Fish Farming System

設計文件

|  |  |
| --- | --- |
| 專案名稱 | FFS養魚系統 |
| 撰寫日期 | 2017/11/28 |
| 發展者 | 劉子詣、陳維哲、胡增豪、許翔淋、柯翊偉 |

**版次變更記錄**

|  |  |  |
| --- | --- | --- |
| 版次 | 變更項目 | 變更日期 |
| 0.1 | 初版 | 2017/11/18 |
| 0.2 | 二版 | 2017/11/21 |
| 0.3 | 三版 | 2017/11/27 |
| 1 | demo版 | 2017/11/28 |
| 1.1 | 修正版 | 2017/12/13 |
| 2 | 發布版 | 2018/1/6 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

目錄

[**版次變更記錄 1**](#_30j0zll)

[**1.**](#_1fob9te) **系統模型與架構(System Model/System Architecture) 3**

[**2.**](#_3znysh7) **介面需求與設計(Interface Requirement and Design) 4**

[**3.**](#_2et92p0) **流程設計(Process Design) 6**

[**4.**](#_tyjcwt) **使用者畫面設計(User Interface Design) 8**

[**5.**](#_3dy6vkm) **資料設計(Data Design) 13**

[**6.**](#_1t3h5sf) **類別圖設計(Class Diagram) 15**

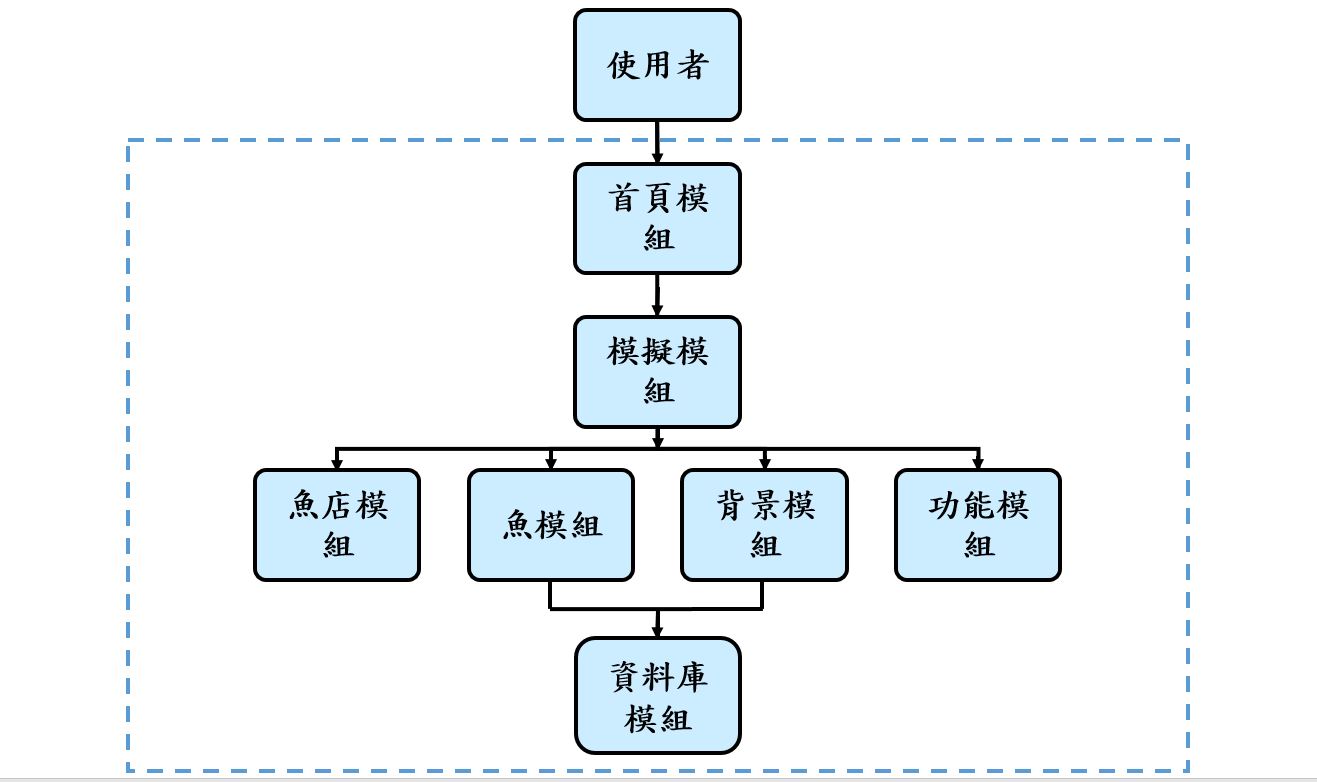
[**7.**](#_4d34og8) **實作技術(Implementation Languages and Platforms) 18**

