|  |  |
| --- | --- |
| **IT Graad 11 PAT 2024 : Healthy Living** | |
| **Naam en Van:** | **Gustav Jones** |
| **Projek Naam:** | **Boereraad** |

# **Task 1A : Define the task**

## Topic:

A library for home use medical remedies. This application will contain many different medical remedies that the average human user can access from the application to apply in their daily lives. It will enable the user to submit and find new solutions to their medical problems while being administrated to ensure correct information.

## Purpose of program:

To create a bridge between users and local knowledge passed on through generations and tested by medical administrators. It will house all the medical knowledge any user will need for home medical needs all in one place. It will also enable the user to keep track and review remedies they have used in the past.

## Possible solution:

A pending changes system. A system where the user has the freedom to add and edit information about home remedies with ease, while an administrator reviews the pending changes. The application will have 2 sides, user and administrator. The user will be able to view and edit remedies as well as create reviews for remedies and submit pending changes stored in text files. The administrator will be able to approve or deny changes as well as freely edit the database itself.

## Scope:

This program will not be a social media. It will not contain user interaction, but strictly factual medical information. It will also not be a way to interact with home doctors, but instead serve as a simple way to enhance your health by using everyday household items.

# **Task 1B : User Stories**

| **WHO** | **WHAT** | **WHY** |
| --- | --- | --- |
| User (Average Human) | Access a list of household remedies. | To treat minor symptoms with easy to find household items. |
| User | Edit remedies information. | To update or correct medical information about a remedy. |
| User | Add a new remedy to the collection | To increase the size of the medical information library of the application |
| User | Create a review of the remedy after use to treat symptoms | To create a history of remedies that can be rated on their effectiveness for future use. |
| Administrator (Medical Professional) | Review newly added remedies by the user. | To add pending additions to the application database if the information is factual. |
| Administrator | Review edited remedies information by the user. | To check for any errors made by the user when editing a remedy's information before updating the database. |
| Administrator | View and edit all user's information from an admin page. | To update user information and to list user attached reviews for remedies |
| Administrator | Freely view and edit remedies directly to the database. | To update the information of any selected remedy without the use of users and pending changes. |

# **Task 1B : User Stories - Acceptance Test**

* The user can view a list of remedies in the application database.
* The user can add a new remedy to the pending changes of the database.
* The user can edit an existing remedy to the pending changes of the database.
* The administrator can review additions and changes to new or existing remedies
* The administrator can view and edit any user account in the database and the attached remedy reviews.

# **Task 2 : Data Dictionary**

# **Data structures**

## Text files:

|  |  |
| --- | --- |
| **Name of text file** | RemedyName.txt |
| **Purpose** | To store pending changes or additions made by the user in the application before being saved to the database. A new text file will be created with the remedy name that was edited and written to with the information when the user makes a change to a new/existing remedy. When the administrator reviews the changes, the data will be read from the text file into an array and then stored in the database. |
| **Format** | Price;  EaseOfUse;  List of symptoms to use for;  Description of the remedy; |
| **Extract of data** | 0.17;  3;  Sore throat, Cough;  Mix 1 tablespoon of honey with the juice of half a lemon in warm water. Drink to soothe a sore throat.; |

## Arrays:

|  |  |
| --- | --- |
| **Declaration** | 1) arrPendingChangeRemedyName : array[1..150] of string;  2) arrPendingChangeRemedyInformation : array[1..150] of string; |
| **Populate arrays** | **arrPendingChangeRemedyName:**  Read the contents of a relative 'changes' directory next to the executable and append the names of each text file as a remedy name to the array.  **arrPendingChangeRemedyInformation:**  Open the corresponding file in the first array and read the content to the element in this array |
| **Processing** | Calculate each field information from the serialization within the array element  Sort the Remedies in alphabetical order  Search for a changed remedy. |

## User defined methods

|  |  |
| --- | --- |
| **Signature** | **Explanation** |
| function CalculateFieldInformation(pFieldName : string; pSerializedChanges : string): string; | Calculate the information for a specific field serialized within the pending changes of a remedy text file and return the field value. |
| function GetUserPassword(pUserID : Integer): string; | Gets the password of a specific user passed as a parameter by ID stored in the database. |
| function InitialiseDatabase() : Boolean; | Runs the code to setup the database file provided with the executable and return True on success and False if the database cannot be found. |
| procedure AddRemedy(); | Stores input from various controls to save a new text file with pending addition changes for an added remedy. |
| procedure AddReview(); | Add a new child table record for the logged in user with input from various controls. |

# **Task 3: Database**

## Role of Database: (List of FIVE functions)

1) Create new user records or authenticate existing users with Log In or Sign Up on the home page.

