Zenlike

Requirements specification

Name: Steven Gaynor

Student Number: 14108038

Email: [x14108038@student.ncirl.ie](mailto:x14108038@student.ncirl.ie)

Higher Diploma in Science in Cloud Computing (HDCLOUD)

10th March 2015

Contents

[Zenlike: Requirements Specification 2](#_Toc413786089)

[1. Introduction 2](#_Toc413786090)

[1.1 Purpose 2](#_Toc413786091)

[1.2 Project Scope 2](#_Toc413786092)

[1.3 Definitions, Abbreviations 3](#_Toc413786093)

[2. User Requirements Definition 3](#_Toc413786094)

[3. Requirements Specification 3](#_Toc413786095)

[3.1 Functional Requirements 4](#_Toc413786096)

[3.2 Non-functional requirements 10](#_Toc413786097)

[Bibliography 11](#_Toc413786098)

# Zenlike: Requirements Specification

## Introduction

### 1.1 Purpose

The purpose of this document is to set out the requirements for the development of **Zenlike**, a proposed web-based app based on the principal of giving the user power over the organisation of their fitness, diet and relaxation in a balanced and sustainable way.

The intended clients are people who are requiring guidance in organising their schedules in an easy way to allow for all of the healthy aspects of life that their bodies demand. Specific target audiences include those wishing to lose some weight, build a more sustainable and effective exercise regime and those who wish to weave some meditative practices into their lives. Evidently, the shear amount of choice of fitness plans and diets on the market at this time can be of confusion to many people. One of the primary aims of this app will be to help sort the wood from the trees in these areas.

Several academic articles over recent times have proven the linkages between healthy foods and exercise as a beneficial aspect of daily life. The effect of exercise on brain activity is proven from clearly defined studies of the chemicals released during physical activity (Hunsberger, et al., 2007). It is also clear from some recent studies that practicing mindfulness for a consistent amount of time per week can improve a person’s mood (Goyal, et al., 2014). Helping people to strike the balance between all of these beneficial activities, and recognising the positive attributes of all of these is something that I believe is important. This is what gave me the idea to develop an app that would assist people in organising their healthy lifestyles in an easy way.

### 1.2 Project Scope

The scope of the project is to develop a web-based application incorporating a planner and easily accessible information on meals, exercise types and relaxing activities. The system shall have a database to store all of this information and will be manipulated through the website GUI by the user. The website administrators will also have the ability to add new items to the database in order to develop the knowledgebase of the site.

### 1.3 Definitions, Abbreviations

**MyHub:** Is the central php webpage that has three main sections: Weekly Planner, MyEvents and the links to the pages with choices of meals, exercises and relaxation events.

**MyEvents:** Is a section of the Planner that displays the user’s choice of events in a stylised CSS text box. They can be dragged onto the calendar.

## 2. User Requirements Definition

In this section, the set of objectives and requirements for the system will be outlined from the customer’s perspective. It is clear that any system that is designed must be very user-friendly and allow the client to easily organise their **Zenlike** events according to their own personal choices from the elements that the app will offer. It is clear that how all of these things balance out and indicate the success of the plan to the user will need to be determined by the objects of the finished software package.

In more concrete terms, the main requirements for the user of this product are the following:

* To survey the vital statistics of the user, such as height weight, gender and age
* To choose from a list of meals according to preferences, place the meal on the planner
* To choose exercises from a list, and place them on the planner
* To choose relaxation events from a list, and place them on the planner
* To show the user how this combination of exercise and diet would impact on the net intake of calories and vitamins and minerals
* To allow administrators of the website to add new meals, exercises and relaxation events

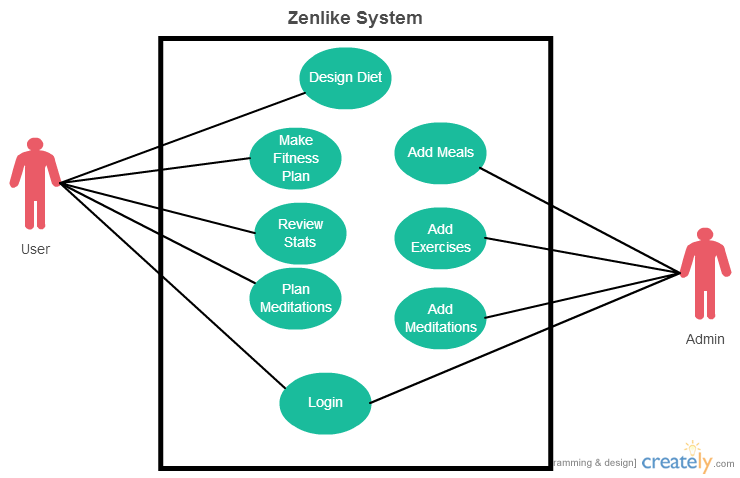
## 3. Requirements Specification

The requirements should be verifiable during the testing phase. The system needs to be one that can be easily learnt within a few minutes. The layout of the GUI and smooth running of the underlying code will be essential to ensuring this. During the testing phase, it is also necessary to ensure that as many errors as possible are routed out. This section will describe in detail both the functional and non-functional requirements of the system.