[**8.**](#_2s8eyo1) **設計議題(Design Issue) 18**

1. **系統模型與架構(System Model/System Architecture)**

系統主要分為7大模組，首頁模組，模擬模組，魚模組，魚店模組，背景模組，功能模組，資料庫模組，每一個模組一個package。

在首頁模組中提供start、continue及Instruction功能，可以重新開始遊戲、繼續上次遊戲及觀看使用說明。在模擬模組中提供了主要介面，供使用者使用各種功能及觀看各種背景資訊、魚資訊。魚的模組提供了魚類的各種資訊，如魚種、年齡、性別…等。魚店模組提供使用者新增魚類至模擬模組，且可以設定性別及年齡。背景模組包含許多背景變數，如水溫、乾淨度、時間…等的變動。功能模組提供各種功能按鈕，如換水、打氣機、調溫器…等。使用者可以直接使用首頁模組，並從首頁進入模擬介面後間接地使用各種下游模組。資料庫模組用來儲存魚資訊跟背景資訊，藉此讓使用者能從上次的紀錄繼續模擬。



1. **介面需求與設計(Interface Requirement and Design)**

|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 01 使用者與首頁模組之介面 | 首頁模組 | 使用者 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| none | none | none |
| **對應之介面需求(Interface Requirement Description)** | | |
| 使用者到首頁介面 | | |

|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 02 首頁模組與模擬模組之介面 | 模擬模組 | 首頁模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | none | none |
| **對應之介面需求(Interface Requirement Description)** | | |
| 從首頁介面進入模擬介面 | | |

|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 03 模擬模組與魚店模組之介面 | 魚店模組 | 模擬模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | 魚的姓名、性別  、年齡、魚種 | 魚的姓名、性別  、年齡、魚種 |
| **對應之介面需求(Interface Requirement Description)** | | |
| 添加魚到模擬介面 | | |

|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 04 模擬模組與魚模組之介面 | 魚模組 | 模擬模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | none | 魚資料 |
| **對應之介面需求(Interface Requirement Description)** | | |
| 在模擬介面呈現出魚的資訊 | | |

|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 05 模擬模組與背景模組之介面 | 背景模組 | 模擬模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | none | 背景資訊 |
| **對應之介面需求(Interface Requirement Description)** | | |
| 在模擬介面呈現出背景的資訊 | | |

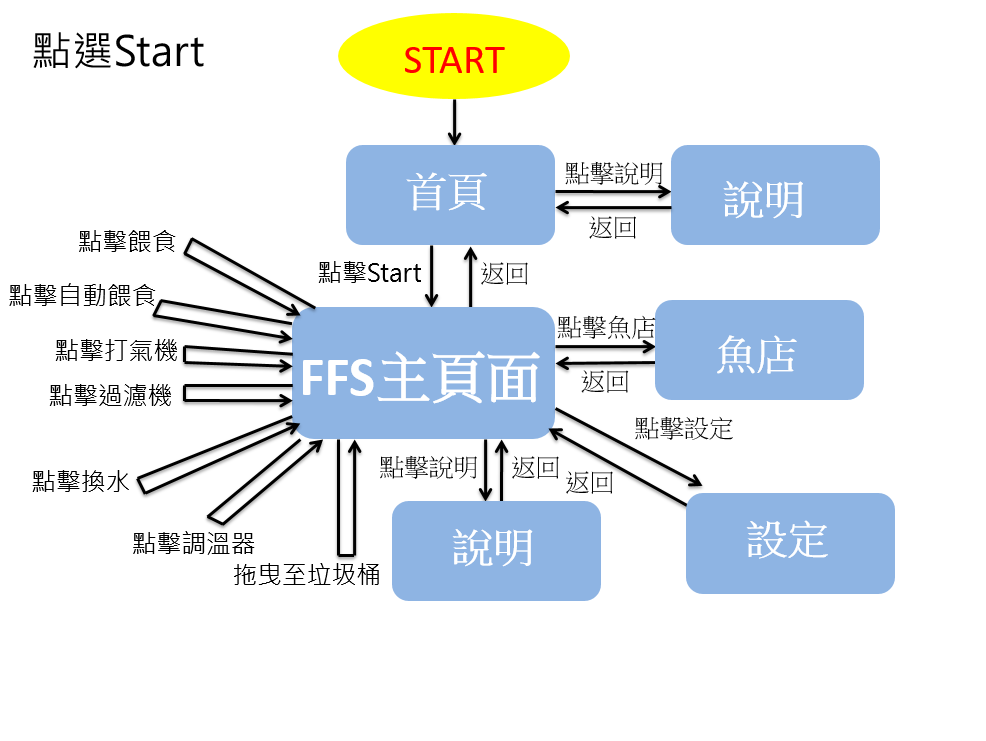
|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 06 模擬模組與功能模組之介面 | 功能模組 | 模擬模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | none | 更改後的魚資訊  、環境資訊 |
| **對應之介面需求(Interface Requirement Description)** | | |
| 使用Method去更改魚和背景的資訊 | | |

