Danielle Bosonnet

Business Information Systems
Year 4

07669402

Funky Fitness

www.funkyfitness.heroku.com

Technical Report

1 Executive Summary

The main objective of this project is to provide an online website which was developed for a personal trainer Carol. It enables the user to use the website to track their fitness online. To begin using the website the user will need to sign up and become a member to avail of the full range of services on offer. Each user will have their own profile page and they can follow (similar to Twitter) other users of my application. The website was built using Ruby on Rails 3 running on Ubuntu Operating system.

The website offers the following features:

- Online fitness programme designed by a personal trainer based on the users age.
- A section on nutritional information, sample menu's for breakfast, lunch , dinner and snacks
- An interactive diagram of the muscles in the body and an explanation of how to exercise each body part correctly
- BMI(Body Mass Index) and BMR(Basal Metabolic Rate) calculators
- Availability to link progress to social network sites – Twitter and Facebook
- Videos of different exercises that do not require equipment
- A feedback survey

2 Introduction

2.1 Background

The student decided to create this application because there is currently not a website that provides all the features needed to track fitness online. Obesity is a growing problem in today's society however accessing information that enables users to gain a greater understanding of fitness and exercise is a major issue and currently not very widely available. As the student has a keen interest in fitness and nutrition they felt that this will enable them to enjoy it more.

As this project was proposed and carried out last year using html, xhtml, css and JavaScript, so the student decided to use Ruby on Rails as the programming language which operates on a variety of different platforms such as Windows, Macintosh and Linux, thus creating an entirely different project but based on the same subject area

The customer is a fully qualified gym instructor in Top Level gym in Skerries. She has been working there for 5 years and also teaches step aerobics and spin classes. The customer was approached by the student in September and asked to be the customer, she accepted and the project began

2.2 Aims

The main aim for this project was to create an online fitness application that fully meets the requirements of Carol Nevin (Customer). Another aim was to develop a website that will be used by people with a wide range of technological abilities and backgrounds. The website also aimed to provide interactivity to the users, around the muscles of the body and different types of exercises.

2.3 Technologies

Due to the nature of the project and the fact that a number of users will access it, the website needed to operate on a variety of different platforms such as Windows, Macintosh and Linux. With this in mind I decided to use Ruby on Rails 3 running on Linux OS. I also used mySqlite3 ad my database to store the user's information.

2.4 Structure

The first chapter contains the Executive summary which will give a short overview to the reader of the whole report

The second chapter outlines the background and the reason the project was decided, the aims of the overall project and the technologies used to implement the project.

The third chapter describes the overall system in terms of Functional Requirements, Data Requirements, Usability Requirements, Design and architecture of system, How the project was implemented, testing carried out, how the Graphical User Interface (GUI) was laid out, Customer testing involved and the evaluation containing how the system was evaluated and what the results were

The fourth chapter contains the conclusions made by the student on completion of the project

The fifth chapter contains the further development or research that is planned by the student.

The Sixth chapter contains the bibliography of the all the resources used to complete the project.

The final chapter contains the Appendix which includes the Project Proposal, the Project Plan, the Requirement Specification and Monthly logbooks and any other materials used.

3 System

3.1 Requirements

3.1.1 Functional requirements

3.1.2 Requirement 1

The user must register.

3.1.2.1 Description & Priority

This requirement is essential as the user must register in order to access the website.
Therefore nothing can be done unless the user is a member.
For this function to work the database must be set up to store the information.
This is a high priority requirement.

3.1.2.2 Requirement Activation

This requirement is essential as when the user first accesses the webpage they will be prompted to register, so if they are not already a member they must sign up to become a member

3.1.2.3 Technical issues

A database must be created in order to satisfy this requirement the database will store all the users information which includes username, passwords etc.

3.1.2.4 Risks

This requirement might not be able to be satisfied if the user unable to log in, this could be caused by the user entering the wrong username or password, the user must ensure then enter the correct details in order for log in to be successful.
An action can be taken to reduce the probability of this happening, on the login page there will be a forgot your password button when upon clicked the user will be asked for their email address, once submitted their password will be sent to their email address.

3.1.2.5 Dependencies with other requirements

If the user is not a member they won’t be able to access the webpage so login/register is essential.

3.1.2.6 Functional Requirements

3.1.3 Requirement 2

Create user profile

3.1.3.1 Description & Priority

Upon registering the user will be asked to provide some personal information in order to create their profile e.g. name and age.

If this information is not provided to the system the system will be unable to create a profile for the member.

3.1.3.2 Requirement Activation

This requirement will allow the user to enter personal information about them.

3.1.3.3 Technical issues

A form will have to be created in order to allow the user to enter their information and this form must be linked to a database which will store the information

3.1.3.4 Risks

The user may enter the wrong information accidently, if this happens the user will be able to click an edit button, once clicked the user can change the information contained in the form/on profile.

