Oblig 3

Generated by Doxygen 1.8.3.1

Mon Mar 18 2013 17:17:37

Contents

Chapter 1

Hierarchical Index

1.1 Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

Chapter 2

Class Index

2.1 Class List

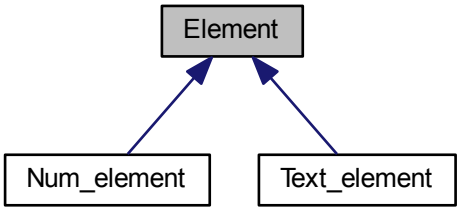
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Class Documentation

3.1 Element Class Reference

Inheritance diagram for Element:



Public Member Functions

- char **get_type** ()
- virtual int **compare** (Element *el)
- virtual void **display** ()

Friends

- class **Num_element**
- class **Text_element**

The documentation for this class was generated from the following files:

- list_tool.h
- list_tool.cpp

3.2 List Class Reference

Public Member Functions

- **List** (listtype li)
- int **is_empty** ()
- int **no_of_elements** ()
- bool **add** (Element ∗el)
- Element ∗ **remove** ()
- Element ∗ **remove** (int no)
- Element ∗ **remove** (const char ∗t)
- Element ∗ **remove_no** (int n)
- bool **destroy** ()
- bool **destroy** (int no)
- bool **destroy** (const char ∗t)
- bool **in_list** (int no)
- bool **in_list** (const char ∗t)
- bool **display_element** (int no)
- bool **display_element** (const char ∗t)
- void **display_list** ()

The documentation for this class was generated from the following files:

- list_tool.h
- list_tool.cpp

3.3 localdb Class Reference

`#include <localdb.h>`

Public Member Functions

- MedList GetDrugs (const std::string search, const int search_type)
- medicine SelectDrug ()
- void AddPrescription ()
- void AddPerson (int type)
- void ListPeople (int type)
- void ListDrugs ()
- void ListPrescriptions (std::list< prescription > pr)
- void PurgeOld (time_t dt)
- void ReadXML ()
- void WriteXML ()
- std::list< prescription > GetPrescribed (std::string search, int how=BY_DRUG)

Public Attributes

- PrescriptionList ∗ **Prescriptions**

3.3.1 Detailed Description

This class is used to map everything – it's the local database Through this class, interaction with the stored data takes place.

3.3.2 Member Function Documentation

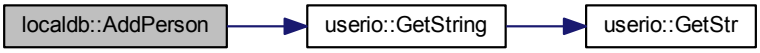
3.3.2.1 void localdb::AddPerson (int *type*)

Requests information from the user to add a new person (DOCTOR or PATIENT as defined by type parameter)

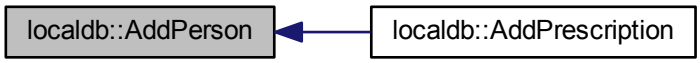
Parameters

int	type Determines if we add a doctor or a patient

Here is the call graph for this function:



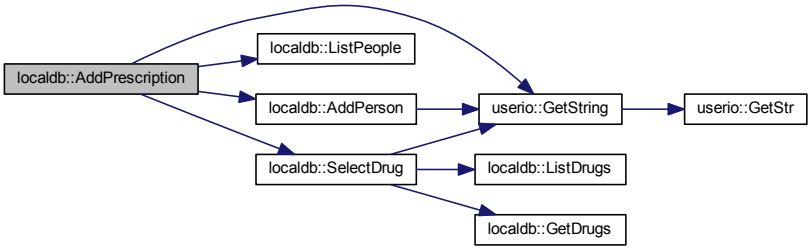
Here is the caller graph for this function:



3.3.2.2 void localdb::AddPrescription ( )

Requests information from the user to add a new prescription. Also adds new doctors / patients if these don't already exist.

Here is the call graph for this function:



3.3.2.3 MedList localdb::GetDrugs (const std::string *search*, const int *search_type*)

Searches rxnorm's medical database for the substance defined by the search parameter. Either by ID, name or estimated name match as defined by the search_type parameter.

