Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Engineering

Bachelor Thesis

Speedcubing cross-platform application

Sanzhar Zhantemirov

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

BACHELOR THESIS ASSIGNMENT

Sanzhar Zhantemirov

Informatics

Thesis title

Speedcubing cross-platform application

Objectives of thesis

The aim of the thesis is to design and implement a cross-platform application for speedcubing. The application will be available on PC, Android smartphones and webpage. Using the application, a user can measure its own time of solving a Rubik's cube or specific algorithm, learn new algorithms and analyze session's statistics including graph. Users will be able to create their accounts in order to save and display their statistics.

Methodology

The thesis consists of two parts – theoretical and practical. The theoretical part of the thesis is based on a study and review of various information sources. Based on the synthesis of the gained knowledge the groundwork for the practical part is defined.

The practical part of the thesis consists of design and implementation of a cross-platform speedcubing application. The application will be deployed, tested and based on the experience from its development and testing feedback the conclusion will be formulated and possible future development options will be outlined. The standard means and tools of the software engineering will be utilized during the process.

The proposed extent of the thesis

35-40 pages

Keywords

Java, C#, JavaScript, Android, PC, Webpage

Recommended information sources

https://developer.android.com/studio/intro
https://firebase.google.com/community/learn
https://www.javatpoint.com/unity-variables-and-functions
https://www.w3schools.com/cs/index.php
https://www.w3schools.com/java/
https://www.w3schools.com/js/default.asp

Expected date of thesis defence

2022/23 SS – FEM

The Bachelor Thesis Supervisor

Ing. Jiří Brožek, Ph.D.

Supervising department

Department of Information Engineering

Electronic approval: 31. 10. 2022

Ing. Martin Pelikán, Ph.D.

Head of department

Electronic approval: 30. 11. 2022

doc. Ing. Tomáš Šubrt, Ph.D.

Dean

Prague on 08. 03. 2023

Declaration

I declare that I have worked on my bachelor thesis titled "Speedcubing cross-platform application" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the bachelor thesis, I declare that the thesis does not break any copyrights.

In Prague on 15.03.2023   _____

Speedcubing cross-platform application

Abstract

The main objective of the thesis is to develop a cross-platform application for speedcubing that will be available on PC, Mobile and Web. Speedcubing is an activity of solving a Rubik's cube as quickly as possible that gained in popularity during the pandemic. The activity requires learning algorithms to solve a cube and a timer to measure solving times.

In order to have a market overview, applications for all three platforms have to be reviewed. Subsequently, the choice of what software, programming language, database and a speedcubing method to use is based on their benefits and drawbacks.

After literature review, the user research is conducted, followed by an application prototype that contains requirements, state machine diagram, use cases and wireframes. Afterwards, the application is developed and released to three platforms.

In conclusion, a further development of the application is defined accordingly to a feedback received from users.

Keywords: Speedcubing, Application, PC, Mobile, Web, Unity Engine, Visual Studio, Firebase.

Multiplatformní aplikace Speedcubing

Abstrakt

Hlavním cílem práce je vyvinout multiplatformní aplikaci pro speedcubing, která bude dostupná na PC, mobilu a webu. Speedcubing je činnost spočívající v co nejrychlejším vyřešení Rubikovy kostky, která si během pandemie získala na popularitě. Aktivita vyžaduje učení algoritmů pro vyřešení krychle a časovač pro měření času řešení.

Pro získání přehledu o trhu je proveden průzkum aplikací pro všechny tři platformy. Následně jsou na základě jejich výhod a nevýhod vybrány programovací jazyk, databáze a využité metody speedcubingu.

Po revizi literatury je proveden uživatelský průzkum, následovaný prototypem aplikace, který obsahuje požadavky, schéma stavového stroje, případy použití a drátové modely. Poté je aplikace vyvinuta a vydána na třech platformách.

Na závěr je na základě zpětné vazby obdržené od uživatelů definován další vývoj aplikace.

Klíčová slova: Speedcubing, Aplikace, PC, Mobilní, Web, Unity Engine, Visual Studio, Firebase.

Table of content

1 Introduction

A Rubik's Cube (1), also known as a Magic cube, is a puzzle toy that was invented in 1974 by Hungarian sculptor and professor of architecture Ernő Rubik. The Rubik's Cube has six faces, each consisting of nine smaller squares in a 3x3 grid. Each face is coloured, typically with white, yellow, blue, green, red, and orange, and to solve a Rubik's cube means to twist and turn the cube so each side can have only one colour.

Figure 1 - Erno Rubik and a Rubik's cube. Source: (1)

The puzzle has many variations, including larger and smaller cubes of different shapes. Most popular are 2x2, 3x3, 4x4, 5x5, 6x6, 7x7, Square-1, Megaminx, Clock and Pyraminx.

The Rubik's Cube has become one of the most popular and recognizable puzzles in the world, which won the "Toy of the Year" award in 1980 and 1981, with total of over 450 million of units sold worldwide.

The subculture speedcubing caused people from over the world to create different methods and practice their algorithms in order to solve the puzzle as quickly as possible. It requires an excellent hand-eye coordination, spatial awareness and finger dexterity. Speedcubing contribute to improving cognitive skills, patience and persistence, encouraging socialization and building positive competition. According to World Cube Association (2), more than 150,000 people participated in more than 8,300 competitions. Some of the top speedcubers can solve a Rubik's Cube in less than 5 seconds, which is truly amazing to watch.

Every speedcuber uses a timer to measure their solving times and review their statistics afterwards. These timers are available on mobile, desktop and web platforms.

2 Objectives and Methodology

2.1 Objectives

The main goal of the thesis is to develop a modern speedcubing application that will be available on Desktop, Mobile and Web platforms. The partial goals are:

• To make an analysis of speedcubing and its trends;

• To analyse speedcubing timer applications for Desktop, Mobile and Web platforms;

• To choose what software, programming language, database and solving method to use in order to develop the application;

• To conduct a user research;

• To write requirements and create state machine diagram, use cases and wireframes;

• To develop and release the cross-platform application for speedcubing;

• To interpret results and draw a conclusion.

2.2 Methodology

The theoretical part of the thesis is based on the study of professional materials, forums of speedcubing and personal experience. Nevertheless, it contains a comparison of advantages and disadvantages of available applications on the market, software and speedcubing methods.

The practical part includes a user research conducted using online surveys and discussions. Based on that, user personas and their preferences are defined. Moreover, the prototype of the application involves writing the requirements, drawing state machine diagram and wireframes. Development of the User Interface of the application and writing scripts are based on personal experience, coding books and online documentations. A feedback from users is collected using online rating and survey followed by further development.

By accumulating knowledge and acquiring experience from theoretical and practical parts of the thesis, conclusion is derived.

3 Literature Review

3.1 What is Speedcubing?

Speedcubing is an activity of solving a Rubik's cube as quickly as possible. It contains a variety of categories: 2x2, 3x3, 4x4, 5x5, 6x6, 7x7, 3x3 one-handed, 3x3-blindfolded, skewb, clock, pyraminx, square-1 and others.

People from all over the world participate in all categories and try to break current records. Speedcubing competitions are organized by World Cube Association. According to WCA website (2), there are 169459 registered speedcubers, who participate in 17 official categories.

On official competitions, cubes are scrambled by computer randomly generated algorithm, so each competitor has equal chances to solve a cube in the shortest time possible.

The table below contains current world records (3):

Category	Type	Result (in seconds)	Name	Citizen of	Year
2x2x2	Single	0.49	Maciej Czapiewski	Poland	2016
	Average of 5	1.02	Zayn Khanani	United States	2022
3x3x3	Single	3.47	Yusheng Du	China	2018
	Average of 5	4.86	Tymon Kolasiński	Poland	2022
4x4x4	Single	16.79	Max Park	United States	2022
	Average of 5	19.88	Max Park	United States	2022
5x5x5	Single	33.02	Max Park	United States	2022
	Average of 5	38.42	Max Park	United States	2022
6x6x6	Single	59.74	Max Park	United States	2022
	Average of 5	1:09.23	Max Park	United States	2022
7x7x7	Single	1:40.89	Max Park	United States	2019
	Average of 5	1:46.57	Max Park	United States	2020
3x3x3 Blindfolded	Single	14.51	Tommy Cherry	United States	2022
	Average of 5	15.24	Tommy Cherry	United States	2021
3x3x3 One-handed	Single	6.20	Max Park	United States	2022
	Average of 5	8.65	Patrick Ponce	United States	2022
Clock	Single	2.87	Yunhao Lou	China	2021
	Average of 5	3.81	Caleb Trelford	United States	2022
Megaminx	Single	25.24	Juan Pablo Huanqui	Peru	2022
	Average of 5	29.27	Leandro Martín López	Argentina	2022
Pyraminx	Single	0.91	Dominik Górny	Poland	2018
	Average of 5	1.66	Jasper Murray	New Zealand	2022
Skewb	Single	0.81	Zayn Khanani	United States	2022
	Average of 5	1.56	Zayn Khanani	United States	2022

Table 1 - Speedcubing World Records. Source: (3)

In 2020, Netflix produced a documentary called "The Speed Cubers" that describes a rivalry between two world champions and friends, Feliks Zemdegs and Max Park (4).

In 2020, during a pandemic, popularity of Speedcubing started to gain momentum. People from over the world were buying different Rubik's cubes and learning how to solve it (5).

In 2021, Red Bull organized World Cup championship (6). The feature of this tournament is that all the speedcubers competed online and it last for 5 days. Max Park, Tymon Kolasinski and Patrick Ponce shared the podium.

Nowadays, speedcubing has become very popular and more people try to break current world records. To do so, they need to create their own or learn existing solving methods.

3.1.1 Solving methods and cube notation

Solving methods contain several algorithms for specific situations. The most popular methods to solve a classic 3x3 Rubik's cube are:

• CFOP/Fridrich Method;

• Roux;

• Petrus.

They will be fully described in **Section "3.9 Speedcubing methods".**

An algorithm (7) can be represented as a sequence of different letters. For example, an algorithm called "OLL 1" looks like "R U B' R B R2 U' R' F R F'" and an algorithm called "OLL 23" looks like "L2 D L' U2 L D' L' U2 L'", where: F - front, B – back, R – right, L – left, U – up and D – down. A letter by itself means a 90-degree clockwise rotation of the side. A letter followed by an apostrophe, also called prime, is a counterclockwise turn.
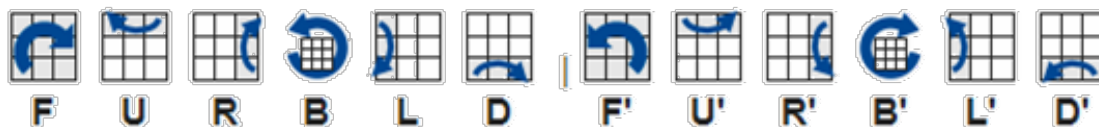

Figure 2 - Rubik's cube algorithm notations. Source: (7)

3.1.2 Positive impact of Speedcubing

Speedcubing is a popular and intellectual activity that has several positive impacts such as:

1. Cognitive Development: Speedcubing can improve cognitive skills such as spatial awareness, memory, problem solving, fine motor skills (8) and decision-making. Solving a Rubik's Cube requires individuals to analyse and manipulate spatial relationships, patterns, and algorithms. In order to solve a 3x3x3 cube in less than 10 seconds using CFOP method, a speedcuber needs to practice more than 100 main algorithms. Each algorithm has more than 3 variations depending which one suits the current situation best;

2. Patience and Persistence: Learning how to solve a Rubik's Cube can be a challenging and frustrating experience, but speedcubing requires patience and persistence to become proficient. This can help individuals develop resilience, persistence, and determination in achieving their goals;

3. Socialization: Speedcubing events and competitions provide opportunities for individuals to socialize with people with similar hobbies, share their experiences, and have new friends from different cultures and backgrounds;

4. Positive Competition: Speedcubing competitions provide opportunities for individuals to compete in a positive and supportive environment. These events often emphasize sportsmanship, fairness, and camaraderie, rather than just winning.

3.2 Current available Android speedcubing applications

After the speedcubing and its trends were defined, current market of applications for three different platforms has to be analysed in order to determine what functions need to be included in my application. These three main platforms are mobile, desktop and web.

The vast majority of speedcubers use applications on their smartphones, so they can measure their solving times at home, office or school. Mobile applications do not require internet connection and keyboard. Two the most popular operating systems for mobile devices are iOS and Android.

The pie chart (9) below represents that Android OS has roughly 2.5 times more users than iOS, therefore the application should focus on Android market first. Google Play (10) is a main online store of Android applications and it needs to be analysed.

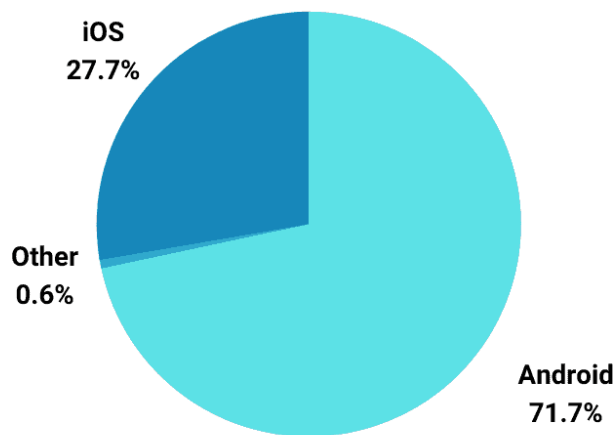Figure 3 - "Which Mobile OS Has the Most Users Worldwide?". Source: (9)

3.2.1 Android application 1: Finger Timer

The most popular application for Android OS is called Finger Timer (11) that was downloaded more than 1,000,000 times. Its interface looks like a real timer used on official speedcubing competitions. Omega Studio released the application on April 5th, 2013 and last updated it on December 10th, 2018.
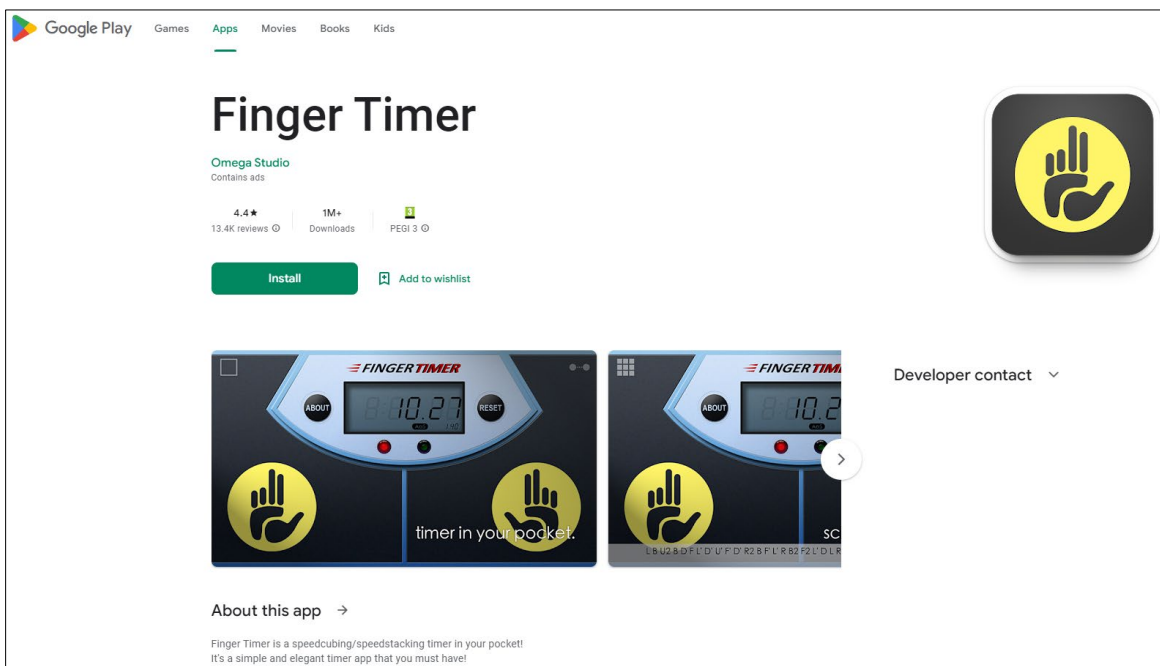

Figure 4 - Finger Timer. Source (11)

3.2.2   Android application 2: SpeedCube Timer – Rubiks Chrono

The second most popular android application for speedcubing is called SpeedCube Timer – Rubiks Chrono (12) that was downloaded by more than 500,000 users. It is developed by Pelayo Rodríguez and its first version was released on June 5th, 2016 and last updated on April 29th, 2022.
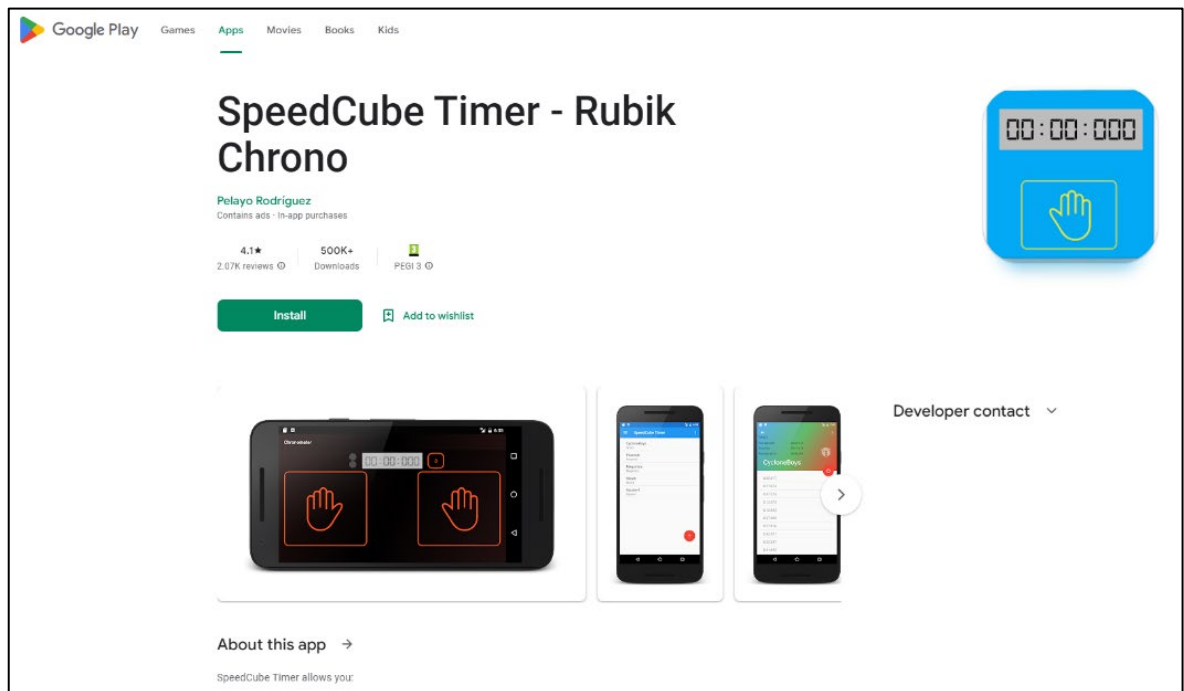

Figure 5 - Rubik Chrono. Source: (12)