3.1.3.5 Dependencies with other requirements

This requirement will interact with the database.

3.1.3.6 Requirement Activation

This requirement will allow the user to enter their personal information.

3.1.4 Data requirements

Data requirements should consider the following:

- The data inputted should be in text format

3.1.5 Requirement

Register

3.1.6 Description & Priority

The user must enter their details in text format.
This is a high priority Requirement.

3.1.7 User requirements

The main requirement of the users is to sign up, once they sign up for the website they have full access to the website and can obtain there free fitness programme.

3.1.8 Environmental requirements

To complete the project the right environment needed to be set up on the students laptop. The student had to partition her hard drive and install Ubuntu operating system, then she had to install ruby on rails 3 and sqlite 3. The student chose to use Ubuntu instead of windows operating systems as it was advised by her supervisor and it is less buggy.

3.1.9 Usability requirements

The main usability requirement was to ensure that the application was completely user friendly and could be used by anyone no matter what there technical abilities are. By conducting the survey it’s was found that users felt the website was easy to use and simple.

3.2 Design and Architecture

To design the website the student used Ruby on Rails 3 which interacted with mySQLite 3 database, which stored all the user information submitted in the form when signing up. The student decided to keep the design of the website simple as it will be used by people with various different technical backgrounds. The student decided to keep the design simple, as user friendly as possible and professional to meet the requirements of the Carol and the users. This application is a web based application so the user will need internet access to use the application.

The first page the user will be presented with is the home page where they can sign up, once they register the will automatically logged in and they will be brought to their profile page which will be customized based on specific criteria like their followers and status feed On this page there are two links so the user’s can share their progress with their friends on social networking site – Twitter and Facebook..
The second page contains a list of different exercise videos which don’t require any equipment so user’s can follow along at home.
The third page contains the users personalize fitness program which is generated using the user’s age.
The fourth page contains sample recipes for Breakfast, Lunch Dinner and Snacks, the page is displayed with four categories (as above), once the user click the desired category a list Of different recipes for that category will be displayed along with the ingredients and instructions on how to cook the dish.
The fifth page will contain a BMI (Body Mass Index) and BMR (Basal Metabolic Rate) calculators. This allows the user to calculate each one and see there result
The sixth page contains an interactive diagram of all the muscles in the body and how to train each one and which equipment to use.
And the final page contains a survey about the website.
The external hardware needed will be a monitor, a CPU (Central Processing Unit), a mouse and a keyboard.

The main algorithm used in the application was the following IF Statement to generate the fitness program for the user :

```
def index
    if @age.to_i <= 25 # use to_i b/c the param will come in as a string
      render "program/under_25"  # this will render under_25.html.erb
    elsif @age.to_i <= 40
      render "program/under_40"
    else
     render "program/over_60"
    end
end
```

The IF statement goes to the database and checks if the age entered by the user is less than or equal to 25, if it is it then a fitness program, specific for users 25 or under is generated for the user and displayed on the programme page of the website, all the processing of the IF statement is done in the background and the user will only see their fitness program. If the age is between 26 and 40 the fitness program specific for users between 26 and 40 is generated for the user and displayed on the programme page of the website. Finally, if the age doesn't meet either of these categories an over 40's program is displayed.

3.3 Implementation

The main function in the users controller code was user authentication (Sign up, in and sign out). It was important to ensure that the user's password was encrypted in the database to protect the user's personal information and in case my database was hacked by an outsider also needed separate authentication method for normal users and admin users, as admin users have administrative privileges which allow them to edit, update, create and delete users. This was achieved by putting a before filter at the top of my controller so the system could identify what privileges the user logging in has. In the user model validations methods were used to ensure users entered the correct information, each validation method had different conditions e.g. password must be between 6 and 40. I used an Active record call-back and passing it to a before save filter, to save and encrypt the users password in my database.

Another important part of the website was the fitness programmes, which were automatically generated, based on the age the user entered when signing up. An IF Statement was used to achieve this. The IF Statement used the age column from the users table in the database to determine the age of the user and then it appended the correct programme based on age.

3.4 Testing

Test specifications attached on the CD in the spec folder. A Framework called RSpec was used for testing purposes using Test Driven Development (TDD) which means writing the test before the application code. RSpec is a domain specific language for describing the behaviour of code. It is used together with a programme called Rspec

In TDD, you have the write a failing test first (e.g. there should be an about page). After this we get the test to pass by adding the about action and the corresponding view. The reason we do it this way is to make sure the feature we are adding is actually tested.

A cycle Known as "Red, Green, and Refactor" was used during testing. The first step is Red, this refers to writing a failing test, and our test tools will indicate with the colour red. The next step refers to Green, this refers to passing a test, and the test tool will indicate this with the colour green. Once we have passed our tests we can refactor our code, i.e. eliminating duplication without actually changing the functionality.

