Traveller's app

1.0

Generated by Doxygen 1.8.17

Chapter 1

Traveller's App

A project for FMI's OOP course

1.1 Idea

Traveller's App is a simple CLI-based application for storing entries of visits to different destinations around the world.

Upon launch, the only available functionality is to register a user or to log in.

Once logged in, we can view added destinations and their ratings, add destinations ourselves, add trips to any of those destinations, add friends and see if any of them have visited the same destinations as us and more.

1.2 Running the application

The application is most-easily run on Linux. All you have to do is execute `build/travellers_app` from the main project directory and the program will launch.

1.3 Doxygen

Doxygen-generated documentation is part of the project. You can access it via `html/index/html` for the html version.

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 Comment Class Reference

A comment to a Trip has the value of the comment itself, and the username of the commenter (i.e. the visitor)

```
#include <comment.hpp>
```

Public Member Functions

- **Comment** (String value, String from)
- bool write_to_bin (std::ofstream &of_stream)

  Write comment string and username of user which posted it.
- bool read_from_bin (std::ifstream &if_stream)

  Read comment string and username of poster.
- String get_value () const

  Get comment string.
- String get_from () const

  Get username string.

3.1.1 Detailed Description

A comment to a Trip has the value of the comment itself, and the username of the commenter (i.e. the visitor)

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

- comment.hpp
- comment.cpp

3.2 Database Class Reference

Database class, responsible for storing currently logged in user and all destinations.

```
#include <database.hpp>
```

Public Member Functions

- Database ()

  Set current user to nullptr and load all destinations from destinations.db.
- Database (const Database &other)

  Copy destinations and make a new instance of user.
- Database & operator= (const Database &other)

  Same as copy constructor, but free memory first.
- ∼Database ()

  Free user memory.
- User ∗ get_user_by_username (const char ∗username) const

  Get user from users.db by name.
- User ∗ get_user_by_email (const char ∗email) const

  Get user from users.db by email.
- Destination ∗ get_destination_by_name (const char ∗name) const

  Get destinaiton by its name, nullptr else.
- const Vector< Destination > & get_destinations () const

  Get an immutable list of the destinations.
- bool add_user (User user) const

  Save new user to users.db and create personal db file.
- bool log_in (const char ∗username, const char ∗password)

  Set current user.
- bool log_out ()

  Save current user's information and log him out.
- User ∗ get_curr_user () const

  Get currently logged-in user.
- bool save_destinations () const

  Write destinations to destinations.db.
- bool save_user () const

  Write user to <username>.db.
- bool add_destination (Destination dest)

  Add new destination to local vector; will be saved to file at close.
- void add_trip_curr_user (Trip trip)

  Add new trip for logged-in user.
- Trip ∗ get_trip_by_user_for_dest (String username, const Destination &existing_destination)

  Get trip by user for destination, if exists (else nullptr)

3.2.1 Detailed Description

Database class, responsible for storing currently logged in user and all destinations.

The rest of the necessary information is gathered from the user file(s).

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

- database.hpp
- database.cpp

3.3 Date Class Reference

A date, represented as day, month and year.

```
#include <date.hpp>
```

Public Member Functions

- Date ()
  Date defaults to 01-01-2000.
- Date (const char ∗date_string)
  Fill out date from ISO 8601 string (only set if valid, else default to 01-01-2000)
- int get_day () const
  Get date's day.
- int get_month () const
  Get date's month.
- int get_year () const
  Get date's year.
- bool set_date (const char ∗date_string)
  Set date via string (only if valid)
- bool set_date (int day, int month, int year)
  Set date via numbers.
- bool is_valid_date_string (const char ∗date_string) const
  Check if date string is valid.
- bool is_valid_date (int day, int month, int year) const
  Check if is a valid date.
- bool is_leap_year (int year) const
  Check if year is a leap year.
- bool write_to_bin (std::ofstream &of_stream)
  Write date as day->month->year in binary file.
- bool read_from_bin (std::ifstream &if_stream)
  Read from binary file as day->month->year.

3.3.1 Detailed Description

A date, represented as day, month and year.

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

- date.hpp
- date.cpp

3.4 Destination Class Reference

Simple destination class, containing its name, number of visits, and rating.

```
#include <destination.hpp>
```

Public Member Functions

- Destination (String name="")

  The most-used constructor, for when creating a new destination. (can be default for array purposes)
- Destination (String name, double acc_rating, int num_visits)

  Full constructor.
- bool read_from_bin (std::ifstream &if_stream)

  Load destination from binary file.
- bool write_to_bin (std::ofstream &of_stream) const

  Write destination to binary file.
- String get_name () const

  Get destination name.
- int get_num_visits () const

  Get number of visits of destination.
- double get_avg_rating () const

  Get average rating of trips to destination.
- bool add_visit (double rating)

  Add a visit entry with a rating.

3.4.1 Detailed Description

Simple destination class, containing its name, number of visits, and rating.

