



**POLITECHNIKA ŚLĄSKA**  
**WYDZIAŁ AUTOMATYKI, ELEKTRONIKI I INFORMATYKI**  
**KIERUNEK INFORMATYKA**

**Projekt inżynierski**

*Opracowanie aplikacji mobilnej wspomagającej prowadzenie  
turnieju szachowego*

Autor: Adam Kincel

Kierujący pracą: dr inż. Krzysztof Dobosz

Gliwice, Styczeń 2020



## Oświadczenie

Wyrażam zgodę/nie wyrażam\* zgody na udostępnienie mojej pracy  
dyplomowej/rozprawy doktorskiej\*

....., dnia .....

.....  
(podpis)

.....  
(poświadczenie wiarygodności podpisu przez Dziekanat)

\* właściwe podkreślić



## **Oświadczenie promotora**

Oświadczam, że praca „Opracowanie aplikacji mobilnej wspomagającej prowadzenie turnieju szachowego” spełnia wymagania formalne pracy dyplomowej inżynierskiej.

Gliwice, dnia .....

.....  
(podpis)



# Spis treści

1.	Wstęp .....	1
1.1.	Wprowadzenie .....	1
1.2.	Cel, zakres i kontekst pracy .....	3
2.	Analiza tematu .....	5
2.1.	Zasady gry w szachy .....	5
2.2.	Systemy rozgrywek w turniejach .....	6
2.3.	Przegląd istniejących rozwiązań .....	10
2.4.	Ustalanie pozycji na liście wyników .....	11
3.	Wymagania i narzędzia .....	13
3.1.	Wymagania funkcjonalne .....	13
3.2.	Wymagania niefunkcjonalne .....	16
3.3.	Opis narzędzi .....	16
4.	Specyfikacja zewnętrzna .....	18
5.	Specyfikacja wewnętrzna .....	27
5.1.	Architektura systemu .....	27
5.2.	Przegląd klas .....	28
5.2.1.	Aktywności .....	28
5.2.2.	Baza danych .....	29
5.2.3.	Mecze .....	30
5.2.4.	Model .....	30
5.2.5.	Klasy pomocnicze .....	30
5.2.6.	System szwajcarski .....	31
6.	Weryfikacja i walidacja .....	33
7.	Podsumowanie i wnioski .....	37
7.1.	Wynik pracy .....	37
7.2.	Możliwości rozbudowy aplikacji .....	38
7.3.	Problemy napotkane w trakcie pracy .....	39
	Bibliografia .....	i
	Spis skrótów i symboli .....	iii
	Zawartość dołączonej płyty .....	iv
	Spis rysunków .....	v





# 1. Wstęp

## 1.1. Wprowadzenie

Szachy są jedną z najstarszych gier planszowych, w związku z tym, że istnieje bardzo duża możliwość scenariuszy czy strategii rozgrywania partii nazywana jest „grą królewską”. Historia szachów sięga aż do lat siedemdziesiątych VI wieku naszej ery [7]. Wtedy, według źródeł pisanych, indyjski radża podarował szachy ówczesnemu władcy Persji Chosrowi I Anoszirwanowi [1]. Warto zaznaczyć, że już wtedy zasady gry były ściśle określone. Po opanowaniu Persji przez Arabów oraz pojawieniu się rozrywki w Europie nastąpiły kolejne ewolucje szachów, które przypominają współczesną grę (wprowadzono pojęcia takie jak: roszada, mat czy promocja). W epoce Odrodzenia szachy zyskały na dużej popularności. Głównie w Hiszpanii oraz Włoszech na przełomie XVI i XVII wieku nastąpił tak zwany „złoty okres” szachów. Powstawały pierwsze podręczniki do nauki oraz analizy gry, zaczęto rozgrywać pierwsze mecze, oraz organizować turnieje, które wzbudzały coraz większe zainteresowanie zarówno wyższych oraz niższych warstw społecznych. Szachy trafiały do większej ilości osób oraz rosły na popularności.

W roku 1924 powstała Międzynarodowa Federacja Szachowa (FIDE), od tego roku rozgrywane są szachowe olimpiady. W 1989 roku doszło do pierwszego pojedynku szachowego pomiędzy komputerem, a człowiekiem [4]. Stworzony przez firmę IBM komputer Deep Thought rywalizował Garrim

Kasparowem, wielokrotnym mistrzem świata, uznawanym przez wielu ekspertów za największą legendę szachów. Gładko zwyciężył Kasparov, jednakże naukowcy z IBM nie ustępowali z celu pokonania legendy i po ulepszeniu maszyny, w 1996 roku doszło do rewanżu. Tym razem Deep Blue (zmodyfikowany komputer) podjął rękawicę i uległ wtedy najmłodszemu mistrzowi tylko 4-2, gdzie o wyniku ostatecznym decydowała ostatnia partia, którą na swoją korzyść rozstrzygnął Kasparov. Kolejny pojedynek odbył się w maju 1997 roku. Kolejne poprawki oraz modyfikacje przyczyniły się do powstania nowej wersji maszyny o nazwie Deeper Blue. Przebieg pojedynku był podobny do poprzedniego jednakże decydującą partię rozstrzygnął na swoją korzyść komputer i w ten sposób stał się pierwszym systemem komputerowym, który wygrał z aktualnym mistrzem świata.

Tak jak wspomniano, już na przełomie XVI oraz XVII wieku zaczęto organizować pierwsze, lokalne turnieje szachowe. Turnieje te nie cieszyły się jednak zbyt wielką popularnością w związku z dopiero rozwijającą się grą jakim były szachy. Pierwsze rozgrywki odbyły się w Madrycie w roku 1575 na dworze króla Hiszpanii Filipa II. Wraz z upływem czasu gra stawiała się coraz bardziej popularna. Organizowano kolejne pojedynki szachowe, które cieszyły się sporym zainteresowaniem. Gospodarzem pierwszego międzynarodowego turnieju szachowego był Londyn. Wydarzenie odbyło się w 1851 roku, a zwycięzcą został niemiecki szachista, urodzony we Wrocławiu, Adolf Anderssen.

W dzisiejszych czasach organizowanych jest mnóstwo turniejów szachowych. Turnieje te mają różne rangi. Międzynarodowe, najbardziej prestiżowe są organizowane przez FIDE [3]. Zawody ogólnopolskie pod egidą Polskiego Związku Szachowego [5]. Istnieją również mniejsze, regionalne turnieje, których organizatorem są wojewódzkie związki szachowe (np. Śląski Związek Szachowy [10]). Do zawodów o najniższej randze należą te, które są organizowane przez lokalną społeczność np. w szkole czy wewnątrz firmy. Wymienione imprezy mają miejsce praktycznie w każdy weekend przez cały rok, ich liczba jest na dobrą sprawę niepoliczalna. Dlatego postanowiono napisać program, który zadba o prawidłowy przebieg turnieju szachowego.

Aplikacja zostanie skierowana przede wszystkim do użytkowników, którzy chcą przeprowadzać turnieje szachowe na szczeblu głównie amatorskim, nie mających funduszy do kupna drogich i bardziej rozwiniętych programów.

## 1.2. Cel, zakres i kontekst pracy

Celem napisania aplikacji mobilnej wspomagającej prowadzenie turnieju szachowego jest wyeliminowanie błędów wynikających z ręcznego kojarzenia zawodników oraz zaoszczędzenie sporej ilości czasu, który organizatorzy mogą spożytkować na inne niezbędne czynności podczas trwania turnieju. W związku z tym, że aplikacja dedykowana jest dla turniejów lokalnych nie jest konieczna komunikacja ze światem zewnętrznym. Program powinien kojarzyć zawodników systemem szwajcarskim.

W kolejnym rozdziale przedstawiona zostanie analiza tematu. Nastąpi wprowadzenie do dziedziny oraz tematyki dotyczącej przeprowadzenia turniejów szachowych. Zostanie porównana specyfika przeprowadzania zawodów z innymi sportami. Następnie omówiony zostanie problem, przed którym postawiono autora. Dokonany zostanie opis przeglądu literatury, który miał miejsce przed oraz w trakcie pisania programu. Przeprowadzona zostanie analiza dotycząca opisu znanych rozwiązań całych aplikacji lub poszczególnych algorytmów, które tą aplikację tworzą. W rozdziale trzecim (wymagania i narzędzia) zostaną przedstawione wymagania funkcjonalne, niefunkcjonalne, przypadki użycia, metodyka pracy nad projektowaniem i implementacją oraz opis narzędzi. W kolejnym rozdziale zostanie opisana specyfikacja zewnętrzna. Zostaną poruszane punkty takie jak: wymagania sprzętowe oraz programowe, sposób obsługi, kategorie użytkowników, kwestie bezpieczeństwa programu. Zaprezentowany zostanie przykład działania programu poparty licznymi zrzutami ekranu ilustrujące wszystkie funkcje oraz scenariusze korzystania z systemu. W rozdziale „specyfikacja wewnętrzna”

przedstawiona zostanie architektura systemu, ważniejsze struktury danych, przegląd klas i algorytmów. Dodatkowo będą pokazane wybrane fragmenty implementacji oraz opisane wzorce projektowe. W tym rozdziale zostanie pokazana struktura bazy danych wraz z opisem tabeli, która się w niej znajduje. Następnie zostanie przedstawiona weryfikacja i walidacja systemu. Opisany będzie sposób testowania aplikacji: przypadki testowe, zakres testowania wzięty pod uwagę oraz wyniki testowania. Wskazane zostaną wykryte i usunięte błędy. W ostatnim rozdziale zostanie przedstawione podsumowanie i wnioski płynące z napisanej pracy. Największe napotkane problemy podczas tworzenia oprogramowania, porównanie postawionych celów przed napisaniem programu, z osiągniętymi po napisaniu aplikacji. Następnie wskazana zostanie dalsza możliwość rozwoju powstałego oprogramowania i ewentualnie stopień ulepszenia oraz optymalizacji otrzymanej aplikacji.