3.2.3   Android application 3: Cube Timer

Next application is called Cube Timer (13) that was developed by Pilow and has more than 500,000 downloads. Its first release was on April 14th, 2013 and last update was on December 30th, 2022.
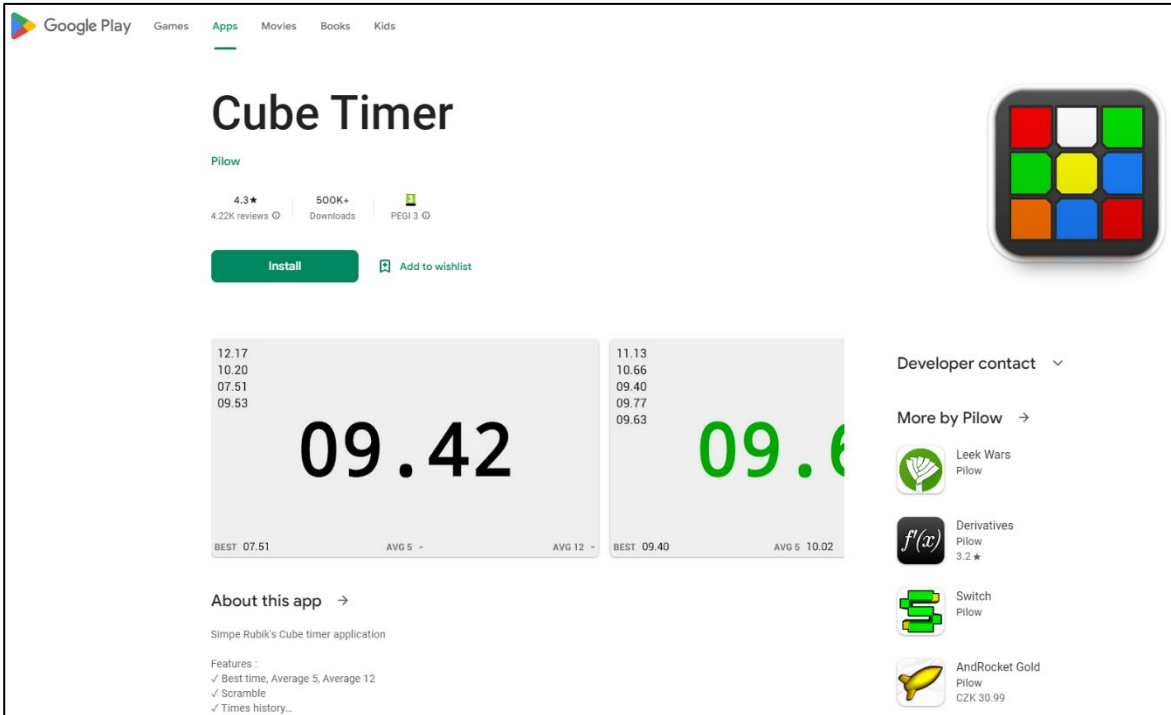
Figure 6 - Cube Timer. Source: (13)

3.2.4 Android application 4: Nano Timer

Nano Timer is an application that closes the list. Nicolas Aubinet developed the application (14) on August, 27th, 2014 that was downloaded by more than 50,000 users. Nano Timer received its last update on March 8th, 2020.
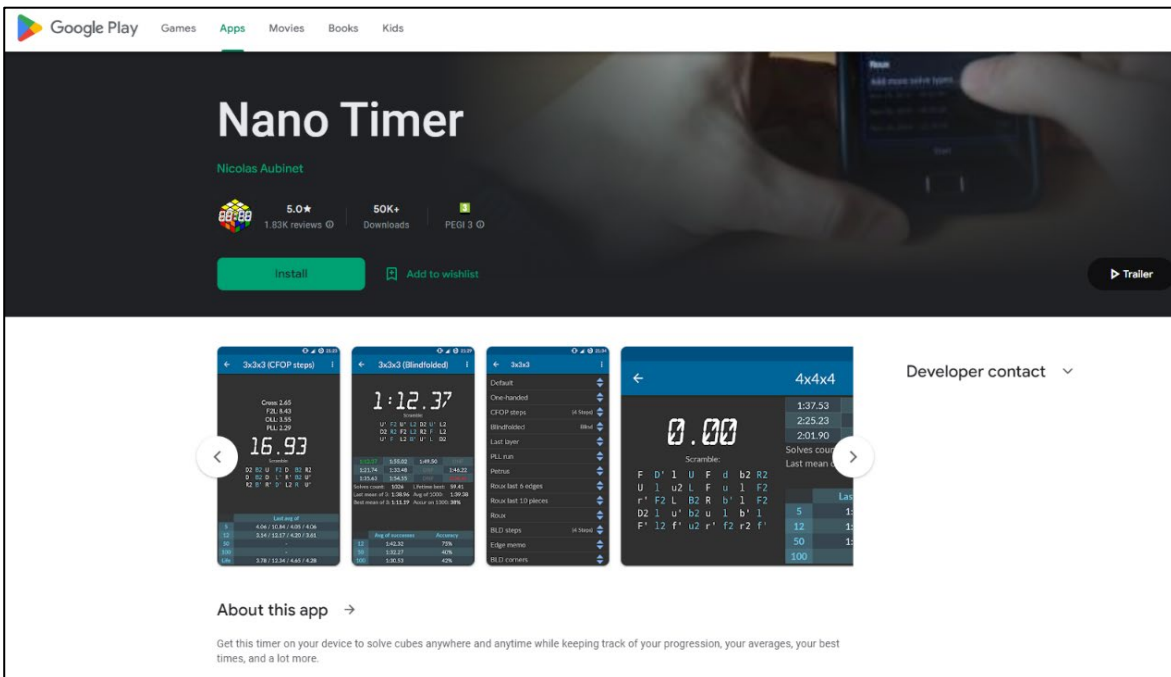

Figure 7 - Nano Timer. Source: (14)

3.2.5   Android applications comparison table

The table below demonstrates benefits and drawbacks of four the most popular applications for Android OS.

Application	Benefits	Drawbacks
Finger Timer	• Very good interface of real competition timer; • Possibility to use web browser as a remote display.	• Does not save statistics; • Does not show graph so user does not see their improvement; • No scramble formula, so a user has to scramble the cube on their own.
Speedcube Timer – Rubik Chrono	• Saves solving time.	• Very simple statistics; • Does not show graph so user does not see their improvement; • Very basic interface; • No scramble so a user has to scramble the cube on their own.
Cube Timer	• Saves solving time.	• Very basic interface; • Very simple statistics; • No scramble formula; • Does not show graph.
Nano Timer	• Contains scramble formula; • Saves solving time.	• Simple statistics; • Simple interface; • Does not show graph so a user does not see their improvement over time.

Table 2 - Android speedcubing applications benefits and drawbacks. Source: author

In conclusion, three applications save solving time and some of them show a scramble formula, which are essential for any type of speedcubing timer. On the other hand, their main drawbacks are their simple statistics, simple interface and no graph representation. They are supposed to only measure time and view the statistics, but a new functionality could be added – displaying tutorial algorithms.

3.3   Current available online speedcubing timers

A majority of speedcubers use online timers on different websites. It requires Internet connection and can be accessed from either PC, tablet or smartphone.

3.3.1   Online timer 1: qqtimer.net

For example, world champion Feliks Zemdegs used a timer on qqtimer.net (15) because it offers a good statistics, shows solving times and stores a session that can be resumed at any time. On the other hand, it does not plot a graph, so users cannot track their improvement.
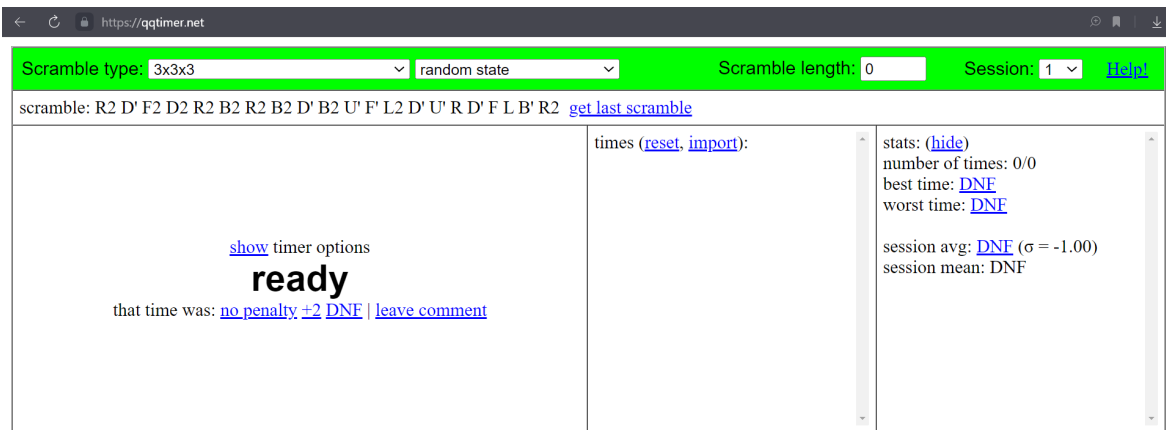

Figure 8 - qqtimer.net. Source: (15)

3.3.2   Online timer 2: cstimer.net

Another popular web timer is called csTimer (16) that was developed by Shuang Chen and Yue Zhang.
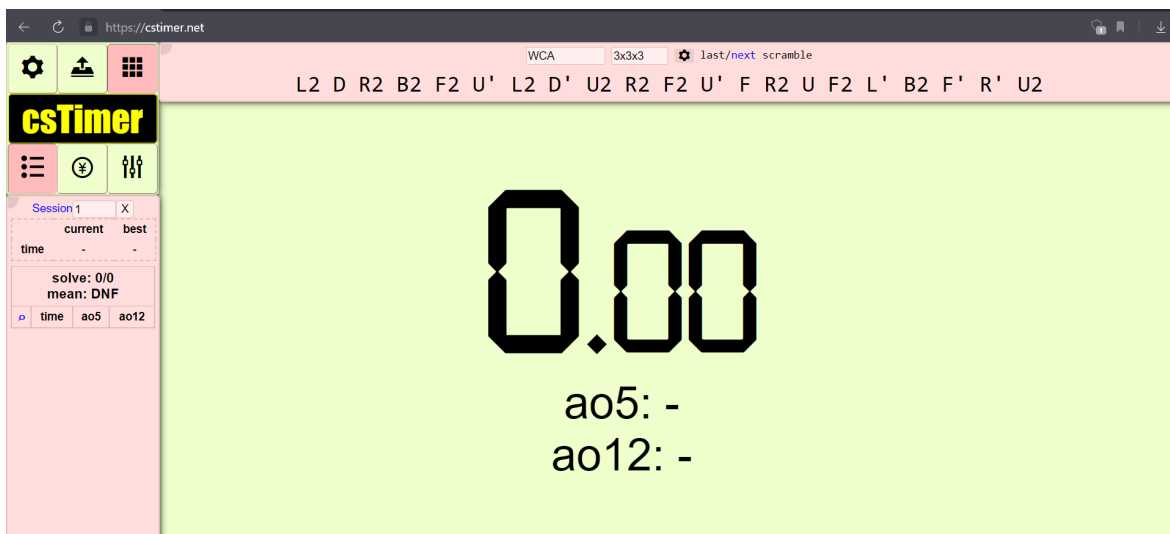

Figure 9 - cstimer.net. Source: (16)

3.3.3 Online timer 3: ruwix.com

Next timer (17) offers decent statistics with a graph, but its interface is average and its elements do not look as a whole.
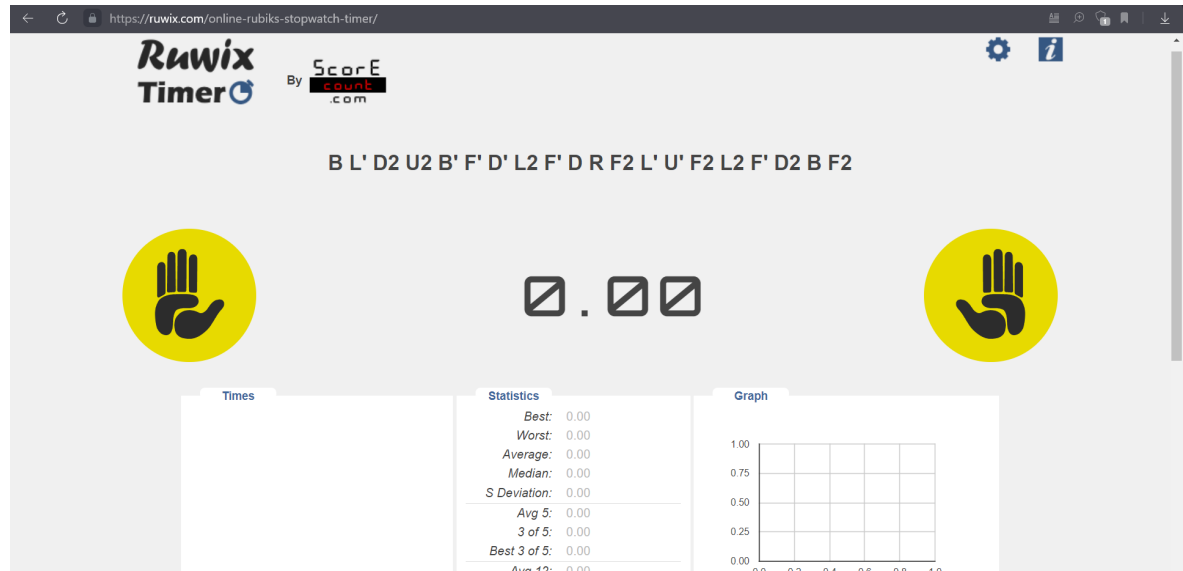

Figure 10 - ruwix.com. Source: (17)

3.3.4 Online timer 4: solvethecube.com

Last web timer (18) contains a page of algorithms that a user could learn, but at the same time there is no detailed statistics and its interface is not user-friendly.
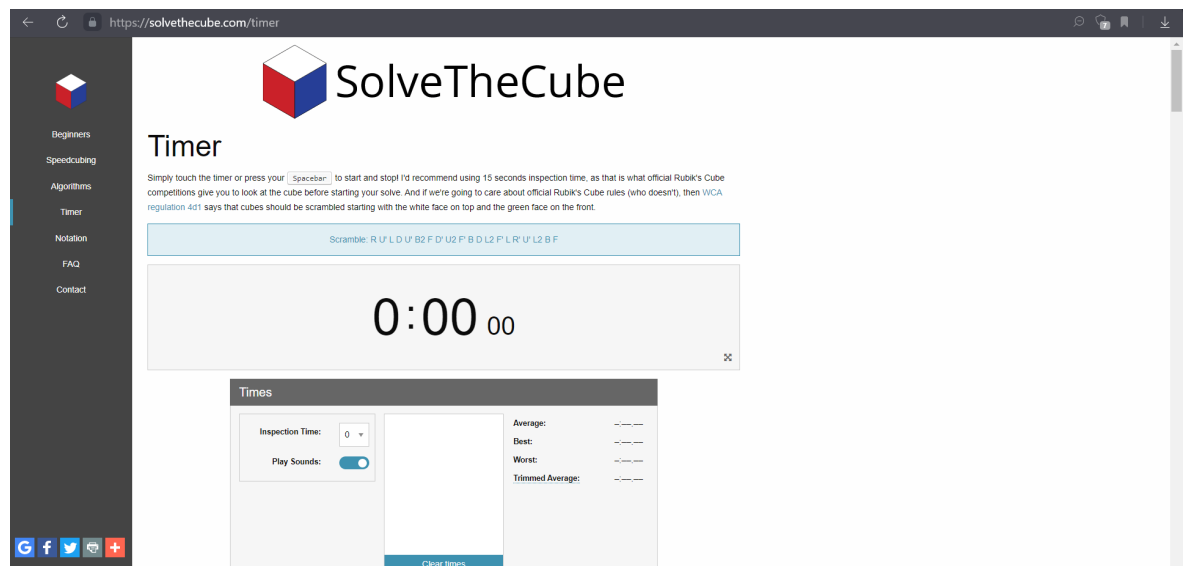

Figure 11 - solvethecube.com. Source: (18)

3.3.5   Online timers comparison table

The table below gives information about advantages and disadvantages of four online timers for speedcubing.

Name	Advantages	Disadvantages
qqtimer	• Very good statistics; • Contains scramble formula.	• Simple interface; • No graph representation.
csTimer	• Contains scramble formula;	• Simple statistics; • No graph representation; • Simple interface.
Ruwix Timer	• Very good statistics; • Contains graph representation; • Contains scramble formula.	• Medium interface.
SolveTheCube	• Contains scramble formula; • Contains graph representation; • Contains a tab to learn new algorithms.	• Simple interface; • Very simple statistics.

Table 3 - Online speedcubing timers' advantages and disadvantages. Source: author

Overall, all four online timers for speedcubing have simple interface and three of them do not include algorithms that a speedcuber could learn. Two of them do not have a good statistics and a graph representation of solving times.

3.4   Current available desktop speedcubing application

Since the application is planned to be cross-platform, a desktop application need to be considered as well. In 2017, software engineer Dallas McNeil from Adelaide, South Australia, developed an application called Block Keeper (19), which is available on Linux, Mac OS and Windows OS.
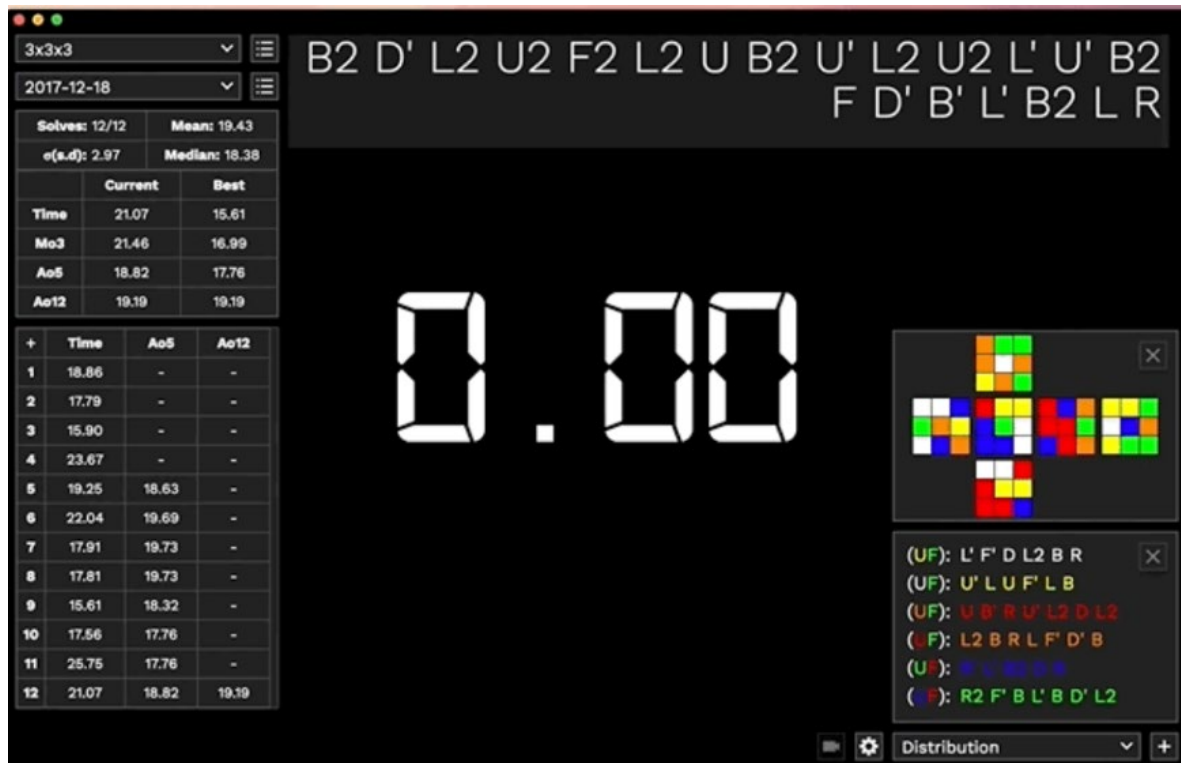
Figure 12 - Block Keeper. Source: (19)

The app has a major advantage of decent statistics and it is available for Windows, Linux and Mac. On the other hand, its disadvantages are lack of graph, algorithms, and its interface.