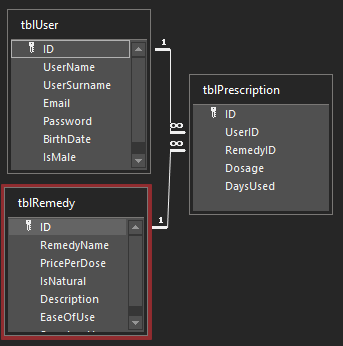
2) List the collection of records of remedies and display them within dynamic tiles on the user side.

3) Create new remedy records created by the user after being authenticated by an admin.

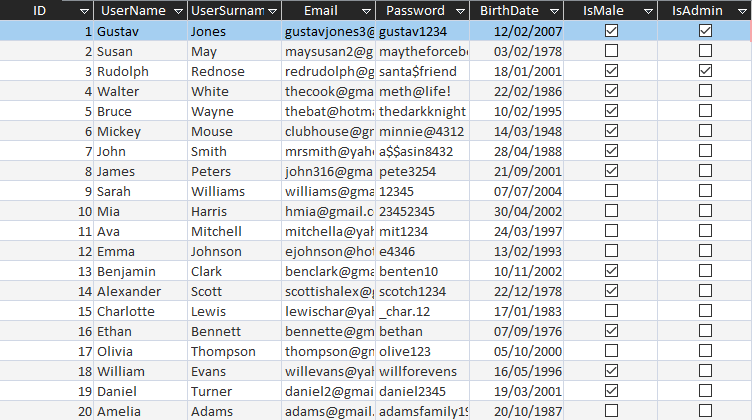
4) Create new review records by the user that links to the user record.

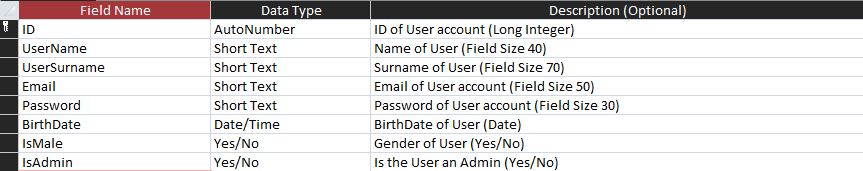
5) Edit user and remedy records by the administrator in the admin page.

## Database design

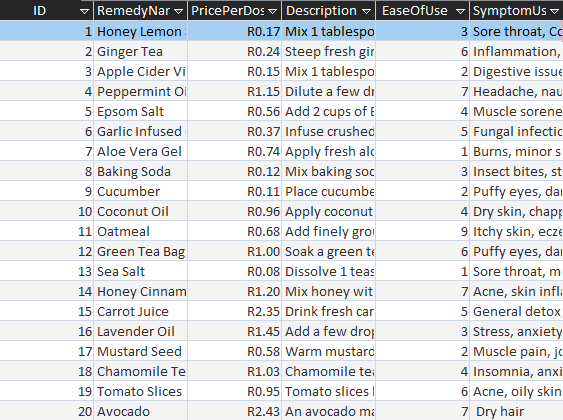
Relations:

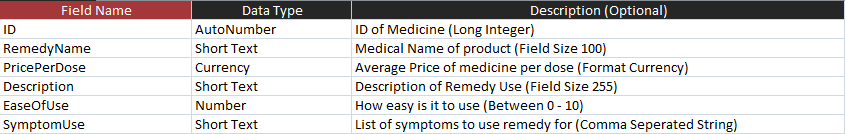
tblUser

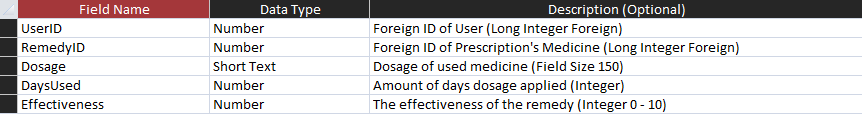
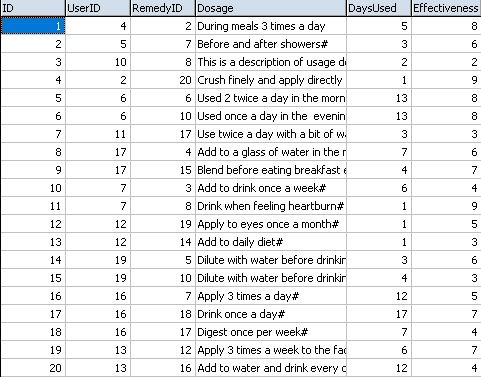




