My Project

Generated by Doxygen 1.8.17

Chapter 1

NBA-Basketball

CS 1D Project 2 NBA basketball

1.1 Requirments

These are a series of modules/tools required for the project to properly run.

- C++ 17
- CMake
- SQL
- SQLite
- Qt 6

1.1.1 Recommendation

The following platforms are suggestions to facilitate working with program and coding in general. These are not required for the project to build/compile, but to simply improve visualization of data structure, tracking work flow, or/and help with systematic coding.

- Clion
- DataGrip
- Git Kraken
- Qt 6

Clion is the recommended IDE for coding any c++ related projects. Since CMake is used as the c++ packet manager, CLion provides different tools that can be used to improve coding along with Cmake CLI. DataGrip is used for SQL codes. The IDE enables programmers to concurrently use multiple different data management systems, such as Oracle RDBMS, MySQL, PostgreSQL, and etc. in a templated project. Git Karaken enables programers to visualize git tracking, commits, and branches.

1.2 Setup

```
git clone https://github.com/PlumLariat/Project-2-CS1D.git
cd Project-2-CS1D
git checkout main
```

1.3 Usage

Your agile team is to write an application that stores the given National Basketball Association (NBA) information into an ordered or unordered map (not using the map or unordered map STL or the QT map). The underlying data structure of the map is the choice of the team. The map should store either the arenas and/or the souvenirs. The application will allow basketball fans to plan their dream vacation.

Chapter 2

Hierarchical Index

2.1 Class Hierarchy

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

Chapter 3

Class Index

3.1 Class List

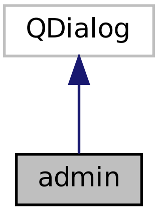
Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 4

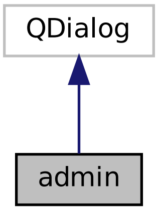
Class Documentation

4.1 admin Class Reference

Inheritance diagram for admin:



Collaboration diagram for admin:



Public Member Functions

- **admin** (QWidget ∗parent=nullptr)

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

- Admin/admin.h
- Admin/admin.cpp

4.2 City Struct Reference

Public Attributes

- std::string **team**
- bool **isVisited** = false

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

- manage_teams/manage_teams.h

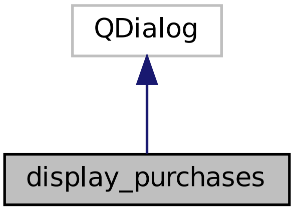
4.3 db Class Reference

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

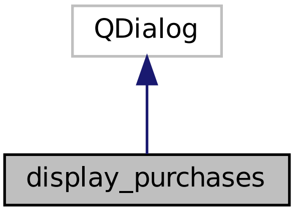
- db.h
- db.cpp

4.4 display_purchases Class Reference

Inheritance diagram for display_purchases:



Collaboration diagram for display_purchases:



Public Member Functions

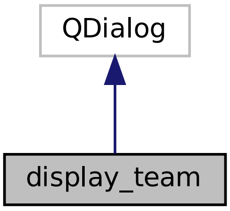
- **display_purchases** (QWidget ∗parent=nullptr)

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

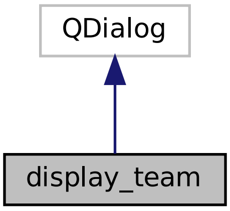
- UI/display_purchases/display_purchases.h
- UI/display_purchases/display_purchases.cpp

4.5 display_team Class Reference

Inheritance diagram for display_team:



Collaboration diagram for display_team:



Public Member Functions

- **display_team** (QWidget ∗parent=nullptr)

Public Attributes

- QSqlDatabase **db**

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

- UI/display_team/display_team.h
- UI/display_team/display_team.cpp

4.6 Distance Struct Reference

Public Attributes

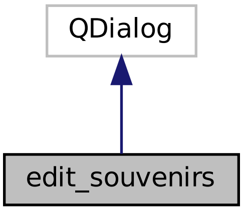
- double **distance**
- bool **discovery** = false

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

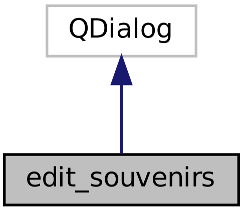
- manage_teams/manage_teams.h

4.7 edit_souvenirs Class Reference

Inheritance diagram for edit_souvenirs:



Collaboration diagram for edit_souvenirs:



Public Member Functions

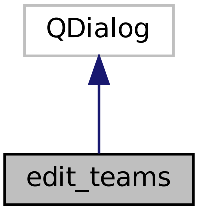
- **edit_souvenirs** (QWidget ∗parent=nullptr)

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

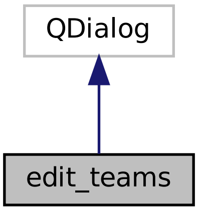
- UI/edit_souvenirs/edit_souvenirs.h
- UI/edit_souvenirs/edit_souvenirs.cpp

4.8 edit_teams Class Reference

Inheritance diagram for edit_teams:



Collaboration diagram for edit_teams:



Public Member Functions

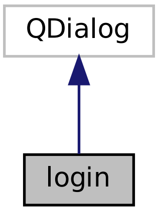
- **edit_teams** (QWidget ∗parent=nullptr)

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

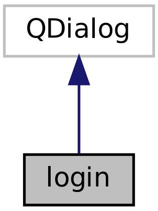
- UI/edit_teams/edit_teams.h
- UI/edit_teams/edit_teams.cpp