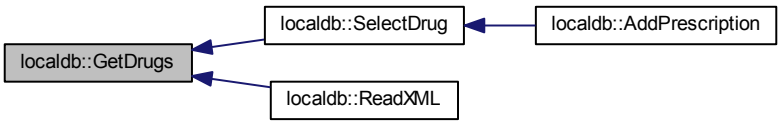
Parameters

const	string search Contains the search string
const	integer search_type How to search the API

Returns

integer

Here is the caller graph for this function:



3.3.2.4 std::list< prescription > localdb::GetPrescribed (std::string *search*, int *how* = BY_DRUG)

This method just routes the call depending on the 'how' parameter. i.e if how == BY_DRUG it will return the returnvalue for GetPrescriptionByDrugID It then returns a list of prescriptions for that particular drug / doctor / patient

Parameters

string	Search string (i.e doctors name, patient SSN, or drug ID)
int	how What criteria to find prescriptions for (BY_DRUG, BY_NAME, BY_DOCTOR)

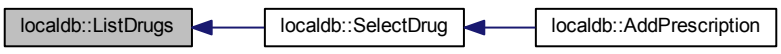
Returns

list<prescription>

3.3.2.5 void localdb::ListDrugs ( )

Lists all drugs in our medical list

Here is the caller graph for this function:



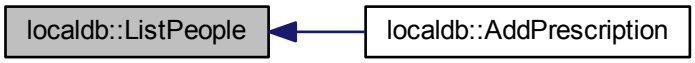
3.3.2.6 void localdb::ListPeople (int *type*)

Lists all people of type (DOCTOR or PATIENT) Lists all people if no type is specified.

Parameters

int	type Indicates if we're listing doctors or patients

Here is the caller graph for this function:



3.3.2.7 void localdb::ListPrescriptions (std::list< **prescription** > *pr*)

Lists all prescriptions in a prescriptionlist

Parameters

list<prescription>	

3.3.2.8 void localdb::PurgeOld (time_t *dt*)

Purges old prescriptions

Parameters

time_t	dt Timestamp-limit – anything older than this will be purged

3.3.2.9 void localdb::ReadXML ( )

Reads the content of our local XML database file and populates the MedLists and People lists accordingly

Here is the call graph for this function:

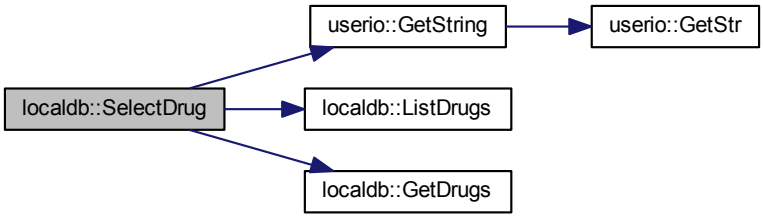
3.3.2.10 medicine localdb::SelectDrug ( )

Lets the user select a drug by entering the drug ID The drug ID can be found online (as provided to the user should he enter 'search') or locally, if a previous prescription has been made for that drug (as provided to the user should he enter 'list'). Returns the selected drug as a medicine struct

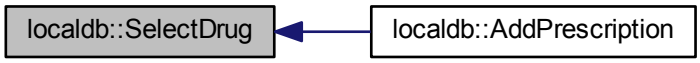
Returns

medicine

Here is the call graph for this function:



Here is the caller graph for this function:



3.3.2.11 void localdb::WriteXML ( )

Outputs all stored data to local XML database file Pretty much the same as the ReadXML() method, but reversed

The documentation for this class was generated from the following files:

- localdb.h
- localdb.cpp

3.4 medicine Struct Reference

Public Attributes

- std::string **name**
- std::string **market_name**
- std::string **rxnorm_id**

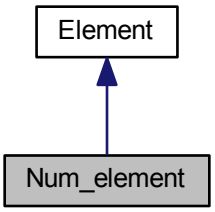
- std::string **concept_id**
- std::string **strength**
- std::string **quantity**

The documentation for this struct was generated from the following file:

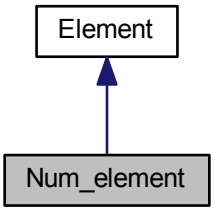
- prescription.h

3.5 Num_element Class Reference

Inheritance diagram for Num_element:



Collaboration diagram for Num_element:



Public Member Functions

- **Num_element** (int no)
- virtual int **compare** (Element *el)

Protected Attributes

- int **number**

The documentation for this class was generated from the following files:

- list_tool.h
- list_tool.cpp

3.6 person Class Reference

Public Member Functions

- std::string **GetFirstName** ()
- std::string **GetLastName** ()
- std::string **GetAddress** ()
- std::string **GetSSN** ()
- std::string **GetZip** ()
- std::string **GetPhone** ()
- void **SetFirstName** (std::string val)
- void **SetLastName** (std::string val)
- void **SetAddress** (std::string val)
- void **SetSSN** (std::string val)
- void **SetZip** (std::string val)
- void **SetPhone** (std::string val)

The documentation for this class was generated from the following files:

- person.h
- person.cpp

3.7 prescription Class Reference

Public Member Functions

- **prescription** (struct tm date)
- **prescription** (person ∗dr, person ∗ptnt, time_t dt=time(NULL))
- **prescription** (person ∗dr, person ∗ptnt, medicine ∗drg, time_t dt=time(NULL))
- time_t **GetDate** ()
- std::string **GetDosage** ()
- person & **GetDoctor** ()
- person & **GetPatient** ()
- medicine ∗ **GetDrug** ()
- void SetHour (int h)
- void SetMinute (int m)
- void SetSecond (int s)
- void SetDay (int d)
- void SetMonth (int m)
- void SetYear (int y)
- void SetDate (time_t d)
- void SetTime (time_t date=time(NULL))
- void **SetDrug** (medicine ∗med)
- void **SetDoctor** (person ∗dr)
- void **SetPatient** (person ∗ptnt)

3.7.1 Member Function Documentation

3.7.1.1 void prescription::SetDate (time_t *d*)

Sets the timestamp manually

Parameters

integer	d Timestamp

Here is the caller graph for this function:

3.7.1.2 void prescription::SetDay (int *d*)

Sets the day specified by integer parameter d

Parameters

integer	d Day of the month to set time to

3.7.1.3 void prescription::SetHour (int *h*)

Sets the hour specified by integer parameter h

Parameters

integer	h Hour to set time to

3.7.1.4 void prescription::SetMinute (int *m*)

Sets the minute specified by integer parameter m

Parameters

integer	m Minute to set time to

3.7.1.5 void prescription::SetMonth (int *m*)

Sets the month specified by integer parameter m

Parameters

integer	m Month to set time to

3.7.1.6 void prescription::SetSecond (int *s*)

Sets the second specified by integer parameter s

Parameters

integer	s Second to set time to

3.7.1.7 void prescription::SetTime (time_t *d* = `time(NULL)`)

Sets the timestamp manually (AND updates tmpDate accordingly)

Parameters

integer	d Timestamp

Here is the call graph for this function:

3.7.1.8 void prescription::SetYear (int *y*)

Sets the year specified by integer parameter y

Parameters

integer	y Year to set time to

The documentation for this class was generated from the following files:

- prescription.h
- prescription.cpp

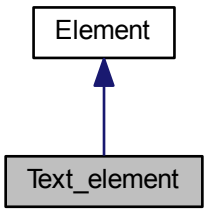
3.8 substance Class Reference

The documentation for this class was generated from the following files:

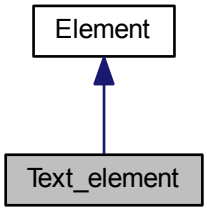
- substance.h
- substance.cpp

3.9 Text_element Class Reference

Inheritance diagram for Text_element:



Collaboration diagram for Text_element:



Public Member Functions

- **Text_element** (const char ∗t)
- virtual int **compare** (Element ∗el)

Protected Attributes

- char ∗ **text**

The documentation for this class was generated from the following files:

- list_tool.h
- list_tool.cpp

3.10 userio Class Reference

Public Types

- enum {
  NOFILTER, **MIN**, **MAX**, **BETWEEN**,
  EQUALS, **NOT**, **CONTAINS** }

Static Public Member Functions

- static char GetChar (std::string pre, int how, char alt1= '-', char alt2= '-')
- static int GetInt (std::string pre, int how, int alt1, int alt2=0)
- static std::string GetString (std::string pre, int how, std::string str1="", const int min=0, unsigned const int max=0)
- static std::string GetString (std::string pre, std::string regex, std::string readable_regex)
- static time_t GetTimestamp (std::string pre)
- static struct tm GetTime (std::string pre)
- static struct tm GetDate (std::string pre)
- static time_t MakeTimestamp (struct tm clock, struct tm date)
- static char GetChar ()
- static int GetNum ()
- static char * **GetpChar** ()
- static std::string GetStr ()

3.10.1 Member Function Documentation

3.10.1.1 char userio::GetChar (std::string *pre,* int *how,* char *alt1 = '-',* char *alt2 = '-'*) [static]

Sanitizes a char (as determined by the interaction enum 'how')

Parameters

string	pre Text to print before catching users input
integer	how Determines how the expected char should be
char	alt1 Used when 'how' is EQUALS, BETWEEN, NOT, or CONTAINS to compare against
char	alt2 Used when 'how' is EQUALS or BETWEEN in conjunction with alt1

Returns

char

Here is the call graph for this function:

3.10.1.2 char userio::GetChar ( ) [static]

Gets a char from the user through the input stream

Returns

char

Here is the caller graph for this function:

3.10.1.3 struct tm userio::GetDate (std::string *pre*) [static],[read]

Creates a timestruct from userprovided date.