3.5 Conclusion on comparison

After analyzing all nine speedcubing timers for desktop, mobile and web platforms, my application should include:

- User-friendly interface;
- Scramble formula;
- Good statistics;
- Graph representation of solving times;
- Algorithms and their visual representation.

Moreover, one function of saving and loading users' results is planned to be implemented using database, so users of the application can see who solves a Rubik's cube quicker:

- User data input.

3.6 Software to use in the application

At this stage, what software to use has to be decided. The most famous software to create an application for Android OS is called Android Studio (20) developed by Google in 2013. Scripts for web version could be written in Visual Studio Code (21) or Atom Code Editor (22). Functionalities of the application on different platforms have different approach. In order to have the application as similar as possible on mobile, desktop and web versions, game engines can be used, because some of them offer functionalities to develop one same product on different platforms.

Nowadays, the most popular development engines are Unity and Unreal.

3.6.1 Unity Engine

Unity (23) is a cross-platform game engine and development platform that allows developers to create 2D and 3D games, simulations, and other applications. Unity was first released in 2005 by Unity Technologies and has since become one of the most popular game engines in the industry, making games available on a wide range of platforms, including desktop, mobile, consoles, virtual reality and web (24).

The engine provides a wide range of tools and features for developers, including an editor with user-friendly interface, writing scripts on C#, physics simulation, and asset management. This makes it easier for developers to create games and interactive content without having to spend a lot of time on low-level tasks.

Figure 13 - Unity Engine logo. Source: (24)

3.6.2 Unreal Engine

Unity Engine's alternative is Unreal Engine (25). It is a powerful game engine and development platform that was created by Epic Games in 1998. Its fifth version demonstrated outstanding capabilities on 'The Matrix Awakens' presentation (26). This engine allows development an application across several platforms including Windows, Android, iOS, web and consoles.

Nowadays, many big budget games such as Fortnite, Gears of War 4, Rocket League, Atomic Heart and Assetto Corsa Competizione were developed on Unreal Engine.

One of the key features of Unreal Engine is its graphical capabilities. It uses advanced rendering techniques, such as physically-based rendering, to create realistic and immersive visuals. It also provides a wide range of tools for creating and editing game environments, including terrain, foliage, and lighting.

In addition to its graphical and scripting capabilities, Unreal Engine also provides a range of tools for creating multiplayer games and networking functionality.

Figure 14 - Unreal Engine logo. Source: (25)

3.6.3   Software comparison table

The table below represents pros and cons of Unity and Unreal Engines (27):

Engine Name	Pros	Cons
Unity Engine	• The engine's cross-platform integration is significantly better. It offers to build a finished product across all major mobile, VR, desktop, console platforms and plus the Web. It supports Android, iOS, Windows Phone, Tizen, Windows OS, Mac, Linux, PS4, Xbox One, PlayStation Vita, Wii U and others. Moreover, it supports Playstation VR, Steam VR, Microsoft HoloLens and Gear VR platforms; • It's simpler to create mobile applications and simple games with Unity; • The Unity Asset Store provides access to thousands of pre-built assets and tools, including 3D models, textures, audio, and scripts, which can save	• Unity Engine is not able to offer the quality of graphics as good as of Unreal Engine; • Rendering process is slower. It uses more memory and resources of a computer.

	developers a lot of time and effort;	
	• Development process is easier, because C# programming language is used in writing scripts;	
	• Unity has a large and active community of developers, who share knowledge and resources through forums, tutorials, and online courses.	
Unreal Engine	• Unreal Engine helps to achieve the ultimate quality, especially in terms of graphics, therefore it is one of the best engines to be used in developing big games with huge budgets;	• The asset uploaded to the Unity store might not be accepted to the Unreal store, because developers choose the submitted assets very carefully;
	• Unreal Engine also provides a powerful visual scripting system called Blueprint. It offers to create game logic and functionality without having to write code, making it easier for non-programmers. On the other hand, in order to complete complex tasks, a developer needs to use C++;	• It's not the best tool for small games, because it is mainly focused on developing games with bug budget;
	• Unreal is open-source engine, whose source code available for use or modification. It makes development more efficient and easier;	• In Unreal community there are not so many experts that can help beginners, because the engine is chosen by very specialized teams.
	• Post processing is really fast, hence its rendering technology is an advantage. Moreover, the engine uses less memory and resources which results a good performance;	
	• Both Unity and Unreal engines have good VR integration.	

Table 4 - Unity and Unreal Engines' pros and cons. Source: author

In conclusion, a speedcubing application that is to be created does not require good graphics. During Programming on C# and Component-based SW Development courses in Czech University of Life Sciences Prague, students learned how to code on C#. Based on previous personal experience in developing games in Unity and gained knowledge in programming on C#, Unity Engine will be perfect choice in order to develop a cross-platform application.

3.7   Programming language to use in the application

Unity Engine scripts are written on C# and this programming language was studied in university, hence it will be chosen for development the application.
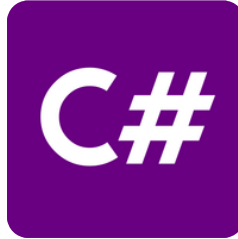

Figure 15 - C# Programming language logo (28)

C# (29) also known as CSharp, is a high-level object-oriented programming language created by Microsoft Corporation in 2000. The language runs on the .NET Framework. It allows programmers to create cross-platform applications including Windows client apps, Web applications, Mobile apps, video games, cloud applications, components, libraries, services and APIs (30).

One of the main characteristics of C# is its system of managing memory. Garbage collector is used to automatically manage memory allocation and deallocation. This feature makes C# more convenient to use than lower-level languages like C and C++, which require manual memory management (31).

C# also has a rich set of frameworks and libraries that make it easier to develop applications. The .NET Framework, for example, provides a wide range of classes and libraries, while the .NET Core framework allows developers to build cross-platform applications ((32), p.593).

CSharp supports modern programming constructs such as lambda expressions, LINQ (Language-Integrated Query), and async/await. These features make it easier to write expressive, concise and efficient code ((33), p.152).

Generic types and methods provide increased type safety and performance. Moreover, C# supports iterators, which enable implementers of collection classes to decide a behavior of a code (34).

Sample code below represents a loop that iterates over an array of numbers and calculates their sum.

```csharp
using System;

  class Progam
  {
    static void Main(string[] args)
    {
      int[] numbers = [1,2,3,4,5];
      int sum = 0;

      for(int i = 0; i <numbers.Length; i++)
      {
        sum += number[i];
      }
      Console.WriteLine("The sum of the numbers is " + sum);
    }
  }
```

Source code 1 - Sample code of C#. Source: author

3.8 Database to use in the application

As mentioned in chapter "**3.5 Conclusion on comparison**" the application will include the function where users will be able to store and share their statistical data with other users. In order to make this function available, database needs to be set. Firebase real-time database (35) is well-known for its functionality of storing and synchronizing data between users. It is a cloud-based NoSQL database powered by Google Firebase. It supports Unity platforms, Android and Web, which means a perfect match for the speedcubing application.

3.9 Speedcubing methods

After analysing technical part, the solving method has to be chosen, so its algorithms could be displayed in the application. There are different Rubik's cube solving methods created to maximise turning speed, lower a number of moves, and consequently reduce solving time.

Most popular methods among speedcubers are:
• CFOP/Fridrich Method;
• Roux;
• Petrus.

3.9.1 CFOP/Fridrich Method

CFOP (Cross, F2L, OLL, PLL) is the most popular method proposed by Jessica Fridrich in 1981 for solving a Rubik's Cube (36). It is an advanced method that is commonly used by speedcubers to solve the cube quickly and efficiently. The method contains 119 algorithms and a cube can be solved in approximately 60 moves. Here is a brief overview of each step of the CFOP method:

1. Cross: The first step is to solve the first layer of the cube by creating a cross on one face. This involves selecting a colour to start with, finding and positioning the edge pieces that match that colour, and then rotating the cube to align the edge pieces with the centre pieces of the adjacent faces;

Figure 16 - CFOP Cross. Source: (36)

2. F2L (First Two Layers): Once the cross is complete, the next step is to solve the first two layers of the cube together. This involves pairing up the corner and edge pieces that belong together, and then inserting them into the correct positions using a series of algorithms;

Figure 17 - CFOP F2L. Source: (36)

3. OLL (Orientation of the Last Layer): Once the first two layers are complete, the next step is to orient the last layer of the cube. In this step, the goal is to make the top face one colour;

Figure 18 - CFOP OLL. Source: (36)

4. PLL (Permutation of the Last Layer): The final step is to permute the last layer of the cube. This involves using a set of algorithms to move the pieces into their correct positions and orientations, resulting in a solved cube.

Figure 19 - CFOP PLL. Source: (36)

3.9.2 Petrus Method

Unlike CFOP, which focuses on solving the cube layer by layer, Petrus Method (37) focuses on a block-building approach, where the first two layers are solved intuitively with no algorithms. The method was created by Lars Petrus in 1981 and it is a second the most popular method for solving a Rubik's cube. It includes 493 algorithms and approximately up to 45 moves are needed to solve a 3x3 cube. Steps of the algorithm are:

1. 2x2x2: The first step is to build a 2x2x2 block anywhere on the cube;

Figure 20 - Petrus 2x2x2. Source: (37)

2. 2x2x3: After 2x2x2 block is built, then next step is to expand it to the 2x2x3 block;

3. F2L: Once previous step is completed, the next step requires to solve edges that are left in order to have first two layers solved;

4. LL: The last step is to solve the Last Layer using any LL approach such as PLL (from CFOP), ZBLL, OCLL and others.

3.9.3 Roux Method

The Roux method (38) is a third popular method for solving the Rubik's Cube that was invented by French speedcuber Gilles Roux in 2003. Roux is a block-building method that involves creating two 1x3 blocks on opposite sides of the cube, and then using a series of moves to solve the remaining pieces. The method contains 42 algorithms and requires from 28 to 48 moves to solve a 3x3 cube. It has next steps to complete:

1. First Block: The first step is to build a 1x3 block on one side of the cube using one corner and two edges. This block should be placed so that the corner is in the correct position and orientation, and the two edges are aligned with the corresponding centre pieces on the adjacent faces;

Figure 21 - Roux First Block. Source: (38)

2. Second Block: Once the first block is complete, the next step is to build a second 1x3 block on the opposite side of the cube using the remaining corner and edges;

Figure 22 - Roux Second Block. Source: (38)

3. CMLL (Corners and Middles Last Layer): With both blocks complete, the next step is to orient and permute the remaining corners and middle-edge pieces on the last layer of the cube. This is done using a series of algorithms to solve the corners and edges in two steps;

Figure 23 - Roux CMLL. Source: (38)

4. LSE (Last Six Edges): The final step is to solve the remaining six edges using a series of moves that preserve the orientation of the corners and the solved blocks. This is often done intuitively, rather than using a set of algorithms.

Figure 24 - Roux LSE. Source: (38)

3.9.4 Speedcubing methods comparison table

Table below represents advantages and disadvantages of three the most popular speedsolving methods (39):

Method	Benefits	Drawbacks
CFOP	• Since the method transitions from the beginner's method, also known as layer-by-layer method, it is easier to learn. It requires to understand how to solve first two layers intuitively and then to solve the third layer using formulas. It is more straightforward than blockbuilding stage in other methods; • CFOP method is used by most of professional and amateur speedcubers for many years, hence there are more researches and computer generated variations of algorithms for specific cases. There are more tutorials and mentors who can provide assistance and give advices; • Statistically, most of 3x3 world records were set using CFOP method.	• CFOP method requires more moves to solve a cube than other methods; • A speedcuber needs to learn 119 different algorithms only for the last layer.

Petrus and Roux	• The Petrus and Roux methods can be used in "Fewest moves" discipline as they focus on solving a cube with the least number of moves possible; • Using PLL, ZBLL, OCLL and other options, methods can help to solve the layer in one algorithm; • They are great methods for "3x3x3 One-Handed" discipline as they can be adapted for movements of right and upper faces.	• For beginners it might be difficult to get better using Petrus Method, because its concept differs from layer-by-layer method; • Blockbuilding approach does not let a speedcuber to use fingertricks to achieve a high value of turns per second.

Table 5 - Speedcubing methods comparison. Source: author

3.9.5 Speedcubing method to use in the application

CFOP method has more significant advantages and it is very suitable for the application, because either amateurs or beginners can open the tab of algorithms, learn new variations and improve their speedcubing results.

Creator of this method, Jessica Fridrich, was born in Czechoslovakia in 1963 and at 17 she started creating her own method of solving a Rubik's cube.

Figure 25 - Jessica Fridrich (40)

"How does it work inside? How can you turn every side and it still holds together? How do you solve it? It's a challenge. It's one of the few puzzles where you don't have to
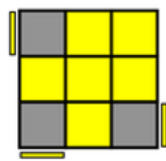
tell anyone any rules. You take the cube, mix it up and everybody knows what their task is. You actually formulate your own rules." – says Jessica Fridrich (40).
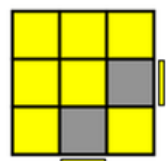
Jessica uploaded her method to website of speedcube enthusiasts and it became very popular among them due to its simplicity and effectiveness. Nowadays, she teaches students at Binghamton University as a professor of the Department of Electrical and Computer Engineering.

Here are the examples of her method's algorithms used in solving the last layer of a Rubik's cube:

OLL - Orientating Last Layer (41)	*PLL – Permutation Last Layer (41)*
R U2 R' U' R U' R'	(x) R' U R' D2 R U' R' D2 R2
(r U R' U') M (U R U' R')	R U R' U' R' F R2 U' R' U' R U R' F'
(R' U' F) (U R U' R') F' R	R' U R U' R2 F' U' F U R F R' F' R2 U'
(R U R' U) (R U' R' U') (R' F R F')	R' U' R U' L R U2 R' U' R U2 L' U R2 U R
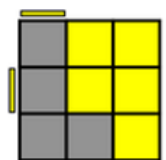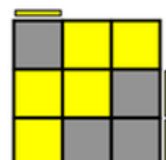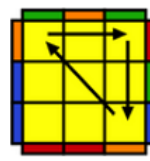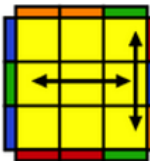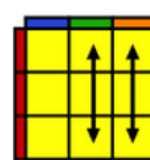
Figure 26 - OLL and PLL algorithms. Source: (41)

3.10 Literature review summary

In the literature review part, an activity and a popularity of speedcubing were described. Moreover, strengths and weaknesses of applications on all three platforms were analysed, so my application could have as less weak sides as possible but more strong sides. Nevertheless, a main cross-platform engine was chosen, consequently it helped to choose a programming language to write scripts. Three the most popular methods of solving a Rubik's cube were compared and the most suitable was selected following by providing examples of its algorithms. This part was written using academic articles, coding books, online documentations, online forums and personal experience.

At this stage, after gaining needed knowledge and having personal experience, development of the application can be started and will be described in practical part.

4 Practical Part

The Practical part contains user research, usecases and scenarios, requirements, state machine diagram, development of the application and its release, a feedback from users and a conclusion.

The application is planned to be used by speedcubers from all over the world and include next functions of: measuring solving time, collecting and creating statistics, plotting a graph, containing algorithms of CFOP method, registering or logging into the system, uploading statistical data and displaying other users' results.

4.1 User Research

4.1.1 Survey and analysis

In order to define target audience and what type of timer they use, online survey was created and published on different Reddit communities such as "r/Cubers", "r/Rubiks_Cubes", "r/cubing", Facebook groups like "Rubik's cube (Speedcubing)", "Speedcubing CZ/SK" and "SpeedCubing Community". Survey includes three questions of multiple-choice type and one question of Fill in the Blank type. All answers are anonymous due to privacy reasons.