3.5 Graphical User Interface (GUI) Layout

The application will interact with the user by using a Graphical User Interface (GUI). Most of the operation in a GUI is by mouse clicks and scrolling. This type of interface will suit the target market and will ensure that the site will suit all types of users and will keep in mind the different levels of technological ability. It consists of a number of different pages where each contains different information

3.6 Customer testing

Date:	Attendees:	Results:
30/10/10	Carol Nevin and Danielle Bosonnet	Carol was happy with the progress on sign up and sign out working but wanted a micro posting similar to twitter instead of a forum.
23/11/10	Carol Nevin and Danielle Bosonnet	From previous meeting micro posting was implemented and carol was happy with it and the progress made so far but wanted me to change the colour scheme used throughout the website.
10/12/10	Carol Nevin and Danielle Bosonnet	Showed Carol new colour scheme she was much happier but wanted me to change the layout a little, she also outlined that she wanted some interactivity within the website
11/02/11	Carol Nevin and Danielle Bosonnet	Carol was happy with the interactive diagram but suggested adding little pictures of the exercise beside the muscle description, we then discussed linking website to social networking websites: Twitter and Facebook.
31/03/11	Carol Nevin and Danielle Bosonnet	Review of the interactive diagram and carol was satisfied with the updates to it. And she was happy with the linkage to Facebook and twitter but asked if the buttons could be moved from the bottom of the page to the top.
26/04/11	Carol Nevin and Danielle Bosonnet	Discussed changes to social networking buttons and carol asked if we could implement a survey that users can take to evaluate the website.
12/05/11	Carol Nevin, Sharron Walsh and Danielle Bosonnet	Survey implemented and carol was happy with it. Final demonstration of fully working website, to the manager of the community centre and Carol

The system evolved over time by identifying Carol's requirements and then implementing them step by step, over the timeline of the project.

3.7 Evaluation

The system was evaluated using an online survey which is attached in the appendix. The survey revealed the following results:

1. 96.6% of the people surveyed said it was easy to navigate around the website.
2. 58.3% of those surveyed said they would visit Funky Fitness again.
3. 3.4% said it was difficult to navigate around the website and 83.3% of those surveyed were satisfied with the website.
4. 88.1% rated the information either ok brilliant or excellent. 78.3% rated the layout and colour scheme as ok, good and excellent.

3.8 Discussions

From the results of the survey the following comments were made:
During the evaluation with Carol and the project supervisor- Paul Stynes a decision was made to keep the amount of information available minimal to avoid reputation of the same code, as a result this led to comments from users that more information e.g. pictures, recipes and exercises were needed. From those surveyed they also commented the following:

1. To move the footer to the top of the page instead of having it at the bottom as when users login they need to scroll to the bottom of the page to use different functions of the website.
2. Breakdown of calories, fat etc in recipes
3. Accessibility options like ability to choose font size
4. Background music

4 Conclusions

Overall I enjoyed developing this website and learning a new programming language, the only disadvantage I found was as I was only learning Ruby on Rails while developing the website, it was difficult to find out how to implement different aspects of the website, also as Rails v3 is relatively new and is written slightly different to Rails v2.3.8 some of the Gems had not been converted and tested with Rails v3. I found that I spent alot of time trying to find out and debug different aspects of the website, whereas I felt that if I had of used a language I had some experience with it would have been easier to implement more into my website.

My customer is seeking approval from her manager to use the website for Top Level gym instead of just having one page on Skerries Community Centres website.

5 Further development or research

I plan to add more information, videos and a forum where users can ask questions like boards.ie. I also want to add music playlists that users can download.

6 Bibliography

Bosonnet, D. (2010). *Simple form.* Available: http://www.ruby-forum.com/topic/552264. Last accessed 30th Nov 2010.

Bosonnet, D. (2011). *My Apps.* Available: https://api.heroku.com/myapps/funkyfitness. Last accessed 04th May 2011.

Carneiro,C and Al Barazi, R (2010). *Beginning Rails 3*. USA: Apress. pg1-345.

Griffiths, D (2008). *Head First Rails: A Learner's Companion to Ruby on Rails* . USA: O'Reilly Media . p1-147.

Hub, G. (2011). *Home.* Available: https://github.com/. Last accessed 19th Mar 2011.

Hartl, M. (2010). *Learn Rails by Example.* Available: http://ruby.railstutorial.org/. Last accessed 23rd March 2011.

Ruby community. (2010). *Ruby on Linux.* Available: http://www.ruby-lang.org/en/downloads/. Last accessed 30th Oct 2010.

Ruby community. (2008). *Ruby Gems.* Available: http://rubygems.org/pages/about. Last accessed 17th Apr 2011

Yoon, M . (2011). *surveyor-dev* . Available: http://groups.google.com/group/surveyor-dev/browse_thread/thread/a4a2378efdb4dd43?pli=1. Last accessed 04th Apr 2011.