## **2. Analiza tematu**

### **2.1. Zasady gry w szachy**

Szachy to dwuosobowa, strategiczna gra planszowa. Rozgrywa się ją na 64- polowej, składającej się z 8 poziomych rzędów oraz 8 pionowych kolumn planszy zwanej szachownicą [8]. Każdemu poziomemu rzędowi przyporządkowane są cyfry od jeden do osiem, natomiast pionowemu litery od a do h. Dzięki takiemu ponumerowaniu można w łatwy określić położenie danej bierki (figury oraz pionka). Na szachownicy rozstawia się 32 bierki w odmiennych kolorach, zazwyczaj czarnych i białych. Każdemu graczowi przysługują 16 bierek, w skład których wchodzi: król, hetman, dwa gońce, dwa skoczki, dwie wieże oraz osiem pionków. Rozgrywka rozpoczyna się od ustalenia, który zawodnik otrzyma białe bierki co wiąże się z rozpoczęciem partii. Istnieją różne sposoby wyłonienia pierwszeństwa. Jedną z nich jest losowanie, natomiast podczas turnieju wybór zawodnika rozpoczynającego partię jest ustalany przez wybrany system rozgrywek. Gracze wykonują ruch bierkami na zmianę zgodnie z zasadami ruchu dla danej bierki. Jeśli bierka danego zawodnika wejdzie na pole, na którym znajduje się figura bądź pionek przeciwnika następuje zabicie. Zbita bierka jest usuwana z szachownicy i nie bierze udziału w dalszej partii. Celem gry jest postawienie króla przeciwnika w takim miejscu na planszy przed którym nie ma możliwości obrony. W takiej sytuacji następuje zwycięstwo zawodnika, który doprowadził do opisanej sytuacji. Rozgrywka może również zakończyć się remisem, występując on w następujących przypadkach:

- gracze uzgodnili taki wynik,
- wystąpił tak zwany pat, czyli sytuacja, w której jeden z zawodników nie może wykonać żadnego posunięcia zgodnego z zasadami poruszania się bierok oraz nie jest atakowany przez żadną z figur bądź pionka,
- pozostanie zbyt mało bierok do pokonania przeciwnika, np. na planszy pozostaną tylko dwa króle,
- taka sama pozycja wystąpi na szachownicy trzykrotnie,
- jeden z graczy wykonuje posunięcie, który zmusza przeciwnika do powtórzenia swoich ruchów.

## 2.2. Systemy rozgrywek w turniejach

Turnieje szachowe, w przeciwieństwie do innych zawodów np. Mistrzostwa Świata w piłce nożnej lub Wielkoszlemowe turnieje tenisowe, charakteryzują się tym, że może się zgłosić różna liczba graczy. W innych sportach, takich jak np. wspomniana wyżej piłka nożna, liczba zespołów jest z góry ustalona przez organizatorów imprezy. W Mistrzostwach Świata bierze udział 32 kraje, a w Mistrzostwach Europy 24 państwa. Wracając do szachów, w zawodach organizowanych przez Międzynarodową Federację zazwyczaj maksymalna liczba graczy jest podana przed rozpoczęciem turnieju. Oczywiście może się tak zdarzyć, że nie zostanie wypełniona cała pula i turniej będzie zmniejszony. W polskich turniejach zawodowych i amatorskich z reguły nie ma podanej maksymalnej liczby zawodników, więc na dobrą sprawę liczba graczy jest nieograniczona. Na taki przypadek aplikacja również musi być przygotowana.

Bardziej deterministyczną cechą przy ustalaniu właściwości rozgrywek szachowych jest liczba rund. Warto dodać, że jest ona ściśle związana z systemem rozgrywek, który jest głównym czynnikiem oraz cechą turnieju

szachowego. Jak informuje kodeks szachowy wydany przez Polski Związek Szachowy [6], system rozgrywek powinien być dostosowany do rangi zawodów, natomiast liczba rund powinna być dobrana tak, aby istniała możliwość wyłonienia zwycięzcy. Dodatkowo PZS podkreśla, że uczestnicy muszą mieć szansę na pokazanie swoich umiejętności w turnieju. Wyklucza to sytuację, w której do zawodów zgłosiło się 100 graczy, a liczba rund wynosi 2. W takiej sytuacji (niezależnie od wyboru systemu) nie jest możliwe wyłonienie najlepszego zawodnika w turnieju, ponieważ zajdzie sytuacja, w której kilkudziesięciu uczestników będzie miało taką samą, maksymalną, liczbę zwycięstw.

W turniejach szachowych istnieje kilka systemów rozgrywek, najbardziej popularne to:

- system kołowy,
- system pucharowy,
- system szwajcarski.

System kołowy polega na tym, że każdy zawodnik rozgrywa w turnieju po jednej partii ze wszystkimi pozostałymi uczestnikami. Z racji na dużą liczbę meczów system ten przeznaczony jest dla niewielkiej liczby uczestników (przeważnie 16-20, rzadko 20). Numery startowe ustalane są losowo, natomiast następne kojarzenia zawodników dokonywane są według porządku ustalonego w tak zwanych tabelach kojarzeń. Tabele te pozwalają tak przeprowadzić turniej, aby dany zawodnik grał naprzemiennie różnym kolorem bierka, co wiąże się z tym, że zagra jednakową partię zarówno białym, jak i czarnym kolorem. W przypadku zgłoszenia się niewielkiej liczby zawodników możliwa jest modyfikacja systemu kołowego na tak zwany system dwukołowy. Polega on na tym, że każdy gracz rozgrywa z pozostałymi mecz i rewanż. Dodatkowym atutem jest fakt, że rewanże są grane odmiennymi kolorami bierka.

System pucharowy charakteryzuje się główną zasadą – kto przegra, ten odpada. Główną cechą systemu jest fakt, że dzięki niemu można

bezproblemowo i szybko przeprowadzić turniej ze sporą liczbą zawodników. Dodatkowo, system pucharowy cieszy się dużą popularnością wśród kibiców, ponieważ rozgrywany turniej jest szybki i dynamiczny. Każdy mecz jest ważny oraz interesujący. Wadą turniejów rozgrywanych tym systemem jest możliwość niesprawiedliwego rozstrzygnięcia zwycięzcy. Dlatego bardzo istotną rzeczą jest odpowiednie rozstawienie graczy przed zawodami. Polega ono na umieszczeniu najlepszych zawodników w osobnych częściach turniejowej drabinki, aby ewentualna partia pomiędzy nimi była w jak najpóźniejszej fazie rozgrywek. Warto zaznaczyć, że systemie pucharowym liczba miejsc w drabince wynosi równowartość jednemu z wyrazów ciągu geometrycznego  $2^n$ , czyli 2, 4, 8, 16 itd. W sytuacji, w której liczba zawodników jest mniejsza niż dany wyraz ciągu geometrycznego możliwe jest pauzowanie najwyżej rozstawionych zawodników w pierwszej rundzie.

System szwajcarski polega na pojedynkach pomiędzy zawodnikami o zbliżonym poziomie umiejętności. Z góry ustalana jest liczba rund, która zostanie rozegrana. Dana liczba musi spełniać równanie (2.1).

$$\begin{cases} x > 0 \\ x \leq \begin{cases} p - 1, & p \bmod 2 = 0 \\ p, & p \bmod 2 = 1 \end{cases} \end{cases} \quad (2.1)$$

gdzie:

- $x$  – możliwa liczba rund,
- $p$  – liczba zawodników

Drugą możliwością jest wykorzystanie wzoru (2.2) przedstawiającego optymalną liczbę rund uwzględniając liczbę graczy.

$$x = \text{ceil}(\log_2 p) \quad (2.2)$$

gdzie:

- $p$  – liczba zawodników

Przed każdym skojarzeniem zawodnicy powinni być uszeregowani na liście według następujących kryteriów:

1. Liczba punktów



2. Ranking międzynarodowy (FIDE)
3. Ranking polski (ewentualnie ranking lokalny, jeśli istnieje)
4. Kolejność alfabetyczna

W pierwszym skojarzeniu zawodnicy mają 0 punktów, więc uszeregowanie graczy rozpoczyna się zaczynając od rankingu międzynarodowego. Po przygotowaniu listy startowej (biorąc pod uwagę sortowanie zawodników względem powyższych zasad) dzielona jest ona na pół. Pierwszy gracz z górnej połówki mierzy się z pierwszym graczem z dolnej i następnymi według tej zasady. W przypadku nieparzystej liczby uczestników, zawodnik z najmniejszym dorobkiem punktowym otrzymuje tak zwany „bye” (wolny los). Przed kolejnymi rundami następuje przyporządkowanie zawodników do grup punktowych. Każdy uczestnik turnieju trafia do grupy, w której znajdują się rywale, którzy mają tyle samo punktów. Dana grupa dzieli się na dwie części. Podobnie jak w przypadku kojarzenia par w pierwszej rundzie, gracz z górnej części trafia na pierwszego gracza z dolnej. Problem jest wtedy, gdy zawodnik nie znajdzie rywala w swojej grupie punktowej. W takiej sytuacji należy gracza przenieść na pierwsze miejsce, niższej grupy i rozpocząć szukanie przeciwnika jak opisano powyżej. Może zdarzyć się sytuacja, w której gracz „spadnie” do najniższej grupy punktowej i tam nie znajdzie przeciwnika. Wtedy należy z powrotem go przenieść do wyższych grup i poszukać dla niego odpowiedniego rywala spełniającego warunki i kryteria systemu szwajcarskiego. W tym miejscu warto zaznaczyć, że istnieją dwie podstawowe zasady systemu, których nie wolno złamać. Jedna z nich mówi o tym, że niedopuszczalne jest kojarzenie dwóch zawodników więcej niż jeden raz. Kolejna informuje, o tym, że dany gracz, w ciągu całego turnieju może otrzymać wolny los tylko jeden raz. Zaletą przedstawionego systemu jest możliwość przeprowadzenia zawodów z większą liczbą uczestników. Przeciwnie do systemu pucharowego, jedna przegrana nie powoduje odpadnięcia z turnieju, a możliwość zmierzenia się z graczami o podobnych umiejętnościach. Należy zaznaczyć, że przedstawiony opis jest jedynie skróconą oraz uproszczoną wersją tego niezwykle

skomplikowanego systemu. Pominięto w nim kwestie takie jak np. algorytm doboru kolorów bierek, informacje o wskaźniku „x” oraz „p”. Znacznie uproszczono opis przemieszczania zawodników pomiędzy grupami punktowymi.

## 2.3. Przegląd istniejących rozwiązań

Przed rozpoczęciem pracy zrobiono przegląd istniejących programów, których zadaniem jest przeprowadzenie turnieju. Istnieje kilka programów komputerowych, o których warto wspomnieć. W angielskiej wersji językowej warto wyróżnić webowy program SWIPS<sup>1</sup>. Użytkownik za darmo otrzyma bardzo okrojoną wersję, w którym największą wadą jest brak wprowadzania dowolnej ilości zawodników i rund. Maksymalna liczba rund wynosi 5, natomiast graczy – 20. Płatna wersja zawiera dodatkowo, między innymi, eksportowanie danych do plików oraz możliwość edytowania kojarzeń. Kolejnym rozwiązaniem webowym jest ChessManager<sup>2</sup>. Program w polskiej wersji językowej (za darmo) dostarcza przeprowadzenie turnieju bez limitów liczby rund i zawodników. Wersja płatna zawiera między innymi następujące funkcjonalności: zmiana sposobu sortowania listy startowej, eksport wszystkich danych zawodników oraz ustalanie punktacji pomocniczej.