Survey can be completed on: https://forms.gle/6gNb5t7HS7b2ewXe7

Responses can be viewed on:

https://docs.google.com/spreadsheets/d/1TdD7aXahJdKM2VpchizytqFhk_X7rV98gVaZJuCxpKk/edit?resourcekey#gid=18738303

Overall, 192 people have filled in the survey and the results were automatically generated by Google Forms (42). They are represented below in the form of diagrams:
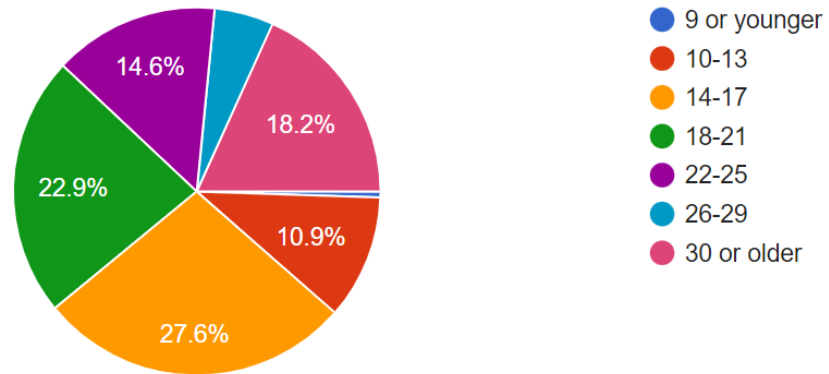
Figure 27 - "How old are you?". Source: author
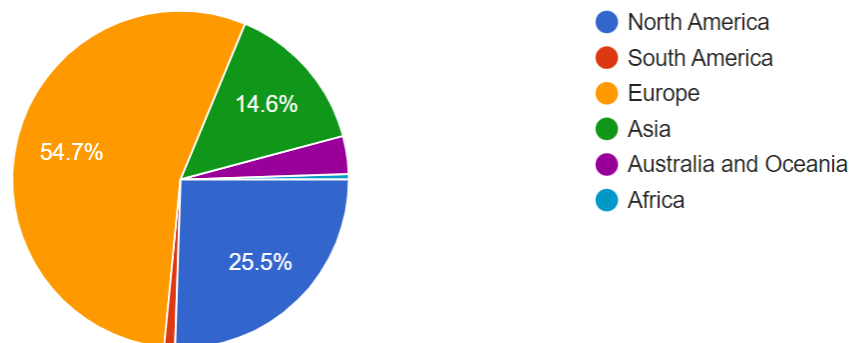
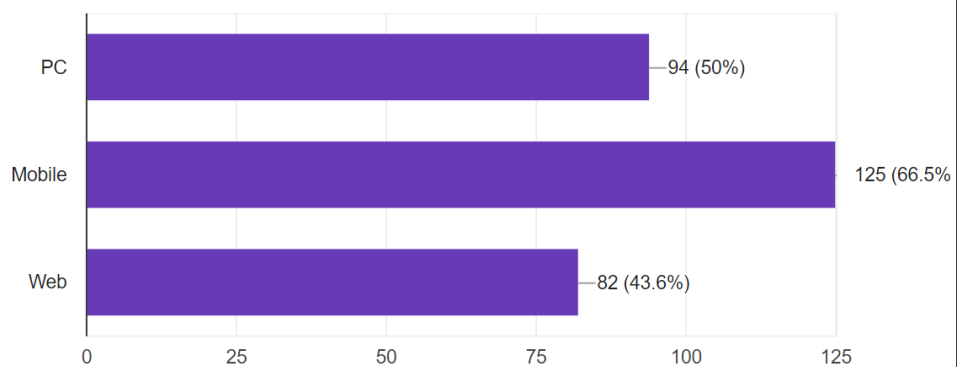Figure 28 - "Where are you from?". Source: author

Figure 29 - "Where do you use a timer?". Source: author

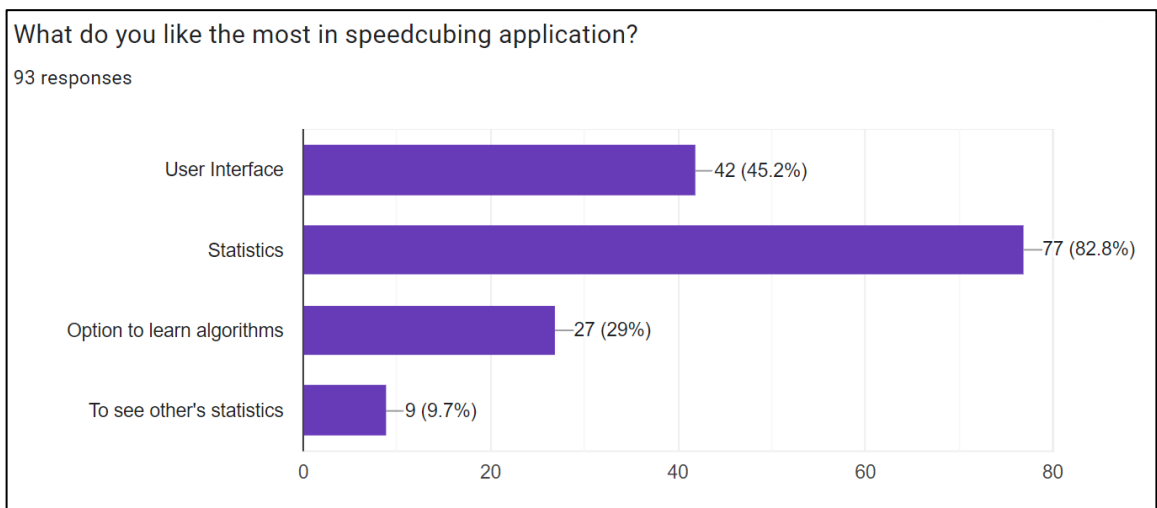Figure 30 - "What is your 3x3 best result?". Source: author

Figure 31 - "What do you like the most in speedcubing application?". Source: author

Based on the information provided in four diagrams above, the table below is created manually. It represents the connection between best solve, worst solve, average time of solving a Rubik's cube and age category among respondents. Age category was sorted out, best solve, worst solve and average time were calculated using SUBTOTAL(5, range), SUBTOTAL(4, range) and SUBTOTAL(1, range) formulas of Microsoft Office Excel 2019 respectively.

Age category (in years old)	Best solve (in seconds)	Worst solve (in seconds)	Average time of solving a Rubik's cube (in seconds)	Number of respondents
9 or younger	11.00	11.00	11.00	1
10-13	3.61	35	17.45	21
14-17	3.46	53	13.19	51
18-21	3.55	34	10.41	43
22-25	5.3	42	15.39	28
26-29	8.8	90	29.62	10
30 or older	7	210	28.57	33

Table 6 - Connection between an age category and time of solving a Rubik's cube. Source: author

4.1.2 Survey results

After analyzing the results from the survey about target community and their preferences, next statements can be formulated:

- Most of respondents are 14-17 and 18-21 years old with a ratio of **27.6 %** and **22.9%** respectively. The fact that **18.2%** of participants are 30 years old or older is surprising, which means that speedcubing is interesting for people of all ages;

- More than a half of surveyees are from Europe (**54.7%**), approximately a quarter are from North America (**25.5%**) and a seventh are from Asia (**14.6%**);

- **66.5%** of respondents use a timer on mobile platform, **50%** and **43.6%** use it on desktop and web respectively;

- For 77 of speedcubers "Statistics" is the most crucial part in the application. Therefore, own product will be focusing on creating decent statistics functionalities. Nevertheless, User Interface is the second important part.

- There is a tendency that people who are 26 years old and older solve a Rubik's cube slower than people who are in 10-25 age category.

4.2 User Personas

User personas (43) are fictional representations of different types of users of a product or service. They are created using an analysis of the survey and are designed to help to understand the preferences of the target audience. They can be categorized into three categories:

- Persona A positive: it is a target and main user, who will use the product as long as possible;
- Persona B, neutral: a potential user who will be the application occasionally;
- Persona C, negative: a person, who will not be using the application.

Their characteristics include name, age, demographics, goals and needs, short bio and frustrations.

Figure 32 - Persona A: James Connor. Source: author

Biography

Lissie Joshua is an amateur photographer who like capturing people on the streets of Manchester. Her friend Suzie demonstrated her how to solve a Rubik's cube. From now, Lissie is really interested in speedcubing and wants to learn how to solve it quickly and try to participate in speedcubing competitions.

Goals

- She wants to learn a CFOP method from zero.
- She wants to solve a Rubik's cube in less than 30 seconds.
- She wants to participate in speedcubing competition to gain experience.

Pain Points

- She does not know where to study CFOP method and its algorithms from.
- She does not know what timer to use and on which platform

Lissie Joshua

Photographer

About

🎂 19   🎓 Student

📍 Manchester   ♀ Female

Extrovert   Social   Patient

Friendly   Intelligent

Figure 33 - Persona B: Lissie Joshua. Source: author

Biography

Anthony Fury is a professional boxer who started the career when he was 15 years old. At his prime age of 32, he won 2 belts of WBO and WBA in middleweight category. His daughter Amanda asked him to buy a Rubik's cube after his training, so she can learn to solve it.

Goals

- He wants to be an absolute champion of the world in middleweight category.
- After he becomes a middleweight champion, he wants change the category to super middleweight.

Pain Points

- He thinks that solving a cube will develop his daughter's logical thinking, but does not understand why speedcubers spend hours on practising to solve a cube in seconds.

Anthony Fury

Professional boxer

About

🎂 32   🎓 Not finished

📍 Detroit   ♂ Male

Introvert   Hard-working   Patient

Motivated   Disciplined
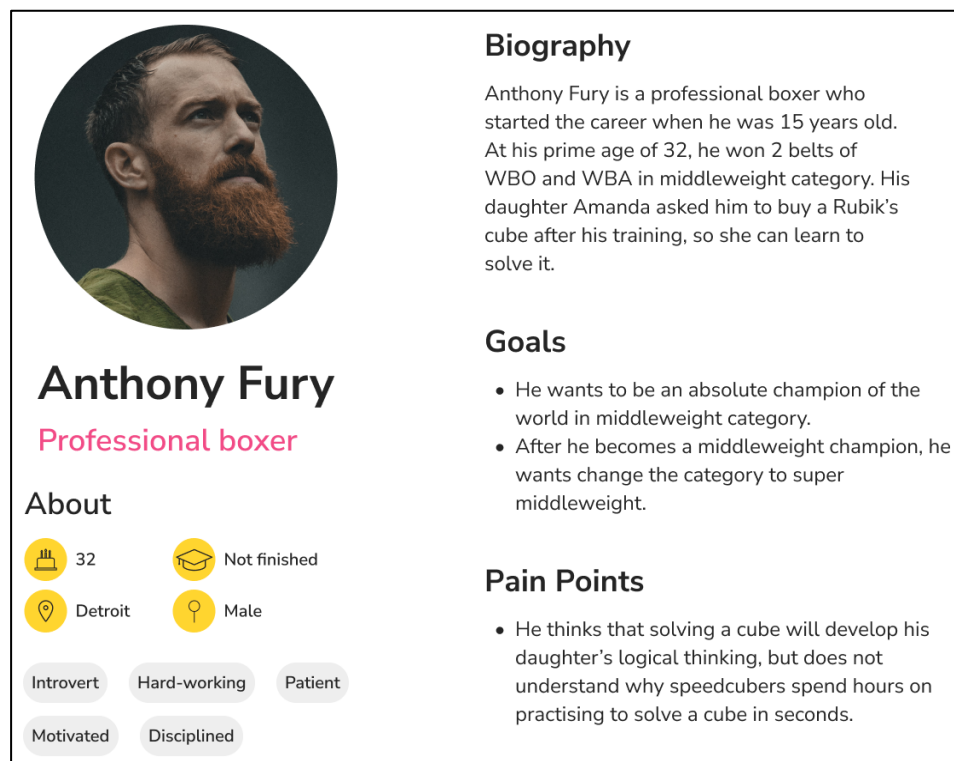
Figure 34 - Persona C: Anthony Fury. Source: author

User Personas were created on Figma.com website using a User Persona Template created by Cheniece Manning (44).

4.3 Usecases and scenarios

Next step is to set usecases and their scenarios, which describe how a user might interact within the application:

Timer/Start page – User Goal 1
Usecase:
On the main page, the user expects:
To see the timer, formula how to scramble a cube, list of solves and basic statistics.
Scenario:
The system shows on the main page:
1. A formula how to scramble a cube on the centered top of the screen;
2. A timer under the scramble formula with big and bold font;
3. Instruction how to start a timer;
4. A list of last solves;
5. A basic statistics containing best solve, worst solve, average of whole session.

Algorithms page – User Goal 2
Usecase:
On the algorithms page, the user expects:
To see the algorithms of CFOP method.
Scenario:
On the algorithms page, the system shows:
1. OLL algorithms of CFOP method that contain:
    a. Name of the algorithm;
    b. Illustrative image;
    c. Variations of the algorithm.
2. PLL algorithms of CFOP method that contain:
    a. Name of the algorithm;
    b. Illustrative image;
    c. Variations of the algorithm.

Statistics page – User Goal 3
Usecase:
On the statistics page, the user expects:
To see the list of all solving times, detailed statistics and a graph
Scenario:
The system shows on the statistics page:
1. A list of all solving times;
2. A statistics including: best solve, worst solve, average of 5 solves, average of 12 solves, average of 100 solves, session mean and number of solves;
3. A plotted graph of all solving times:
    a. With order number of solve on X-axis;
    b. With time of solve on Y-axis;
    c. White dots, which represent a solving time;
    d. Blue line connecting all dots.

User login page – User Goal 4
<u>Usecase:</u>
On the login page, the user expects:
To login to the system.
<u>Scenario:</u>
The system shows on the login page:
1.   Field to enter an email;
2.   Field to enter a password;
3.   "Login" button, by clicking on a user will be logged in;
4.   "Register" button, by clicking on "Register" page will be opened.

The system offers an unregistered user to register by providing a "Register" button.

User register page – User Goal 5
<u>Usecase:</u>
On the register page, the user expects:
To register into the system if they do not have an account yet.
<u>Scenario:</u>
The system shows on the register page:
1.   Field to enter a username;
2.   Field to enter an email;
3.   Field to enter a password;
4.   Field to confirm a password;
5.   "Register" button, by clicking on a user will be registered;
6.   "Back" button, by clicking on "Login" page will be opened.

User Data Input page – User Goal 6
<u>Usecase:</u>
On the User Data Input page, the user expects:
To be able to enter their solving times into required fields.
<u>Scenario:</u>
The system shows on the User Data Input page:
1.   Field to enter a username;
2.   Field to enter a best solve;
3.   Field to enter an average of 5 solves;
4.   Field to enter an average of 12 solves;
5.   "Save" button, by clicking on all entered data will be saved;
6.   "Scoreboard" button, by clicking on "Table of users" page will be opened;
7.   "Sign Out" button, by clicking on a user will be signed out.

Table of users page – User Goal 7
<u>Usecase:</u>
On the Table of users page, the user expects:
To be able to see the scoreboard table containing their and other users' solving results.
<u>Scenario:</u>
The system shows on the Table of users page:
1.   Column of all usernames;

2. Column of all best solves;
3. Column of all averages of 5 solves;
4. Column of all averages of 12 solves;
5. "Back" button, by clicking on a "User Data Input" page will be opened;
6. "Sign Out" button, by clicking on a user will be signed out.

4.4 Requirements

A table below represents requirements for the speedcubing application. They are conditions that need to be met to make sure that a project is completed successfully.

ID	Requirement
	State 0: Timer Page
1	If a user presses a "Space" button, timer's value resets
10	And timer starts counting
11	If a user presses a "Space" button after timer starts counting, timer stops counting
110	And average time of all solving times is calculated
111	And if a number of solves is greater than or equal to five, then average time of last five solves
112	And if a number of solves is greater than or equal to twelve, then average time of last twelve solves
113	And if a number of solves is greater than or equal to hundred, then average time of last hundred solves
114	And new scramble formula is randomly generated
12	When timer stops counting, the solving time is being saved
13	If the current saved solving time is the lowest, then it is the best time
14	If the current saved solving time is the highest, then it is the worst time
15	List of last solves is shown on the left bottom side
16	Basic statistics is shown on the right bottom side
	Transition 01: Timer page Statistics page
20	If a user clicks on "Statistics" tab, system will switch to "Statistics" page
21	If a user clicks on "Timer" tab, system will switch to "Timer" page
	State 1: Statistics Page
30	If a solving time is saved, it is shown on the graph as a dot
31	If there are two or more solving times dots, an orange line connects dots.
32	Full list of solves is shown under the graph
33	Best solve is highlighted with green colour and shown on the right side of the screen
34	Worst solve is highlighted with orange colour and shown on the right side of the screen
35	Statistics is shows under best and worst solves and on the left side from the list of solves
	Transition 02: Timer page Algorithms page
40	If a user clicks on "Timer" tab, system will switch to "Timer" page
41	If a user clicks on "Algorithms" tab, system will switch to "Algorithms" page
	State 2: Algorithms page
50	If a user chooses "OLL" from a drop-down menu, then all OLL algorithms are shown
500	If a user clicks on specific OLL algorithm, pop-up windows appears and displays 5 variations of the algorithm with an image
501	If a user click on OLL pop-up window, then it disappears
51	If a user chooses "PLL" from a drop-down menu, then all PLL algorithms are shown
510	If a user clicks on specific PLL algorithm, pop-up windows appears and displays 5 variations of the algorithm with an image
511	If a user click on PLL pop-up window, then it disappears

Transition 12: Statistics page Algorithms page	
60	If a user clicks on "Timer" tab, system will switch to "Statistics" page
61	If a user clicks on "Algorithms" tab, system will switch to "Algorithms" page
State 3: Login page	
70	If a user does not enter email and presses "Login" button, then a system shows "Missing Email" message
71	If a user does not enter password and presses "Login" button, then a system shows "Missing Password" message
72	If a user enters wrong password and presses "Login" button, then a system shows "Wrong Password" message
73	If a user enters invalid email and presses "Login" button, then a system shows "Invalid Email" message
74	If a user enters email and/or password that do not exist and presses "Login" button, then a system shows "Account does not exist" message
Transition 34: Login page Register page	
80	When a user presses a "Register" button, then a system displays "Register" window
81	When a user presses a "Back" button, then a system displays "Login" window
State 4: Register page	
90	If a user does not enter email and presses "Register" button, then a system shows "Missing Email" message
91	If a user does not enter username and presses "Register" button, then a system shows "Missing Username" message
92	If a user does not enter password and presses "Register" button, then a system shows "Missing Password" message
93	If a user creates a password that is shorter than 6 symbols and presses "Register" button, then a system shows "Weak Password" message
94	If a user confirms a password that does not match first entered password and presses "Register" button, then a system shows "Password does not match"
95	If a user enters email that is already registered and presses "Register" button, then a system shows "Email Already In Use" message
Transition 35: Login page User Data page	
100	When a user presses a "Login" button, then a system displays "User Data" window
101	When a user presses a "Sign Out" button, then a system displays "Login" window
State 5: User Data page	
110	A user enters value to "username" label
111	A user enters value to "bestsolve" label
112	A user enters value to "avg5" label ("avg5" means "average of 5")
113	A user enters value to "avg12" label ("avg12" means "average of 12")
114	If a user presses on "Scoreboard", then scoreboard window is shown
Transition 56: User Data page Firebase	
120	If a user presses on "Save" button, then user's data on Firebase is being updated
State 6: Firebase	
130	"Username" value on Firebase is updated by "username" value entered by the user
131	"Best solve" value on Firebase is updated by "bestsolve" value entered by the user
132	"Avg5" value on Firebase is updated by "avg5" value entered by the user
133	"Avg12" value on Firebase is updated by "avg12" value entered by the user

Table 7 - Requirements. Source: author

4.5 State machine diagram

State machine diagram provides a visual representation of how the application operates within its functions. Requirements ID are written on the right side of blocks and a requirement itself is on the left side. The diagram was created on draw.io website (45).
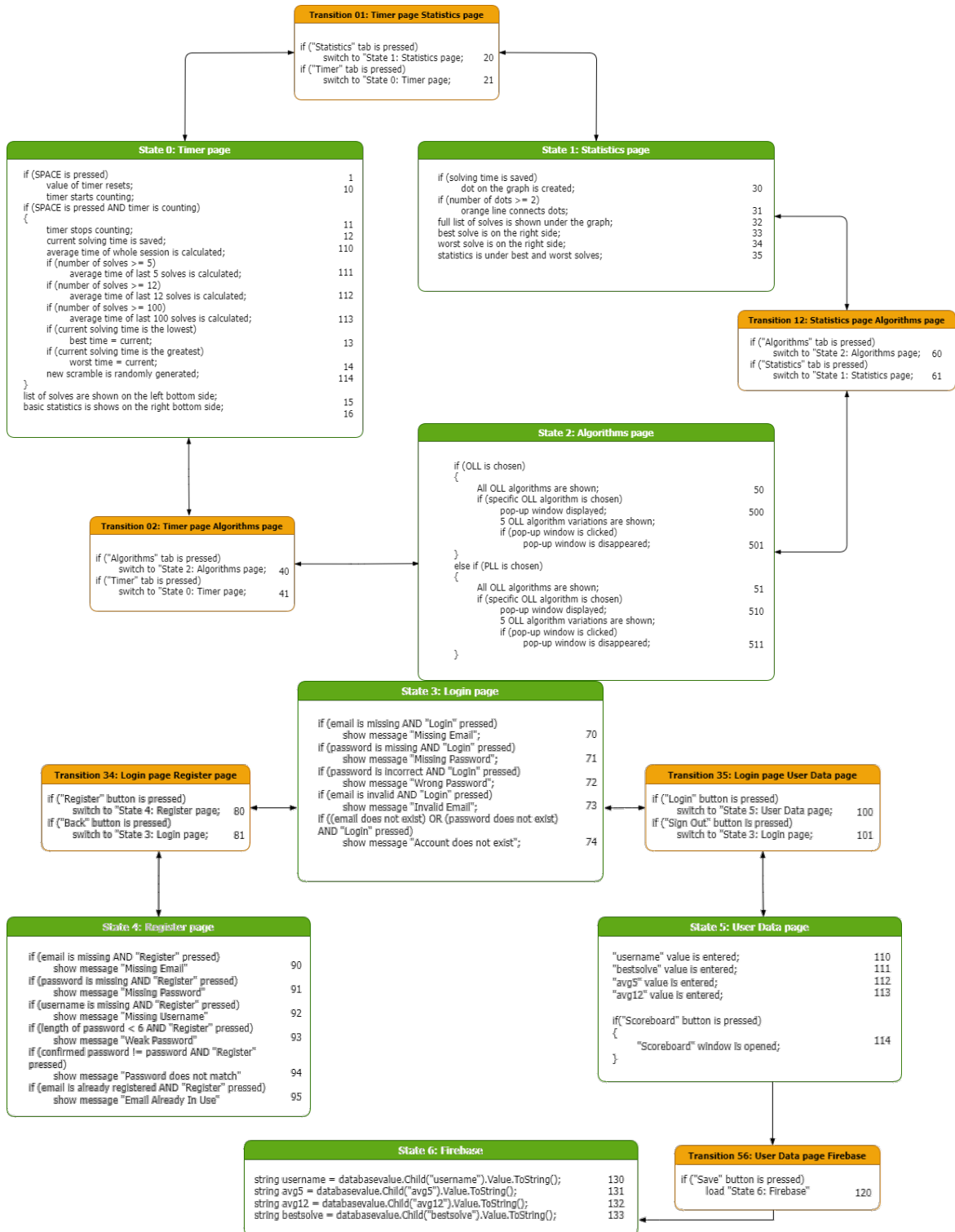

Figure 35 - State Machine diagram. Source: author

4.6 Wireframes

Next step is to create a high fidelity wireframes in order to have a visual preview of how the application will possibly look like. They were created in MS PowerPoint (46). The list of wireframes:

1) Timer page;
2) Statistics page;
3) Algorithms OLL/PLL page;
4) Algorithms OLL/PLL page - choose the algorithm;
5) Table of users page.

