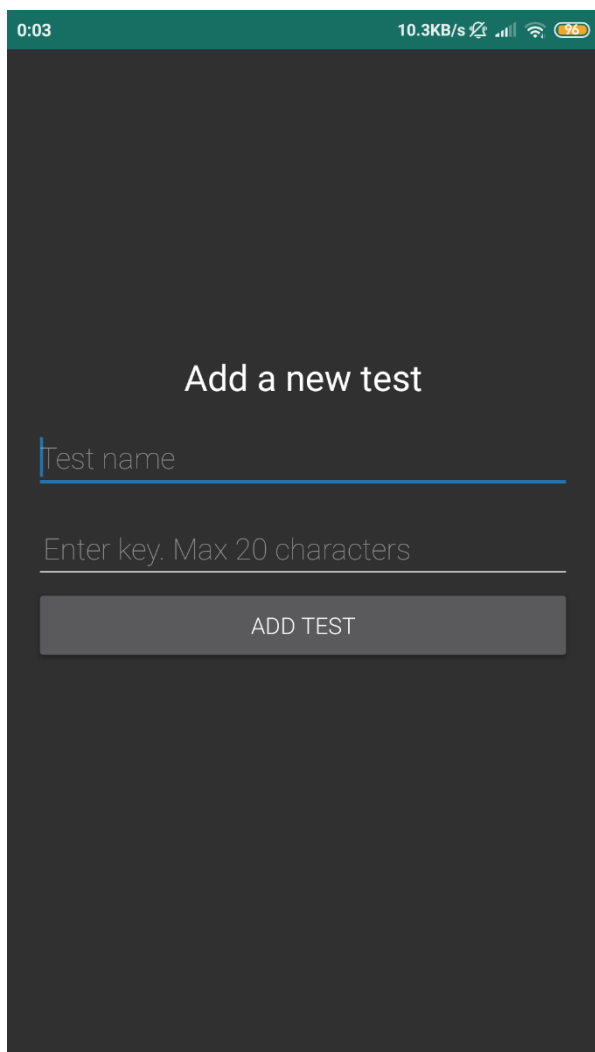


Rysunek 1 – ekran widoczny przy pierwszym uruchomieniu aplikacji na danym urządzeniu – może być również przywołany z ekranu głównej aktywności – jest to instrukcja obsługi.

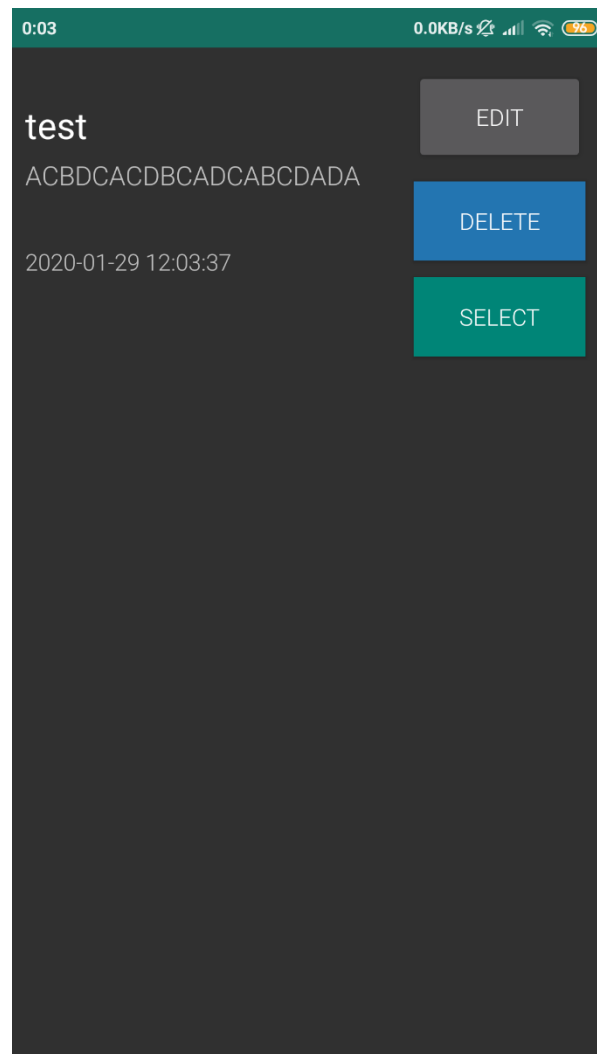


Rysunek 2 – ekran główny. Zawiera przyciski przenoszące do odpowiednich widoków. Treść tekstową stanowią informacje o aktualnie wybranym kluczu/braku wybranego klucza.