Na rynku istnieje kilka aplikacji desktopowych. Jedną z nich jest UTU Swiss<sup>3</sup>. Za darmo oferuje jedynie wersję próbną, w której można wprowadzić maksymalnie 5-rundowy turniej z 16 zawodnikami. Program dostępny jest w wersji angielskiej. Innymi znalezionym rozwiązaniami są programy w polskiej wersji językowej: ChessArbiter PRO<sup>4</sup> oraz OSwiss<sup>5</sup>. Pierwszy z nich jest programem płatnym, a za darmo można pracować tylko w trybie

---

<sup>1</sup> <https://chess.swips.eu>

<sup>2</sup> <https://www.chessmanager.com/>

<sup>3</sup> <https://www.utuswiss.co.uk>

<sup>4</sup> <http://www.chessarbiter.com>

<sup>5</sup> <http://oswiss.ryuu.eu>

demonstracyjnym, który niesie ograniczenia takie jak: maksymalnie 15 zawodników, maksymalnie 5 rund. Drugi jest całkowicie darmowy i oferuje użytkownikom wszystkie niezbędne funkcje do prawidłowego przeprowadzenia turnieju szachowego (kojarzenie graczy, przeliczanie punktów, brak limitów na liczbę graczy oraz rund).

W wersji na urządzenia mobilne znaleziono odpowiednik programu komputerowego SWIPS Chess Tournament Manager<sup>1</sup>. Również aplikacje takie jak: Tournament Manager<sup>2</sup> czy Tournament Maker<sup>3</sup> służą do pomocy przy organizowaniu turnieju szachowego. Wymienione programy są darmowe, w wersji angielskiej. W polskiej wersji językowej nie znaleziono żadnej aplikacji mobilnej na system Android.

## 2.4. Ustalanie pozycji na liście wyników

Czynnikiem, który należy wziąć pod uwagę przy projektowaniu aplikacji wspierającej organizację turnieju szachowego jest ustalanie kolejności miejsc. Zawsze o kolejności zajętych miejsc rozstrzyga liczba zdobytych punktów. Jednakże w przypadku, w którym dwóch (lub więcej) zawodników będzie miało taką samą liczbę punktów należy posłużyć się kryterium wyboru wyżej sklasyfikowanego zawodnika. W tym przypadku z pomocą przychodzą następujące metody, które rozstrzygają o zdobytych miejscach:

- Punktacja Sonneborna–Bergera – zawodnikowi przypisuje się sumę liczby punktów przeciwników, z którym dany zawodnik wygrał oraz połowę sumy liczby punktów, z którym przegrał. Metoda ta jest wykorzystywana w turniejach kołowych.

---

<sup>1</sup> [https://play.google.com/store/apps/details?id=sk.swips.swips\\_application](https://play.google.com/store/apps/details?id=sk.swips.swips_application)

<sup>2</sup> <https://play.google.com/store/apps/details?id=com.nestsport.tournament>

<sup>3</sup> <https://play.google.com/store/apps/details?id=org.eniblo.all.leagues>

- Punktacja Buchholza – zawodnikowi przypisuje się liczbę punktów, która jest sumą zdobytych punktów przez wszystkich jego przeciwników. Jeśli zawodnik podczas turnieju miał wolny los to suma zostaje pomniejszona o 0.5 punktu.
- Punktacja średnia Buchholza – analogicznie do punktacji Buchholza, jednak nie bierze pod uwagę przeciwników, którzy zdobyli największą oraz najmniejszą liczbę punktów.
- Metoda progresji – zawodnikowi przypisuje się sumę punktów, będącą sumą punktów po każdej kolejnej rundzie.
- Suma rankingów – stosowana przez FIDE, bierze się średni ranking przeciwników, którym dany zawodnik rozgrywał partię.
- Wynik bezpośrednich partii – rozstrzyganie o kolejności pomiędzy dwoma zawodnikami.
- System Koyi – zawodnikowi przypisuje się liczbę punktów, z graczami, którzy zdobyli przynajmniej 50% punktów do zdobycia. System jest przeznaczony wyłącznie dla turniejów kołowych.

## **3. Wymagania i narzędzia**

### **3.1. Wymagania funkcjonalne**

Aplikacja służąca za przeprowadzenie turnieju szachowego ma za zadanie wspomóc organizatorów zawodów w bezproblemowy oraz niezawodny sposób doprowadzić turniej do jego finalizacji. Minimalnymi wymaganiami funkcjonalnymi cechującą aplikację, która sprawne i niezawodnie pomoże przy organizacji turnieju są:

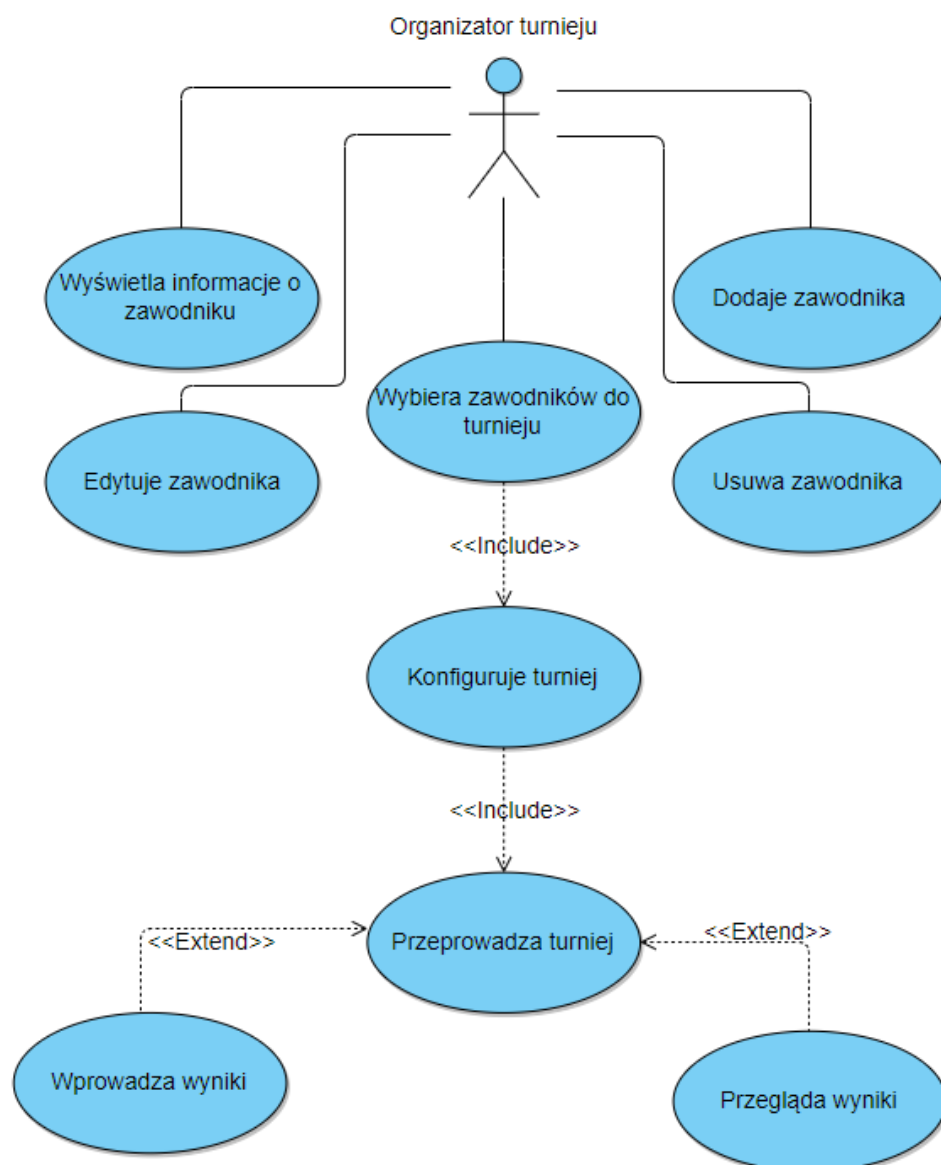
- dodanie dowolnej liczby zawodników do turnieju,
- możliwość wprowadzenia dowolnej liczby rund,
- zdefiniowanie systemu rozgrywek,
- ustalenie metody decydującej o końcowym rezultacie w przypadku równej ilości punktów.

Po przeanalizowaniu specyfiki turniejów szachowych postanowiono wybrać następujące ustawienia zawodów. Aby aplikacja była jak najbardziej uniwersalna wybrano nielimitowaną liczbę zawodników biorących udział w turnieju. Użytkownik nie będzie miał obostrzenia i może wprowadzać tyle zawodników ile zgłosi się do rozgrywek. Systemem rozgrywek wybrano system szwajcarski. Stwierdzono, że jest to najbardziej uniwersalny i sprawiedliwy system. Niewątpliwie zaletą jest to, że dzięki niemu można sprawnie przeprowadzić turniej ze sporą ilością zawodników. Jego

dodatkowym atutem jest fakt, że nawet pomimo porażki, gracz dalej bierze udział w rozgrywce i każdy z uczestników zagra tyle samo meczów. Warto tutaj dodać, że zaimplementowany w aplikacji system szwajcarski jest nieodzwierciedlony jeden do jeden zgodnie z przepisami kodeksu szachowego. Uproszczono w nim między innymi sposób przemieszczania się zawodników pomiędzy grupami w przypadku, gdy dany gracz nie znajdzie w swojej grupie punktowej przeciwnika. Dodatkowo zmodyfikowano algorytm przydzielania bierek. Kolejną cechą, którą trzeba poruszyć jest ustawienie ilości rund w turnieju. Użytkownik ma dwie możliwości. Pierwszą z nich jest ustawienie dowolnej liczby rund spełniającą równanie (2.1). Drugą możliwością jest skorzystanie z gotowego wzoru (2.2) przedstawiającego optymalną liczbę rund uwzględniając liczbę graczy.

Ostatnim czynnikiem jaki wpływa na jakość aplikacji wspomagającej organizację turnieju szachowego jest wybór metody ustalania kolejności miejsc. Tutaj wybrano najbardziej popularną i najczęściej używaną w systemie szwajcarskim punktację Buchholza. Zdecydowano się na nią, ponieważ odzwierciedla umiejętności zawodnika. Nie ma wątpliwości, że łatwiej wygrać ze słabszym przeciwnikiem niż z mocnym. Metoda ta obrazuje z jak dobrymi graczami dany uczestnik turnieju musiał się mierzyć. Dodatkowo zaimplementowana została metoda średniej Buchholza.

Aplikacja nie wymaga podziału użytkowników na role z różnymi uprawnieniami.. Stwierdzono, że nie potrzeba kogoś w rodzaju administratora, ponieważ system nie komunikuje się ze światem zewnętrznym, więc nie istnieje wymiana danych pomiędzy użytkownikami korzystającymi z aplikacji. Dany klient ma dostęp do każdej funkcji istniejącej w systemie. Zajmuje się on wprowadzaniem zawodników do turnieju, konfiguracją oraz przeprowadzaniem zawodów. Jego rolą jest przede wszystkim poprawne wpisanie wyników meczów danej rundy. Wymagania funkcjonalne zostały przedstawione w postaci diagramów przypadków użycia (Rys. 3.1).