4.6.1   Wireframe 1: Timer page
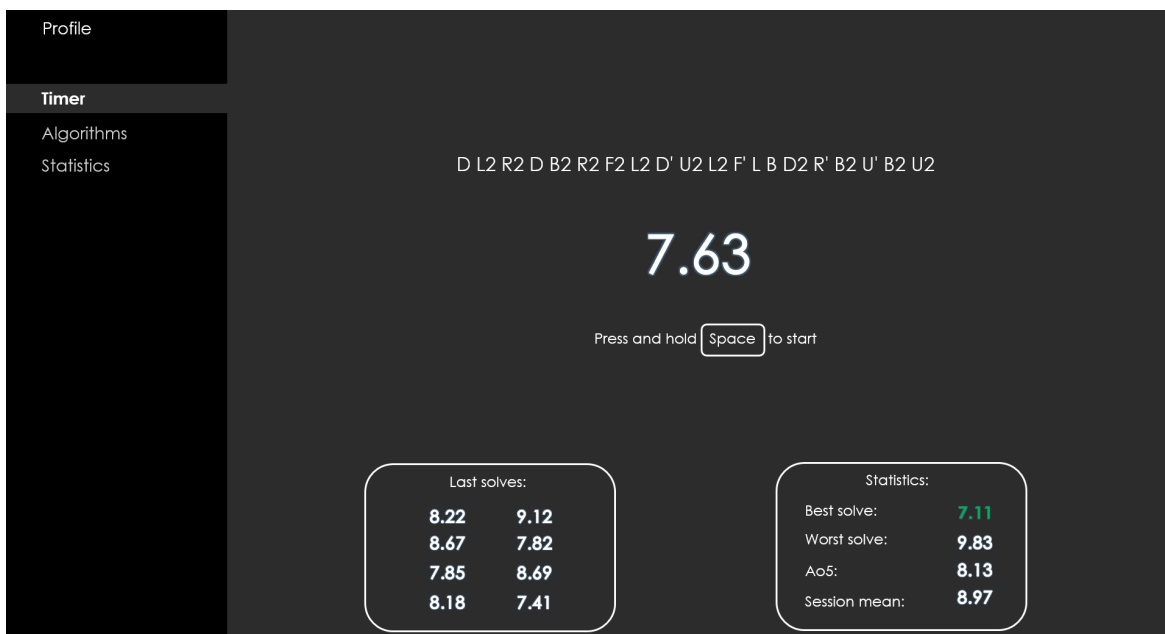
Figure 36 - Wireframe 1: Timer page. Source: author

"Timer" page is to contain scramble formula on the top, a stopwatch in the center, instruction how to start a stopwatch, short list of last solves and a basic statistics on the bottom. There is a menu on the left hand side in order to change the page to "Algorithms", "Statistics" or "Profile". The background is grey, because grey colour represents elegance, intelligence, neutrality, balance and does not disturb users from solving a cube. Its RGB hexadecimal code is 2B2B2B. Font of „Gothic" is planned to be used due to its minimalistic and elegance look. Menu on the left has a background of black colour that separates it from main screen.

4.6.2   Wireframe 2: Statistics page
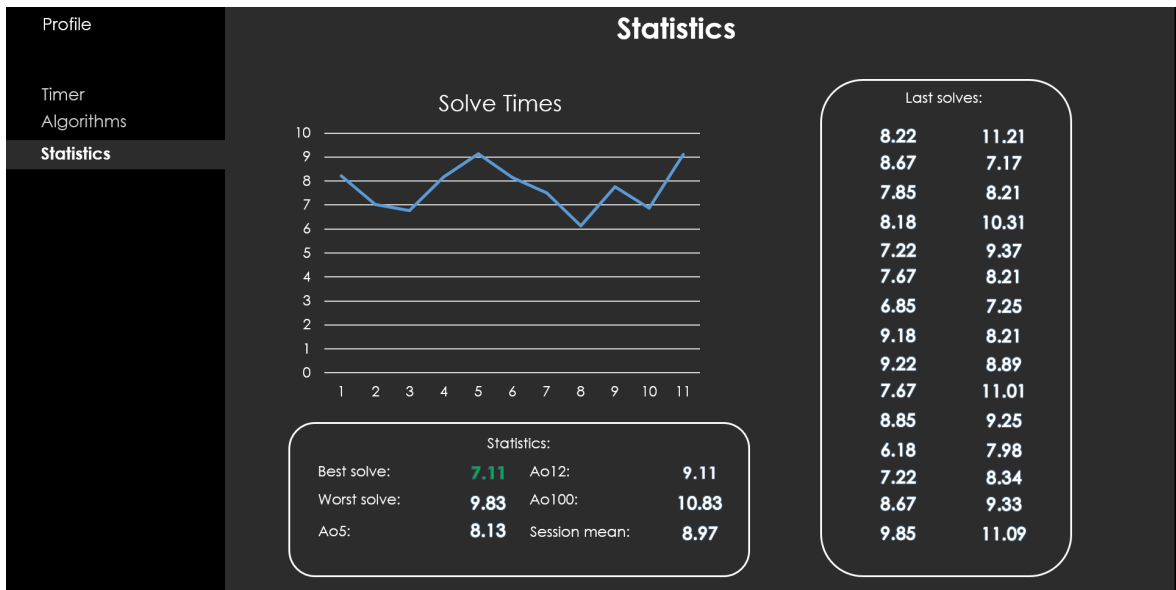
Figure 37 - Wireframe 2: Statistics page. Source: author

The page is to contain a graph plotted based on the latest solves, detailed statistics on the bottom and full list of last solves on the right hand side.

4.6.3   Wireframe 3: Algorithms OLL/PLL page
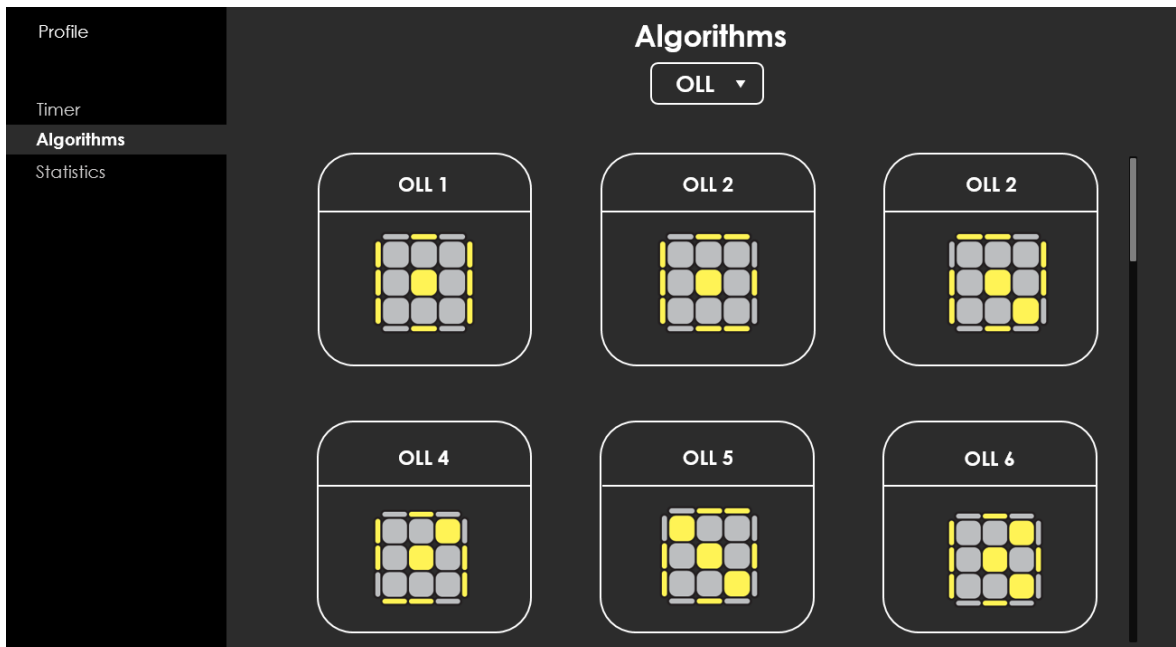
Figure 38 - Wireframe 3: Algorithms OLL/PLL page. Source: author

The „Algorithms" page will contain a drop-down menu on the top providing an option to choose between OLL and PLL algorithms. In the centre there will be scrollable content of

OLL or PLL algorithms with a vertical scrollbar on the right side, both of them will detect finger touch and mouse scroll button.

4.6.4   Wireframe 4: Algorithms OLL/PLL page – choose the algorithm
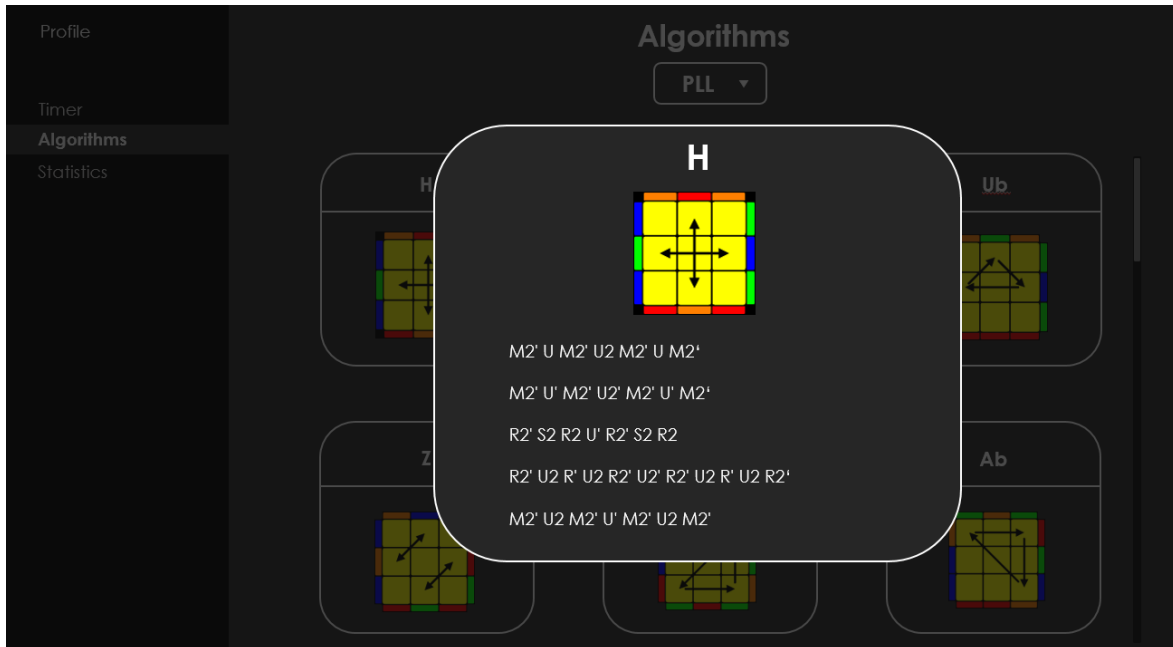
Figure 39 - Wireframe 4: Algorithms OLL/PLL page - choose the algorithm. Source: author

The wireframe has a pop-up window containing an image and variations of one algorithm.

4.6.5   Wireframe 5: Table of users page
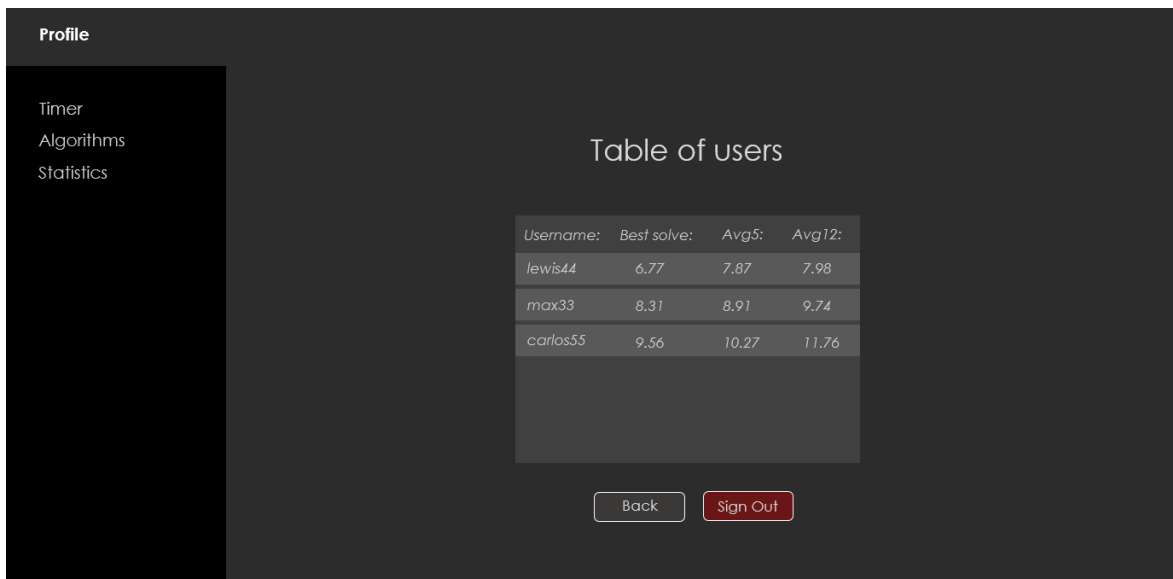
Figure 40 - Wireframe 5: Table of users page. Source: author

Table of users will contain a list of users and their personal records in the application.

4.7 Development of the application

This chapter describes the process of development the application. It includes screenshots of Unity Engine User Interface, project structure, C# scripts, database configuration and creation of name, splash image and logo.

4.7.1 Creating User Interface on Unity Engine
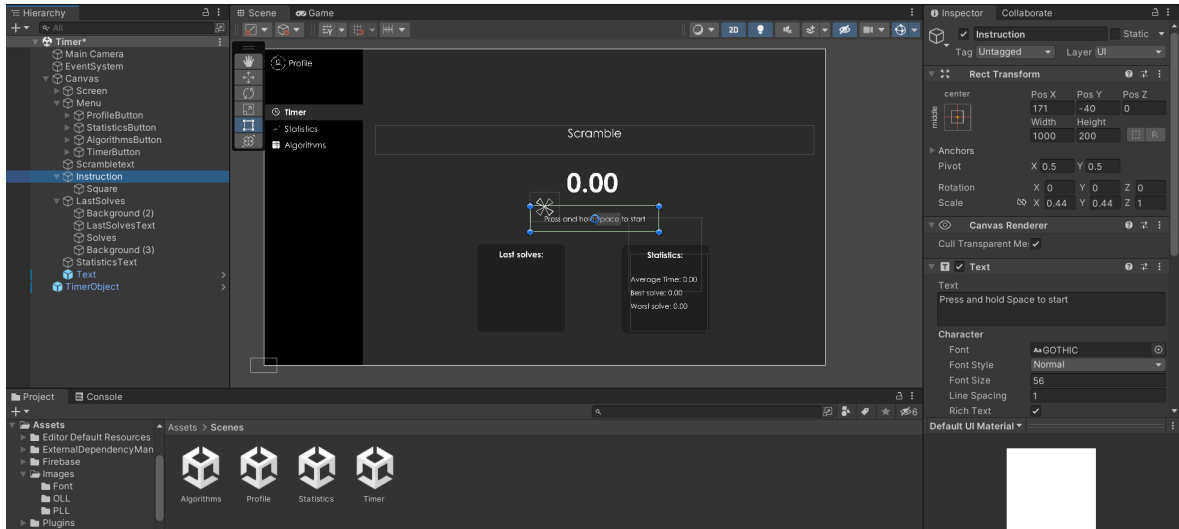
Figure 41 - "Timer" scene development. Source: author

The screenshot above represents the structure of a "Timer" scene, its elements on the left side and their properties on the right side. On the left bottom corner, there are files of the project such as assets, scripts, images, fonts, plugins and others. The scene contains a randomly generated scramble formula on the top, main timer with bold font in the center, statistics and last solves elements on the bottom, and the menu on the left hand side.
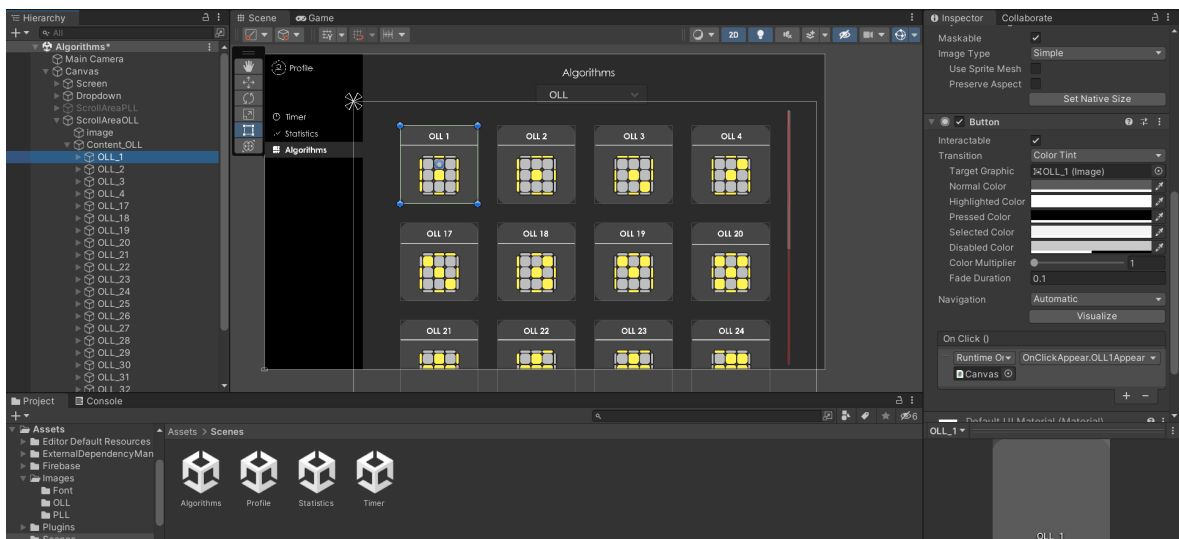
Figure 42 - "Algorithms" scene development. Source: author