### 3.1 Functional Requirements

This section will describe the functional requirements of the Zenlike system. The functional requirements describe “the system function in detail, its inputs and outputs, exceptions” (Sommerville, 2006, p. 120). The use case model diagram below gives an overview of what the system seeks to achieve at a highly conceptual level.

#### 3.1.1 Use Case Diagram



The following section will have a detailed description for each use case roughly in order of importance.

#### 3.1.2 Use Case Description – “Make Fitness Plan”

**Scope**

The scope of this use case is to allow the **User** actor to design a fitness plan using a planner calendar.

**Description**

This use case describes the ability the User will have to choose from a list of exercise types, according to how many calories that they burn. The user will be able to put these exercises onto their planner calendar.

**Flow Description**

The User actor will be presented with a screen that lists exercise types, perhaps with pictures and a brief description of the exercise. The user will click on the exercise “button”/”window”. Clicking this will save this exercise to the user’s list of chosen events. The user will then click on the calendar (in an easily accessible area of the screen) and will drag and drop a CSS designed box with the exercise onto a chosen time on the user’s calendar. The drop of this CSS box will be saved and indicated on the time choses by the user.

**Precondition**

User will be on a webpage that will act as a central hub for co-ordinating all of their Zenlike activities. It will have links to the Zenlike calendar, and the database of meals, exercises and relaxations.

**Activation**

This use case starts when the **User** actor clicks on “Choose Exercises”. This will take them to a webpage which lists the exercises in styled HTML sections.

**Main flow**

1. The system presents the user with the Exercises PHP page.
2. The User clicks on an exercise name to see a picture of the exercise (optional), how many calories burned, and a description. An area of text named something like “Add to Events” can be clicked if the user wants this event.
3. The system registers the user’s choice of this event by making a database entry in the Exercise events table.
4. The User can choose more events if required

**Alternate flow**

1. The system already has that event saved to the Events of the User
2. The User receives a message stating that he/she has already chosen this event.
3. The use case continues at position 2 of the main flow

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

The system presents the calendar with the events saved successfully in the time slots.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 3.1.3 Use Case Description – “Design Diet”

**Scope**

The scope of this use case is to allow the **User** actor to design a diet from a bank of healthy meals using a planner calendar.

**Description**

This use case describes the ability the User will have to choose from a list of meals, according to how many calories that they contain and their vitamin/mineral content etc. The user will be able to put these meals onto their planner calendar.

**Flow Description**

The User actor will be presented with a screen that lists meals, perhaps with pictures and a brief description of the meal. The user will click on the meal “button”/”window”. Clicking this will save this meal to the user’s list of chosen events. The user will then click on the calendar (in an easily accessible area of the screen) and will drag and drop a CSS designed box with the meal onto a chosen time on the user’s calendar. The drop of this CSS box will be saved and indicated on the time chosen by the user.

**Precondition**

User will be on a webpage that will act as a central hub for co-ordinating all of their Zenlike activities. It will have links to the Zenlike calendar, and the database of meals, exercises and relaxations.

**Activation**

This use case starts when the **User** actor clicks on “Choose Meals”. This will take them to a webpage which lists the meals in styled HTML sections.

**Main flow**

1. The system presents the user with the Meals PHP page.
2. The User clicks on a meal name to see a picture of the meal (optional), how many calories it contains, vitamin/mineral content and a description. An area of text named something like “Add to Events” can be clicked if the user wants this event.
3. The system registers the user’s choice of this event by making a database entry in the Meals events table.
4. The User can choose more events if required

**Alternate flow**

1. The system already has that event saved to the MyEvents of the User
2. The User receives a message stating that he/she has already chosen this event.
3. The use case continues at position 2 of the main flow

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

The system presents the calendar with the events saved successfully in the time slots.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 3.1.4 Use Case Description – “Plan Meditations”

**Scope**

The scope of this use case is to allow the **User** actor to plan meditations from a bank of relaxing/meditative events using a planner calendar.

**Description**

This use case describes the ability the User will have to choose from a list of meditations according to their lifestyle preferences. The user will be able to put these events onto their planner calendar.