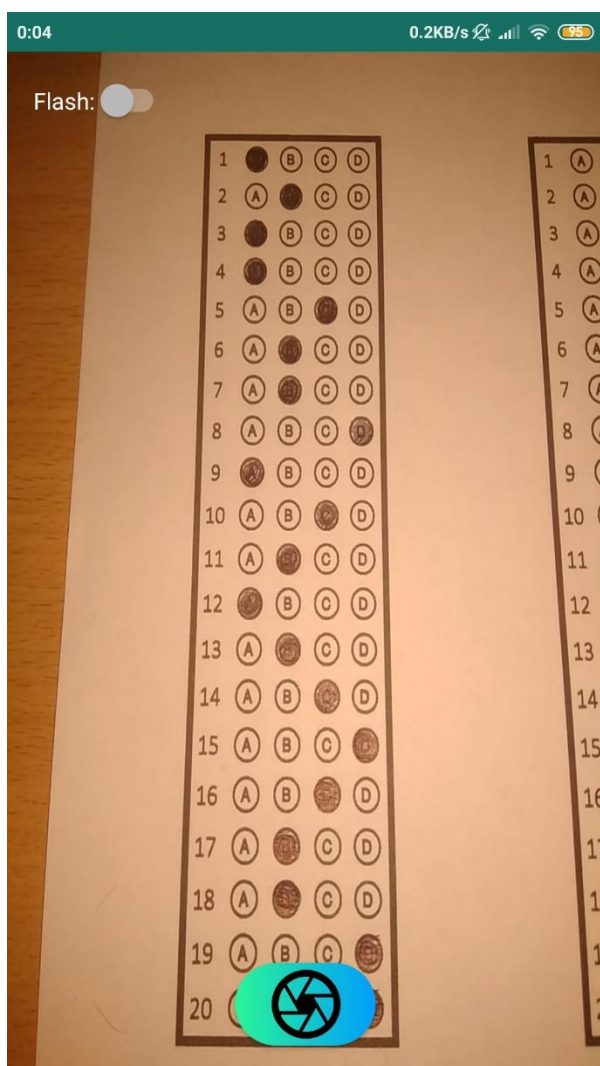
Elementy interfejsu realizujące operacje CRUD dla testów:



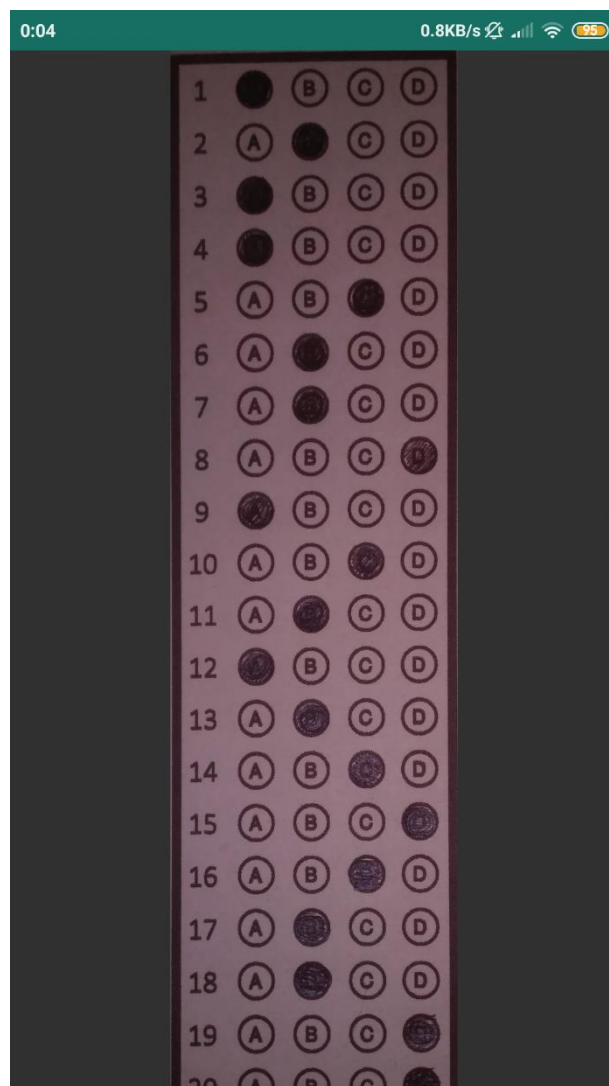
Rysunek 4 – ekran dodania nowego testu. Aby przejść do listy testów trzeba najpierw zainicjalizować tę listę dodaniem testu.