The screenshot above illustrates the structure of an "Algorithms" scene. It contains two main elements called "Content_OLL" and "Content_PLL". They are switched between each other when a user chooses what algorithms they want to learn of using a drop-down menu on the top of the scene. Moreover, it contains Scroll down element on the right side, so a user can scroll the content and view all the algorithms that are hidden.
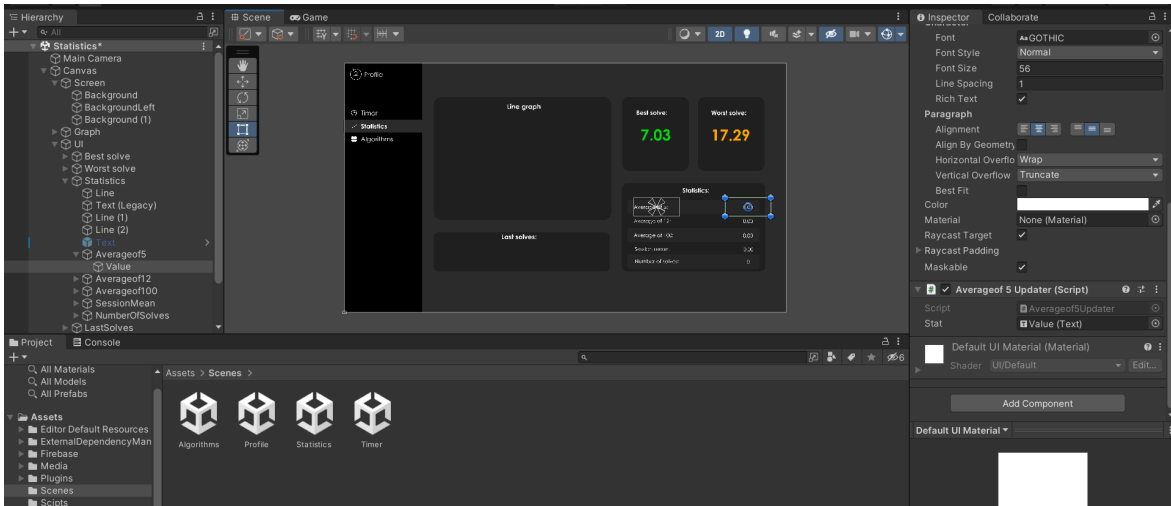
Figure 43 - "Statistics" scene development. Source: author

The figure above shows how "Statistics" scene is built. Main elements are line graph on the left hand top corner, best solve and worst solve on the right hand top corner, statistics and list of last solves on the bottom. As shown on the right side of the screenshot, value of "Averageof5" element is updated using "Averageof5 Updater (Script)".
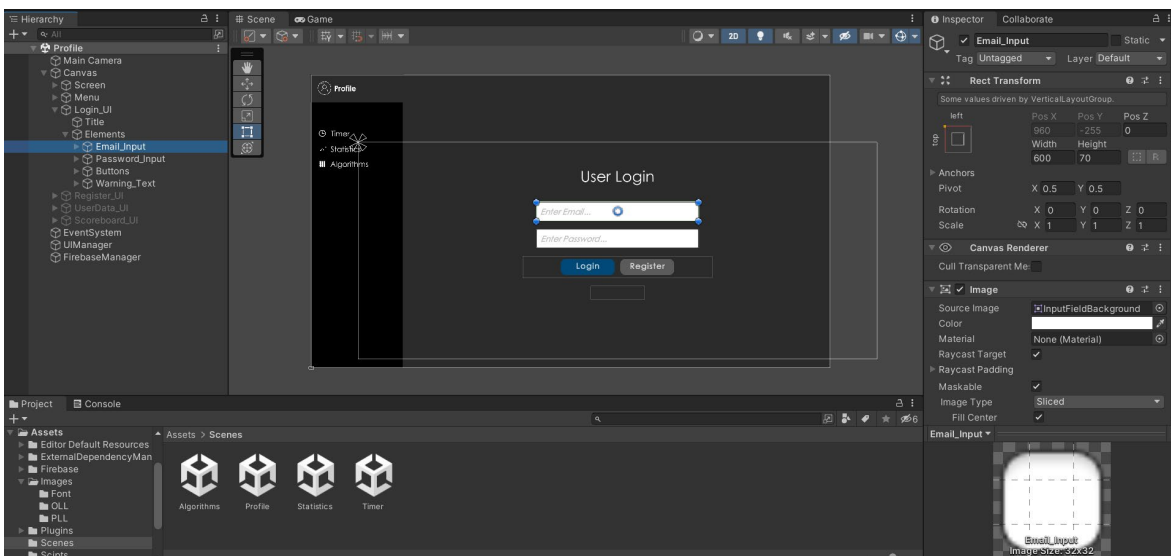
Figure 44 - "User Login" scene development. Source: author

The image above demonstrates elements of "Profile" scene. On the left side in File Hierarchy there are four user interface elements called "Login_UI", "Register_UI",

"UserData_UI" and "Scoreboard_UI". They contain relevant information according to their names. Interesting part is that there is no need to duplicate the scene in order to switch between UI elements. It can be realized using UIManager class and a command "UIManager.instance.ScoreboardScreen();"

4.7.2   Writing scripts in Microsoft Visual Studio 2019

Unity Engine offers lots of interesting functionalities that can be used for the product. On other hand, it requires writing scripts, so all elements can interact with each other.

```csharp
if (Input.GetKeyUp(KeyCode.Space) && status == false)
    {
        CreateScramble();
        avgsession();
        results.Insert(index, (float)Math.Round(value, 2));
        index++;
        lastsolves.text = string.Join(" ", results);
        best = results.Min();
        worst = results.Max();
        besttime.text = best.ToString("0.00");
        worsttime.text = worst.ToString("0.00");
        averageof5Function();
        averageof12Function();
        averageof100Function();
    }
```

Source code 2 - "Space" key pressed. Source: author

The script above describes the process when a "Space" button is pressed to stop a timer, then new scramble is randomly generated and a system calculates "Average of the whole session", "Average of 5 last solves", "Average of 12 last solves", "Average of 100 last solves", "Best solve" and "Worst solve". It rounds every solving time to two decimal numbers and creates a list of last solves.

```csharp
public void CreateScramble(int scramblelength = 19)
{
    int _scramblelength = scramblelength - 1;
    string randomscramble = "";
    string[] characters = new string[] { "R ", "R' ", "R2 ",
        "L ", "L' ", "L2 ", "F ", "F' ", "F2 ", "B ", "B' ",
        "B2 ", "U ", "U' ", "U2 ", "D ", "D' ", "D2 " };
    for (int a = 0; a <= scramblelength; a++)
    {
        randomscramble = randomscramble +
characters[UnityEngine.Random.Range(0, characters.Length)];
    }
    scrambletext.text = randomscramble;
}
```

Source code 3 - "CreateScramble" function. Source: author

The "CreateScramble" function describes how scramble formula is generated. It uses Unity class "Random" and randomly chooses letters from characters of string array (47).

```
private IEnumerator RegisterProcess(string _emailValue, string _passwordValue, string
_usernameValue){
    if (_usernameValue == ""){
        RegisterWarningText.text = "Missing Username";
    }
    else if (RegisterInputPasswordField.text != RegisterInputPasswordVerifyField.text){
        RegisterWarningText.text = "Password Does Not Match!";
    }
    else{
        var RegisterTask = FirebaseAuth.CreateUserWithEmailAndPasswordAsync(_emailValue,
_passwordValue);
        yield return new WaitUntil(predicate: () => RegisterTask.IsCompleted);

        if (RegisterTask.Exception != null){
            FirebaseException FirebaseException = RegisterTask.Exception.GetBaseException() as
FirebaseException;
            AuthError ErrorCode = (AuthError)FirebaseException.ErrorCode;
            string ErrorMessage = "Register Failed!";
            switch (ErrorCode){
                case AuthError.EmailAlreadyInUse:
                    ErrorMessage = "Email Already In Use";
                    break;
                case AuthError.MissingEmail:
                    ErrorMessage = "Missing Email";
                    break;
                case AuthError.MissingPassword:
                    ErrorMessage = "Missing Password";
                    break;
                case AuthError.WeakPassword:
                    ErrorMessage = "Weak Password";
                    break;
            }
            RegisterWarningText.text = ErrorMessage;
        }
        else {
            FirebaseUser = RegisterTask.Result;

            if (FirebaseUser != null {
                UserProfile userProfile = new UserProfile { DisplayName = _usernameValue };
                var ProfileTask = FirebaseUser.UpdateUserProfileAsync(userProfile);
                yield return new WaitUntil(predicate: () => ProfileTask.IsCompleted);

                if (ProfileTask.Exception != null){
                    RegisterWarningText.text = "Username Set Failed!";
                }
                else{
                    UIManager.window.LoginWindowOnClick();
                    RegisterWarningText.text = "";
                    LoginFieldsClean();
                    RegisterFieldsClean();
                }
```
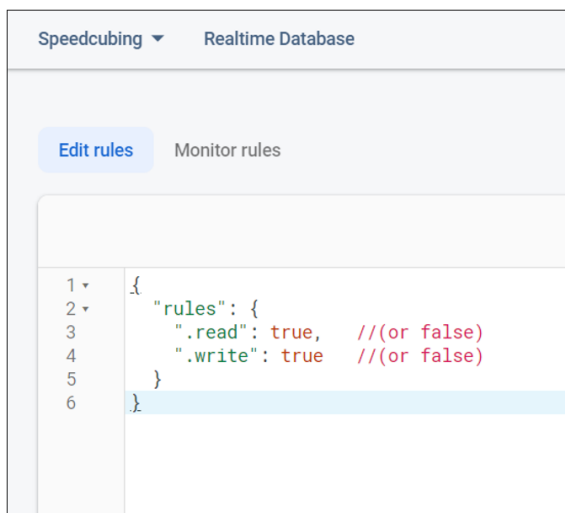
Source code 4 - "RegisterProcess" function. Source: author

The "RegisterProcess" function describes the process of user registration into Firebase Database. In case of any error, corresponding error message will be displayed. Otherwise, a new user will be registered into the system and able to view other users' statistics in the "Table of Users" window of the application (48).
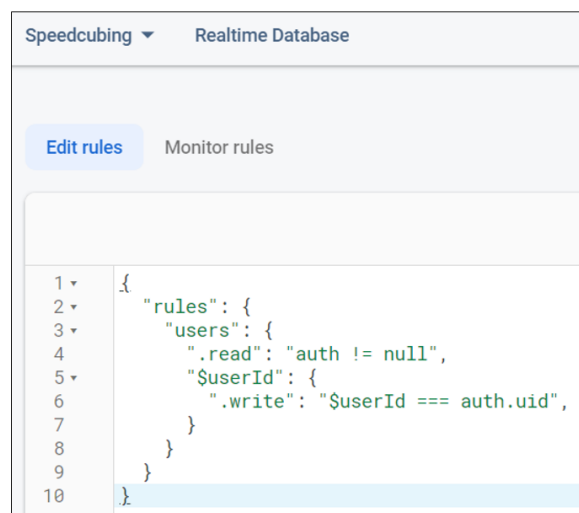
4.7.3 Database rules

While initializing, Firebase offers to make a real-time database either fully public or fully private. The rules had to be changed so only authorized users could read data from database. If a user current ID matches their authorized ID, then a user is able to write data into and read data from the database.

Before: After:

Figure 45 - Database rules "Before" and "After". Source: author

4.7.4 Creation of name, splash image and icon

The application is called "sCube – speedcubing timer", where "s" stands for superb. Icon is displayed on the shortcut, while splash image is shown during launch of the application. Both of them contain grey background colour because of the main screen of the application. A cube is drawn in minimalistic style with five opposite colours that represent sides of a Rubik's cube.

Figure 46 - The application icon. Source: author
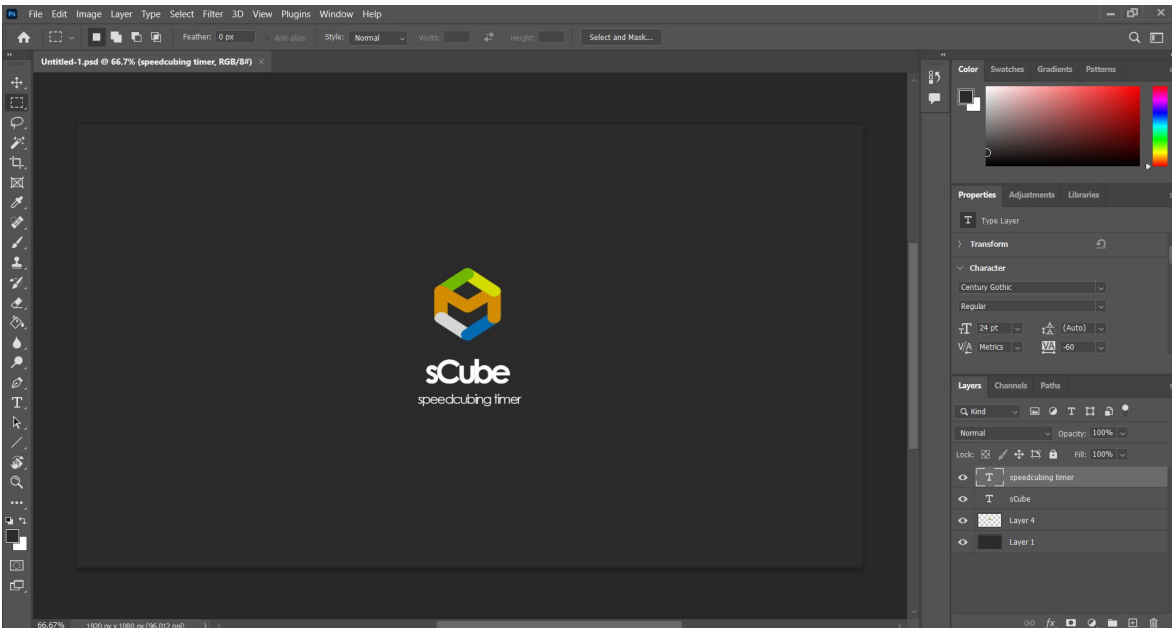
Figure 47 - The application splash image. Source: author

The Icon and Splash Image of the app were created in Adobe Photoshop 2022 (49).

4.8 Application release

The application is developed and is ready to be built for three platforms and released for public access.

4.8.1 Build settings

Build settings allow developers to choose what platform to export an application for. On the left bottom side there is a "Player Settings" (or "Project Settings") option, which allows configuring icon, splash image, name, resolution and other setting for building the product.
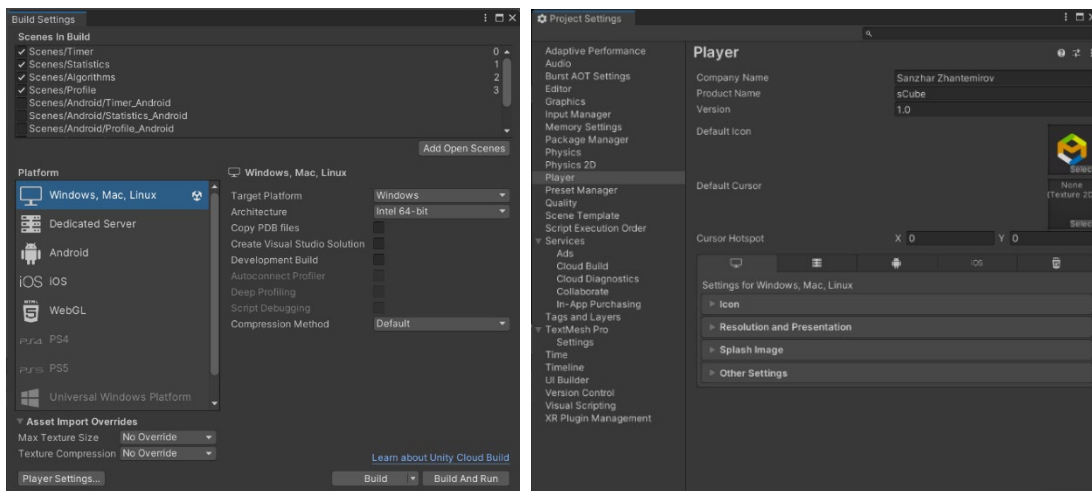
Figure 48 - Build settings. Source: author

