

---

# **BikeTripsManager**

**Bartosz Krygiel**

**Jun 23, 2021**

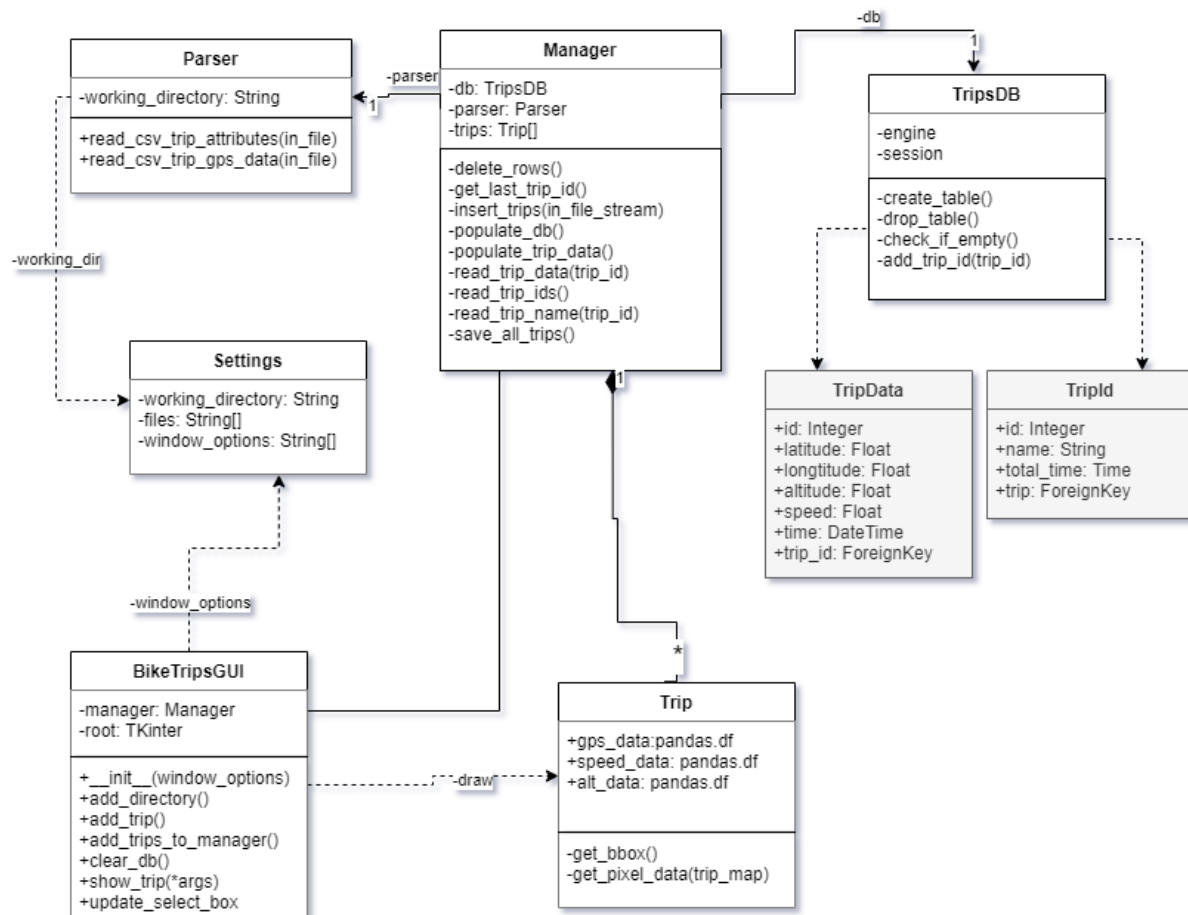
**CONTENTS:**

<b>1</b>	<b>Program Description</b>	<b>1</b>
<b>2</b>	<b>UML Diagram</b>	<b>2</b>
<b>3</b>	<b>Modules</b>	<b>3</b>
3.1	BikeTripsGui . . . . .	3
3.2	Manager . . . . .	4
3.3	TripsDB . . . . .	5
3.4	Trip . . . . .	6
3.5	CSVParser . . . . .	7
3.6	Settings . . . . .	7
3.7	App . . . . .	7
	<b>Python Module Index</b>	<b>8</b>

## **PROGRAM DESCRIPTION**

BikeTripsManager is a simple application with graphical interface to store and manage bike trips. Currently supports only csv files generated from a mobile app called Cycledroid. Users can store bike trips in a database and use the application to plot individual trip's gps data. Application uses a model-view-controller design, where model is a database, view is a graphical interface and controller is a manager

## UML DIAGRAM



### 3.1 BikeTripsGui

class gui.BikeTripsGui

Graphical interface. Acts like a View in a Model-View-Controller application. This GUI lets user choose which trip to plot, select folder with csv files, add new trip and clear db tables

Attributes:

manager: Manager object

root: Tkinter main window

Methods:

add\_directory()

Adds main directory of files and populates database based on those files

add\_trip()

Add new trips chosen from directory

add\_trips\_to\_manager()

Sets manager's trips property

clear\_db()

Command to clear tables and update select box

show\_trip(\*args)

Plots gps data of a trip using Smopy.

Route is colored based on previously selected colorbar type

**Parameters** args – contains trip's id and colorbar type

update\_select\_box()

Updates select box after adding new trips

## 3.2 Manager

class manager.Manager

Manager is an interface connecting database with gui.