|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 07 魚模組與資料庫模組之介面 | 資料庫模組 | 魚模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | 魚資料 | none |
| **對應之介面需求(Interface Requirement Description)** | | |
| 將魚資訊記錄到資料庫中 | | |

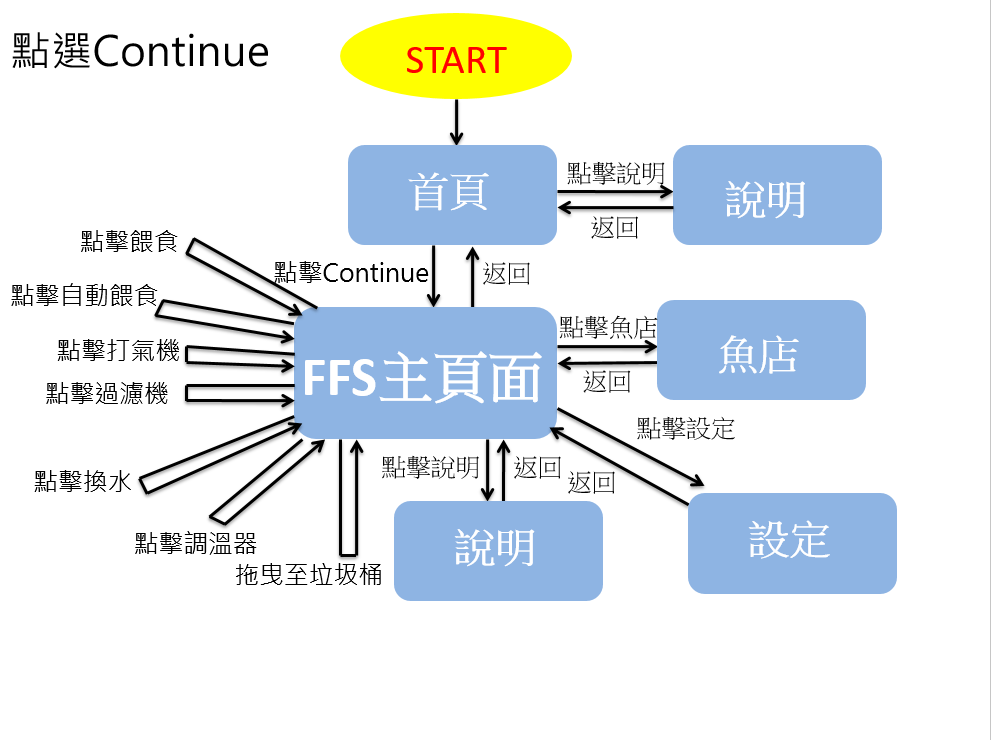
|  |  |  |
| --- | --- | --- |
| **介面名稱**  **(Interface Name)** | **介面提供者**  **(Interface Provider)** | **介面使用者**  **(Interface Consumer)** |
| 08 背景模組與資料庫模組之介面 | 資料庫模組 | 背景模組 |
| **連結方式**  **(Connection Type)** | **輸入資料**  **(Input Data)** | **輸出資料**  **(Output Data)** |
| Java Method Call | 背景資料 | none |
| **對應之介面需求(Interface Requirement Description)** | | |
| 將背景資訊記錄到資料庫中 | | |

1. **流程設計(Process Design)**

3-1



3-2



**狀態描述(State description)**

|  |  |
| --- | --- |
| **Name** | 首頁 |
| **Definition** | 可以點選Start或Continue進入FFS主頁面，亦提供系統說明。 |

|  |  |
| --- | --- |
| **Name** | FFS主頁面(模擬頁面) |
| **Definition** | 提供使用者進行模擬養魚。 |

|  |  |
| --- | --- |
| **Name** | 說明 |
| **Definition** | 提供系統說明。 |

|  |  |
| --- | --- |
| **Name** | 魚店 |
| **Definition** | 提供增加飼養魚，並設定魚的姓名、性別、年齡、魚種功能。 |

