UnityRun

Software Requirement Specifications - v1.0

**S.R.S.**

**Submitted by:**

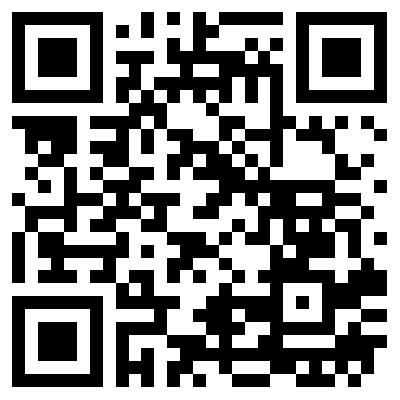
**Mullifiers**

**PARNEET KAUR 401503006**

**PULKIT GARG 101503172**

**Group: COE-12**

(Third Year)

****

**Link to github repository**



**COMPUTER SCIENCE ENGINEERING DEPARTMENT**

**THAPAR UNIVERSITY, PATIALA-147004, PUNJAB**

**INDIA**

**October 2017**

**TABLE OF CONTENTS**

**Chapter No. Topic Page No.**

1. Introduction

1.1 Purpose of this Document 3

1.2 Scope of the Development Project 3

1.3 References 4

1.4 Overview 4

2. Overall Description 5

2.1 Product Perspective 5

2.2 Product functions 6

2.3 User Characteristics 6

2.4 General Constraints, Assumptions and Dependencies 6

2.5 Apportioning of the requirements 7

3. Specific Requirements 8

3.1 External Interface Requirements 8

3.2 Detailed Description of Functional Requirements 8

3.2.1 Functional Requirements for Student Welcome Screen 9

3.2.2 Functional Requirements for Staff Welcome Screen 9

3.3 Performance requirements 10

3.4 Quality attributes 10

3.5 Other requirements 10

4. Change History 10

5. Document Approvers 10

1. **Introduction**
   1. **Purpose of this Document**

The purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality.

* 1. **Scope of the Development Project**

it is intended for unifying all the people around (even 7 seas apart ) to provide motivation for them to run (or jogging in case) in order to kill obesity or remain physically fit and active

this app is real-time multiplayer platform and is integrated with facebook much alike the famous, 8-ball pool game. if n number of players are playing at a given instance, each each of them will receive phychological remarks n tricks such as :-

1. 5m 4m 3m 1m to beat {player name}
2. congratulations on crossing over
3. {player name} is walking and got his/her points reduced by {count} ;P

there will be a showcase of certain tracks to choose from and each will have some different kind of challenege:

1. basis on number of jumps one has to perform in between to score more points
2. runnning speed consistency
3. beating maximum speed with sprint
4. more to be added..… in next versions of srs

after selection of track from main menu, one can send invites through facebook n check if the opponent is online or not otherwise notification will be sent.