Rysunek 3 – ekran listy testów. Zawiera informacje o nazwie i kluczu odpowiedzi z nim związanym. Zawiera akcje edycji, usunięcia i aktywowania.



Rysunek 6 – ekran modułu Kamery. Zawiera podgląd na żywo skanowanego obrazu.



Rysunek 5 – zeskanowany obraz o skorygowanej perspektywie będzie prezentował się bardzo podobnie do powyższego.

0:04				
0.0KB/s				
Answers recognized:				
ABAACBBDDCBABCD CBDD				
Answers from key:				
ACBDCACDBCAD CABCDADA				
Points scored:				
6				
#	Answer	Key		
1	A	A	+	
2	B	C		
3	A	B		
4	A	D		
5	C	C	+	
6	B	A		
7	B	C		
8	D	D	+	
9	D	B		
10	C	C	+	
11	B	A		
12	A	D		
13	B	C		
14	C	A		
15	D	B		

Rysunek 7 – po przewinięciu niżej, pod zeskanowanym zdjęciem znajduje się podsumowanie – zeskanowane odpowiedzi, klucz dla porównania oraz zdobyta na podstawie porównania liczba punktów. Dostępna jest także tabela z porównaniem dla każdego pytania z osobna.

7. Struktura programu

Aplikacja składa się z następujących aktywności:

- MainActivity – główna aktywność, która umożliwia przejście do konfiguracji testów (CRUD), skanowania testu oraz do ekranu pomocy.
- KeyActivity – aktywność pozwalająca na dodanie testu
- ExamsActivity – aktywność realizująca operację Read-Update-Delete dla testów
- CameraActivity – aktywność udostępniająca za pośrednictwem Camera2 API dostęp do kamery a tym samym skanowanie Arkuszy testowych.
- TransitionActivity – aktywność przejściowa, odpowiada ona za realizację algorytmów korekcji oraz rozpoznawania
- ViewActivity – aktywność wyświetlająca rezultat prac aplikacji

Dodatkowo wykorzystane zostały klasy pomocnicze, realizujące połączenie z bazą danych, wypełniające listę testów, klasy zawierające algorytmy.

7.1. Dane wykorzystywane przez program

Aplikacja na wejściu potrzebuje poprawnie wykonanego, dobrze doświetlonego zdjęcia arkusza odpowiedzi z zaznaczonymi wszystkimi odpowiedziami. Uprawnienia jakich potrzebuje to dostęp do systemu plików (aby móc zapisać zrobione zdjęcie) oraz zezwolenie na wykorzystanie hardware'u w postaci kamery