4.9 login Class Reference

Inheritance diagram for login:



Collaboration diagram for login:



Public Member Functions

- void **connClose** ()
- bool **connOpen** ()
- **login** (QWidget ∗parent=nullptr)

Public Attributes

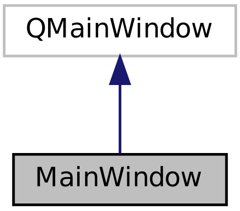
- QSqlDatabase **informationDb**
- QSqlDatabase **distancesDb**

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

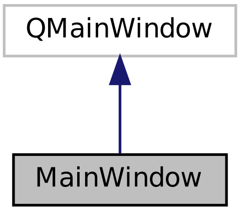
- UI/login/login.h
- UI/login/login.cpp

4.10 MainWindow Class Reference

Inheritance diagram for MainWindow:



Collaboration diagram for MainWindow:



Public Member Functions

- **MainWindow** (QWidget ∗parent=nullptr)

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

- UI/mainwindow/mainwindow.h
- UI/mainwindow/mainwindow.cpp

4.11 manage_teams Class Reference

Public Types

- enum **Teams** {
  Atlanta_Hawks, **Boston_Celtics**, **Brooklyn_Nets**, **Charlotte_Hornets**,

Chicago_Bulls, **Cleveland_Cavaliers**, **Dallas_Mavericks**, **Denver_Nuggets**, **Detroit_Pistons**, **Golden_State_Warriors**, **Houston_Rockets**, **Indiana_Pacers**, **Los_Angeles_Clippers**, **Los_Angeles_Lakers**, **Memphis_Grizzlies**, **Miami_Heat**, **Milwaukee_Bucks**, **Minnesota_Timberwolves**, **New_Orleans_Pelicans**, **New_York_Knicks**, **Oklahoma_City_Thunder**, **Orlando_Magic**, **Philadelphia_76ers**, **Phoenix_Suns**, **Portland_Trail_Blazers**, **Sacramento_Kings**, **San_Antonio_Spurs**, **Toronto_Raptors**, **Utah_Jazz**, **Washington_Wizards**, **endofTeams** }

Public Member Functions

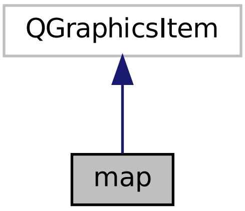
- void **initialize** ()
- std::vector< std::string > **getTeamNames** ()
- double **shortest_path** (int initial, int target)
- double **minimum_spanning_tree** ()
- double **depth_first_search** (Teams team)
- double **breadth_first_search** (Teams team)
- std::deque< City > **getTravelPlan** ()
- double **createCustomPlan** (const std::vector< std::string > &, bool order)
- void **addNewVertex** ()

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

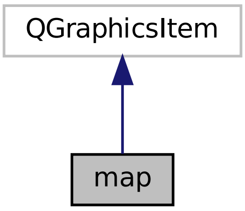
- manage_teams/manage_teams.h
- manage_teams/manage_teams.cpp

4.12 map Class Reference

Inheritance diagram for map:



Collaboration diagram for map:



Public Member Functions

- **map** (const std::string &name, int x, int y)
- QRectF **boundingRect** () const
- void **paint** (QPainter ∗painter, const QStyleOptionGraphicsItem ∗option, QWidget ∗widget) override
- std::string **GetLocation** () const
- void **setPressed** (bool isPressed)
- void **setAvailability** (bool available)
- void **setIsSelected** (bool isDotSelected)

Static Public Member Functions

- static std::string **getStartingCity** ()

Static Public Attributes

- static std::vector< map ∗ > **euroMap**

Protected Member Functions

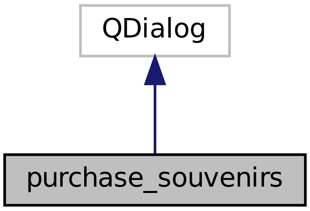
- virtual void **mousePressEvent** (QGraphicsSceneMouseEvent ∗event) override

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

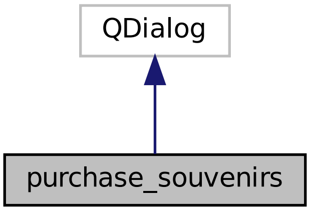
- map/map.h
- map/map.cpp

4.13 purchase_souvenirs Class Reference

Inheritance diagram for purchase_souvenirs:



Collaboration diagram for purchase_souvenirs:



Public Member Functions

- **purchase_souvenirs** (const std::deque< City > &myTeam, QWidget ∗parent=nullptr)

Public Attributes

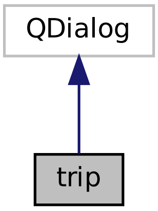
- double **grandTotal**
- double **localTotal**
- int **localNumSouv**

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

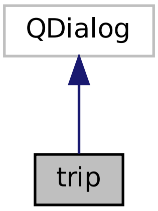
- UI/purchase_souvenirs/purchase_souvenirs.h
- UI/purchase_souvenirs/purchase_souvenirs.cpp

4.14 trip Class Reference

Inheritance diagram for trip:



Collaboration diagram for trip:



Public Member Functions

- **trip** (QWidget ∗parent=nullptr)

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

- UI/trip/trip.h
- UI/trip/trip.cpp

Index