tblRemedy

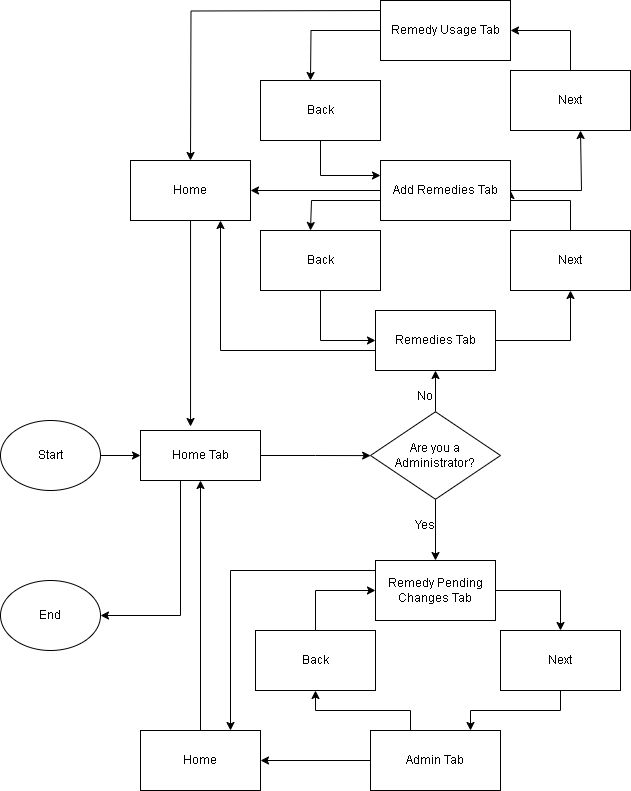


tblReview



# **Task 4 : Navigation & GUI Design**

# **Navigation**