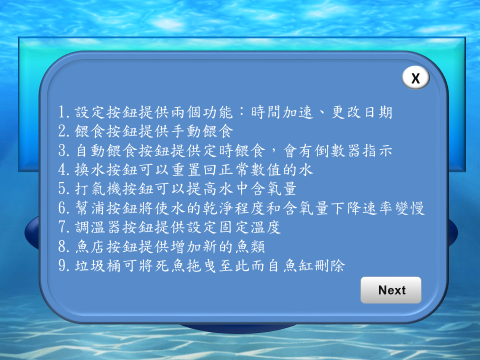
|  |  |
| --- | --- |
| **Name** | 設定 |
| **Definition** | 提供返回首頁、時間加速、設定日期和時間的功能。 |

1. **使用者畫面設計(User Interface Design)**

4-1:首頁-初始畫面



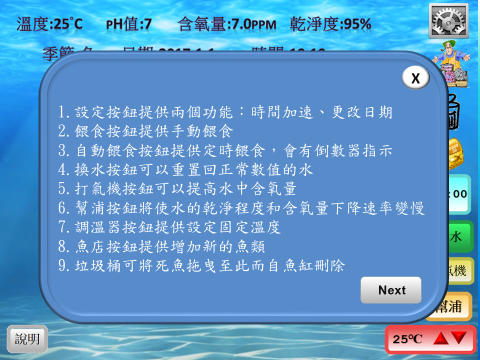
4-2:首頁-說明



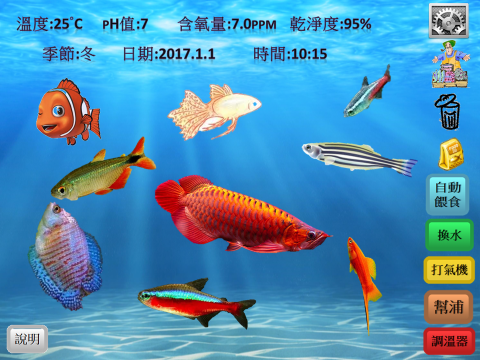
4-3:模擬介面:初始畫面



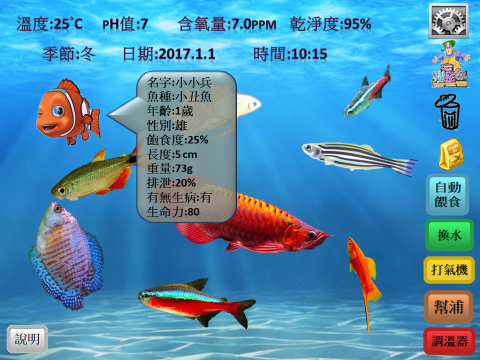
4-4:模擬介面:按鈕全開



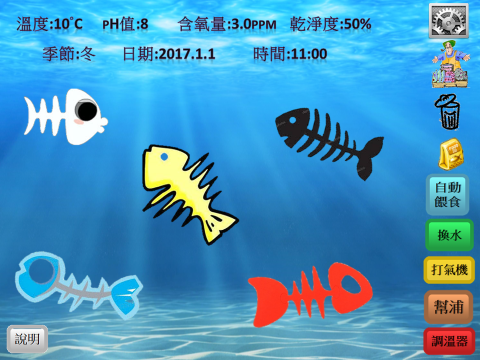
4-5:模擬介面-新增魚



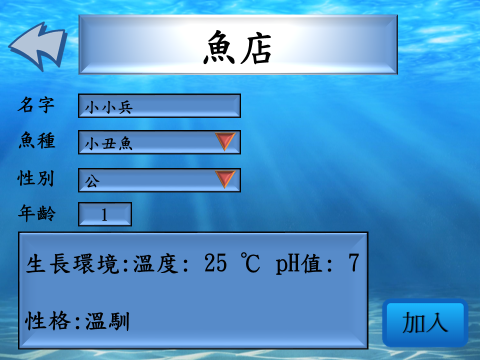
4-6:模擬介面-魚資訊



4-7:模擬介面-魚死亡後



4-8:魚店介面



4-9:設定



1. **資料設計(Data Design)**

**5-1 initial fish Table**

**CREATE TABLE** initial\_fish (

category **varchar(10) NOT NULL**,

length **int(10)** **NOT NULL**,

weight **int(10)** **NOT NULL**,

temperature\_max **double(20,10)** **NOT NULL**,

temperature\_min **double(20,10)** **NOT NULL**,

pHValue\_max **double(20,10)** **NOT NULL**,

pHValue\_min **double(20,10)** **NOT NULL**,

oxygenContent **double(20,10)** **NOT NULL**,

cleanliness **int(10)** **NOT NULL**,

attack **tinyint(1)** **NOT NULL**,

**PRIMARY KEY** (category)

);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位名稱** | **欄位代號** | **定義** | **型態** | **長度** | **備註** |
| **種類** | category | 魚的種類 | varchar | 10 | Primary Key |
| **長度** | length | 魚的長度 | int | 10 | Not Null |
| **重量** | weight | 魚的重量 | int | 10 | Not Null |
| **最大溫度** | temperature\_max | 適合環境溫度最大值 | double | 20,10 | Not Null |
| **最小溫度** | temperature\_min | 適合環境溫度最小值 | double | 20,10 | Not Null |
| **最大pH值** | pHValue\_max | 適合環境的pH最大值 | double | 20,10 | Not Null |
| **最小pH值** | pHValue\_min | 適合環境的pH最小值 | double | 20,10 | Not Null |
| **含氧量** | oxygenContent | 適合環境的含氧量 | double | 20,10 | Not Null |
| **乾淨度** | cleanliness | 適合環境的乾淨度 | int | 10 | Not Null |
| **攻擊** | attack | true代表會攻擊false代表不會攻擊 | tinyint | 1 | Not Null |

**5-2 fish Table**

**CREATE TABLE** fish (

name **varchar(10) NOT NULL**,

category **varchar(10) NOT NULL**,

