**PHASE 1**

**Project Spécification Document**

**Rubik’s Cube Timer**

Picture

Developed by Nsovo Ngobeni

**Signoffs**

|  |  |  |  |
| --- | --- | --- | --- |
| **Phase** | **Teacher** | **Date** | **Signature** |
| 1. Project Specification Document | Mrs Rajamany |  |  |
|  |  |

**1. Problem Summary**

Cubers are people who compete to solve the Rubik’s cube as fast as possible. This app is made to help professional and amateur cubers achieve this goal. The app will help them by tracking their best times and calculating their average time; showing them where they need to improve.

The app will also function as a practice tool for cubers; helping cubers both scramble and solve the rubik's cube efficiently. Scrambles must be done to tournament standards in order for the time to be valid. This app will provide tournament standard Rubik’s cube scrambles for the user to solve, helping them both practice and improve their skills.

The app will first display a scramble - a set of moves that scrambles the cube sufficiently before it can be solved - then display a timer that will be started/stopped using the spacebar. The time and the scramble associated with it will be tracked and recorded for further processing. The app will accumulate all of the solves recorded by the user and display it in a way that will be useful for cubers. It will display the average of your last 5 solves - in order to track your progress -, the average of all your solves, and the average for your current session. All this data will be easily accessible in their corresponding screens.

**2. Motivation and Research**

**Motivation**

I would like to make a cube timer app as cubing has been a hobby of mine for 6 years now. Throughout these 6 years I have improved a lot but I was unable to find a good app to help me practice and reach my goal of participating in an official cubing competition. This app is here to help people like me who want to find a way to effectively practice and improve their times but were unable to find a good way to do it.

**JPERM.NET**

jperm.net is a website that aims to achieve the same basic goal as my website, to help cubers improve their times and hone their skills. However the app comes at the challenge in a completely different way. Instead of providing functionality to time and complete the entire cube (CFOP), it tackles every step individually.

The solving of a rubik's cube can be broken down into 4 distinct parts, CROSS, F2L, OLL and PLL (referred to collectively as CFOP). jperm.net times one step at a time which allows cubers to hone in on the specific areas where they think they can improve. My app on the other hand will time the entire process of solving the cube, not just a specific part of it.

While focusing on specific problem areas is a great way to improve, completing the entire process is more representative of an actual tournament/competition and the ultimate goal of my app is to help users best prepare for these competitions. It is for this reason that my app will ultimately differ from jperm.net

**Cube Timer and Scrambler LITE**

This is an ios app that approaches the goal of helping cubers improve in the same way as my app does. Both apps aim to help cubers by tracking their times and providing copious amounts of practice.

However this app is missing key performance indicators and much of the data recorded by the app is not made available to the user. My app will have more functionality in terms of how the data recorded is processed and shown to the user. My app will allow users to see all their recorded times and the scramble associated with them, they will be able to see all their averages and the average of their last couple times, providing insightful information on their general performance and improvement.

These additional features are there to help the user get a better understanding of their overall performance and all will come in handy when practicing.

**3. Features of the program**

**2.1. Specifications of Program Functions**

**The program will be able to do the following:**

|  |  |
| --- | --- |
| **Program Functions** | **Justification** |
| Add/Delete Time | Users must be able add recorded times to the database. Added times are stored and for further calculations. The users must also have the option to delete a time in the database. |
| DNF Current Time | Sometimes times must be marked as DNF’s  (did not finish). This happens when the user  decides to skip the current solve or is unable to  finish it. |
| Display Averages (ao5, ao7, session average) | Averages provide useful insight into general  trends and will provide the user with extra  insight on their performance and improvement  over their solve times. The ao5 is also a WCA  competitive standard which is used to decide  the winners. |
| Display Recorded Times | Users must be able to access the their records  and data for all times they have recorded. |
| Sort Data | Users must be able to see their data in easy  and convenient ways. The data should be sorted by either the date the time was completed (asc/desc) or the time taken to solve (asc/dec) |

**2.3. Specifications of User Interface**

**The user-interface will have the following:**

* The user

###### Timer Screen (Main Screen):

● A digital timer that will count up when triggered

● A button to start/stop the timer (can also use the space-bar)

● A button to mark current time as a DNF

● A button to switch to data screen

**Stats Screen:**

● Scrollbar that will let users choose what average they would like to display ● A list of all times recorded as well as their corresponding times

● A scrollbar that let users choose how they want the data to be displayed ● A button next to the scrollbar that sorts the data based on the scrollbar selection ● A button to change to the Timer Screen / Main Screen

**Practice Screen:**

* A scrollbar that lets users scroll through all the different algorithms available
* Each algorithm can be clicked to open a popup that allows users to edit the data
  + The popup

**Help Screen:**

**2.3. Specifications of Help**

**The following help features will be available:**

* The app should be intuitive to use, using common design principles which make for a better, simpler user experience. The use of common app design principles and an intuitive user interface will decrease the likelihood of confusion and ambiguity around the apps functions.
* A help page will also be accessible to the user that will describe key concepts which need to be understood and a guide on how to use the entirety of the app's functionality. The page will be accessible through a button in the upper left corner of the screen (indicated by a “?” symbol). This page will contain instructions, accompanied by appropriate visual guides, that will help users better understand app functionality
* The help page will have guides that go over:
  + How to start/stop the timer
  + How to add/delete/amend a time
  + How to access different screens (Stats/Main/Help)
  + How to use the data sorting functions
  + Explanation of competitive DNF rules and how they are applied within the app

**2.4. Specifications of Permanent Data Storage**

**The following data storage will be available:**

A single database table with four fields - an ID, the scramble algorithm, the time taken to solve, and the date and time of the solve - will be used.

|  |  |
| --- | --- |
| **Field** | **Data Type** |
| ID | AutoNumber |
| Algorithm | Short Text |
| SolveTime | Number |
| TimeSolved | Date/Time |

**3. Hardware, Software and Installation Requirements**

**Minimum Hardware Requirements:**

Processor: Minimum Pentium 2 266 MHz processor

RAM: 2 GB

Disk Space: 256 MB

**Minimum Software Requirements:**

OS: Windows 8 / Mac OS X 10.8.3

Java 8: version 8u51 or above