7 Appendix

7.1 Project Proposal

Project Proposal

TITLE

Funky Fitness

Danielle Bosonnet
07669402
daniellleb5989@gmail.com

BSc (Hons) in Business Information Systems

Date
24/09/2010

1 Objectives

The main objective of this project is to provide an online website which is developed for a personal trainer, which enables the user to use the website to track their fitness online. This tracking system will be similar to the Wii fit, which enables the user to set goals and view their progress.

When the user opens my website they will need to sign up and become a member to avail of the full range of services available. Each user will have their own profile page which will be customized based on specific criteria relevant to them.

The website will offer the following features:

- Online fitness programme designed by a personal trainer based on their age, height, level of fitness, weight and time available to train.
- A forum which allows users to ask general fitness questions which can then be answered by other users or the personal trainer.
- A section on nutritional information, sample food plans and calories burned doing different exercises
- An online diary where users can track the time spent doing physical activity   and a food diary which will be developed using Google Calendar
- Tutorials that allows user to test their knowledge on the correct way the use all gym equipment
- An interactive diagram of the muscles in the body and an explanation of how to exercise each body part correctly
- BMI(Body Mass Index) and BMR(Basal Metabolic Rate) calculator
- Availability to link progress to social network sites
- Videos of different exercises
- Site map and site search

This website will be developed with an attractive user interface. As the site will be used by people with a wide range of background and technological abilities' I will design the site to suit all types of users and will keep in mind the different levels of technological ability.

If time constraints allow or subsequently post project completion date I will include a section on:

- Exercises for people with different health conditions which effect the way that they can exercise.

2 Background

The reason I decided to create this application was because I have never come across a website that provides all the features needed to track your fitness and as obesity is a growing problem in today’s society I feel that this application will help provide users with a greater understanding of fitness and exercise. Also I have a keen interest in fitness and nutrition so I feel that this will enable me to enjoy it more.

As this project was proposed and carried out last year using html, xhtml, css and JavaScript, I am going to use Ruby on Rails as my programme language which operates on a variety of different platforms such as Windows, Macintosh and Linux, thus creating an entirely different project but based on the same subject area

3 Technical Approach

Once I had my idea and concept for the project outlined, I began to focus on the technologies I will use. I looked at various different languages and platforms. Due to the nature of my project and the fact that a number of will access it, the site will need to operate on a variety of different platforms such as Windows, Macintosh and Linux. With this in mind I need to find a language that supports many different platforms. So I decided to use Ruby on Rails with Linux operating system.

4 Special resources required

To implement my project I will have to learn Ruby on Rails and will be using a number of books and online resources. Outlined below are the main resources I have discovered but this will change throughout the lifespan of my project, any additional resources will be referenced accordingly throughout the project.

- Programming Ruby 1.9 by Dave Thomas
- Rails for Java Developers By Stuart Halloway and Justin Gehtland
- Advanced Rails Recipes by Mike Clarke
- Head First Rails by David Griffiths
- The JavaScript anthology 101 essential tips, tricks and hacks by James Edwards and Cameron Adams
- http://www.rubyonrailstutorials.com/
- http://rubyonrails.org/screencasts/rails3/

5 Project Plan

Project Milestones	Status	Completion date
Install and Configure Ruby, rails, my SQL and Net beans		24/09/10
Develop Framework		01/10/10
Develop login in and log out functionality		25/10/10
Develop members profile page		01/11/10
Develop fitness programmes		15/11/10
Develop Forum		13/12/10
Develop tutorials and videos		27/12/10
Develop section for nutritional information, sample food plans and calories burned doing different exercises		10/01/11
Develop Calculators		24/01/11
Develop Diary section		31/01/11
Develop Interactive Muscle Diagram		28/02/11
Develop social network link		07/03/11
Develop search facility/ site map		21/03/11

6 Technical Details

This project will be written In Ruby on Rails. The main part of the site will be its login system and specific user profiles. Each user’s profile will be customized based on specific criteria relevant to them and their information will be held on a database.

7 Consultation 1

Paul Stynes.
Firstly, I consulted Paul with my idea to build this application and he advised me to look at a project carried out last year and to ensure that my concept is not identical.
Secondly he advised me to use Ruby on Rails as my programming Language as it is becoming increasing popular and is compatible on many different platforms.
Finally he suggested creating the application on Linux instead of Windows.

8 Consultation 2

Orla Lahart

Danielle Bosonnet 24/09/10
Signature of student and date

7.2 Requirement Specification

Title Requirements Specification (RS)

Document Control

Revision History

Date	Version	Scope of Activity	Prepared	Reviewed	Approved
14/10/2005	1	Create	AB	X	X
21/10/05	2	Update	CD		

Distribution List

Name	Title	Version
Paul Stynes	Lecturer II	2

Related Documents

Title	Comments
Title of Use Case Model	
Title of Use Case Description	

Table of Contents

1 Introduction

1.1 Purpose

The purpose of this document is to set out the requirements for the development of Funky Fitness gym application. I will use this as a template in order to create my project.
The intended customer of this application is the general public who are looking for an easier way to access fitness information and programmes quickly from home or on the go.