It works like a controller in a typical Model-View-Controller application

Attributes:

db : TripsDB object

Instance of TripsDB. Used to connect with database

parser : CSVParser object

Instance of CSVParser. Used to parse csv files

trips : dict

Dictionary to hold all trips from database

Methods:

delete\_rows()

Clears tables

get\_last\_trip\_id()

Reads last added trip's id

**Returns** int trip id

insert\_trips(*in\_file\_stream*)

Adds new trips to database

**Parameters** in\_file\_stream (list) – list of file names

populate\_db(*filenames*)

Creates tables for database and populates using parsed data from files

**Parameters** filenames (list) – list of file names

populate\_trip\_data(*filenames*)

Parses data and adds to database

**Parameters** filenames – list of file names

read\_trip\_data(*trip\_id*)

Reads trip data from database

**Parameters** trip\_id (int) – trip id

**Returns** Trip trip\_data: Trip object created from database data

read\_trip\_ids()

Reads all trip ids from database

**Returns** pandas.Series trip\_ids: series of all trip ids

read\_trip\_name(*trip\_id*)

Reads trip name from database

**Parameters** trip\_id (int) – trip id

**Returns** str trip\_name: name of a trip

save\_all\_trips()

Sets trips property using data from database

### 3.3 TripsDB

`class database.TripsDB(db='sqlite:///trips.db')`

Interface for connecting with sqlite database. Acts like a Model in a Model-View-Controller application

Attributes:

engine: SQLAlchemy engine object Session: SQLAlchemy session object

`add_trip_id(name, total_time)`

Adds trip to database

**Parameters**

- name (str) – name of a trip
- total\_time (str) – total time of a trip

`check_if_empty()`

Checks if TripId table is empty

**Returns** bool

`create_table()`

Creates tables

`drop_table()`

Drops tables

## 3.4 Trip

`class trip.Trip(trip_data)`

Class for storing trip's data. Later used in GUI

Attributes:

`geo`: `pandas.DataFrame` stores gps data

`speed`: `pandas.Series` stores speed data

`altitude`: `pandas.Series` stores altitude data

Methods:

`get_bbox()`

Gets boundaries of a box for plotting map

**Returns** tuple of 4 gps points

`get_pixel_data(trip_map)`

Converts gps data to pixel for plotting

**Parameters** `trip_map` – gps data

**Returns** `pandas` dataframe converted from lat/lon to pixels



## 3.5 CSVParser

class csv\_parser.CSVParser

Class used for parsing csv to pandas dataframe

Attributes:

wd: str working directory

read\_csv\_trip\_attributes(*in\_file=""*)

Reads first line from file to get trip name and trip time

**Parameters** in\_file (str) – name of a csv file

**Returns**

str trip\_name: name of a trip

str trip\_total\_time: total time in H:M:S format

read\_csv\_trip\_data(*in\_file=""*)

Reads trip data from file, gps data, speed, altitude, time

**Parameters** in\_file (str) – name of a csv file

**Returns** pandas.DataFrame trip\_data: parsed data from csv file

## 3.6 Settings

class settings.Settings

class used to store working directory and file names

Attributes:

WD: str working directory

FILES: list of file names

## 3.7 App

app.app()

Runs application

## PYTHON MODULE INDEX

### a

app, [7](#)

## INDEX

\spxentryadd\_directory()\spxextragui.BikeTripsGui  
method, 3

\spxentryadd\_trip()\spxextragui.BikeTripsGui method, 3

\spxentryadd\_trip\_id()\spxextradatabase.TripsDB  
method, 5

\spxentryadd\_trips\_to\_manager()\spxextragui.BikeTripsGui  
method, 3

\spxentryapp  
  \spxentrymodule, 7

\spxentryapp()\spxextrain module app, 7

\spxentryBikeTripsGui\spxextraclass in gui, 3

\spxentrycheck\_if\_empty()\spxextradatabase.TripsDB  
method, 5

\spxentryclear\_db()\spxextragui.BikeTripsGui method, 3

\spxentrycreate\_table()\spxextradatabase.TripsDB  
method, 5

\spxentryCSVParser\spxextraclass in csv\_parser, 7

\spxentrydelete\_rows()\spxextramanager.Manager  
method, 4

\spxentrydrop\_table()\spxextradatabase.TripsDB method,  
5

\spxentryget\_bbox()\spxextratrip.Trip method, 6

\spxentryget\_last\_trip\_id()\spxextramanager.Manager  
method, 4

\spxentryget\_pixel\_data()\spxextratrip.Trip method, 6

\spxentryinsert\_trips()\spxextramanager.Manager  
method, 4

\spxentryManager\spxextraclass in manager, 4

\spxentrymodule  
  \spxentryapp, 7

\spxentrypopulate\_db()\spxextramanager.Manager  
method, 4

\spxentrypopulate\_trip\_data()\spxextramanager.Manager  
method, 4

\spxentryread\_csv\_trip\_attributes()\spxextracsv\_parser.CSVParser  
method, 7

\spxentryread\_csv\_trip\_data()\spxextracsv\_parser.CSVParser  
method, 7

\spxentryread\_trip\_data()\spxextramanager.Manager  
method, 4

\spxentryread\_trip\_ids()\spxextramanager.Manager  
method, 4

\spxentryread\_trip\_name()\spxextramanager.Manager  
method, 4

\spxentrysave\_all\_trips()\spxextramanager.Manager  
method, 4

\spxentrySettings\spxextraclass in settings, 7

\spxentryshow\_trip()\spxextragui.BikeTripsGui method,  
3

\spxentryTrip\spxextraclass in trip, 6

\spxentryTripsDB\spxextraclass in database, 5

\spxentryupdate\_select\_box()\spxextragui.BikeTripsGui  
method, 3