age **int(10)**  **NOT NULL**,

gender **varchar(10)** **NOT NULL**,

satiation **int(10)** **NOT NULL**,

length **double(20,10)** **NOT NULL**,

weight **double(20,10)** **NOT NULL**,

excretion **int(10) NOT NULL**

sick **tinyint(1)** **NOT NULL**,

life **double(20,10)** **NOT NULL**,

**PRIMARY KEY** (name)

);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位名稱** | **欄位代號** | **定義** | **型態** | **長度** | **備註** |
| **名字** | name | 魚的名字 | varchar | 10 | Primary Key |
| **種類** | category | 魚的種類 | varchar | 10 | Not Null |
| **年齡** | age | 魚的年齡 | int | 10 | Not Null |
| **性別** | gender | 魚的性別 | varchar | 10 | Not Null |
| **飽食度** | satiation | 魚的飽食度 | int | 10 | Not Null |
| **長度** | length | 魚的長度 | double | 20,10 | Not Null |
| **重量** | weight | 魚的重量 | double | 20,10 | Not Null |
| **排泄度** | excretion | 魚的排泄狀態 | int | 10 | Not Null |
| **生病** | sick | 魚的健康狀態 | tinyint | 1 | Not Null |
| **生命力** | life | 魚的生命力 | double | 20,10 | Not Null |

**5-3 background Table**

**CREATE TABLE** background (

temperature **double(20,10) NOT NULL**,

pHValue **double(20,10) NOT NULL**,

oxygenContent **double(20,10)**  **NOT NULL**,

cleanliness **double(20,10)** **NOT NULL**,

season **varchar(10)** **NOT NULL**,

date **bigint(20) NOT NULL**,

time **bigint(20) NOT NULL,**

endTime **bigint(50) NOT NULL,**

);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位名稱** | **欄位代號** | **定義** | **型態** | **長度** | **備註** |
| **溫度** | temperature | 環境的溫度 | double | 20,10 | Not Null |
| **pH值** | pHValue | 環境的pH值 | double | 20,10 | Not Null |
| **含氧量** | oxygenContent | 環境的含氧量 | double | 20,10 | Not Null |
| **乾淨度** | cleanliness | 環境的乾淨度 | double | 20,10 | Not Null |
| **季節** | season | 環境的季節 | varchar | 10 | Not Null |
| **日期** | date | 結束時之日期 | bigint | 20 | Not Null |
| **時間** | time | 結束時之時間 | bigint | 20 | Not Null |
| **結束之毫秒** | endTime | 結束時之實際毫秒 | bigint | 50 | Not Null |