1.2 Project Scope

The scope of the project is to develop a web application which allows users to become members so they can access there free fitness programme, track their progress and set goals, linking this to social network sites, a forum, exercise videos, BMI and BMR calculators, quizzes to test their knowledge on the correct way to use all gym equipment, sample recipes and nutritional information and interactive diagram of the muscles in the body and how to train them.
The system will enable users to log in, and once logged in they will be brought to there fitness program which will contain all there information, this will allow them to keep there information up to date and share there progress with there friends on social network sites.
To develop this application I will use sqlite3 as a database to store all the members' information and will use Ruby on Rails running on Ubuntu to create the system.

2 User requirements definition

From the customers perspective the system will allow users to log in and access they own free fitness program, they can track their progress and goals and share these with their friends on social network sites.
The user will also be able to take a quiz to test their knowledge on how to use all gym equipment correctly, access to a forum where the user can ask questions, sample meal plans, interactive diagram of all the muscles in the body and how to train each one, calculate their BMI (Body Mass Index) and BMR (Basal Metabolic Rate) and watch exercise videos.

3 System architecture

The system I will develop will be user friendly as much as possible and with the site search and map, this will allow the user to quickly find and search the site for what they are looking for, this will ensure that it is easy to navigate through.
This application is a web based application so the user will need internet access to use the application.
The first page the user will be presented with is the log in page, once they log in using a specific username and password which will be stored in a database they will be brought to their profile page which will be customized based on specific criteria relevant to them such as age, height, weight and time available to exercise. On this page there will also be a link so the user can share their progress with their friends on social networking sites.
The second page will be a forum where user can ask questions which can be answered by other users and or the administrator or customer.
The third page will be an interactive diagram of all the muscles in the body and how to train each one.

The fourth page will contain a quiz to test the users knowledge on how to use all gym equipment correctly, this quiz will contain multiple choice questions and the user will have to select the correct answer from a choice of four answers, at the end of the quiz the user will be presented with a review of the quiz so they can see which questions they got wrong and the correct answers.
The fifth page will contain information on nutritional information and healthy recipes for breakfast, lunch, dinner and also snacks.
The sixth page will contain a BMI (Body Mass Index) and BMR (Basal Metabolic Rate) calculators. This will allow the user to calculate each one and see there result.
And the final page will contain videos of different exercises.
The site will also contain a site map and site search.
The external hardware needed will be a monitor, a CPU (Central Processing Unit), a mouse and a keyboard.

4 Requirements specification

4.1 Interface requirements

My application will interact with the user by using a Graphical User Interface (GUI). Most of the operation in a GUI is by mouse clicks and scrolling. This type of interface will suit my target market and will ensure that the site will suit all types of users and will keep in mind the different levels of technological ability. It will consist of a number of different pages where each contains different information which will be developed using Ruby on Rails.
The input and output will be going to the one system which will be developed using RoR. The information must be entered using text fields otherwise the information will be invalid. The data gathered must be stored in a database so that the users profile and information is saved.

4.2 Functional requirements

4.2.1 Requirement 1

The user must register.

4.2.1.1 Description & Priority

This requirement is essential as the user must register in order to access the website. Therefore nothing can be done unless the user is a member.
For this function to work the database must be set up to store the information.
This is a high priority requirement.

4.2.1.2 Requirement Activation

This requirement is essential as when the user first accesses the webpage they will be prompted to register, so if they are not already a member they must sign up to become a member

4.2.1.3 Technical issues

A database must be created in order to satisfy this requirement the database will store all the users information which includes username, passwords etc.

4.2.1.4 Risks

This requirement might not be able to be satisfied if the user unable to log in, this could be caused by the user entering the wrong username or password, the user must ensure then enter the correct details in order for log in to be successful.
An action can be taken to reduce the probability of this happening, on the login page there will be a forgot your password button when upon clicked the user will be asked for their email address, once submitted their password will be sent to their email address.

4.2.1.5 Dependencies with other requirements

If the user is not a member they won’t be able to access the webpage so login/register is essential.

4.2.1.6 Functional Requirements

4.2.2 Requirement 1

Create user profile

4.2.2.1 Description & Priority

Upon registering the user will be asked to provide some personal information in order to create their profile e.g. age, height, sex and weight.
If this information is not provided to the system the system will be unable to create a profile for the member.

4.2.2.2 Requirement Activation

This requirement will allow the user to enter personal information about them.

4.2.2.3 Technical issues

A form will have to be created in order to allow the user to enter their information and this form must be linked to a database which will store the information

4.2.2.4 Risks

The user may enter the wrong information accidently, if this happens the user will be able to click an edit button, once clicked the user can change the information contained in the form/on profile.