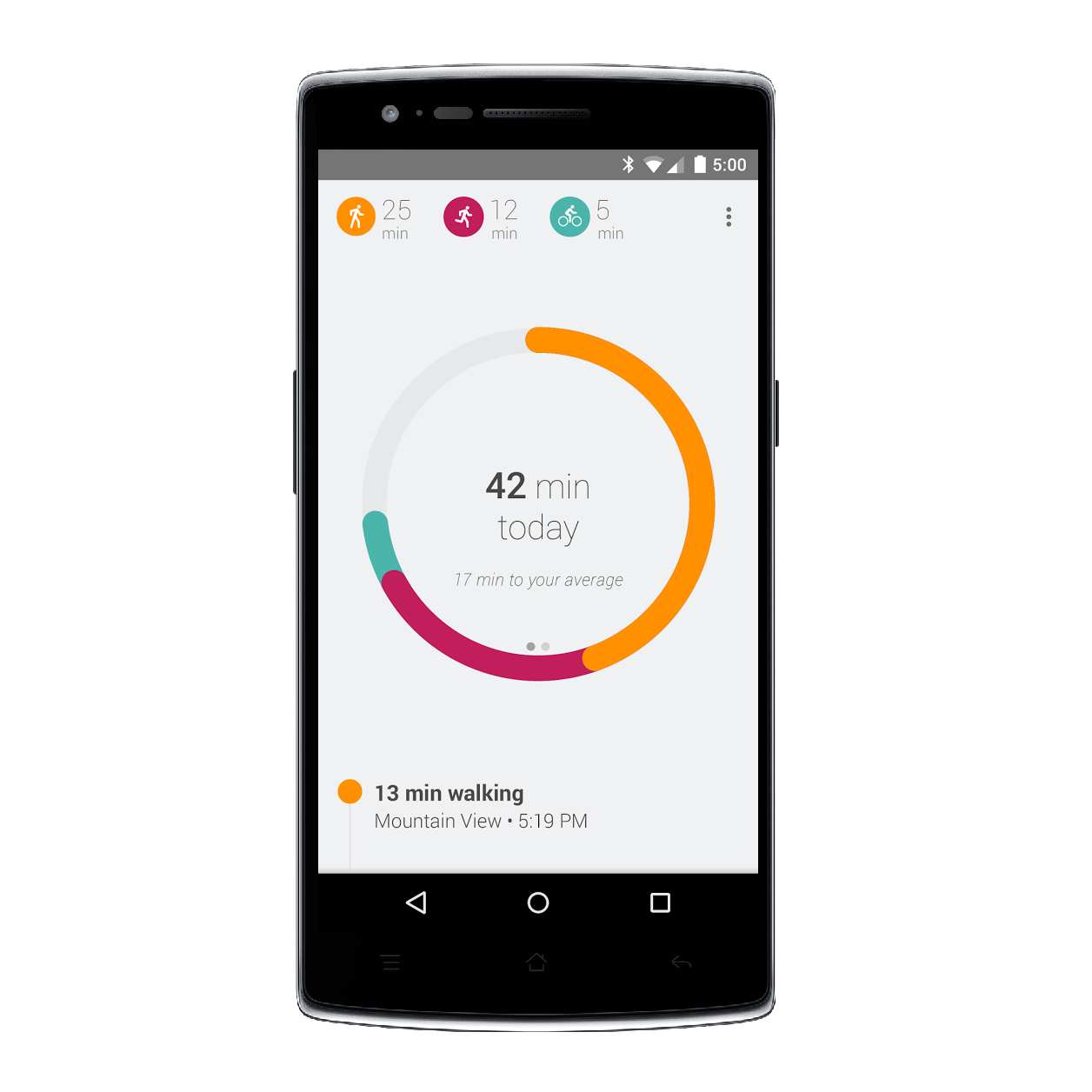
* 1. **References**
     1. <https://docs.unity3d.com/Manual/UNetLobby.html>
     2. <https://developers.facebook.com/docs/graph-api/>
     3. <https://store.unity.com/products/unity-personal>
     4. https://play.google.com/store/apps/details?id=com.google.android.apps.fitness&hl=en
  2. **Overview**

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 2 gives the functional requirements, data requirements and constraints and assumptions made while designing the system. It also gives the user viewpoint of product use. Section 3 gives the specific requirements of the product. Section 3.0 also discusses the external interface requirements and gives detailed description of functional requirements.

1. **Overall Description**
   1. **Product Perspective**

The product does not require any additional device that a person does not have. It only needs an android phone or any ios device ( as it is a cross-platorm app ) to use all the features of the product. The current makes and the hardware builds since 2013 for every device have all the sensors required for the product to work effeicently

Below are some diagrams showing the flow of the application:



This figure shows the basic status screen to show a user scrores and health remarks to the application (ios/Android).