Rys.3.1. Diagram przypadków użycia

## 3.2. Wymagania niefunkcjonalne

Aplikacja działa na urządzeniach mobilnych z systemem Android w wersji 9 lub wyższej. Została opracowana na tablety z minimalną przekątną 10 cali. Posiada prosty i przejrzysty interfejs użytkownika. Jego ważną cechą jest intuicyjność. Odnosi się to szczególnie do wprowadzania wyników spotkań. Kolorystyka jest oparta na różnych odcieniach brązu. Rozmiar aplikacji wynosi 2,65MB. Dzięki testowaniu programu można stwierdzić, że jest on niezawodny. Najważniejszym punktem programu jest kojarzenie par w systemie szwajcarskim. Zostało one zaprojektowane zgodnie z ustalonymi wcześniej wytycznymi. Kolejną cechą jest możliwość rozbudowy o, między innymi, dodanie innych systemów rozgrywek lub generowanie raportów po zakończonych rundach.

## 3.3. Opis narzędzi

Do opracowania aplikacji zostało wykorzystane środowisko programistyczne jakim jest Android Studio. Jest to najpopularniejsza platforma systemowa do projektowania, tworzenia oraz debugowania programów dla urządzeń mobilnych. Oferuje wiele przydatnych funkcji, które niewątpliwie pomagają użytkownikom w pracy z programem. Formatowanie kodu, kolorowanie składni, łatwość w debugowaniu, prostota w tworzeniu interfejsu graficznego to tylko niektóre z licznych zalet jakie posiada w swojej ofercie Android Studio. Dodatkowym autem jest bezpłatność programu. Dzięki wbudowanemu narzędziu do kontroli wersji w bezproblemowy sposób można projekt wrzucać na platformę GitHub, z której również skorzystano. Do testowania aplikacji posłużono się wirtualnym emulatorem wbudowanym w środowisko programistyczne. Wybrano tablet o wymiarach 10.1", rozdzielczości WXGA (panoramiczna rozdzielczość, powstała na bazie XGA). Aplikacja została napisana w języku Java. Jako system zarządzania bazą danych użyto SQL Lite, która najbardziej nadaje się do korzystania z

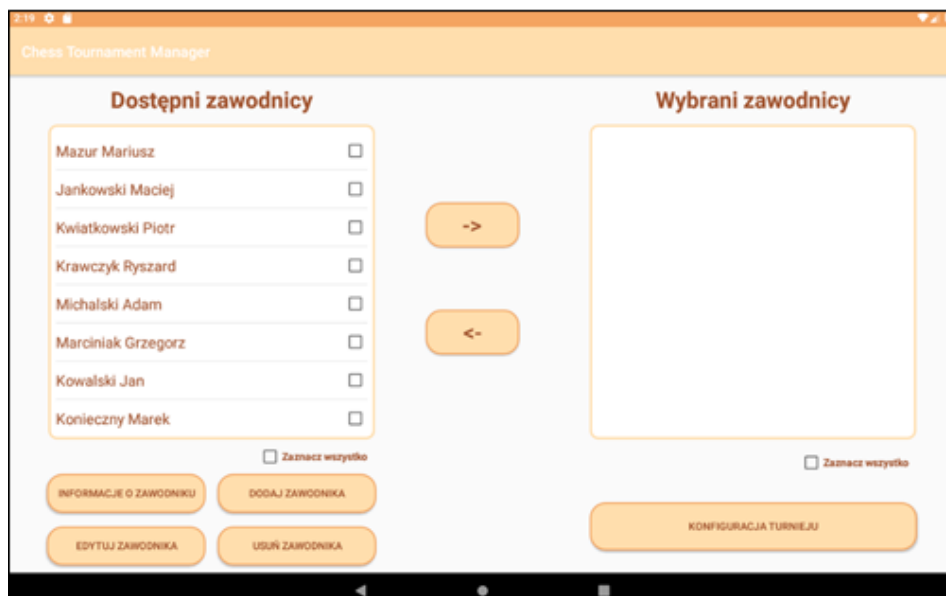


wbudowanymi systemami jakim jest np. tablet. Do mapowania obiektowo relacyjnego (ORM) użyto biblioteki Room [9]. Dzięki niej można w łatwy sposób komunikować się z bazą danych z poziomu kodu. Swoją składnią przypomina Hibernate. Niewątpliwie zaletą użytej biblioteki jest prostota w zrozumieniu oraz wykonaniu zapytań do bazy. Można uruchamiać kolejne wątki, które łączą się z bazą danych nie blokując jednocześnie wyświetlania interfejsu graficznego. Daje to duże pole do popisu przy aplikacjach dynamicznych. Jediną wadą jest brak formatu daty, dlatego został napisany konwerter, który umożliwia jej zapis oraz odczyt.

## 4. Specyfikacja zewnętrzna

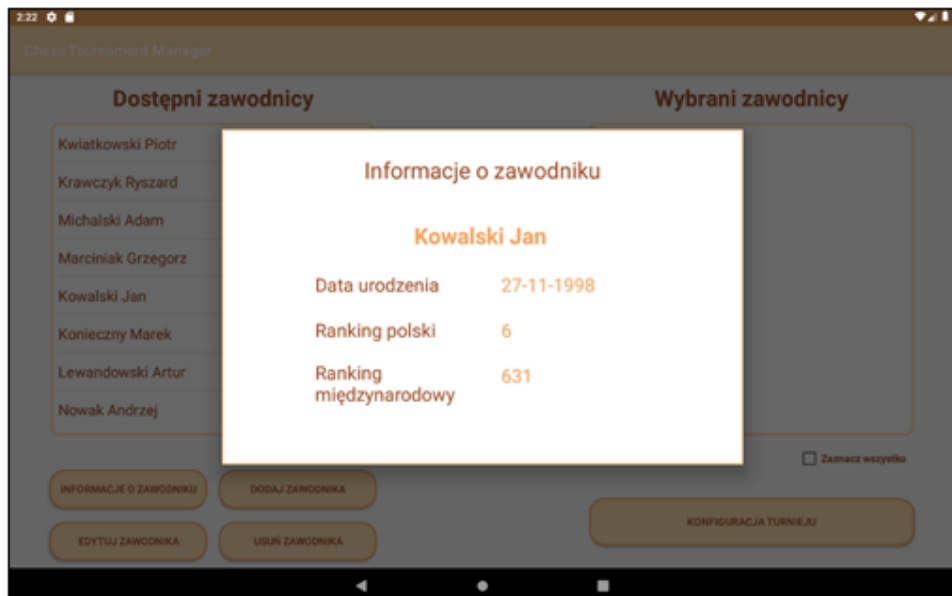
Program do przeprowadzania turnieju szachowego jest dedykowany na urządzenia mobilne z systemem Android od wersji 9.0. Projekt „Opracowanie aplikacji mobilnej wspomagającej prowadzenie turnieju szachowego” przeznaczony jest na tablety o rozdzielczości około 10 cali. Na jednej aktywności wyświetlanych jest dużo różnych informacji. Aby użytkownikowi wygodnie się pracowało zalecana jest instalacja oprogramowania na urządzeniach mobilnych o większych rozmiarach. Do korzystania z aplikacji wymagana jest jej instalacja na tablecie, na którym będzie ona używana. Po zainstalowaniu program jest gotowy do użycia, należy kliknąć na ikonę aplikacji na urządzeniu mobilnym.

Pierwszym ekranem (Rys.4.1) jaki się wyświetla po uruchomieniu programu jest wybór zawodników do turnieju. uruchomieniu programu jest wybór zawodników do turnieju. W liście znajdującej się po lewej stronie, zatytułowanej „Dostępni zawodnicy” istnieje możliwość zaznaczenia pojedynczego zawodnika lub wszystkich naraz. Wybranych graczy należy dodać do listy, która leży po prawej stronie ekranu („Wybrani zawodnicy”). W niej znajdują się zawodnicy, którzy zostaną dodani do turnieju po kliknięciu przycisku „konfiguracja turnieju”. Z listy wybranych uczestników również można przerzucić zawodników z powrotem do listy dostępnych graczy. Działa to na identycznej zasadzie jak w przypadku dodania zawodników z prawego do lewego panelu, jednakże należy nacisnąć odpowiedni przycisk (lewa strzałka).

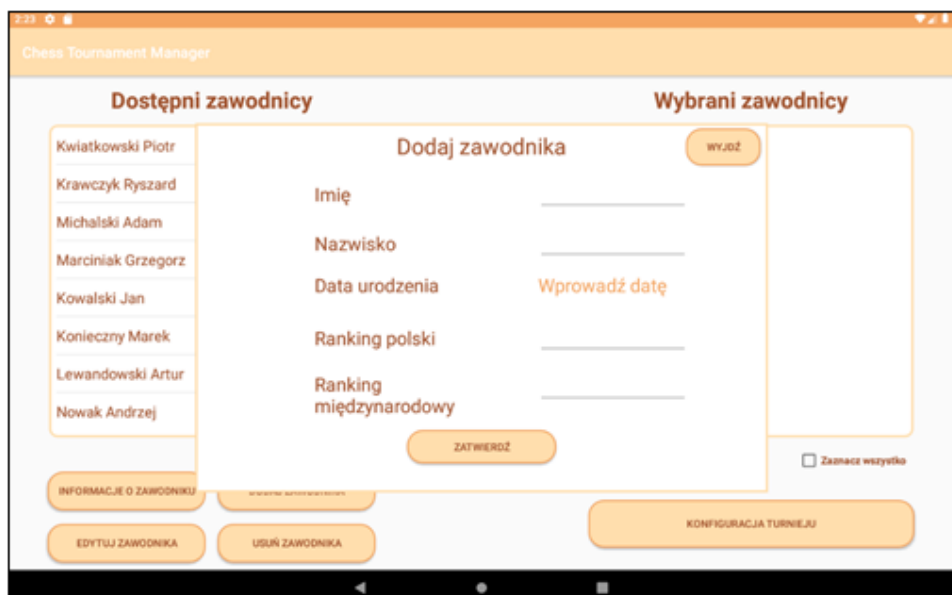


Rys.4.1. Wybór zawodników do turnieju

Pod listą dostępnych szachistów znajdują się cztery przyciski o następujących funkcjach. Pierwszą z nich jest wyświetlanie informacji o zawodniku. Po otwarciu okienka istnieje możliwość zobaczenia wszystkich wiadomości na temat zaznaczonego zawodnika (Rys.4.2). Po prawej stronie widnieje przycisk, który służy do dodania gracza do systemu (Rys.4.3). Dzięki niemu najpierw zawodnik zostanie dodany do bazy danych, następnie wyświetlony zostanie w lewym panelu. Wymagane jest wypełnienie imienia, nazwiska oraz wybranie daty z kalendarza, który pokaże się po kliknięciu w tekst „Wprowadź datę”. Ranking polski oraz międzynarodowy może pozostać pusty jeśli wprowadzany zawodnik go nie posiada. Aby potwierdzić dodanie nowego szachisty należy nacisnąć przycisk „Zatwierdź”, natomiast aby anulować i zamknąć okno trzeba użyć przycisku „Wyjdź” widniejącego w prawym, górnym rogu.