4.2.2.5 Dependencies with other requirements

This requirement will interact with the database.

4.2.2.6 Requirement Activation

This requirement will allow the user to enter their personal information.

4.3 Documentation requirements

The two documents that are required are

1. A Technical report
2. User Manual

4.3.1 Requirement 1

Technical Report

4.3.2 Description & Priority

This document will describe the process, progress, or results of technical or scientific research or the state of a technical or scientific research problem.
This is a high priority requirement

4.3.3 Requirement 2

User Manual

4.3.4 Description & Priority

This document will is a technical communication document intended to give assistance to the users of the web application.
This is a high priority requirement

4.4 Data requirements

Data requirements should consider the following:

- The data inputted should be in text format

4.4.1 Requirement

Register

4.4.2 Description & Priority

The user must enter their details in text format.
This is a high priority Requirement.

4.5 Non-Functional Requirements

4.5.1 Security requirement

In the system there is a security requirement which is user login. A stable database must be created in order to store all the information entered by the users.

5 System models

See Appendices for use case diagram

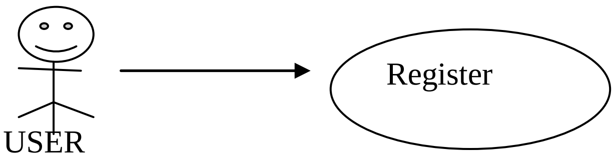
6 System evolution

The system could evolve over time by adding new aspects to the application like Exercises for people with different health conditions which effect the way that they can exercise.

7 Appendices

7.1 Use case 1

Use case
Register
Scope in
The scope of this use case is to allow the user to register.
Description
This use case describes the process of how the user creates an account which allows them to login by entering their username and password.
Use Case Diagram



Flow Description
Precondition
The system is in initialisation mode and the user will be greeted with a main page where they can login or register.
Activation
This use case starts when the user enters their details into the text boxes and clicks sign up.
Main flow

1. The user submits their details
2. The system identifies the information entered by the user.
3. The system checks details and ensures they meet the required fields to become a member.
4. The user becomes a member and can enter the website.

Alternate flow
A1 : User enters incorrect or not the required details

1. The system does not register the user.
2. The user will be denied access to the site.
3. The use case continues at position 3 of the main flow

Exceptional flow
E1 : The system decides if any fields in the form were left blank by the user

4. The system checks username and password entered by the user to ensure it is correct.
5. The user waits for the system to confirm access.
6. The use case continues at position 4 of the main flow

Termination
The system presents the welcome page if login is successful.

Post condition
The system goes to a wait state until user clicks next button.

7.2 Use case 2

Use case

Get Fitness Program

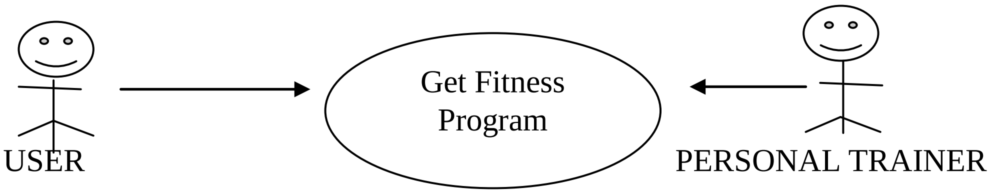
Scope

The scope of this use case is to allow the user to obtain a free fitness program.

Description

This use case allow the user to enter their personal information and based on the information entered the system will create a fitness program for the user.

Use Case Diagram



Flow Description

Precondition

The system is in idle state and is waiting for the user to enter their personal information into the form provided.

Activation

This use case starts when the user enters their details into the text boxes and clicks enter.

Main flow

1. The user submits their details.(See A1)
2. The system identifies the information entered by the user is correct.
3. The system checks details and ensures they meet the requirements.(See E1)
4. The system creates fitness program if details correct

Alternate flow

A1 : The user submits incorrect details

7. The system will check details and ensures they meet the requirements.
8. The user will be prompted to enter the correct details.
9. The use case continues at condition 4 of the main flow.

Exceptional flow

E1 : System decides if details are correct

10. The system checks username and password is correct.
11. The user waits for the system to give fitness program.

Termination

The system gets fitness program for user.

Post condition

The system goes to a wait state until user clicks next link.

7.3 Use case 3

Use case

Update Details

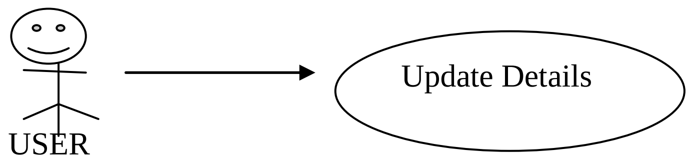
Scope

The scope of this use case is to update user details.