4.8.2   Desktop version

"Build Settings -> Windows, Mac, Linux -> Build -> Choose the folder to export the project to" steps have to be completed in order to build the app for Desktop platform. After building the project, File Explorer will automatically open the folder and the application can be launched. Desktop version looks very similar to wireframes created earlier. Colours and font are the same, icons on the menu are added. On "Statistics" page (Figure 50), a graph is orange coloured, because it looks brighter and has a better match with grey colour. Locations of elements were changed and now interface looks as a whole. Screenshots of the desktop application are represented below.
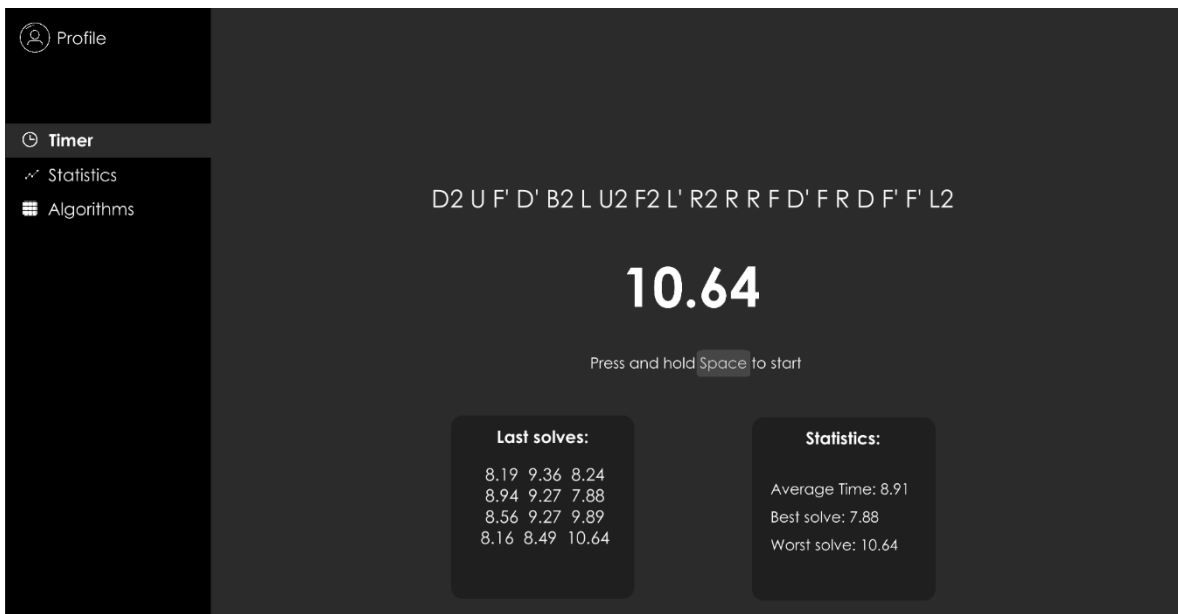
Figure 49 - "Timer" page. Source: author
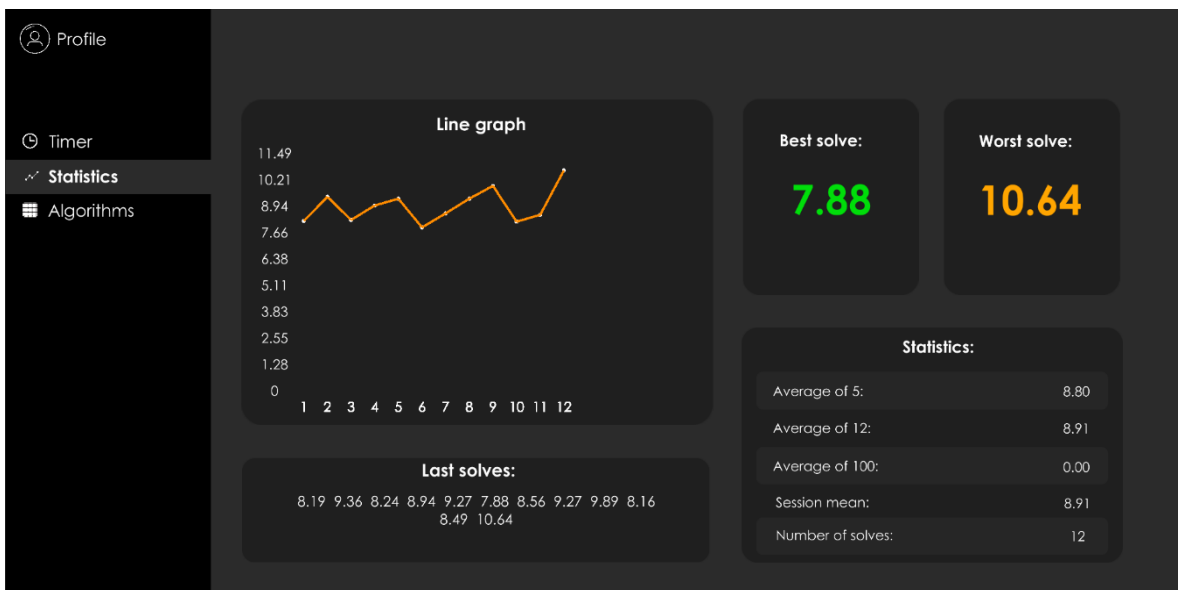
Figure 50 - "Statistics" page. Source: author
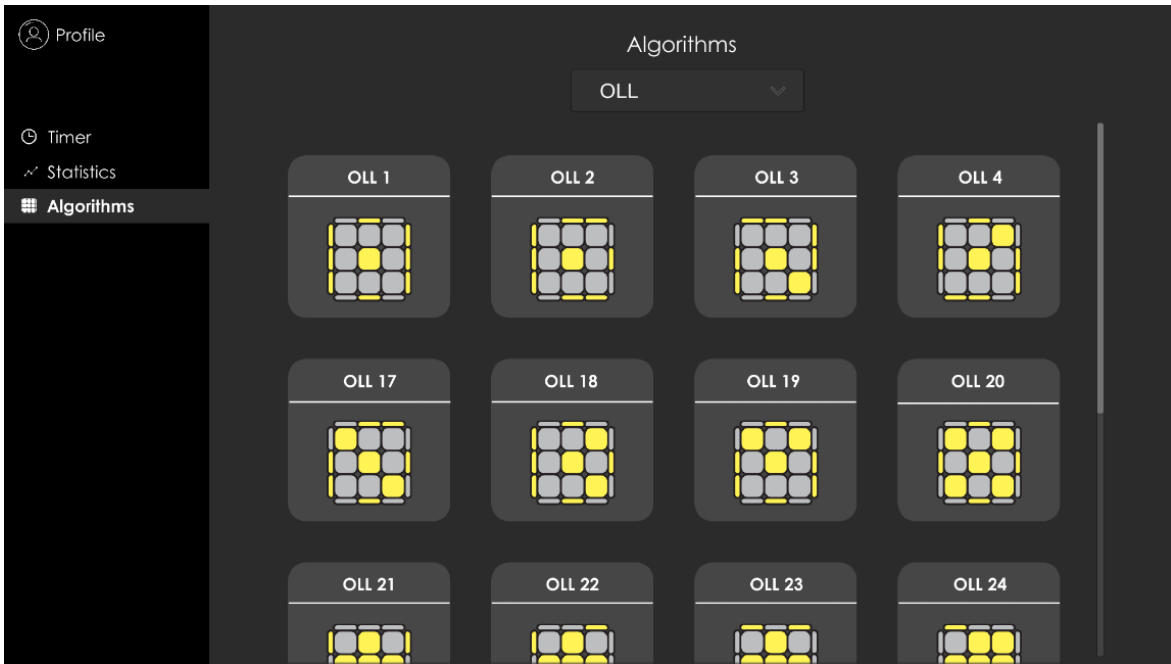
Figure 51 - "Algorithms" page. Source: author
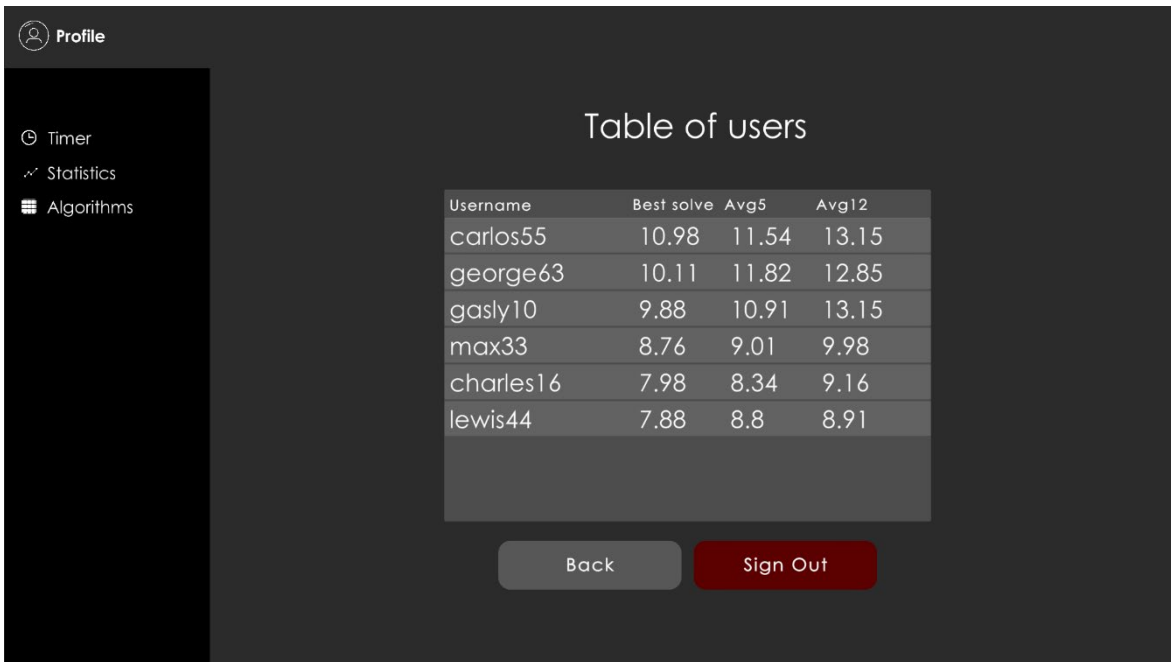
Figure 52 - "Table of Users" page. Source: author

The installation file was uploaded to Google Drive (50) and is available for downloading from this link:

https://drive.google.com/file/d/1aefE6AGk75fi2M7FlU48ZB0h4J4V0Mw-/view?usp=sharing

Desktop version contains all necessary tabs and all functions work properly. One platform is covered, hence the app has to be built for Android and Web.

4.8.3   Android version

With the aim of building the application for Android OS, the project requires to have its keystore, minimum API level of 31, 64-bit and file extension as App Bundle (.aab). An extension of APK file is not appropriate, since the app has to be uploaded to Google Play. Build procedure: "Build settings -> Android -> Player Settings -> Other Settings -> Create keystore -> Change minimum API to 31 -> Choose "Scripting Backend" as "IL2CPP" -> Tick ARM64 -> Build App Bundle (Google Play) -> Build".

After successful build, the .aab file is created and mobile application is ready to be tested. The app was installed on local device and the screenshots are represented below:
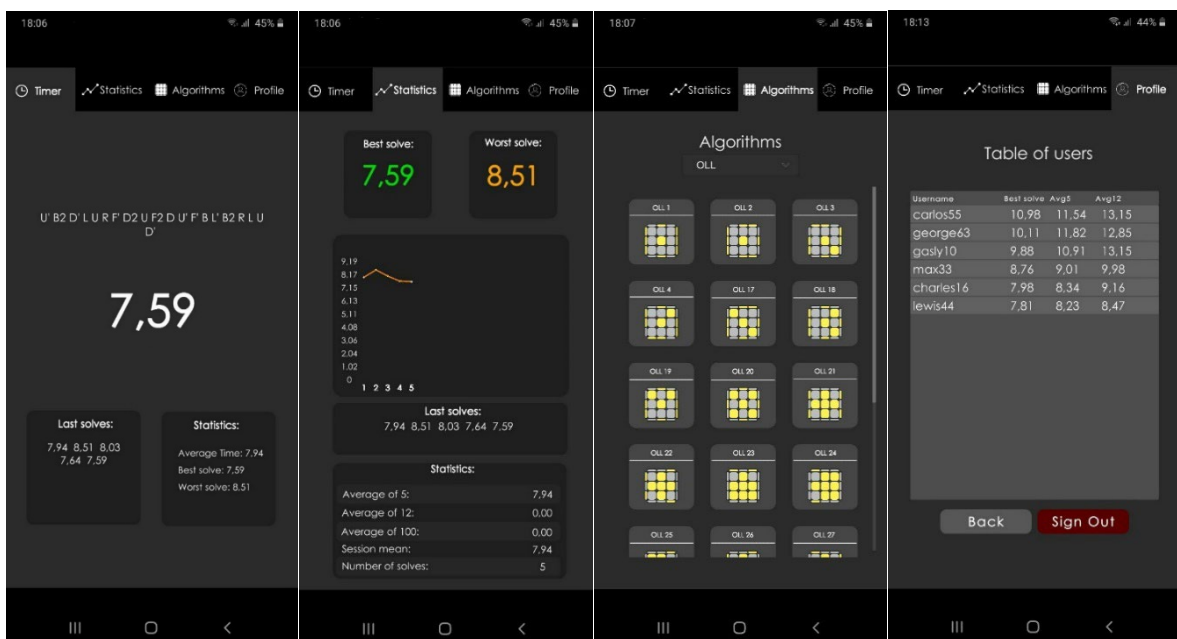
Figure 53 - Android version screenshots. Source: author

After verifying that the application works perfectly on local device, it is time to upload it to Google Play. In order to do that, the account with developer mode is required, which costs 25 USD. After activating the account and uploading the application (Figure 54), it needs to be reviewed by Google.
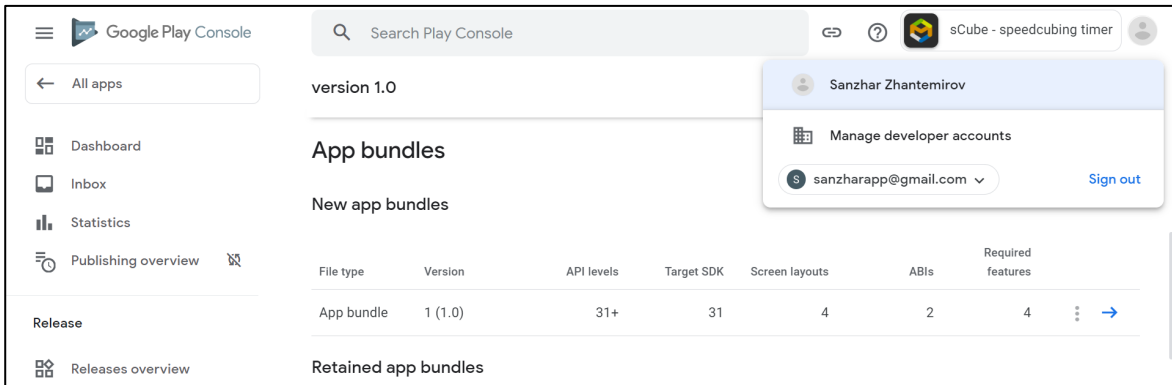
Figure 54 - Google Play Developer Console. Source: author

The review process took four business days and after successful verification, "sCube – speedcubing application" is available for downloading by everyone for free.

Link to the application:

https://play.google.com/store/apps/details?id=com.DefaultCompany.Speedcubing
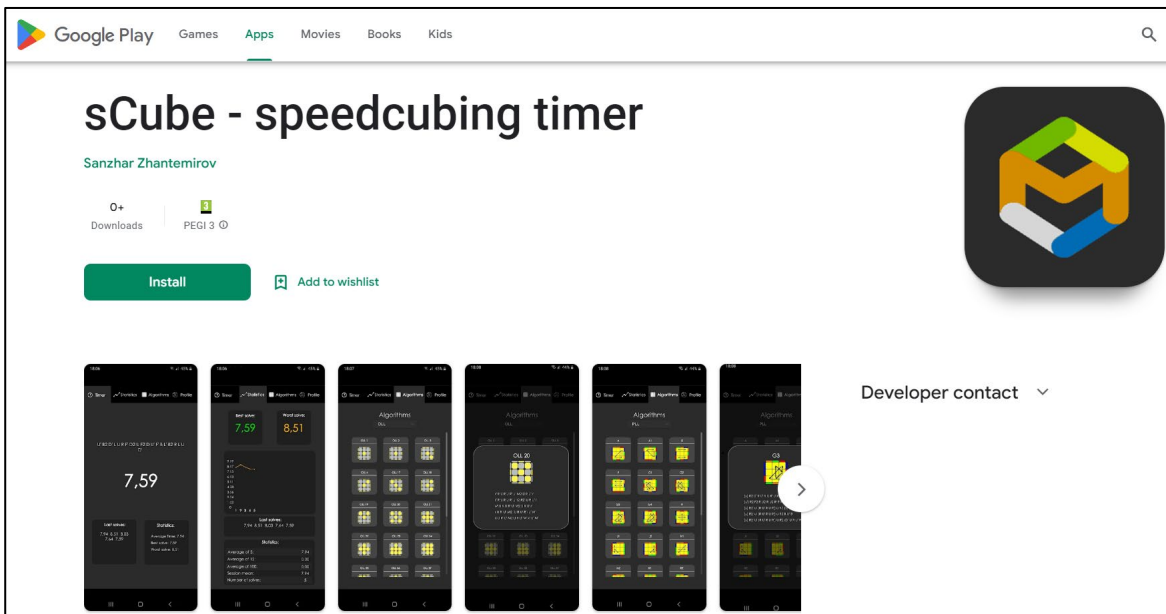
Figure 55 - The application on Google Play. Source: author

4.8.4   Web version

In order to build the application for Web, next procedure needs to be completed: "Build Settings -> WebGL -> Player Settings -> Publishing Settings -> Tick "Decompression Fallback" -> Build"

The application is now in WebGL version and is ready to be uploaded to website in these steps:

- Create a personal account with a username "Speedcubing" on https://itch.io website (Figure 56);

- Upload .zip file of the project;

- Set characteristics of a project such as name, URL, short description, category, price, access rights, screenshots and tags.
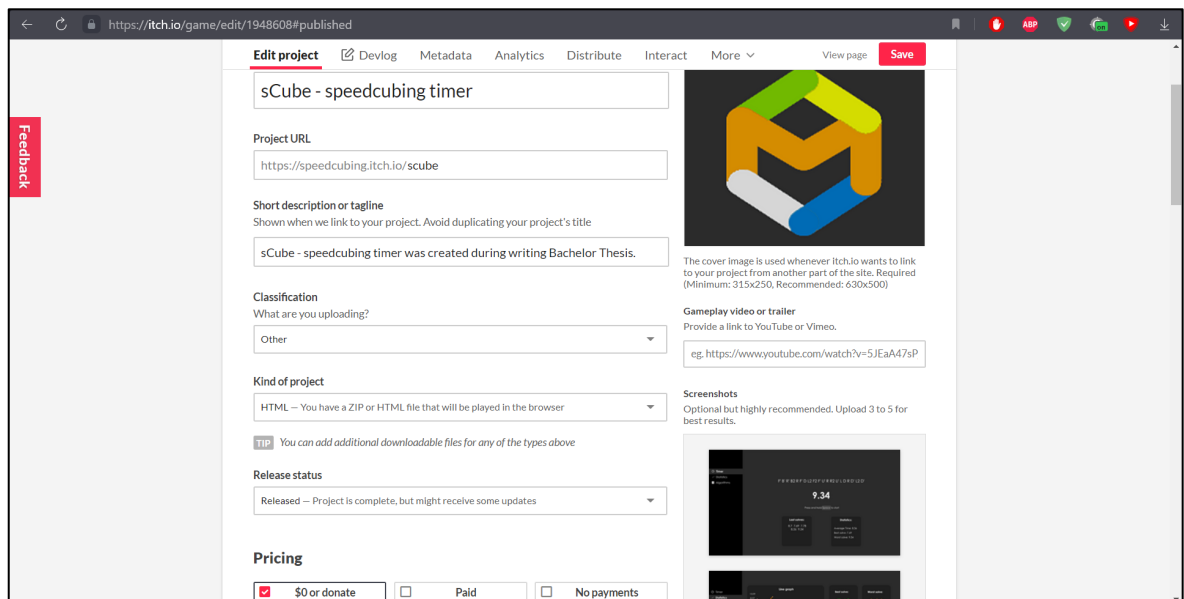
Figure 56 - Uploading the application to Web. Source: author
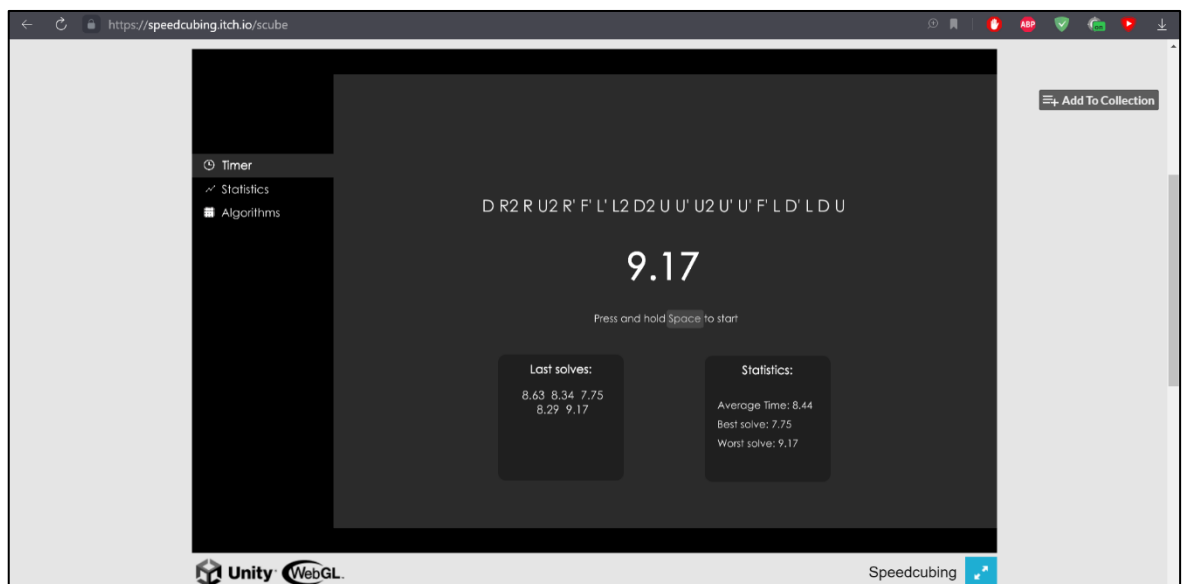
Figure 57 - The application on Web. Source: author

Web version is available on the following link: https://speedcubing.itch.io/scube.
Moreover, the application can be opened in full screen mode.