Rys.4.2. Wyświetlanie informacji o zawodniku



Rys.4.3. Dodanie zawodnika

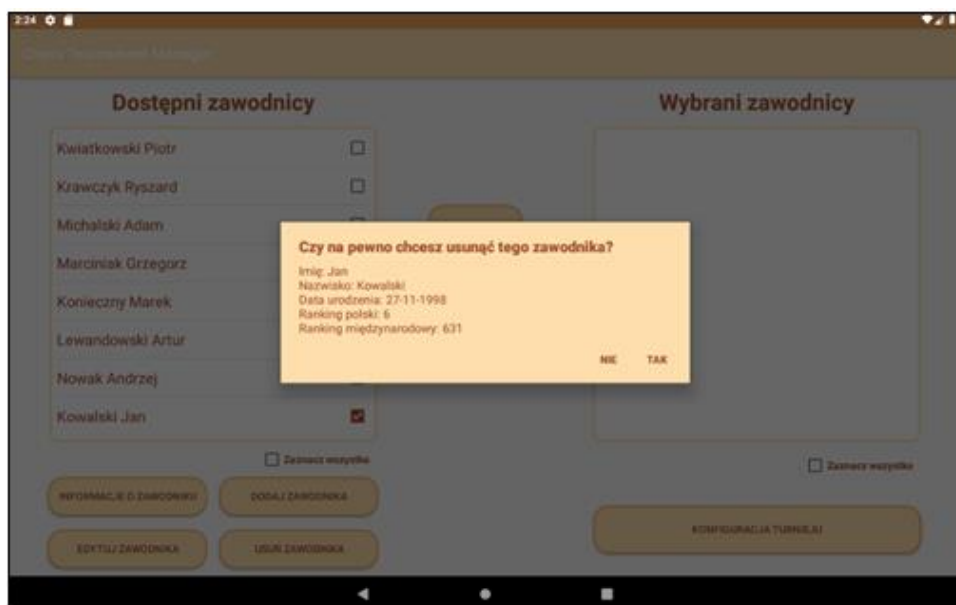
Kolejną funkcją jest edycja gracza (Rys.4.4).

The screenshot shows the 'Chess Tournament Manager' application interface. The top bar is orange with the title 'Chess Tournament Manager'. Below it, the main area is divided into three columns. The left column, titled 'Dostępni zawodnicy', contains a list of player names. The middle column, titled 'Edytuj zawodnika', contains a form with five input fields for player details. The right column, titled 'Wybrani zawodnicy', is currently empty. At the bottom, there are four orange buttons: 'INFORMACJE O ZAWODNIKU', 'EDYTUJ ZAWODNIKA', 'USUŃ ZAWODNIKA', and 'KONFIGURACJA TURNIEJU'. A checkbox labeled 'Zaznacz wszystkie' is located in the bottom right area.

Dostępni zawodnicy	Edytuj zawodnika	Wybrani zawodnicy
Kwiatkowski Piotr	Imię: Jan	
Krawczyk Ryszard	Nazwisko: Kowalski	
Michalski Adam	Data urodzenia: 27-11-1998	
Marciniak Grzegorz	Ranking polski: 6	
Kowalski Jan	Ranking międzynarodowy: 631	
Konieczny Marek		
Lewandowski Artur		
Nowak Andrzej		

Rys.4.4. Edytowanie zawodnika

Po zaznaczeniu jednego człowieka na liście dostępnych zawodników należy kliknąć w pole oznaczone tekstem „edytuj zawodnika”. Ukazane zostanie okno identyczne do poprzedniego, jednak w polu wypełnienia informacji o zawodniku zostaną podstawione dane zaznaczonego gracza. Jeśli szachista nie posiada rankingu pokaże się informacja „Brak rankingu” w odpowiednim miejscu. Ostatnim przyciskiem widniejącym pod listą dostępnych zawodników jest „usuń zawodnika” (Rys.4.5). Należy zaznaczyć dowolnych graczy oraz kliknąć na przycisk. W momencie wybrania jednego człowieka wyświetlona zostanie o nim informacja oraz zapytanie czy na pewno go usunąć. Jeśli zostanie zaznaczonych więcej pokaże się zapytanie czy na pewno usunąć bez wyświetlania informacji o kasowanych zawodnikach. Kasacja graczy oznacza wymazania ich z listy znajdującej się po lewej stronie i usunięcie z bazy danych. Należy więc z tej funkcji korzystać w sposób przemyślany.



Rys.4.5. Usuwanie zawodnika

Aby przejść do następnego ekranu (Rys.4.6) należy wcisnąć przycisk „konfiguracja turnieju”. Po prawej stronie wyświetlona zostanie lista wszystkich wybranych zawodników do turnieju wraz z rankingiem międzynarodowym oraz polskim. Z lewej strony widnieją ustawienia zawodów. Pierwsza z nich to możliwość wyboru rund. W systemie szwajcarskich rundy wybierane są przez organizatorów rozgrywek. Niewłaściwy wybór może w negatywny sposób wpłynąć na rozstrzygnięcia turnieju. Do wyboru rundy istnieją dwie możliwości. Pierwsza z nich (domyślna) to wybranie standardowej liczby rund, która wyznaczana jest ze wzoru 2.2. Druga to ręczne wpisanie liczby faz turnieju jednakże musi ona spełniać założenia określone wzorem 2.1. W przeciwnym wypadku zostanie wyświetlony komunikat informujący użytkownika o błędzie oraz z poprawnym zakresem liczby rund. Druga oraz trzecia opcja nie podlega zmianie i ma charakter informujący o liczbie zawodników w turnieju oraz uszeregowaniu ich przed pierwszym kojarzeniem. Sortowanie to jest następujące. W pierwszej kolejności patrzy się na ranking międzynarodowy, kolejno na ranking polski, a na końcu uszeregowanie odbywa się kolejnością alfabetyczną. Ostatnią rzeczą jaką można ustawić przy konfigurowaniu

ustawień zawodów jest wybór metody przy jednakowych wynikach. Możliwości są dwie: metoda Buchholza oraz średni Buchholz.

The screenshot shows the 'Chess Tournament Manager' app interface. It is divided into two main sections: 'Ustawienia turnieju' (Tournament Settings) on the left and 'Lista startowa' (Starting List) on the right.

**Ustawienia turnieju (Tournament Settings):**

- Ręczny wybór liczby rund:** A dropdown menu showing 'Standardowa liczba rund: 4'.
- Liczba zawodników:** A text input field containing the number '10'.
- Uzeregowanie zawodników:** A list of sorting options: '1. Ranking międzynarodowy', '2. Ranking polski', and '3. Alfabetycznie'.
- Metoda przy jednakowych wynikach:** A dropdown menu showing 'Metoda Buchholza'.

**Lista startowa (Starting List):**

Zawodnik	Ranking międzynarodowy	Ranking polski
1. Konieczny Marek	888	66
2. Kowalski Jan	631	6
3. Michalski Adam	99	144
4. Nowak Andrzej	72	619
5. Jankowski Maciej	5	9
6. Marciniak Grzegorz	1	44
7. Krawczyk Ryszard	Bez rankingu	Bez rankingu
8. Kwiatkowski Piotr	Bez rankingu	Bez rankingu
9. Lewandowski Artur	Bez rankingu	Bez rankingu
10. Mazur Mariusz	Bez rankingu	Bez rankingu

At the bottom right of the 'Lista startowa' section, there is a button labeled 'ROZPOCZNIJ TURNIEJ'.

Rys.4.6. Konfiguracja turnieju

Po zakończeniu konfiguracji ustawień turnieju należy nacisnąć przycisk „rozpocznij turniej” znajdujący się w prawym, dolnym rogu. Po jego naciśnięciu ukaże się rezultat kojarzenia pierwszej rundy (Rys.4.7). W nagłówku widnieje informacja o trwającej rundzie. Pierwsza kolumna to szachiści rozgrywający partię białym kolorem. Druga służy do wprowadzenia wyniku. Trzecia to zawodnicy grający czarnym kolorem. Osoba zajmująca się przeprowadzaniem turnieju po zakończonej rundzie musi rozważyć wprowadzić wyniki klikając na listę rozwijaną przy danym meczu.

Gracz 1 (białe)	Wynik	Gracz 2 (czarne)
Michalski Adam	Wprowadź wynik	Marciniak Grzegorz
Nowak Andrzej	Wprowadź wynik	Krawczyk Ryszard
Kowalski Jan	Wprowadź wynik	Kwiatkowski Piotr
Konieczny Marek	Wprowadź wynik	Lewandowski Artur
Jankowski Maciej	Wprowadź wynik	Mazur Mariusz

NASTĘPNA RUNDA

Rys.4.7. Wpisywanie wyników meczów

Istnieją trzy możliwości rozstrzygnięcia wyniku spotkania. Gdy gracz, który gra białymi wygra zostaje mu przypisany wynik „1-0”. W przypadku jego porażki rezultat to „0-1”. Jeśli partia zakończy się remisem należy wybrać opcję „½-½”. Podczas wprowadzania wyników zwycięzca zostanie zaznaczony kolorem brązowym, przegrany – czerwonym, a remisujący zostaną poddani efektowi pochylenia (kursywa). Po uzupełnieniu wszystkich wyników wymagane jest wciśnięcie przycisku „następna runda” w celu kojarzeń drugiej rundy. Podczas trwania turnieju istnieje możliwość przeglądania wyników spotkań z poprzednich rund, aktualnej klasyfikacji oraz wyjścia z turnieju. W tym celu należy kliknąć na znak menu (trzy poziome kreski) znajdujący się w lewym, górnym rogu. Wyświetlone zostanie menu nawigacyjne. Aby zobaczyć wyniki jednej z poprzednich rund należy kliknąć na dany numer rundy. Po przejściu na tą stronę widać wszystkie mecze i wyniki danej rundy (Rys.4.8).