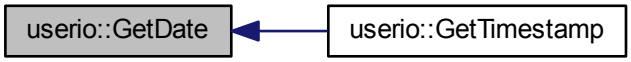
Parameters

string	pre Pretext to show before taking input

Returns

struct tm

Here is the caller graph for this function:



3.10.1.4 int userio::GetInt (std::string *pre,* int *how,* int *alt1,* int *alt2* = 0) [static]

Sanitizes an integer (as determined by the interaction enum 'how')

Parameters

string	pre Text to print before catching users input
integer	how Determines how the expected integer should be
int	alt1 Used when 'how' is EQUALS, BETWEEN, NOT, or CONTAINS to compare against
int	alt2 Used when 'how' is EQUALS, BETWEEN or NOT in conjunction with alt1

Returns

int

Here is the call graph for this function:

3.10.1.5 int userio::GetNum ( ) [static]

Gets an integer from the user through the input stream

Returns

integer

Here is the caller graph for this function:

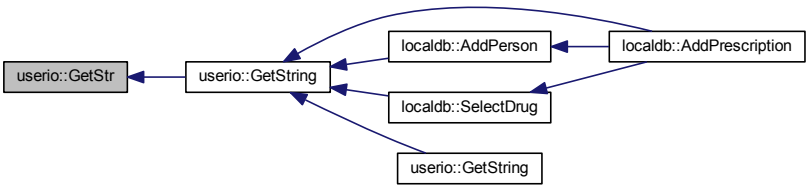
3.10.1.6 std::string userio::GetStr ( ) [static]

Gets a string from the user through the input stream

Returns

string

Here is the caller graph for this function:



3.10.1.7 std::string userio::GetString (std::string *pre*, int *how*, std::string *str1* = " ", const int *min* = 0, unsigned const int *max* = 0) [static]

Sanitizes a string (as determined by the interaction enum 'how')

Parameters

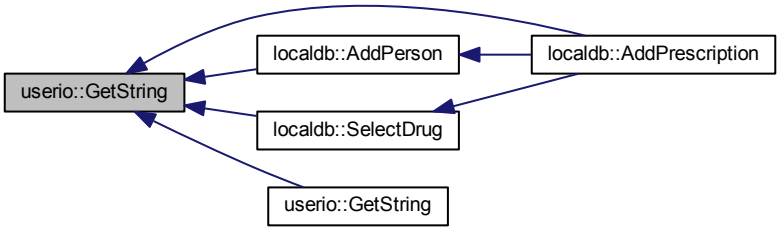
string	pre Text to print before catching users input
integer	how Determines how the expected integer should be
int	str1 Used when 'how' is not set to NOFILTER to compare against
int	min Used when how is set to MIN or BETWEEN
int	max Used when how is set to MAX or BETWEEN

Returns

string

Here is the call graph for this function:

Here is the caller graph for this function:



3.10.1.8 std::string userio::GetString (std::string *pre*, std::string *regex*, std::string *readable_regex*) [static]

Gets a string from the user through the input stream only if it matches the specified regex pattern.

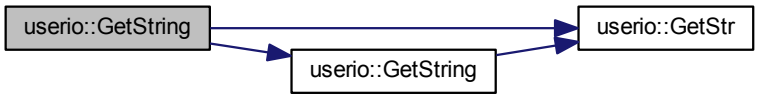
Parameters

string	pre
string	regex

Returns

string

Here is the call graph for this function:



3.10.1.9 struct tm userio::GetTime (std::string *pre*) [static], [read]

Creates a timestruct from userprovided hour and time.

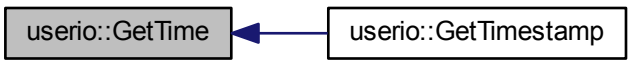
Parameters

string	pre Pretext to show before taking input

Returns

struct tm

Here is the caller graph for this function:



3.10.1.10 time_t userio::GetTimestamp (std::string *pre*) [static]

Creates requests date and time from user, and returns timestamp

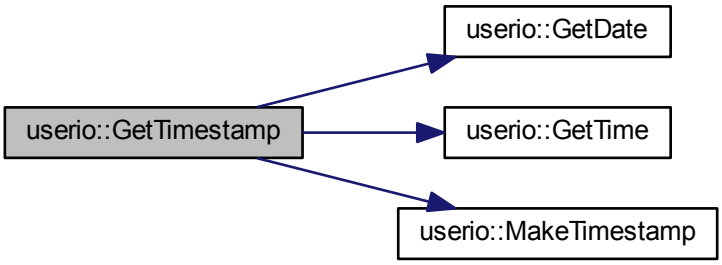
Parameters

string	pre Pretext to show before taking input

Returns

time_t

Here is the call graph for this function:



3.10.1.11 time_t userio::MakeTimestamp (struct tm *clock*, struct tm *date*) [static]

Merges time and date from two different tm structs

Parameters

struct	tm clock Time struct
struct	tm date Date struct

Returns

time_t

Here is the caller graph for this function:

The documentation for this class was generated from the following files:

- userio.h
- userio.cpp

Index