The user is asked to enter the basic details while registering like login with facebook account, height to weight ratio in order to tell recommendation of challenges by computing BMI , before this screen appears

All this data gets saved in the facebook graph database when the game seession is terminated

*Figure-1 User Status Screen and flow of the application*

* 1. **Product Functions**

The product should be able to perform the following operations:

* + 1. It must be able to authenticate the user by facebook login and able to get friends list through facebook graph APIs and set the score back into thier id’s progress database using Oauth2.0.
    2. Must be able to provide real time information about the players distant apart accurately and without much latency in communication.
    3. The server is to provide processed data from consolidation of orientaion, completion % , other sensory data from all the players in a session and asynchronously able to drive and inittiate, all event based messages across the players
    4. Should ideally provide an intuitive GUI for all components, such as for users to submit documents and for administrators to oversee scheduling.
  1. **User Characteristics**

This app is essentially made for any person from a a tennager to adult termed as unity-player and the player should able to use the interface of app well enough, have an account on facebook, able to grant app access permissions and, a know a brief introduction of app’s graphic user interface although it will be highly intuitive but still should be aware enough to get the intuition too.

Its well suited to any type of audience or players suchas atheletes , obesse people, or high school students, even cycling players (A feature request implementable in SRSv2.0).

**2.4 General Constraints, Assumptions and Dependencies**

The following list presents the constraints, assumptions, dependencies or guidelines that are imposed upon implementation of the system:

* + 1. The software has to be integrated onto the reader-writer terminal that in turn has an extremely small form factor and has support for limited capability APIs only.
    2. Due to the small form factor, only limited graphics can be supported on the display screen.

1. There are memory requirements for documents uploaded and to keep a track of the user’s login information.
2. The system must have a user friendly interface that is simple enough for all types of users to understand.
3. Response time for loading the software and for retrieving a document should be no longer than five seconds.
4. A general knowledge of basic computer skills and of basic working of login based system is required to use the product.
5. The central database server and backup database servers should be updated regularly. This updating and replication of data from central database server to the backup database server can introduce additional latency in the working of the system.

Due to the lack of time, we are unable to fully implement security features in this product.

**2.5 Apportioning of requirements**

The Resource Sharing System is to be implemented in the following three phases:

1. **Pilot Phase:** Here the login based system including document sharing will be implemented among a batch of 100 people (of which 16 are teachers and remaining are students). Initially we will be providing access privileges for two types of users: student and teachers as they will be ones most involved in this phase.
2. **Institute wide deployment:** Following the successful completion of the pilot phase, we plan to deploy the same across other batches in the college. Apart from document sharing, attendance monitoring functionality can also be added in the system which will help the students to keep a track of their attendance and stay well above the mark of 75%. And also help teachers to get rid of the attendance registers and reduce their paperwork.
3. **Extension of Resource Sharing System to other applications:** In the future we can have a single student portal for example serving different purposes like purchasing a textbook at the Institute’s Co-operative store or paying the semester fees, an application (like the webkiosk) that will really boost the utility value of the system. Here the same functionalities will be implemented in each phase; the only difference will be the number of transactions being carried out and the scale of implementation.

**3. Specific Requirements**

**3.1 External Interface Requirements**

The following list presents the external interface requirements:

* The product requires a limited and not so fancy graphics usage with a login with facebook screen , fb friends invitation send/receive portal, main menu to do selection of tracks / challenges and a user statics interface
* The product requires an apt usage of sound through earphones or other similar external accessories. The hardware and operating system requires a screen resolution not more than 320 x 240 pixels (owing to the small form factor).
* Animation is not an essential feature but it can be considered for future variants of the system in the form of a home screen wherein the user will select the tracks/challenges by scrolling sideways , also, analog and some moden designed animated UI elements preferably from graphicsriver, envato market corp.

**3.2 Detailed Description of Functional Requirements**

The table below shows a template that we’ll be using to describe functional requirements for two types of users: student and teachers as one can easily deduce the functional requirements for other user types with this template.

|  |  |
| --- | --- |
| **Purpose** | A description of the functional requirements and its reasons |
| **Inputs** | What are the inputs; in what form will they arrive; from what sources can the inputs come; what are the legal domains of each input. |
| **Processing** | Describes the outcome rather than the implementation; includes any validity checks on the data, exact timing of operation (if needed), how to handle unexpected or abnormal situations |
| **Outputs** | The form, shape, destination and volume of output; output timing; range of parameters in the output; unit of measure of the output; process by which output is stored or destroyed; process for handling error message produced as output. |