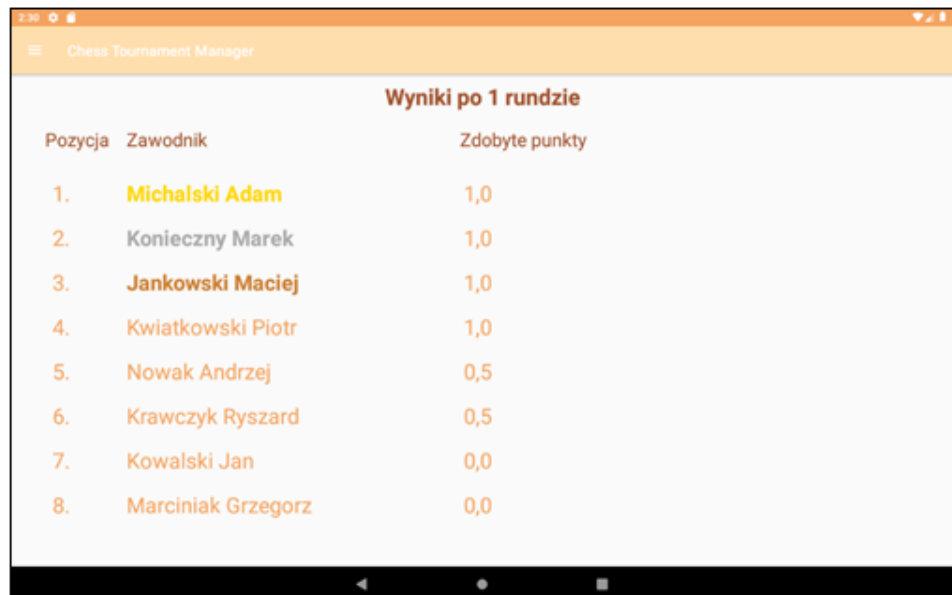




Runda 1			
Lp.	Gracz 1 (białe)	Gracz 2 (czarne)	Wynik
1.	Michalski Adam	Marciniak Grzegorz	1-0
2.	Nowak Andrzej	Krawczyk Ryszard	½-½
3.	Kowalski Jan	Kwiatkowski Piotr	0-1
4.	Konieczny Marek	Lewandowski Artur	1-0
5.	Jankowski Maciej	Mazur Mariusz	1-0

Rys.4.8. Przeglądanie wyników poprzednich rund

Chcąc zobaczyć wyniki po ostatniej skończonej rundzie w menu należy kliknąć w „Wyniki” (Rys.4.9). W ostatniej rundzie zamiast przycisku „następna runda” pokaże się przycisk „pokaż wyniki końcowe” informująca użytkownika, że dana runda jest rundą finalną. Po jej skończeniu ukazuje się strona z końcowymi wynikami (Rys.4.10). Wyświetlane zostaną pozycje danych zawodników w turnieju, liczbę punktów, którą zdobyli oraz wybrana metoda decydująca o kolejności miejsc. Po zakończeniu turnieju można przeglądać wyniki poszczególnych rund. Istnieją dwie możliwości wyjścia z turnieju. Pierwszą z nich jest kliknięcie systemowej strzałki wstecz. Drugą opcją jest wyjście z poziomu menu. Każdorazowa chęć wyjścia z turnieju wymaga potwierdzenia użytkownika w celu uniknięcia sytuacji przypadkowego opuszczenia turnieju.



Pozycja	Zawodnik	Zdobyte punkty
1.	<b>Michalski Adam</b>	1,0
2.	Konieczny Marek	1,0
3.	<b>Jankowski Maciej</b>	1,0
4.	Kwiatkowski Piotr	1,0
5.	Nowak Andrzej	0,5
6.	Krawczyk Ryszard	0,5
7.	Kowalski Jan	0,0
8.	Marciniak Grzegorz	0,0

Rys.4.9. Przeglądanie wyników po ostatnio zakończonej rundzie



Pozycja	Zawodnik	Zdobyte punkty	Metoda Buchholza
1.	<b>Konieczny Marek</b>	3,5	6,5
2.	<b>Jankowski Maciej</b>	3,0	8,0
3.	<b>Michalski Adam</b>	2,5	8,5
4.	Krawczyk Ryszard	2,5	8,0
5.	Kwiatkowski Piotr	2,0	10,5
6.	Mazur Mariusz	2,0	6,5
7.	Kowalski Jan	1,5	8,0
8.	Lewandowski Artur	1,0	8,0

Rys.4.10. Wyświetlanie wyników końcowych

## **5. Specyfikacja wewnętrzna**

### **5.1. Architektura systemu**

Projekt „Opracowanie aplikacji mobilnej wspomagającej prowadzenie turnieju szachowego” jest dedykowany na urządzenia mobilne z systemem Android. Napisany jest w języku Java, wykorzystane środowisko programistyczne to Android Studio. Do przechowywania zawodników użyto bazy danych dedykowanej na potrzeby urządzeń mobilnych – SQL Lite. System wykorzystuje lokalną bazę danych SQL Lite. W dużej mierze jest ona wykorzystywana przy wbudowanych systemach. Pozwala na korzystanie między innymi zapytań zagnieżdżonych, transakcji, widoków, wyzwalaczy oraz nakładaniu więzów integralności. Widać, że nie ustępuje większym systemom zarządzania relacyjnymi bazami danych takimi jak Oracle czy MySQL. Baza zawiera jedną tabelę, która przechowuje dane graczy, którzy dodawani są przez użytkownika. Pola tej tabeli to: id, imię, nazwisko, ranking polski oraz międzynarodowy i data urodzenia. Stwierdzono, że są to wszystkie niezbędne dane do gromadzenia informacji o szachistach.

## 5.2. Przegląd klas

Projekt podzielono poszczególne pakiety. W pierwszym znajdują się klasy reprezentujące aktywności użyte w aplikacji. Do najważniejszych należą te, które są odpowiedzialne za ekrany wyświetlające wprowadzanie zawodników do turnieju, konfigurację zawodów oraz ich przebieg.

### 5.2.1. Aktywności

W aktywności wprowadzania graczy istnieje metoda, która wywoływana jest przy ładowaniu ekranu i służy do pobrania zawodników z bazy danych. Biblioteka Room daje możliwości komunikacji z bazą danych w osobnych wątkach. Dzięki temu aplikacja nie jest spowalniana przez łączenie oraz pobieranie, usuwanie bądź aktualizowanie rekordów w bazie danych. W tej klasie istnieje również mechanizm wyświetlania informacji o zawodników w formie wyskakującego okna. Ponadto zaimplementowane są mechanizmy przemieszczające zawodników pomiędzy dwoma listami: dostępnych oraz wybranych graczy. Niezbędne są metody, które powodują przejście do kolejnych widoków.

W następnej aktywności, którą zdecydowano się opisać jest klasa służąca do konfiguracji turnieju. Ciekawym mechanizmem jest użycie przełącznika, za pomocą którego można ustawić liczbę rund w turnieju. W odpowiedniej metodzie zaimplementowana jest reakcja na włączenie przełącznika. Gdy zmienia swój stan podmieniany jest wygląd. Na starcie jest nieaktywny – ustawiane jest pole tekstowe stworzone w panelu projektowania wyglądu w aplikacji. W momencie zmiany stanu przełącznika, w kodzie tworzone jest edytowalne pole tekstowe, które jest zastępowane polem stworzonym statycznie w panelu projektowania. W opisywanej aktywności zaprojektowano listę zawodników uporządkowaną w pierwszej kolejności według rankingu międzynarodowego, polskiego, a w ostatniej alfabetycznie. Do zaprojektowania listy użyto mechanizmu do tworzenia tabel. W osobnym pliku został napisany nagłówek listy: zawodnik, ranking międzynarodowy,

ranking polski, a w osobnym rzędy odpowiadające nagłówkom. Dzięki temu dane w liście są sformatowane.

Następna aktywność, która pełni istotną rolę w systemie służy do wprowadzania wyników meczów. Na starcie załadowany jest widok pierwszej rundy oraz tworzone jest menu, dzięki któremu można zobaczyć wyniki z poprzednich rund, wyniki po ostatnio skończonej rundzie bądź wyjść z turnieju. Najważniejszą metodą w tej klasie jest budowa widoku do wprowadzania rezultatów spotkań. Jest on tworzony w sposób dynamiczny w zależności od liczby zawodników oraz rund. W pierwszej kolejności został przygotowany układ, który jest przewijany w razie większej ilości meczów. Do niego w pętli dodawane są dynamicznie dwa pola tekstowe (zawodnicy) oraz lista rozwijana z trzema opcjami wyboru (zwycięstwo szachisty grającego białym, czarnym kolorem lub remis). Dodano zabezpieczenie się w razie sytuacji kiedy liczba zawodników jest nieparzysta. Wtedy w ostatnim meczu jeden uczestnik ma wolny los i zamiast listy rozwijanej na sztywno wyświetlany jest wynik, natomiast przeciwnik to „Bye”. Wszystkie pola tekstowe oraz listy rozwijane przechowywane są w tablicach inicjalizowanych przy starcie aktywności. Posłużono się tą strukturą danych, ponieważ poprzez indeksy łatwo jest się dostać do pola czy listy. Na zmianę stanu list nałożono słuchacza, dzięki któremu przypisywany jest wynik poszczególnego meczu oraz zmiana kolorów zawodników.

### 5.2.2. Baza danych

Kolejny pakiet służy do opisu bazy danych. Znajdują się w nim trzy klasy. W pierwszej tworzona jest instancja bazy. Użyto tutaj wzorca projektowego jakim jest singleton, ponieważ w całym programie korzysta się tylko z jednej instancji bazy. Dodatkowo w tej klasie występuje pole typu interfejsowego, który zdefiniowany jest w tym samym pakiecie, a którego implementacja jest dostarczana przez bibliotekę Room. W tym interfejsie znajdują się metody operujące na bazie danych (dodawanie, usuwanie, aktualizacja). Ostatnią klasą znajdującą się w tym pakiecie jest klasa, która konwertuje typ Date na Long i na odwrot. Została ona zaimplementowana, ponieważ biblioteka nie

wspiera obsługi typu Date. W zamian za to można posłużyć się adnotacją TypeConverter przed metodą konwertującą.

### 5.2.3. Mecze

W następnym pakiecie zaimplementowano klasę oraz typ wyliczeniowy, które służą do opisu meczu. Klasa posiada cztery pola: liczba rund, dwóch zawodników, wynik meczu. Rezultat spotkania jest reprezentowany przez typ wyliczeniowy posiadający trzy pola – zwycięstwo, remis, przegrana.

### 5.2.4. Model

Do ważnego pakietu należy zaliczyć model. Podzielono go na dwie klasy. W jednej występuje klasa reprezentująca zawodnika, w której znajduje się mapowanie do tabeli w bazie danych. Dzięki bibliotece Room mapowanie jest bardzo proste, przypominające znaną bibliotekę jaką jest Hibernate. Za pomocą adnotacji Entity odwzorowana jest tabela z bazy, natomiast ColumnInfo – kolumna. Różnicą jest brak wcześniej wspomnianego formatu daty. Z pomocą przychodzi adnotacja TypeConverters, w której podaje się nazwę klasy do konwertera. Następna jest klasa reprezentująca zawodnika biorącego udział w turnieju. Dziedziczny ona po klasie zawodnika opisaną powyżej. Posiada pola charakterystyczne dla zawodów np. liczbę zdobytych punktów, poprzedni przeciwnicy oraz kolory bierok używanych we wcześniejszych rundach. W pakiecie znajduje się typ wyliczeniowy, który opisuje kolor bierok. Gracz może otrzymać biały, czarny albo żaden kolor w przypadku wolnego losu. Dodatkowo zaimplementowana została metoda, która zwraca przeciwny kolor do otrzymanego. Jest ona używana przy przydzielaniu kolorów zawodnikom.