Due to the nature of the file structure, trips are not stored as part of this class. Trips are stored for each user, instead of for each destination.

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

- destination.hpp
- destination.cpp

3.5 IOHandler Class Reference

A helper class to handle parsing commands and arguments.

```
#include <io_handler.hpp>
```

Public Member Functions

- void input_command ()

  Get first string from the console.
- void input_args (std::istream &i_stream)

  Get arguments, separated by space.
- String get_command () const

  Get current command.
- Vector< String > get_args () const

  Get current arguments in a list.
- bool check_number_of_arguments (int num_of_args) const

Returns whether the number of current arguments is 'num_of_args'.

- void print_message (String message, String prefix="") const

  Print message.
- void print_prompt () const

  Print shell prompt.
- void print_usage (String command, String usage="", bool with_prefix=true) const

  Print usage of command.
- void print_error (String desc) const

  Print error with description.
- void print_not_logged_in () const

  Print that there is no logged in user.
- void print_error_explain (String desc) const

  Same as print error, but without the "Error: " prefix.
- void print_unknown_command () const

  Print unknown command.
- void print_success (String message) const

  Print success with message.
- void print_help () const

  Print help on how to use the app.

3.5.1 Detailed Description

A helper class to handle parsing commands and arguments.

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

- io_handler.hpp
- io_handler.cpp

3.6 TravellersApp Class Reference

The outermost class of the application.

```
#include <travellers_app.hpp>
```

Public Member Functions

- void run ()

  Run app's main loop.

3.6.1 Detailed Description

The outermost class of the application.

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

- travellers_app.hpp
- travellers_app.cpp

3.7 Trip Class Reference

Contains information about a Trip from a User.

```
#include <trip.hpp>
```

Public Member Functions

- Trip (String dest_name, Date start_date, Date end_date, int rating, Comment comment, Vector< String > photos)

  Constructor for creating a new Trip.
- Trip (std::ifstream &if_stream)

  Constructor for creating a Trip instance out of a file.
- bool write_to_bin (std::ofstream &if_stream)

  Write the trip's information to a binary file.
- bool read_from_bin (std::ifstream &of_stream)

  Load the trip's information from a binary file.
- Comment get_comment () const

  Get copy of the trip's comment.
- Vector< String > get_photo_names () const

  Get copy of photo filenames list.
- Date get_start_date () const

  Get beginning date of trip.
- Date get_end_date () const

  Get end date of trip (can only be >= beginning)
- String get_destination_name () const

  Get copy of the destination's name.
- int get_rating () const

  Get copy of the rating of this Trip.

3.7.1 Detailed Description

Contains information about a Trip from a User.

Each trip has a destination name, a time period (two Date objects), rating (1-5), a Comment object and a list of photo filenames. The comment has the username of the User who made the visit, allowing identification later on.

The comment can be empty, as well as the photo filenames list. The beginning date can match the ending date for a trip period of a single day.

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

- trip.hpp
- trip.cpp

3.8 User Class Reference

The user class, which stores its username, email, password hash(!), trips and friends.

```
#include <user.hpp>
```

Public Member Functions

- User ()

  Create an all-empty user (for array purposes)
- User (String username, String email, String password_hash)

  Create new user.
- bool set_username (String username)

  Set username for user (only if valid and free)
- bool set_email (String email)

  Set email for user (only if valid)
- bool set_password (String password)

  Set password (hash) for user (only if valid)
- String get_username () const

  Get username.
- String get_email () const

  Get email.
- String get_password_hash () const

  Get password hash.
- Vector< String > get_friends_usernames () const

  Get copy of usernames of user's friends.
- Vector< Trip > get_trips () const

  Get copy of trips for user.
- bool append_to_bin (std::ofstream &of_stream) const

  Append user username, email, and password hash to binary file.
- bool is_correct_password (const char *password) const

  Return whether password matches user's.
- void add_trip (const Trip trip)

  Add trip.
- void add_friend (String new_friend_username)

  Add friend.
- bool save () const

  Save information of user to file w/ same name.
- bool save_trips (std::ofstream &of_stream) const

  Save only trips to file.
- bool save_friends (std::ofstream &of_stream) const

  Save friends to file.
- bool save_as_friend (std::ofstream &of_stream) const

  Save current user as friend in file.
- bool load ()

  Load trips and friends from user file.
- bool load_trips (std::ifstream &if_stream)

  Load only trips from file.
- bool load_friends (std::ifstream &if_stream)

  Load only friends from file.

3.8.1 Detailed Description

The user class, which stores its username, email, password hash(!), trips and friends.

Trips are loaded from the 'personal database' of the user, i.e. the file with the same name at path db/users/

Friends are not stored as user pointers, but instead as simply their usernames, in order to keep the personal database of the user lean. It is Database's responsibility to load users from files, all a user knows about his friends are their usernames.

The passwords are hashed via bcrypt and stored as such.

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

- user.hpp
- user.cpp

Index