7.2. Opis plików zewnętrznych

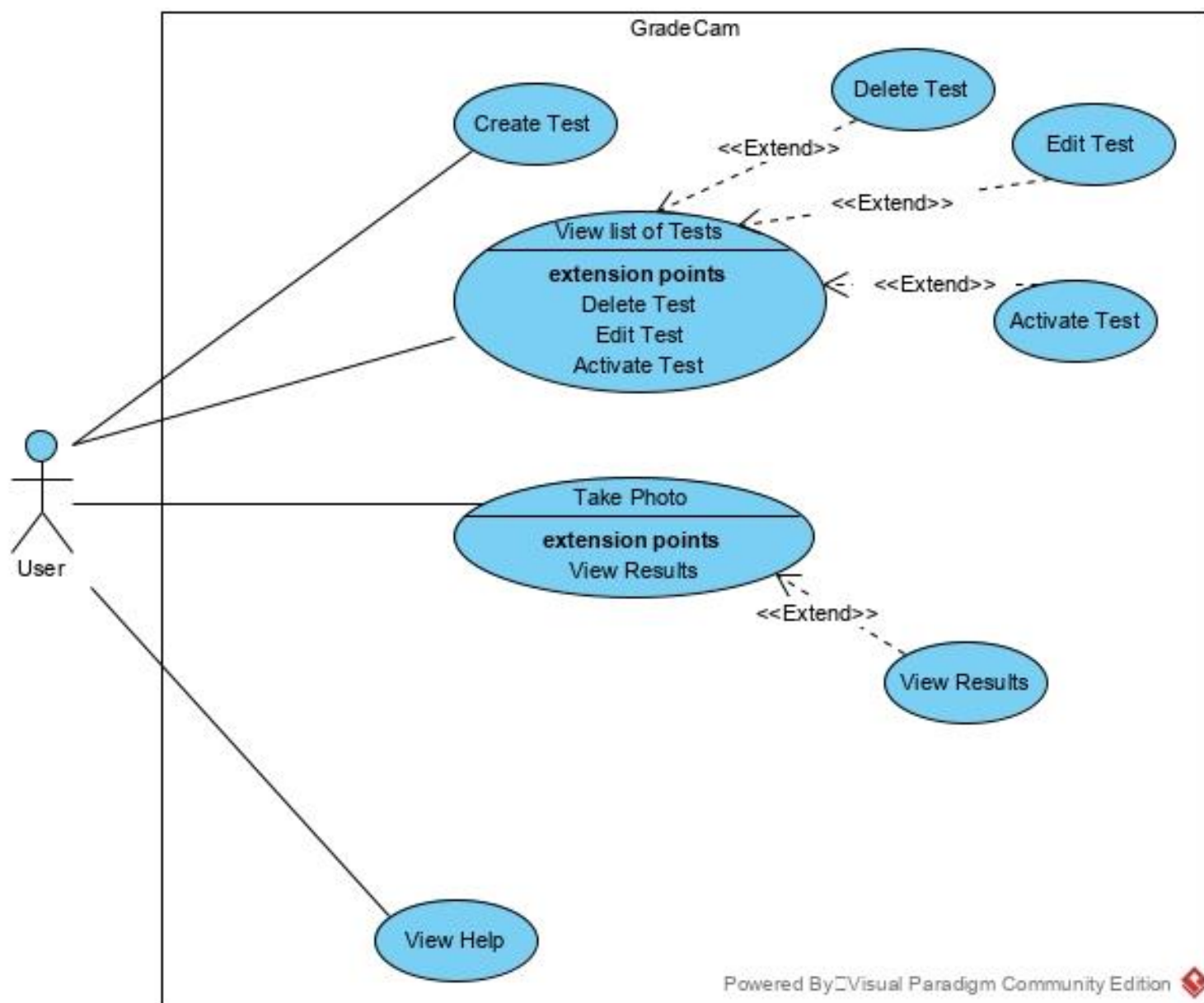
Aplikacja oprócz zapisywanego w pamięci (i nadpisywanego za każdym razem) zdjęcia zrobionego za pomocą aparatu nie korzysta z żadnych plików wewnętrznych.