### 5.2.5. Klasy pomocnicze

W następnym pakiecie znajdują się klasy „pomocnicze”, używane w innych klasach. Przykładem takiej klasy są adaptory, które zaimplementowane zostały, aby stworzyć listę z nagłówkiem przedstawiających zawodników biorących udział w turnieju. Dodatkowo

napisano klasy budujące okna dialogowe. Dzięki nim w wywołują się metodę z tej klasy z odpowiednimi wiadomościami oraz przyciskami i okno jest wyświetlane. Nie ma potrzeby za każdym razem budować okien dialogowych od zera. Kolejna klasa służy do opisu stałych spoza aktywności, które są zdefiniowane w odpowiednim pliku (strings.xml).

### 5.2.6. System szwajcarski

Do najważniejszego pakietu należy pakiet, w którym znajduje się klasa opisująca system szwajcarski. W pierwszej kolejności przypisywane są wartości pól podane przez użytkownika: liczba rund oraz metoda decydująca o miejscu w przypadku równej ilości punktów. Dodatkowo tworzona jest lista zawodników turniejowych na bazie listy graczy wybranych do zawodów. Sprawdzana jest parzystość liczby zawodników, która decyduje czy któryś z nich będzie miał w danej rundzie wolny los. Zważywszy na różnice zaimplementowanego systemu szwajcarskiego z systemem zapisanym w kodeksie szachowym postanowiono opisać napisany algorytm. Pierwszym krokiem wypracowanego algorytmu jest kojarzenie pierwszej rundy. Postanowiono napisać osobną metodę, ponieważ ta faza charakteryzuje się różnymi zasadami w porównaniu do pozostałych. Listę zawodników dzieli się na pół i następuje kojarzenie według porządku: pierwszy gracz z górnej części spotyka się z pierwszym graczem z części dolnej. Analogicznie pozostali zawodnicy są kojarzeni. Wyjątkiem jest sytuacja, w której występuje nieparzysta liczba zawodników. Wtedy ostatni zawodnik ma wolny los, dostaje punkt za wygraną, ale nie jest mu przydzielany kolor bierek. Kolejne rundy kojarzone są według następującego algorytmu. Najpierw przygotowane są grupy punktowe. Zawodnicy przyporządkowani są do grup o tej samej liczbie zdobytych oczek. Znajdowany jest gracz, który dostanie wolny los. Poszukiwania zawodnika, który będzie pauzował rozpoczyna się od znajdującego się na najniższym miejscu w ostatniej grupie. Jeśli nie otrzymał on wcześniej wolnego losu to zostaje wyznaczony do pauzowania. W przeciwnym wypadku bierze się następnego zawodnika w kolejności do momentu, w którym nie otrzymał wcześniej wolnego losu. Znalezione szachista jest usuwany z grupy punktowej, ponieważ nie bierze on udziału w

dalszym kojarzeniu. Zaczynając od pierwszego zawodnika z najwyższej grupy punktowej szuka się dla niego przeciwnika w następujący sposób. W pierwszej kolejności sprawdza się czy rozpatrywany gracz posiada rywala w swojej grupie. Aby zaistniał dobór zawodników muszą zostać spełnione dwie zasady. Gracz nie posiada już przeciwnika oraz znaleziony przeciwnik nie rozgrywał z nim meczu w poprzednich rundach. Znajdowanie rywala w grupach punktowych przebiega według następującego algorytmu. Najpierw szuka się go w przeciwnej połowie grupy (rozpoczynając od pierwszej pozycji), następnie w swojej połowie w sposób analogiczny. W przypadku dalszego nie znalezienia oponenta, szachista łądję do innej grupy. W tym momencie wszystkie mecze, które zostały przypisane do zawodników z grupy, do której spadł gracz zostaną usunięte. Jeśli nie spadł do grupy ostatniej to spada do niższej. W przeciwnym wypadku kierowany jest do wyższej, chyba że znajduje się w pierwszej wtedy spada do niższych grup i szuka w nich rywala. Podczas dodawania meczu następuje przydzielanie kolorów bierok. Oblicza się ile ostatnich partii dani zawodnicy rozegrali jednym kolorem. W zależności, który rozegrał więcej spotkań jednym kolorem otrzymuje on kolor przeciwny. Przebieg algorytmu kojarzenia zawodników powtarzany jest dopóki wszyscy zawodnicy zostaną skojarzeni. Po zakończeniu ostatniej rundy następuje sortowanie w pierwszej kolejności według liczby zdobytych punktów, w następnej według wybranej przez użytkownika metody wyłonienia lepszego w przypadku równej ilości oczek. Warto zaznaczyć, że opracowany algorytm jest jedynie uproszczoną wersją oryginalnego systemu szwajcarskiego.



## 6. Weryfikacja i walidacja

Jedną z najważniejszych cech oprogramowania jest jego niezawodność. Użytkownik korzystający z aplikacji wymaga, żeby jej działanie było zgodnie z ustalonymi wcześniej wytycznymi. Dodatkowo bardzo ważna jest bezawaryjność funkcjonowania. Program nie może generować błędów czy w sposób nagły zakończyć swojej pracy. Do uniknięcia powyżej przedstawionych sytuacji wykonuje się testowanie aplikacji.

Podczas implementacji program do przeprowadzania turnieju szachowego był testowany na bieżąco. Każda nowo dodana funkcjonalność została sprawdzana pod kątem poprawności jej działania. Przeanalizowano przypadki skrajne, niepoprawne oraz brak wymaganych danych. Dodatkowo każda nowa aktywność była testowana pod kątem przejścia z poprzedniego widoku. Dane, które przekazywane są z poprzedniej aktywności powinny zostać prawidłowo odebrane oraz przetworzone w nowej. W programie istnieje komunikacja z lokalną bazą danych. Każdorazowe wstawianie, usuwanie oraz edytowanie danych zostało przetestowane. Również pobieranie rekordów z SQL Lite zostało poddane testom. Podanie błędnych danych przez klienta zostaje zakomunikowane odpowiednim komentarzem wraz z uzasadnieniem oraz podaniem rozwiązania problemu. Niemożliwym jest jednak zabezpieczenie aplikacji przed błędnym wprowadzeniem wyników z rzeczywistymi rozstrzygnięciami na szachownicy.

Najważniejszym punktem programu było opracowanie algorytmu do systemu szwajcarskiego. Podczas implementacji bardzo uciążliwe było jego testowanie z racji jego złożoności. Gracze powinni grać z rywalami o tej

samej lub najbardziej zbliżonej liczbie punktów. Każdy zawodnik może grać z drugim tylko raz podczas trwania całego turnieju. Tylko raz uczestnik turnieju może dostać wolny los. Dany szachista powinien grać zbliżoną liczbę partii kolorem czarnym i białym. Te warunki znacznie utrudniały implementację, a w późniejszym etapie testowanie. Zauważono, że obie te czynności zajęły podobną ilość czasu, ponieważ algorytm musiał być zaimplementowany w sposób bezbłędny. Przeanalizowano wszystkie możliwe warunki, które mogą się zdarzyć podczas turnieju.

Napisane zostało 6 testów jednostkowych, które powinny być jak najbardziej czytelne [2]. Testy podzielono na dwie grupy. Pierwszą z nich jest inicjowanie losowej liczby zawodników oraz liczby rund z podanego zakresu. Warto zaznaczyć, że wpisanie poprawnej liczby rund (nie większej niż liczby graczy) jest testowane we wcześniejszej fazie programu i nie jest poddawane testom, ponieważ istnieje gwarancja, że dane są poprawne. Drugą grupą testową jest losowanie liczby zawodników z podanego zakresu, natomiast liczba rund jest dostosowana do maksymalnej możliwej wartości w zależności od wprowadzonych zawodników (wzór 2.1). Wszystkie testy napisane są w sposób następujący. W pierwszej kolejności wykonywane jest inicjowanie liczby rund oraz szachistów. Następnie na konsolę wypisane są powyższe informacje wraz z nazwą testu. Później symulowany jest przebieg turnieju. Kojarzone są następne rundy, natomiast wyniki są losowane w celu zautomatyzowania testów. Po zakończonych zawodach dla każdego zawodnika wykonywana jest asercja, która ma zwrócić negatywny wynik testu dla wartości przeciwnej do oczekiwanej. Pierwsze dwa testy obejmują sprawdzanie warunku o niepowtarzaniu się tych samych meczów w całym turnieju. Kolejne dwa sprawdzają czy któryś zawodnik nie otrzymał wolnego losu więcej niż jeden raz. Ostatnie dwa nadzorują poprawną kolejność zawodników po ostatniej rundzie. Wyniki testów jednostkowych przedstawione są na rysunku 6.1, a podany zakres liczby zawodników to 20 - 50.

1. repeatableOpponentRandom ()  
Dla liczby zawodników = 31  
Dla liczby rund = 26
2. repeatableOpponentMaxRoundsNumber()  
Dla liczby zawodników = 46  
Dla liczby rund = 45
3. checkByeRandom()  
Dla liczby zawodników = 47  
Dla liczby rund = 41
4. checkByeMaxRoundsNumber()  
Dla liczby zawodników = 34  
Dla liczby rund = 33
5. correctFinalOrderRandom()  
Dla liczby zawodników = 37  
Dla liczby rund = 21
6. correctFinalOrderMaxRoundNumber()  
Dla liczby zawodników = 44  
Dla liczby rund = 43

Rys.6.1. Wyniki testów jednostkowych

Po przetestowaniu algorytmu szwajcarskiego w sposób automatyczny postanowiono sprawdzić poprawność kojarzenia się par spełniając założenia systemu. Z racji złożoności oraz wysokiego stopnia skomplikowania algorytmu nie znaleziono sposobu na zaimplementowanie testów automatycznych. Postanowiono drogą manualną sprawdzić napisany algorytm pod kątem sposobu doboru zawodników w pary. Przeanalizowane zostało kojarzenie dla 15 zawodników, na maksymalnym dystansie rundowym (15). Po każdej rundzie wypisano grupy punktowe, aktualną klasyfikację generalną oraz wygenerowane mecze. Następnie każda runda została prześwietlona pod kątem poprawności kojarzeń, spadków do niższej grupy bądź awansów do

wyższej. Po zakończeniu turnieju zostaje wyświetlona historia gier, która dla każdego zawodnika wypisuje wszystkich jego przeciwników oraz kolory rozgrywanych partii. Zauważono, że kolory rozkładają się w miarę równomiernie, co również jest warunkiem poprawnego działania systemu szwajcarskiego. Po tej wnikliwej analizie stwierdzono poprawność działania oraz zakończono pracę nad implementacją algorytmu.