Description

This use case allows the System to update details change by the user.

Use Case Diagram



Flow Description

Precondition

The system is in idle state and is waiting for the user to change/update their details.

Activation

This use case starts when the user changes/updates their details.

Main flow

1. The user changes/updates details.
2. The system identifies the changes made.
3. The user waits for the system to update details.
4. The system updates users details
5. The system ensures the users information is saved.

Alternate flow

A1 : The user makes changes

1. The system asks user if they want to share with a social network site.
2. The user says yes
3. System shares information with social network site.
   Use case continues at position 5 of the main flow.

Exceptional flow

E1 : the user makes changes

4. The system asks user if they want to share with a social network site.
5. The user says no
6. System doesn't share information with social network site.
7. Use case continues at position 5 of the main flow.

Termination

The system updates users details.

Post condition

The user's details are updated.

Use case 4

Use case

Log out

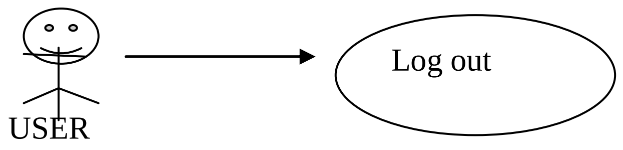
Scope

The scope of this use case is to allow the user to exit the system by clicking log out button.

Description

This use case allows the user to log out of the system securely.

Use Case Diagram



Flow Description

Precondition

The system is in idle state and is waiting for the user to click the logout button.

Activation

This use case starts when the user clicks the favourite's button.

Main flow

1. The user clicks logout button.
2. The system identifies the logout button is pressed.(See A1)
3. The user waits for the system to load logout page.
4. The system ensures the users information is saved
5. System logs user out.

Alternate flow

A1 : The user waits for system to logout.

6. The system ensures profile is saved correctly.
7. The use case continues at position 5 of the main flow.

Exceptional flow

E1 : System decides if details are correct

8. The system checks username and password is correct.
9. The system ensures all profile details are saved correctly.
10. If not the system will prompt the user to save profile changes.
11. System will logout