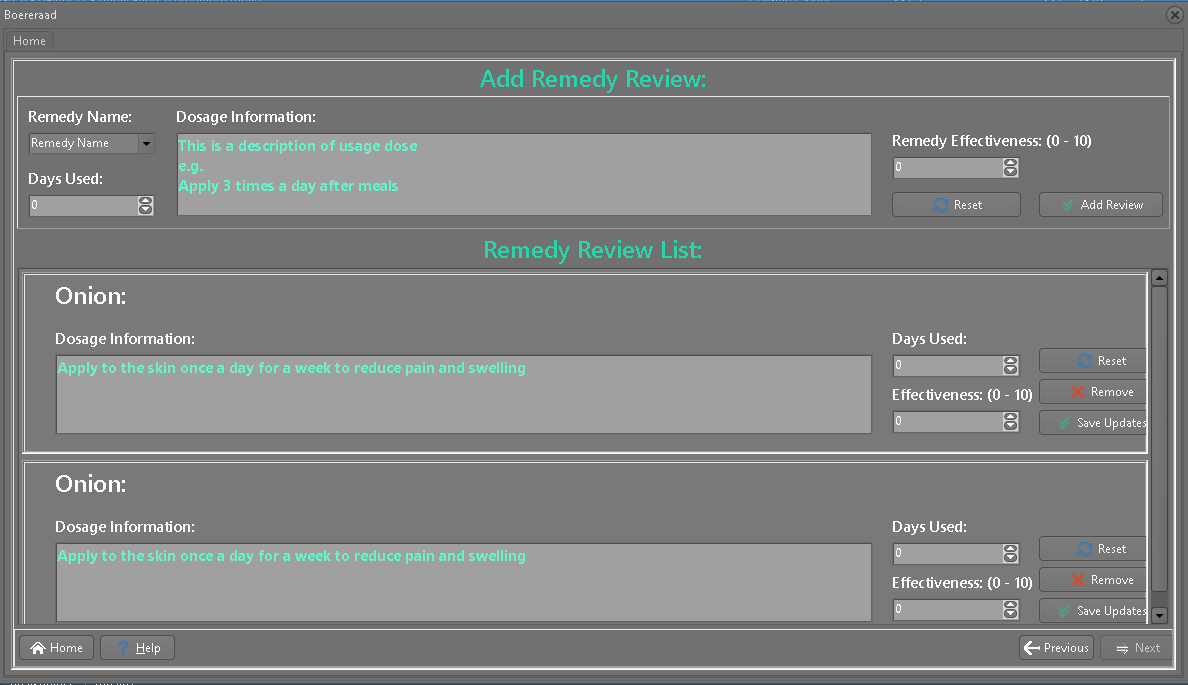


# **GUI Design**

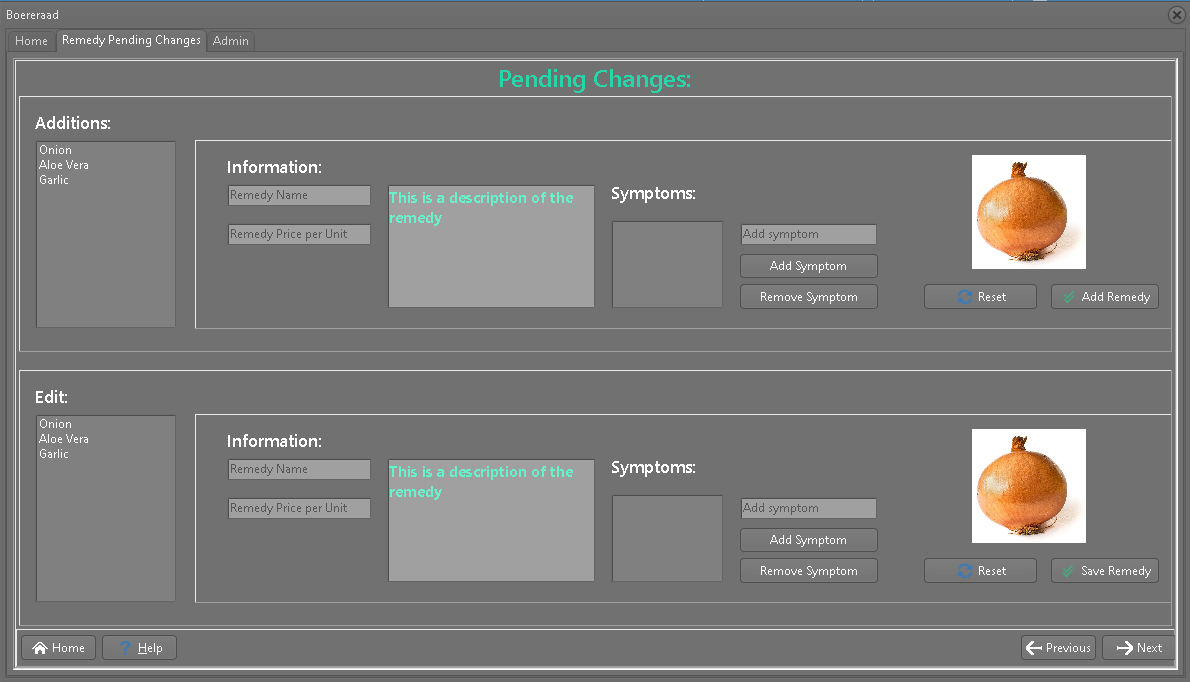
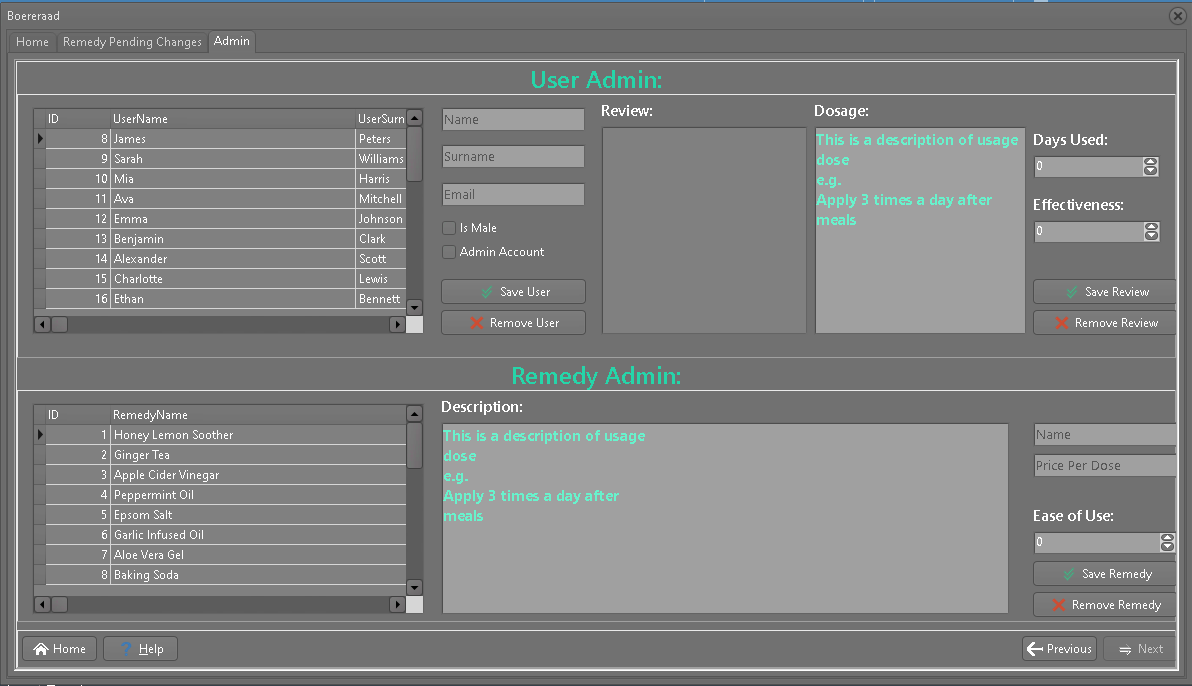
## User:



## 



## Admin:



# **IPO – Software Design Tool (Task 5)**

## INPUT: tsAddRemedy

| **Source** | **Variable name & data type** | **Format (type, size)** | **GUI Object Name** | **Local/Global variable** |
| --- | --- | --- | --- | --- |
| Keyboard | sRemedyName : String | None, String Input | edtAddRemedyInputsRemedyName | Local |
| Keyboard | rRemedyPrice : Real | Currency, 2 decimals | edtAddRemedyInputsPrice | Local |
| Keyboard, Mouse Click | iEaseOfUse : Integer | None, Integer Input | sedAddRemedyInputsEaseOfUse | Local |
| Keyboard | sDescription : String | Concat Each Line | redAddRemedyInputsDescription | Local |
| Keyboard, Mouse Click | arrSymptoms : array of string | For each element in cltAddRemedyInputsSymptoms | cltAddRemedyInputsSymptoms | Local |

## INPUT: tsRemedyPendingChanges

| **Source** | **Variable name & data type** | **Format (type, size)** | **GUI Object Name** | **Local/Global variable** |
| --- | --- | --- | --- | --- |
| Mouse Click | iRemedyIndex : Integer | Selected object in lstRemedyPendingChangesAdditionsRemediesList | lstRemedyPendingChangesAdditionsRemediesList | Local |
| Keyboard | sRemedyName : String | None, String Input | edtRemedyPendingChangesAdditionsInfoName | Local |
| Keyboard | rRemedyPrice : Real | Currency, 2 Decimals | edtRemedyPendingChangesAdditionsInfoPrice | Local |
| Keyboard | sDescription : String | Concat Each Line | redRemedyPendingChangesAdditionsInfoDescription | Local |
| Keyboard, Mouse Click | arrSymptoms : array of strings | For each element in cltRemedyPendingChangesAdditionsInfoSymptoms | cltRemedyPendingChangesAdditionsInfoSymptoms | Local |

## VALIDATION:

| **Input** | **Validation technique and description** | **Associated error message** |
| --- | --- | --- |
| edtAddRemedyInputsRemedyName | Presence, check if no value was given as name | Please enter a name for the remedy |
| edtAddRemedyInputsPrice | Type Check, check if input can be parsed to currency real value | Please enter a valid currency amount |
| lstRemedyPendingChangesAdditionsRemediesList | File Exists Check, check if the item that was selected in the listbox has a corresponding  text file | Remedy Change information not found |
| redRemedyPendingChangesAdditionsInfoDescription | Length Check, check if the description given is within the accepted length (255) of characters | Description may not be longer than 255 characters |