7.3. Podział na moduły, komunikacja między modułami

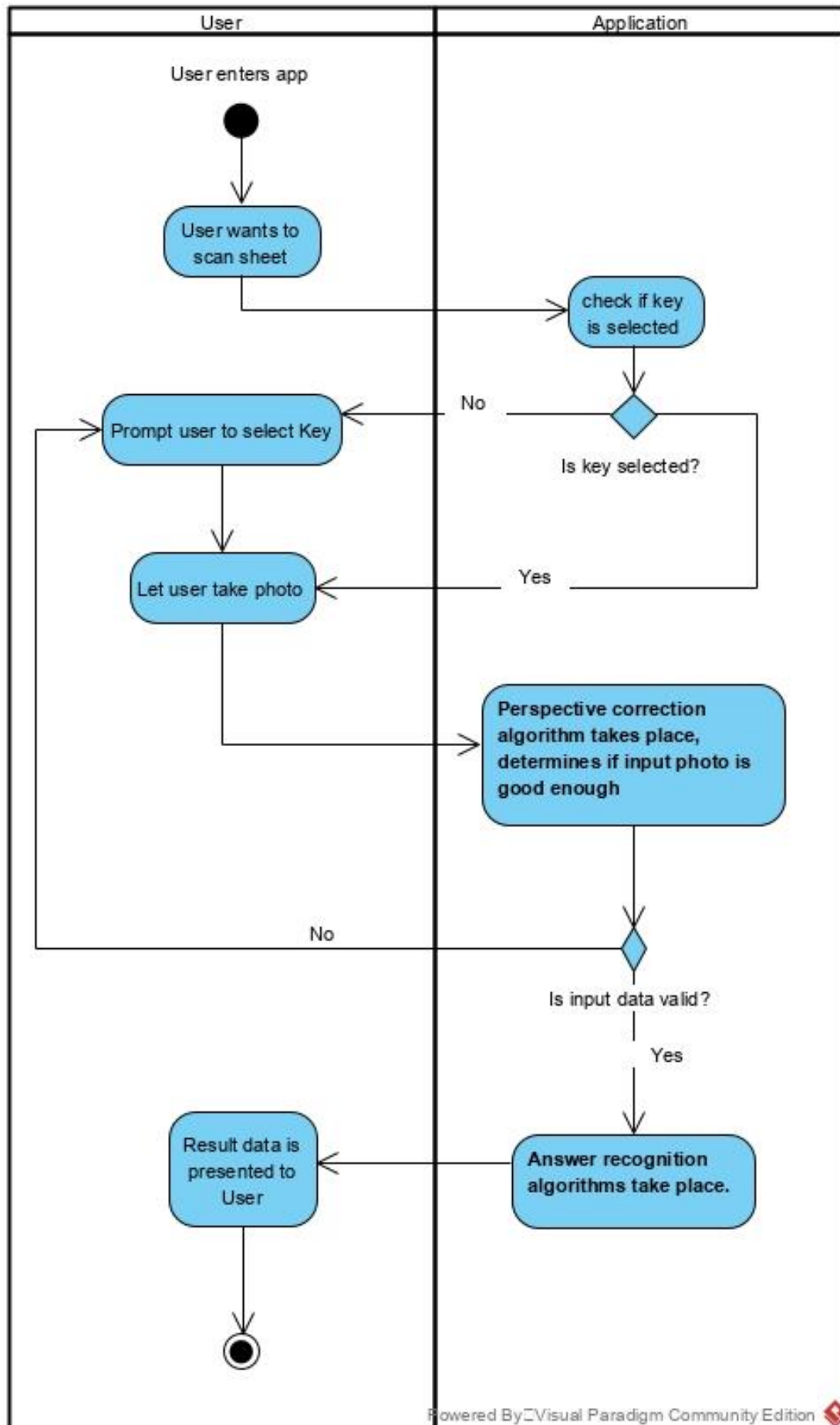
W aplikacji jedyny podział na moduły o jaki można by się pokusić to podział na API kamery oraz „resztę” aplikacji (algorytmy, interfejs). Aplikacja za pomocą algorytmów obrabia wykonane za pomocą API kamery zdjęcie oraz wykorzystując GUI prezentuje wyniki.

8. Diagramy UML

8.1. Diagram przypadków użycia



8.2. Diagram czynności / aktywności



9. Literatura

- Oficjalna dokumentacja OpenCV - <https://docs.opencv.org/>
- Dokumentacja Android - <https://developer.android.com/docs>
- StackOverflow
- Youtube