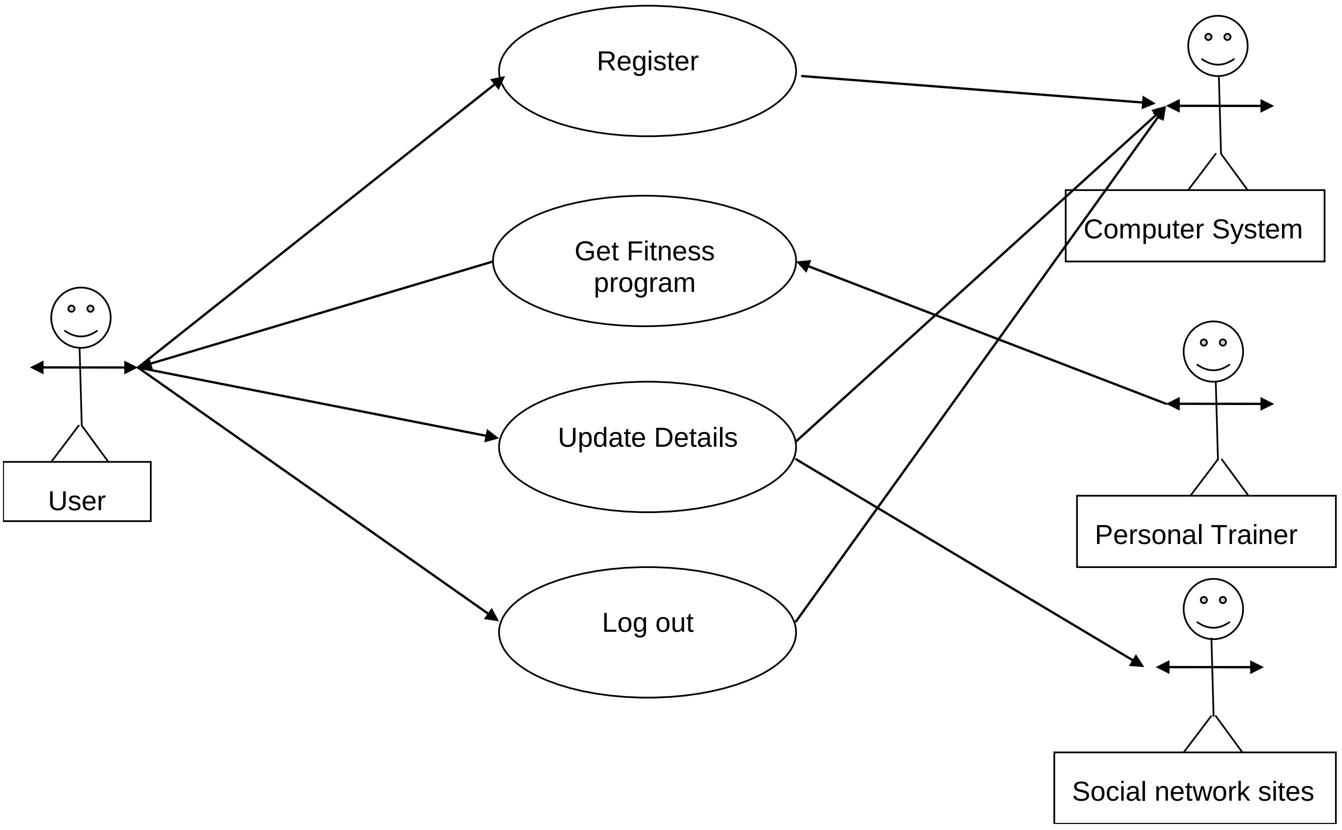
Termination

The system logs out user.

Post condition

The system goes to login in page

Use Case Diagram:



7.1 Monthly log book

7.2 Other material Used

Survey monkey Questions:

1. How did you find out about Funky Fitness?

□ How did you find out about Funky Fitness? Newspaper or magazine article

□ Friend/colleague

□ Google search

□ Surfing the web

□ Facebook

□ Other

2. How likely are you to visit Funky Fitness again?

□ How likely are you to visit Funky Fitness again? Very Likely

□ Somewhat Likely

□ Netural

□ Somewhat unlikely

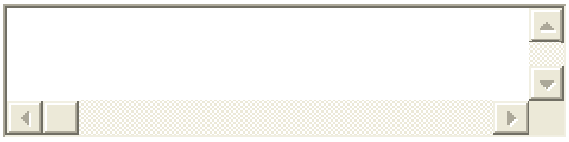
□ Very unlikely

3. Is it easy to navigate around the website?

□ Is it easy to navigate around the website? Yes

□ No

4. If no, Please specify why



If no, Please specify why

5. How satisfed are you with our website? (1 meaning disappointed 5 meaning very impressed)

☑ 1

□ 2

□ 3

□ 4

□ 5

6. How would you rate the information provided?

□ How would you rate the information provided? Poor

□ Ok

□ Brilliant

□ Excellent

7.How would you rate the layout and color scheme? (1 = bad, 5= Excellent)

□ 1

□ 2

□ 3

□ 4

□ 5

8. If you could add something to the site what would it be?

If you could add something to the site what would it be?

Done

Responses:

Response Summary

Total Started Survey: 60
Total Completed Survey: 60 (100%)

PAGE: 1

DownloadCreate Chart1. How did you find out about Funky Fitness?

	answered question	**60**
	skipped question	**0**

	Response Percent	Response Count
Newspaper or magazine article	0.0%	0
Friend/colleague	30.0%	18
Google search	3.3%	2
Surfing the web	5.0%	3
Facebook	6.7%	4
Other	**58.3%**	**35**

DownloadCreate Chart2. How likely are you to visit Funky Fitness again?

	answered question	**60**

DownloadCreate Chart1. How did you find out about Funky Fitness?		
	skipped question	**0**
	Response Percent	**Response Count**
Very Likely	8.3%	5
Somewhat Likely	20.0%	12
Netural	**30.0%**	**18**
Somewhat unlikely	15.0%	9
Very unlikely	26.7%	16

DownloadCreate Chart3. Is it easy to navigate around the website?		
	answered question	**59**
	skipped question	**1**
	Response Percent	**Response Count**
Yes	**96.6%**	**57**
No	3.4%	2

Download4. If no, Please specify why

answered question	3
skipped question	57

	Response Count
Show Responses	3

DownloadCreate Chart5. How satisfed are you with our website? (1 meaning disappointed 5 meaning very impressed)

answered question	60
skipped question	0

	Response Percent	Response Count
1	3.3%	2
2	13.3%	8
3	**35.0%**	**21**

DownloadCreate Chart5. How satisfed are you with our website? (1 meaning disappointed 5 meaning very impressed)

4	**35.0%**	**21**
5	13.3%	8

DownloadCreate Chart6. How would you rate the information provided?

	answered question	**59**
	skipped question	**1**

	Response Percent	Response Count
Poor	11.9%	7
Ok	**50.8%**	**30**
Brilliant	30.5%	18
Excellent	10.2%	6

DownloadCreate Chart7. How would you rate the layout and color scheme? (1 = bad, 5= Excellent)

	answered question	**60**

DownloadCreate Chart7. How would you rate the layout and color scheme? (1 = bad, 5= Excellent)

	skipped question	**0**
	Response Percent	**Response Count**
1	5.0%	3
2	18.3%	11
3	**40.0%**	**24**
4	28.3%	17
5	10.0%	6

Download8. If you could add something to the site what would it be?

answered question	**58**
skipped question	**2**
	Response Count
Show Responses	58