## PROCESSING:

|  |  |
| --- | --- |
| **What processing will be done** | **Algorithms, formulas, etc.** |
| **List:** | **Algorithms:** |
| 1. Creating a new user record with the sign up button |  |
| 2.Add a review as a child object to database |  |
| 3. Authenticating an existing user to the database information including email and password |  |
| 4.Populate a listbox for the pending changes from text file information |  |
| 5. Read and display remedies from database into a combobox (populate combobox from database) | Algorithm 1:  TableIndex <- dmBoereraad.tblRemedy.RecNo  cmbRemedies.Items.Clear  dmBoereraad.tblRemedy.First  while not dmBoereraad.tblRemedy.Eof  Remedy <- dmBoereraad.tblRemedy['RemedyName']  cmbRemedies.Items.Add(Remedy)  dmBoereraad.tblRemedy.Next  end while  dmBoereraad.tblRemedy.RecNo <- TableIndex  Output cmbRemedies |
| 6. Database user search by email | Algorithm 2:  Email <- Input  ID <- 0  dmBoereraad.tblUser.First  while not dmBoereraad.tblUser.Eof  if dmBoereraad.tblUser['Email'] = Email  ID <- dmBoereraad.tblUser['ID']  end while  Output ID |
| 7. Calculate fields from serialized information stored in text files | Algorithm 3:  SerialisedInformation <- Input from text files  SeperatorIndex <- Pos(';', SerialisedInformation)  Price <- StrToFloat(Copy(SerialisedInformation, 1, SeperatorIndex - 1))  Delete(SerialisedInformation, 1, SeperatorIndex)  SeperatorIndex <- Pos(';', SerialisedInformation)  EaseOfUse <- StrToInt(Copy(SerialisedInformation, 1, SeperatorIndex - 1))  Delete(SerialisedInformation, 1, SeperatorIndex)  SeperatorIndex <- Pos(';', SerialisedInformation)  Symptoms <-nCopy(SerialisedInformation, 1, SeperatorIndex - 1)  Delete(SerialisedInformation, 1, SeperatorIndex)  Description <- SerialisedInformation |
| 8.Add an object to global pending changes arrays | Algorithm 4:  Count <- Global  RemedyName <- Input from text file  RemedyInformation <- Input from text file  Inc(Count)  arrPendingChangeRemedyName[Count] <- RemedyName  arrPendingChangeRemedyInformation[Count] <- RemedyInformation  Output parallel arrays |

## OUTPUT: tsRemedies

| **Data for Output** | **Format (Type, size)** | **Output Component** |
| --- | --- | --- |
| Remedy Name | String. Start with capital, with lowercase letters. | Label |
| Remedy Image | Small Resized PNG image of the remedy | Image |
| Remedy Description | Text paragraph with a max length of 255 characters | Richedit |
| Symptoms Use | A list of symptoms to use the remedy for in a listbox | ListBox |
| Natural Remedy | A boolean value to indicate if the remedy is natural | Checkbox |
| Remedy Price | A real value formatted to currency, rounded to 2 decimals indicating the remedy price | Edit |

## OUTPUT: tsAdmin

| **Data for Output** | **Format (Type, size)** | **Output Component** |
| --- | --- | --- |
| User Database Information | All user records saved in the database sorted by rows with columns as fields selectable to display in other controls | DBGrid |
| User Name | Display the user record name as a string | Edit |
| User Surname | Display the user record surname as a string | Edit |
| User Email | Display the user record email as a string | Edit |
| User Gender | Display the user record gender as a boolean. True if male, false if female | Checkbox |
| User Type | Display the user record type as a boolean. True if Admin, False if only User. | Checkbox |
| User Reviews | Display the user record connected reviews from the database in a list as strings | Listbox |
| Review Dosage | Display the dosage of the selected prescription as a string | Richedit |
| Review Days Used | Display the amount of days a prescription was used | SpinEdit |
| Review Effectiveness | Display the effectiveness (0 - 10) of the remedy in the review | SpinEdit |