**5-4 background Table**

**CREATE TABLE** button (

autofeed **boolean NOT NULL**,

autoFeedTime **int (10) NOT NULL**,

inflator **boolean**  **NOT NULL**,

changeTem **boolean** **NOT NULL**,

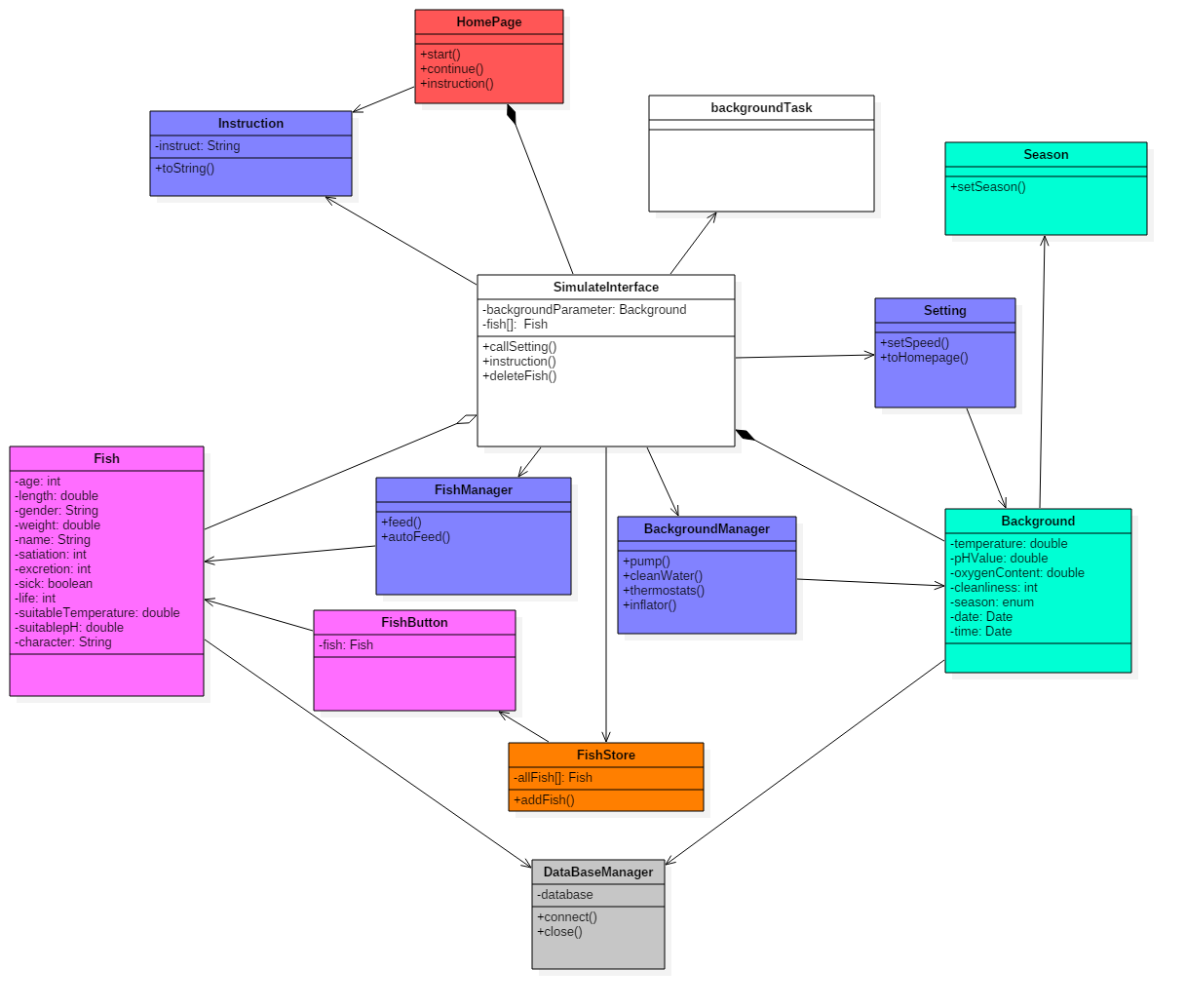
changeTemDegree **int(10)** **NOT NULL**,

filter **boolean NOT NULL**,

);

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **欄位名稱** | **欄位代號** | **定義** | **型態** | **長度** | **備註** |
| **自動餵食按鈕** | autofeed | 判斷自動餵食按鈕有無點擊 | boolean | - | Not Null |
| **自動餵食時間** | autoFeedTime | 紀錄自動餵食時間 | int | 10 | Not Null |
| **打氣機按鈕** | inflator | 判斷打氣機按鈕有無點擊 | boolean | - | Not Null |
| **調溫器按鈕** | changeTem | 判斷調溫器按鈕有無點擊 | boolean | - | Not Null |
| **調溫器溫度** | changeTemDegree | 記錄調溫器溫度 | int | 10 | Not Null |
| **過濾機** | filter | 判斷過濾機按鈕有無點擊 | boolean | - | Not Null |