## **7. Podsumowanie i wnioski**

### **7.1. Wynik pracy**

Celem pracy było stworzenie aplikacji mobilnej na urządzenia z systemem Android, która wspomaga przeprowadzenie turnieju szachowego. Zostały spełnione wszystkie postawione wymagania co pozwoliło na osiągnięcie celu. Dzięki temu programowi można było przejść przez cały rozwój stworzenia oprogramowania. Zaczęto od zapoznania się z postawionym problemem. Zainteresowano się przebiegiem oraz przeprowadzeniem turnieju szachowego w warunkach rzeczywistych. Zawody turniejowe mają przebieg zazwyczaj dynamiczny, więc starano się wytworzyć aplikację, która odpowiada potrzebom organizatorów. Następnie zdefiniowano wymagania funkcjonalne oraz нефunkcjonalne. Najważniejsze z nich to wprowadzenie zawodników do aplikacji oraz prawidłowy przebieg turnieju. Programu cechuje niezawodność, dzięki której nie dojdzie do błędów między innymi przy kojarzeniu zawodników. Taka pomyłka miałaby negatywny wpływ na dalszy przebieg turnieju. Program jest obsługiwany przez osoby, które przeprowadzają turnieje szachowe. Powinni być to ludzie, którzy znają podstawowe przepisy gry oraz specyfikę turnieju szachowego. Spora liczba turniejów jest przeprowadzana w sposób dynamiczny, partie trwają kilka minut, zawodnicy są w rytmie meczowym. Dlatego aplikacja jest szybka w obsłudze. Szczególnie wprowadzanie wyników pomiędzy rundami jest intuicyjne oraz łatwe. Dzięki temu turniej przebiegnie sprawnie i szybko, a gracze nie będą musieli zwlekać na rozpoczęcie kolejnego meczu, co niewątpliwie

zaburzyłoby ich koncentrację i przygotowanie do partii. Kolejną cechą aplikacji jest duża możliwość rozbudowy. Przy konfigurowaniu turnieju można dodać wiele różnych możliwości, które uatrakcyjniłyby system. Program można rozbudować o zwiększenie wyboru systemu rozgrywek. Został zaimplementowany najbardziej skomplikowany - szwajcarski. Jednakże dodanie systemów takich jak pucharowy, kołowy czy scheveningen zwiększyłoby możliwości pomysłodawców zawodów przy organizacji i przeprowadzeniu turnieju.

Wraz z upływem czasu wstępny projekt ewoluował jak to często bywa przy wytworzeniu oprogramowania. Jako przykład może posłużyć fakt, że podczas projektowania aplikacji pominięto funkcjonalności takie jak: usuwanie, edytowanie oraz przeglądanie informacji o danym zawodniku, który widnieje w systemie. Dopiero w trakcie implementacji zauważono, że wymienione funkcje są niezbędne. Weryfikacja przygotowanego oprogramowania to kolejny etap, który trzeba było przejść, żeby otrzymać w pełni niezawodną aplikację. Dzięki wnikliwemu testowaniu stwierdzono, że program jest uodporniony i na błędy powodujące złe dane wprowadzone przez użytkownika, i na zagrożenia typu: nieprawidłowe skojarzenie zawodników.

## 7.2. Możliwości rozbudowy aplikacji

Pomimo spełnienia wszystkich wymagań postawionych przed napisaniem pracy możliwa jest dalsza rozbudowa powstałej aplikacji. Przede wszystkim można stworzyć konta dla użytkowników korzystających z aplikacji. Dany klient loguje się na swoje konto gdzie zapisana będzie historia rozegranych turniejów czy przegląd meczów w rozegranych zawodach. Dodatkowo przy wprowadzaniu zawodników można pobierać ranking międzynarodowy z oficjalnej strony Międzynarodowej Organizacji Szachowej, a polski z Polskiego Związku Szachowego. Ponadto istnieje rozbudowa systemów rozgrywek. W turniejach szachowych toczą się zawody w systemie:

kołowym, pucharowym oraz scheveningen. Po ich dodaniu aplikacja zyskałaby na atrakcyjności. Kolejną rozbudową byłaby możliwość tworzenia raportów podczas trwania turnieju. W trakcie zawodów można by wygenerować taki raport z wynikami poprzedniej rundy oraz aktualną klasyfikacją generalną. Powinno to być sporym ułatwieniem dla organizatorów, którzy mogliby np. po każdej rundzie drukować dany raport i wywieszać go w ogólnodostępnym dla zawodników miejscu.

### **7.3. Problemy napotkane w trakcie pracy**

Podczas pisania pracy największym problemem było zrozumienie oraz zaimplementowanie systemu szwajcarskiego. Jest to najbardziej skomplikowany i złożony z systemów, którym rozgrywane są turnieje szachowe. Na początku należało dotrzeć do źródła, które w najdrobniejszych szczegółach opisuje dany system. Znalaziono internetową wersję kodeksu szachowego wydane przez Polski Związek Szachowy w 2007 roku. Znajdują się w nim przepisy gry, regulaminy i system rozgrywek. Opisany w nim system przedstawiony jest w sposób zwięzły i zrozumiały. Dużym plusem jest przedstawienie przykładowego turnieju, który jest rozgrywany w porządku szwajcarskim. Dzięki temu zrozumiano ideę oraz wszystkie zasady konieczne do prawidłowego przeprowadzania zawodów. Po rozpoczęciu implementacji napotkano na problemy w prawidłowym odzwierciedleniu systemu. Do największych należą odwzorowanie kojarzenia w par zawodników pod względem koloru bierka jakim powinni grać w danej rundzie. Stwierdzono, że zostanie wymyślona uproszczona wersja, która w pełni zachowa podstawowe zasady systemu. Do kolejnego problemu można zaliczyć obsługę bazy danych. Sam wybór systemu zarządzania bazą danych nie należał do trudnych. Jednakże sposób łączenia się do niej wymagał poświęcenia sporej ilości czasu. Dodatkowo podjęto próbę wykorzystania biblioteki wspierającą mapowanie obiektowo relacyjne (Room). Tutaj również pracowano na nowym narzędziu i samo zrozumienie go zajęło trochę

czasu. Po zagłębieniu się w temat bazy danych na urządzeniu mobilnym nie napotkano większych problemów. Małą niedogodnością jest brak formatu daty w bibliotece Room. Poradzono sobie w ten sposób, że napisano klasę, która służy do konwertowania typu Date na Long i na odwrót. Do kłopotów, z którym trzeba było się zmierzyć to wpisywanie danych wejściowych bezpośrednio do kodu źródłowego w klasach, które nie reprezentują aktywności. W nich nie można odnieść się do pliku strings.xml, który służy do zdefiniowanych nazw stałych. Istnieją różne obejścia służące do ominięcia tego problemu tworząc klasy udostępniając kontekst aplikacji. Nie jest to ładne rozwiązanie, więc postanowiono napisać osobną klasę, która posiada stałe, statyczne pola z nazwami używane przez klasy spoza aktywności. Jednym z większych problemów było tworzenie wyglądu aktywności przedstawiający przebieg turnieju. Trudność polegała na tym, że wygląd musiał zostać zaprojektowany dynamicznie w zależności od liczby zawodników biorących udział w turnieju. Pierwszym pomysłem było stworzenie jednego elementu, który przedstawiałby jeden mecz i wykorzystanie go tyle razy ile jest meczów. Niestety w środowisku Android Studio nie jest możliwe wykorzystanie danego widoku wewnątrz drugiego widoku więcej niż jeden raz. Postanowiono ominąć ten problem tworząc dynamiczne widoki w kodzie źródłowym.







## Bibliografia

- [1] Jerzy Giżycki, Władysław Litmanowicz. *Szachy od A do Z, tom I*. Wydawnictwo Sport i Turystka, Warszawa, 1986.
- [2] Robert C. Martin. *Clean Code: A Handbook of Agile Software Craftsmanship*. Prentice Hall, 2008.
- [3] *Calendar*.  
<https://www.fide.com/calendar> [data dostępu 2019-11-24]
- [4] *Jak komputer pokonał człowieka w szachy - Nauka - polskieradio.pl*.  
<https://www.polskieradio.pl/23/266/Artykul/843208,Jak-komputer-pokonala-czlowieka-w-szachy>  
[data dostępu 2019-11-24]
- [5] *Kalendarz Szachowy*. <http://www.ks.prv.pl/>  
[data dostępu 2019-11-24]
- [6] *kodeks\_szachowy\_2017.pdf*.  
[http://pliki.pzszach.pl/komisje/kolegium\\_sedziow/pzszach.org.pl/kodeks\\_szachowy\\_2007.pdf](http://pliki.pzszach.pl/komisje/kolegium_sedziow/pzszach.org.pl/kodeks_szachowy_2007.pdf) [data dostępu 2019-12-03]
- [7] *Królestwo szachów: Historia szachów*.  
<http://szachoman.blogspot.com/2009/12/historia-szachow.html>  
[data dostępu 2019-12-03]
- [8] *Polska wersja Fide rules*  
<http://pzszach.pl/wp-content/uploads/2017/05/2017-07-01-Polska-wersja-Przepisow-Gry-FIDE.pdf>  
[data dostępu 2019-12-14]

- [9] *Room Persistence Library / Android Developers* .  
<https://developer.android.com/topic/libraries/architecture/room> [data dostępu 2019-12-03]
  
- [10] *Śląski Związek Szachowy*. <https://szs.org.pl/>  
[data dostępu 2019-12-15]

## Spis skrótów i symboli

<i>FIDE</i>	Międzynarodowa Federacja Szachowa (fr. <i>Fédération Internationale des Échecs</i> )
<i>ORM</i>	Mapowanie obiektowo – relacyjne (ang. <i>Object-Relational Mapping</i> )
<i>API</i>	Interfejs programowania aplikacji (ang. application programming interface)
<i>PZS</i>	Polski Związek Szachowy

## **Zawartość dołączonej płyty**

Na płycie DVD dołączonej do dokumentacji znajdują się następujące materiały:

- praca w formacie pdf,
- źródła programu,
- zbiory danych użyte w eksperymentach.

## Spis rysunków

- Rys.3.1 Diagram przypadków użycia
- Rys.4.1 Wybór zawodników do turnieju
- Rys.4.2 Wyświetlanie informacji o zawodniku
- Rys.4.3 Dodanie zawodnika
- Rys.4.4 Edytowanie zawodnika
- Rys.4.5 Usuwanie zawodnika
- Rys.4.6 Konfiguracja turnieju
- Rys.4.7 Wpisywanie wyników meczów
- Rys.4.8 Przeglądanie wyników poprzednich rund
- Rys.4.9 Przeglądanie wyników po ostatnio zakończonej rundzie
- Rys.4.10 Wyświetlanie wyników końcowych
- Rys.6.1 Wyniki testów jednostkowych