4.9 User Interface and Functionalities differences and similarities

The application is available on all three platforms. The table below represents similarities and differences of the application's user interface and functionalities on different versions.

Platform	Similarities	Differences
Desktop	All platforms support next functions: • Measure solving time; • Generate scramble; • Show basic statistics on main screen; • Show detailed statistics and plot a line graph; • Show OLL and PLL algorithms with drop-down menu and vertical scroll bar; • Show the algorithms with pop-up window of five different formulas. Both Desktop and Android versions support Firebase authentication and real-time database functions.	Designed for Windows, Mac and Linux; Can be downloaded and installed as an executable file.
Android		Has to be installed via Google Play; The User Interface is adapted for users of mobile phones. Button are larger, everything is centralized and statistics tab is structured differently; The main screen of timer requires a user to touch the screen instead of pressing "Space" button, therefore the part of script was changed. System recognizes when a user touches invisible button on the main screen and disables a function of pressing "Space" button.
Web		Can be accessed on the Internet website; Firebase is not supported by Unity WebGL, therefore functions of authentication and uploading and showing results of other users are disabled.

Table 8 - The application similarities and differences on three platforms. Source: author

4.10 Feedback from users

In order to receive a feedback from users, online survey was created using Google Forms and published on the same Reddit and Facebook groups that were mentioned in **"4.1.1 Survey and analysis"** chapter. A feedback survey contains three questions of linear-scale type, two multiple-choice type questions and one question of Fill in the Blank type. All answers are anonymous.

Survey can be completed on: https://forms.gle/uThyMRqq27LnW9ev5

Responses can be viewed on:

https://docs.google.com/spreadsheets/d/1joXunE5xBOTTwtv-p8DQ0c2pcaCPZCgt-3W_eR7EHF8/edit?usp=sharing

Overall, 23 people have filled in the survey and the results of the survey are automatically generated by Google Forms. The most interesting parts are represented below as diagrams:
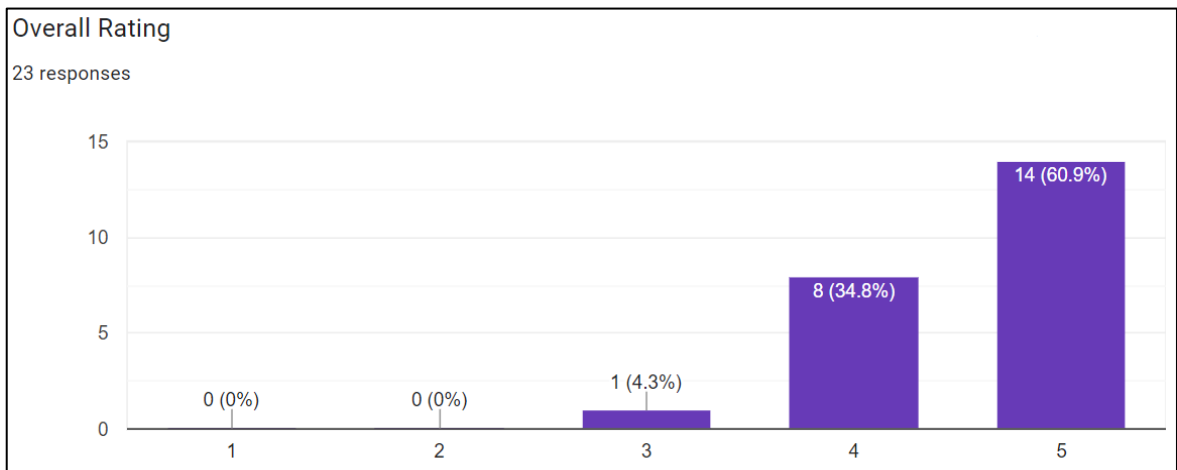
Figure 58 - "Overall Rating". Source: author
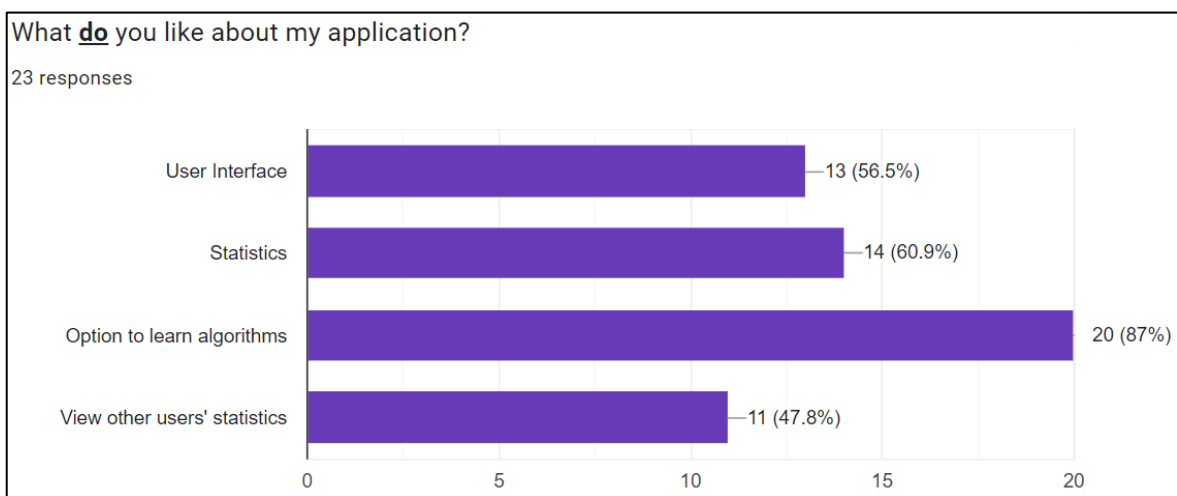
Figure 59 - "What do you like about my appication?". Source: author

Any recommendations?

8 responses

keep going you are doing great

It needs a stackmat timer style like ruwix's timer and I think you should also have an 'add times manually' button. Sometimes I time with other devices than upload them to ruwix website or whatever. Very cool timer, with a few adjustments it might be one of the best timers online!

Try to add custom settings like for holding the timer to start it and not just by clicking .

Add new languages, poor English I have, Czech Language I want

Add light theme

Figure 60 - "Any recommendations?". Source: author

4.10.1 Feedback summary

According to collected data from feedbacks, first version had a good start. Its average rating is 4.57 out of 5 received from 23 reviewers. Their recommendations will be taken into consideration, and with my personal opinion, new features will be added in further development.

4.11 Further development

After receiving feedback from experienced speedcubers, next version of the application would have these upgrades:

- An option to add cubes from 2x2 to 7x7, pyraminx, megaminx, skewb and others, so a user can choose what cube to record statistics of;
- A visual 2D representation how a cube should look like after it is being scrambled;
- Automatic update of users' statistics on real-time database, which includes best solve, worst solve, average of 5, average of 12, average of 100;
- An option to choose language;
- An option to change theme, font and background;
- Release a version for iOS;
- Alternative of Firebase that is supported by Unity WebGL;
- An option to add or remove solving time;
- An option to measure time for specific scramble or algorithm, so users can compare their skills under the same conditions.

5 Conclusion

The main objective was to develop a cross-platform speedcubing application and it was achieved successfully. The application is available on Android, PC and Web platforms. It received a good feedback with some recommendations from speedcubers.

The goal was achieved using literature review of current available online timers and applications for desktop and mobile devices. It gave a total overview on the present market with its strong and weak sides. This part also included the comparison of two main development engines, which conducted to a choice of a programming language. Moreover, the most suitable speedcubing method for either professionals or beginners was chosen from three the most popular methods and its algorithms were included in the application.

Starting working in practical part, target audience with its preferences and needs were defined using online surveys that was filled out by 192 respondents. Afterwards, the prototype of the application including requirements, state machine diagram, use cases and wireframes was created. It marked the beginning of the application development. Built product matched its prototype and all requirements were met, therefore it was uploaded to all three different platforms.

Finally, the application was tested by speedcubers from all over the world and received their feedback for the further development.

The project application is a very good opportunity for all speedcubers with different level of skills to learn new algorithms, measure solving times, review and share the statistics, and compete with each other.

6 References

1. About Rubik's cube. *Rubik's.* [Online] Spin Master Ltd. [Citace: 10. 8 2022.] https://rubiks.com/en-US/.

2. Persons. *World Cube Association.* [Online] 2022. [Citace: 10. 8 2022.] https://www.worldcubeassociation.org/persons?page=1&region=all&search=.

3. Records. *World Cube Association.* [Online] 2022. [Citace: 8. 10 2022.] https://www.worldcubeassociation.org/results/records.

4. The Speed Cubers. *Netflix.* [Online] Netflix, 2020. [Citace: 9. 10 2022.] https://www.netflix.com/cz-en/title/81092143.

5. Kemp, Emma. "The popularity has just completely exploded": Rubik's Cube's second coming. *The Guardian.* [Online] 10. 12 2021. [Citace: 9. 10 2022.] https://www.theguardian.com/sport/2021/dec/10/the-popularity-has-just-completely-exploded-rubiks-cubes-second-coming.

6. Cripps, Lucy. Get your speedcubing fix from the Rubik's Cube World Cup 2021 right here. *Red Bull.* [Online] Red Bull, 4. 12 2021. [Citace: 9. 10 2022.] https://www.redbull.com/nz-en/red-bull-rubiks-cube-world-cup-live-updates.

7. Ferenc, Denes. Rubik's Cube Algorithms. *Ruwix.* [Online] [Citace: 10. 10 2022.] https://ruwix.com/the-rubiks-cube/algorithm/#:~:text=To%20describe%20operations%20on%20the,apostrophe%20is%20a%20counterclockwise%20turn..

8. Elbrecht, Cornelia. The Sensorimotor Hands-Brain Connection. *Sensorimotor art therapy.* [Online] 29. 07 2020. [Citace: 10. 10 2022.] https://www.sensorimotorarttherapy.com/blog/2020/7/29/the-sensorimotor-hands-brain-connection.

9. Turner, Ash. Android vs. Apple Market Share: Leading Mobile Operating Systems (OS) (Mar 2023). *Bankmycell.* [Online] 03 2023. [Citace: 1. 03 2023.] https://www.bankmycell.com/blog/android-vs-apple-market-share/#:~:text=Android%20continues%20to%20reign%20in%20the%20global%20market%20with%20a,used%20by%20many%20smartphone%20brands.

10. How Google Play works. *Google.* [Online] [Citace: 12. 10 2022.] https://play.google.com/about/howplayworks/.

11. Omega Studio. Finger Timer. *Google Play.* [Online] 10. 12 2018. [Citace: 4. 1 2023.] https://play.google.com/store/apps/details?id=air.tw.url.omega.FingerTimer..

12. Rodriguez, Pelayo. SpeedCube Timer - Rubik Chrono. *Google Play.* [Online] 29. 4 2022. [Citace: 4. 1 2023.] https://play.google.com/store/apps/details?id=com.projects.pelayo.speedcubetimer.

13. Pilow. Cube Timer. *Google Play.* [Online] 30. 12 2022. [Citace: 4. 1 2023.] https://play.google.com/store/apps/details?id=com.avelsoft.cubetimer.

14. Aubinet, Nicolas. Nano Timer. *Google Play.* [Online] 8. 5 2020. [Citace: 4. 1 2023.] https://play.google.com/store/apps/details?id=com.cube.nanotimer.

15. Gottlieb, Michael. *qqtimer.* [Online] [Citace: 4. 1 2023.] https://qqtimer.net/.

16. Zhang, Shuang Chen and Ye. *CsTimer.* [Online] [Citace: 4. 1 2023.] https://cstimer.net/.

17. Scorecount.com. Ruwix Timer. *Ruwix.* [Online] [Citace: 4. 1 2023.] https://ruwix.com/online-rubiks-stopwatch-timer/.

18. Mark. Timer. *SolveTheCube.* [Online] [Citace: 4. 1 2023.] https://solvethecube.com/timer.

19. McNeil, Dallas. Block Keeper. *Dallas McNeil.* [Online] 2017. [Citace: 4. 1 2023.] https://dallasmcneil.com/projects/blockkeeper/.

20. Google. Android Studio. *Android.* [Online] [Citace: 4. 1 2023.] https://developer.android.com/studio?gclid=Cj0KCQiAx6ugBhCcARIsAGNmMbhRQwL Y80daTECg- 385arme_s7c3BwmipdjynwQ1Q1jxu0okkXQV88aAsdJEALw_wcB&gclsrc=aw.ds..

21. Microsoft. Visual Studio Code. *Visual Studio.* [Online] Microsoft. [Citace: 4. 1 2023.] https://code.visualstudio.com/.

22. Atom. *Softonic.* [Online] GitHub, 7. 6 2022. [Citace: 4. 1 2023.] https://atom.en.softonic.com/?ex=DINS-635.2.

23. Unity Technologies. Make it faster. Share it sooner. *Unity.* [Online] Unity Technologies, 2023. [Citace: 4. 1 2023.] https://unity.com/.

24. Schardon, Lindsay. What is Unity? - A Guide for One of the Top Game Engines. *Gamedevacademy.* [Online] 13. 1 2023. [Citace: 1. 2 2023.] https://gamedevacademy.org/what-is-unity/.

25. The world's most open and advanced real-time 3D creation tool. *Unreal Engine.* [Online] Epic Games. [Citace: 1. 2 2023.] https://www.unrealengine.com/en-US.

26. Introducing The Matrix Awakens: An Unreal Engine 5 Experience. *Unreal Engine.* [Online] 9. 12 2021. [Citace: 1. 2 2023.] https://www.unrealengine.com/en-US/blog/introducing-the-matrix-awakens-an-unreal-engine-5-experience.

27. N-iX. Unity vs. Unreal: How to Choose the Best Game Engine. *Medium.* [Online] 11. 9 2018. [Citace: 1. 2 2023.] https://medium.com/@N_iX/unity-vs-unreal-how-to-choose-the-best-game-engine-d3dbb4add73c.

28. C# Tutorial. *Javatpoint.* [Online] https://www.javatpoint.com/c-sharp-tutorial.

29. Curator. A Complete C# Tutorial For Beginners to Advanced. *C-sharpcorner.* [Online] 13. 2 2023. [Citace: 16. 2 2023.] https://www.c-sharpcorner.com/article/C-Sharp-tutorial/.

30. Bill Wagner, tdykstra, rodrimc, v-kents, scottaddie. A tour of the C# language. *Microsoft.* [Online] 13. 2 2023. [Citace: 16. 2 2023.] https://learn.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/.

31. C# Introduction. *W3schools.* [Online] [Citace: 16. 2 2023.] https://www.w3schools.com/cs/cs_intro.php.

32. Andrew Stellman, Jennifer Greene. *Head First C#: A Learner's Guide to Real-World Programming with C# and .NET Code 4th Edition.* místo neznámé : O'Reilly Media, 2021. 978-1491976708.

33. Price, Mark J. *C# 10 and .NET 6 – Modern Cross-Platform Development: Build apps, websites, and services with ASP.NET Core 6, Blazor, and EF Core 6 using Visual Studio 2022 and Visual Studio Code, 6th Edition 6th ed. Edition.* místo neznámé : Packt Publishing, 2021. 978-1801077361.

34. Chan, Jamie. *Learn C# in One Day and Learn It Well: C# for Beginners with Hands-on Project (Learn Coding Fast with Hands-On Project).* místo neznámé : CreateSpace Independent Publishing Platform, 2015. 978-1518800276.

35. Firebase Realtime Database. *Google.* [Online] 3. 3 2023. [Citace: 4. 3 2023.] https://firebase.google.com/docs/database.

36. CFOP method. *Speedsolving.* [Online] 20. 10 2022. [Citace: 4. 11 2022.] https://www.speedsolving.com/wiki/index.php/CFOP_method.

37. Petrus Method. *Speedsolving.* [Online] 2025. [Citace: 4. 11 2022.] https://www.speedsolving.com/wiki/index.php/Petrus_Method.

38. Roux Method. *Speedsolving.* [Online] 24. 11 2022. [Citace: 6. 12 2022.] https://www.speedsolving.com/wiki/index.php/Roux_method.

39. Ghodgaonkar, Abhijeet. Which is Better: CFOP or ROUX? *Cubelelo.* [Online] 29. 11 2022. [Citace: 7. 12 2022.] https://www.cubelelo.com/blogs/cubing/which-is-better-cfop-or-roux.

40. Kocher, Chris. Watson professor known as puzzle-solving pioneer. *Binghamton University.* [Online] 17. 5 2021. [Citace: 8. 12 2022.] https://www.binghamton.edu/news/story/3060/puzzle-solving-pioneer.

41. CFOP 3x3 Rubiks Cube Algorithms. *Speedcube.* [Online] [Citace: 16. 1 2023.] https://www.speedcube.com.au/pages/cfop-3x3-rubiks-cube-algorithms.

42. Google Forms. *Google.* [Online] Google. [Citace: 12. 2 2023.] https://www.google.com/forms/about/.

43. Faller, Patrick. Putting Personas to Work in UX Design: What They Are and Why They're Important. *Adobe.* [Online] 17. 12 2019. [Citace: 12. 1 2023.] https://xd.adobe.com/ideas/process/user-research/putting-personas-to-work-in-ux-design/.

44. Manning, Cheniece. User Persona Template. *Figma.* [Online] 2022. [Citace: 24. 2 2023.] https://www.figma.com/community/file/1015759429756106784.

45. The easiest way for Confluence teams to collaborate using diagrams. *Draw.io.* [Online] [Citace: 25. 2 2023.] https://drawio-app.com/.

46. Microsoft PowerPoint. *Microsoft.* [Online] Microsoft Corporation. [Citace: 21. 2 2023.] https://www.microsoft.com/en-ww/microsoft-365/powerpoint.

47. Random. *Unity Documentation.* [Online] 3. 3 2023. [Citace: 4. 3 2023.] https://docs.unity3d.com/ScriptReference/Random.html.

48. Add Firebase to your Unity project. *Firebase.* [Online] [Citace: 26. 2 2023.] https://firebase.google.com/docs/unity/setup.

49. Everyone can. Photoshop. *Adobe.* [Online] [Citace: 27. 2 2023.] https://www.adobe.com/products/photoshop.html.

50. Easy and secure access to your content. *Google.* [Online] [Citace: 24. 2 2023.] https://www.google.com/drive/.

7   List of figures, tables and source code

7.1   List of figures

7.2   List of tables

7.3   List of source code