1. **類別圖設計(Class Diagram)**



紅色：首頁模組 白色：模擬模組 橘色：魚店模組 粉紅色：魚模組

湖水綠：背景模組 藍色：功能模組 灰色：資料庫模組

|  |  |
| --- | --- |
| **Class Name** | HomePage |
| **Module** | 首頁模組 |
| **Description** | 開啟程式的第一個頁面，有Start、Continue、Instruction可以點選 |
| **Relationship with other classes** | Instruction、SimulateInterface |
| **Attributes** | 無 |
| **Traceability with other use case** | 首頁、模擬介面 |

|  |  |
| --- | --- |
| **Class Name** | SimulateInterface |
| **Module** | 模擬模組 |
| **Description** | 應用程式主頁面，顯示魚缸資訊、魚資訊，並提供各種功能按鈕來控制魚缸資訊、魚資訊 |
| **Relationship with other classes** | HomePage、Instruction、Setting、FishManager、BackgroundManager、Fish、FishStore、Background |
| **Attributes** | backgroundParameter、fish[] |
| **Traceability with other use case** | 魚缸資訊、魚類資訊、魚店介面 |

|  |  |
| --- | --- |
| **Class Name** | Instruction |
| **Module** | 功能模組 |
| **Description** | 說明頁面，說明應用程式的操作方法 |
| **Relationship with other classes** | HomePage、SimulateInterface |
| **Attributes** | instruct |
| **Traceability with other use case** | 首頁、模擬介面 |

|  |  |
| --- | --- |
| **Class Name** | Setting |
| **Module** | 功能模組 |
| **Description** | 提供設定日期及遊戲速度的功能，並可以返回首頁 |
| **Relationship with other classes** | SimulateInterface、Background |
| **Attributes** | 無 |
| **Traceability with other use case** | 模擬介面、魚缸資訊 |

|  |  |
| --- | --- |
| **Class Name** | FishManager |
| **Module** | 功能模組 |
| **Description** | 提供有關魚的操作 |
| **Relationship with other classes** | SimulateInterface、Fish |
| **Attributes** | 無 |
| **Traceability with other use case** | 魚類資訊、餵食 |

|  |  |
| --- | --- |
| **Class Name** | BackgroundManager |
| **Module** | 功能模組 |
| **Description** | 提供有關背景的操作 |
| **Relationship with other classes** | SimulateInterface、Background |
| **Attributes** | 無 |
| **Traceability with other use case** | 魚缸資訊、調溫、換水、打氣機、垃圾桶 |

|  |  |
| --- | --- |
| **Class Name** | Fish |
| **Module** | 魚模組 |
| **Description** | 儲存各種魚類資訊 |
| **Relationship with other classes** | SimulateInterface、FishManager、FishStore、FishButton |
| **Attributes** | age、length、gender、weight、name、satiation、excretion、sick、  life、suitableTemperature、suitablepH、character |
| **Traceability with other use case** | 魚類資訊 |

|  |  |
| --- | --- |
| **Class Name** | Background |
| **Module** | 背景模組 |
| **Description** | 儲存各種背景資訊 |
| **Relationship with other classes** | Setting、BackgroundManager |
| **Attributes** | Temperature、pHValue、oxygenContent、cleanliness、season、date、time |
| **Traceability with other use case** | 魚缸資訊 |

|  |  |
| --- | --- |
| **Class Name** | FishStore |
| **Module** | 魚店模組 |
| **Description** | 讓使用者新增魚至模擬介面 |
| **Relationship with other classes** | SimulateInterface、Fish、FishButton |
| **Attributes** | allFish[] |
| **Traceability with other use case** | 魚類資訊、魚店介面 |

|  |  |
| --- | --- |
| **Class Name** | Database |
| **Module** | 資料庫模組 |
| **Description** | 儲存魚資訊及背景資訊 |
| **Relationship with other classes** | Fish、Background |
| **Attributes** | 無 |
| **Traceability with other use case** | 魚缸資訊、魚類資訊 |

|  |  |
| --- | --- |
| **Class Name** | FishButton |
| **Module** | 魚模組 |
| **Description** | 撰寫魚的監聽事件 |
| **Relationship with other classes** | Fish、FishStore |
| **Attributes** | fish |
| **Traceability with other use case** | 模擬介面、魚類資訊、魚店介面 |

1. **實作技術(Implementation Languages and Platforms)**

本系統將以視窗的方式呈現，前、後端皆採用Java撰寫，系統後台之資料庫建構為MySQL環境，預計使用JDBC套件連接MySQL資料庫，並透過SQL的方式取得所需的資料集。

1. **設計議題(Design Issue)**

**議題一：**

系統要做成遊戲還是模擬介面？

**解決方案：**

模擬介面

**理由：**

因為市面上已經有不少以遊戲為主的養魚程式，鮮少以模擬為主，因此，本系統注重模擬，讓想要養魚的使用者可以先了解真實的情況。

**議題二：**

是否使用金錢功能？

**解決方案：**

不使用

**理由：**

因為本系統所注重的是"模擬"，因此使用者不需擔心金錢的問題，即可模擬現實的情況。

**議題三：**

是否增加繼續遊戲的選項？

**解決方案：**

實作繼續遊戲及資料庫

**理由：**

本系統注重"模擬"，要實現真實養魚的情況，使用者必須長期使用本系統，為了讓使用者可以不必一直守在電腦前觀察，本系統實作資料庫儲存當前的資料，讓使用者可以在下次進入系統時，回復到上次的紀錄當中。