**Flow Description**

The User actor will be presented with a screen that lists relaxing things to do, perhaps with pictures and a brief description of the event. The user will click on the meditation “button”/”window”. Clicking this will save this meditation to the user’s list of chosen events. The user will then click on the calendar (in an easily accessible area of the screen) and will drag and drop a CSS designed box with the meditation onto a chosen time on the user’s calendar. The drop of this CSS box will be saved and indicated on the time chosen by the user.

**Precondition**

User will be on a webpage that will act as a central hub for co-ordinating all of their Zenlike activities. It will have links to the Zenlike calendar, and the database of meals, exercises and relaxations.

**Activation**

This use case starts when the **User** actor clicks on “Choose Relaxing Events”. This will take them to a webpage which lists the relaxations in styled HTML sections.

**Main flow**

1. The system presents the user with the Relax PHP page.
2. The User clicks on an event name to see a picture of the event (optional). An area of text named something like “Add to Events” can be clicked if the user wants this event.
3. The system registers the user’s choice of this event by making a database entry in the Meditations events table.
4. The User can choose more events if required

**Alternate flow**

1. The system already has that event saved to the MyEvents of the User
2. The User receives a message stating that he/she has already chosen this event.
3. The use case continues at position 2 of the main flow

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

The system presents the calendar with the events saved successfully in the time slots.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 3.1.5 Use Case Description – “Add Meals, Exercises, Meditations”

**Scope**

The scope of this use case is to allow the **Admin** actor to login as a user with admin privileges and easily add new meals, exercises and mediations to the bank of events.

**Flow Description**

The Admin actor will fill in fields and optionally add a picture in order to allow more events for the user to choose from

**Precondition**

Logged in as an Admin user. Ability to “Add” and “Edit” the events in each of the three events pages.

**Activation**

This use case starts when the **Admin** actor clicks on Add or Edit

**Main flow**

1. The Admin clicks Add event
2. Fills out Name, Description and calorie/mineral details etc.
3. The System saves these new details to database
4. The Admin checks to see these events appear correctly without error.

**Alternate flow**

A1

The exercise type is duplicate of one already there.

**Exceptional flow**

E1 :

1. The system loses contact with the MySQL database.
2. The Admin receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues after database connectivity restored.

**Termination**

The events saved successfully on the events pages

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 3.1.6 Use Case Description – “Review Stats”

**Scope**

The scope of this use case is to allow the **User** actor to review how many calories they have consumed and/or burned off in a given week, and to review vitamin/mineral intake.

**Description**

This use case describes the ability the User will have to view stats of their calorie count given the meals/exercises etc. they have chosen.

**Flow Description**

The User actor will be presented with a stat box that contains the above described details

**Precondition**

The events have been successfully placed on the calendar

**Activation**

This use case starts when the **User** actor has started placing events on their calendar.

**Main flow**

1. The User has at least one event on his/her calendar.
2. The system feeds info into a stat table.
3. This info is outputted into a display on the MyHub page.

**Alternate flow**

n/a

**Exceptional flow**

1. The system loses contact with the MySQL database.
2. The User receives a message indicating a temporary server issue, and that they may re-attempt in a few moments.
3. The use case continues at position 2 of the main flow.

**Termination**

If the user chooses to remove his/her events from the calendar.

**Post condition**

The system displays the MyHub webpage and allows the user to navigate to any other page as required.

#### 3.1.7 Use Case Description – “Login”

This will allow the User and Admin to see their own personalised MyHub page that fulfils all of the use cases as appropriate above. It allows the functionality that a User can save events to their calendar and return at a later time to view/amend as required.

### 3.2 Non-functional requirements

Non-functional requirements are “quality of service” and “non-behavioural requirements” that determine the aspects of the system’s running that do not relate to items such as use cases or functional requirements, basically determining how the system controls itself and interacts with other systems (Stellman & Greene, 2005, p. 113).

The non-functional requirements for Zenlike essentially encapsulate that the system should give responses in good time. For example, a drag-and-drop event should not take more than 5 seconds to complete. Lag should kept to a minimum. This will be ensured during the testing phase.

The website should also have excellent cross-browser functionality and be fully mobile responsive.

# Bibliography

Goyal, M., Singh, S. & Sibinga, E. M. S., 2014. Meditation Programs for Psychological Stress and Well-being: A Systematic Review and Meta-analysis.. *JAMA Intern Med.,* 174(3), pp. 357-368.

Hunsberger, J. G. et al., 2007. Antidepressant actions of the exercise-regulated gene VGF. *Nature Medicine,* Volume 13, pp. 1476 - 1482.

Sommerville, I., 2006. *Software Engineering.* 8th ed. Boston: Addison-Wesley.

Stellman, A. & Greene, J., 2005. *Applied Software Project Management.* s.l.:O'Reilly Media.