**Table 1: Template for describing functional requirements**

**3.2.1 Functional Requirements for Facebook Login Screen**

The table below gives the functional requirements for Student Welcome Screen.

|  |  |
| --- | --- |
| **Purpose** | This screen thus provides information specific to each user signedin via facebook oauth2.0 API, upon the successful identification of the ID no. and the access code with the values stored in the central indevice database |
| **Inputs** | A user need to provide his/her fb userID-email/phone no. And the password in the external browser window thus opened and grant the permission access to the app |
| **Processing** | User will get signedin via facebook oauth2.0 API, upon the successful identification of the ID no. and the access code with the values and session cookies, stored in the central indevice database. |
| **Outputs** | The screen will then redirect to facebook friends invitation screen with a message output as “successfully loggedin with facebook” |

**Table 2: Functional Requirements for Facebook Login Screen**

**3.2.1 Functional Requirements for User health and scores stats screen**

The table below gives the functional requirements for Student Welcome Screen.

|  |  |
| --- | --- |
| **Purpose** | This screen thus provides information like score statistics, health remarks such as calories burned, number of jumps remaining to be performed for completing the challenge etc.. specific to each player in a multiplayer realtime virtual room upon the successful identification of the ID no. and the access code with the values stored in the central database server. |
| **Inputs** | A student can view a page of information by choosing from one of the options given on the welcome screen. Selection is performed with a simple keypad. |
| **Processing** | This will process the event based data streamed from the unity server and make inferences about health remarks and updates the current score indevice database. |
| **Outputs** | It will provide information like score statistics, health remarks such as calories burned, number of jumps remaining to be performed for completing the challenge etc.. to the output screen enhanced by stunning modern UI elements |

**Table 2: Functional Requirements for User Stats Screen**

**3.2.2 Functional Requirements for Friends Invitaion Screen**

The table below gives the functional requirements for Teacher Welcome Screen.

|  |  |
| --- | --- |
| **Purpose** | To make a virtual compettion between the players remotely, it will provide people (fb friends) to choose as opponents and challenge them. |
| **Inputs** | A user needs to select the rival/opponent from the friend list, there will two options either send invite or challenge |
| **Processing** | It sends challenge to user only if has online status and running the app in background otherwise, will send fb notification with the use of fb graph APIs to the opponent as much as like a candy crush request. |
| **Outputs** | The screen will be redirected to track/challenge selection portal then |

**Table 5: Functional Requirements for Friends invitation Screen**

**3.3 Performance Requirements**

* The system is designed for the smartphone with any of the below listed platforms : Ios/apple ,android/google and custom android based ROM variants.
* The app needs to send data in an interval of 500 miliseconds to the multiplayer realtime server (provided by unity inc.) with datagram networking approach so,not maintaining a good consistent internet connection may be a pitfall.
* All the voice recordings used for motivating the user will be in the indevice database only, which account for a 100 mb additional game data to be downloaded , through usually, any game 2D or 3D requires a handful amount of texture and ambience game data. Our goal is to minimize the rom space used by aid of compression via rar and zip preferably, if possible
* Our primary motive in the performance enhancing section is to minimize manipulation errors such as counting a jump if the user get on a pleateu kind surface, noices in the sensory data by aid of a user action which will do the well calibration of sensors.
* Not exceeding the limits and bounds of request processed by the unity realtime server, to avoid any hidden costs encountered to the company .Unity personal pack provides 14,000 requests in a session of 1 hour within one multiplayer game-room

**3.4 Quality Attributes**

The product is target towards a specific variety of users such as students, staff, etc. The product must load quickly and work well on a variety of terminals and must be cross-platform without bugs. It must also tolerate wide variety of input possibilities from a user, such as cycling intead of running which may bring crap data onto common leaderboard database portal.

**3.5 Other Requirements**

None at this time. Any modifications will totally depend on the amount of traffic which will be handled by the system in the pilot phase.

**4. Change History**

Version 1.0 – Initial Release

**5. Change History**

SRS for realtime multiplayer jogging remote interaction system (including scores and leaderboard) approved by:

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Name:**

**Designation:**

**Date:**