**議題四：**

是否讓使用者可以幫魚取名字？

**解決方案：**

是

**理由：**

同種魚種可能有不只一隻，為了方便使用者可以辨識不同的魚，本系統提供幫魚取名字的功能，同時也有利於資料庫的建立使用。

**議題五：**

是否實作背包功能？

**解決方案：**

不實作

**理由：**

原本設想將欲刪除的魚，先放進背包，再從背包將魚丟掉，但由於背包的功能單一，無其他功能，多此一舉！因此只實作垃圾桶功能，將魚直接拖曳到垃圾桶即可完成刪除動作。

**議題六：**

是否實作加速功能？

**解決方案：**

實作加速功能

**理由：**

為了讓使用者可以快速得知養魚的成果，本系統提供加速功能，讓使用者得到快速模擬的體驗。

**議題七：**

關閉系統後，魚缸資訊、魚資訊是否要繼續運行？

**解決方案:**

實作

**理由:**

因為此系統注重"模擬"，因此即使使用者關閉系統，此系統還是能依據現實情況使魚缸資訊、魚資訊隨著時間改變。

舉例：若現在虛擬時間為12/13 7:00，且加速倍率為4倍，關閉系統後，會將此時資料存進資料庫，且速度變為和現實生活相同，若12/14 7:00再度開啟系統，則會讀取關閉時的資料，並做一天的變化計算非四天。

**議題八：**

是否加入暫停功能

**解決方案:**

是

**理由:**

為提供更彈性的使用者體驗，使忙碌的使用者即使多天未使用此系統，也不至於讓魚死亡。

**議題九：**

模擬系統演算法

**解決方案:**

按鈕：

餵食一次：每一隻魚飽食度+5，滿是100

若餵食過多(每一隻魚飽食度皆100)，多一次餵食，ph值上升0.1，

乾淨度下降2%

自動餵食：可於設定中設定每多少分鐘餵食一次(飽食度變化同上)，可主動關閉

打氣：依溫度決定含氧量上升到最大值如下，速度為每小時+0.2ppm

0-10度：11ppm 11-20：9ppm 21-30：7ppm 31-40：6ppm

調溫器：隨時間趨向設定之溫度，趨近速度為每秒0.001度

過濾機：乾淨度緩慢上升趨近90%，速度為每秒+0.001%，

ph值趨近於7.0，每秒0.001

換水：直接將乾淨度設為100%，含氧量依溫度給予預設值如下，ph值設為7.0

0-10度：11ppm 11-20：9ppm 21-30：7ppm 31-40：6ppm

魚：

年齡：每12天增加一歲

生命力：依年齡決定生命力最高值如下，若溫度不在適溫內、不在適合ph值中

、乾淨度低於40%、含氧量低於4ppm

其一成立則生命力下降速率，每秒下降1，若兩者成立速率為2，依此

類推，若低於40則會生病

0-10歲100 11-20歲90 21-30歲80 31-40歲70 41-50歲60

50以上50

長度：每秒增加0.001-0.003公分

重量；每秒增加0.001-0.005g

排泄：飽食度非0%時隨時間每秒增加0.5%，100%時排泄，造成ph值上升0.02

，排泄完歸零

魚種衝突：紅十字、非洲慈鯛會攻擊其他所有魚種，若此兩種養在一起，體重大

者會攻擊體重小者，被攻擊者生命力每秒下降1-3

生病：生命力下降速率為平常的兩倍

飽食度：飽食度高於80且環境為適溫、適ph、適乾淨度、適含氧量則每秒生命

力增加0.5，最高增至為生命力上限

隨時間飽食度每分鐘下降1

背景(條件為不使用任何功能)：

ph值：不隨時間更動，但隨著死魚的數量(n隻)，每秒下降0.001\*n

乾淨度：不隨時間更動，但隨著死魚的數量(n隻)，每秒下降0.001\*n

含氧量：若魚的數量為n，則速率為每小時下降0.012\*n

溫度：趨近季節的溫度，隨時間每分鐘趨近0.1度

春20 夏25 秋20 冬15

**議題十：**

魚的資料是否要分種類與實體兩層去紀錄？

**解決方案:**

初始魚的資料存在 initial fish table

新new的魚種存在fish table

**理由：**

各種種類魚的初始值和實體魚的屬性不盡相同，實體魚的某些屬性不允許null，而各種魚屬性的初始值可能為null，如name