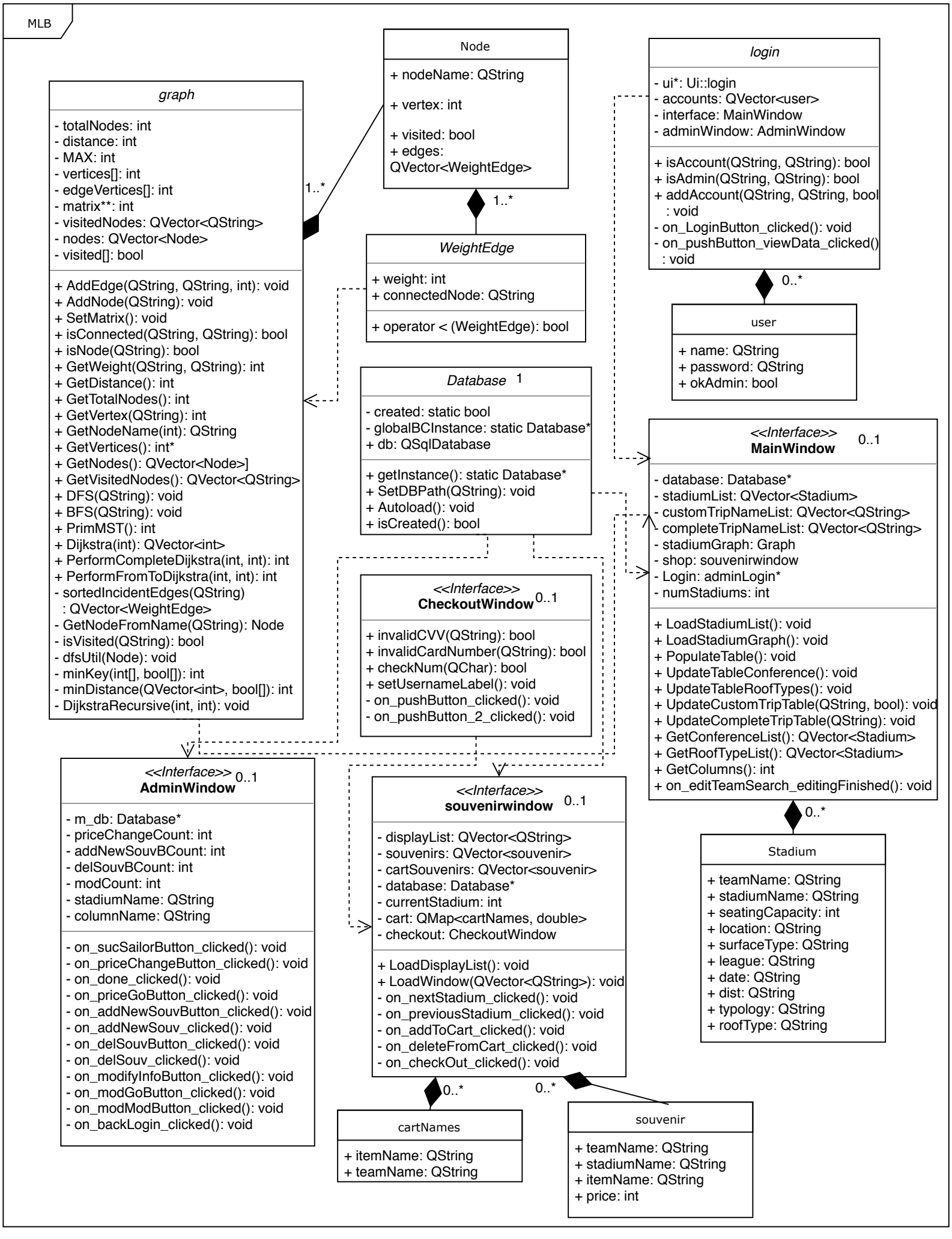
MLB
Node
+ nodeName: QString
+ vertex: int
+ visited: bool
+ edges: QVector<WeightEdge>
1..*
1..*
graph
- totalNodes: int
- distance: int
- MAX: int
- vertices[]: int
- edgeVertices[]: int
- matrix**: int
- visitedNodes: QVector<QString>
- nodes: QVector<Node>
- visited[]: bool
+ AddEdge(QString, QString, int): void
+ AddNode(QString): void
+ SetMatrix(): void
+ isConnected(QString, QString): bool
+ isNode(QString): bool
+ GetWeight(QString, QString): int
+ GetDistance(): int
+ GetTotalNodes(): int
+ GetVertex(QString): int
+ GetNodeName(int): QString
+ GetVertices(): int*
+ GetNodes(): QVector<Node>]
+ GetVisitedNodes(): QVector<QString>
+ DFS(QString): void
+ BFS(QString): void
+ PrimMST(): int
+ Dijkstra(int): QVector<int>
+ PerformCompleteDijkstra(int, int): int
+ PerformFromToDijkstra(int, int): int
- sortedIncidentEdges(QString) : QVector<WeightEdge>
- GetNodeFromName(QString): Node
- isVisited(QString): bool
- dfsUtil(Node): void
- minKey(int[], bool[]): int
- minDistance(QVector<int>, bool[]): int
- DijkstraRecursive(int, int): void
WeightEdge
+ weight: int
+ connectedNode: QString
+ operator < (WeightEdge): bool
login
- ui*: Ui::login
- accounts: QVector<user>
- interface: MainWindow
- adminWindow: AdminWindow
+ isAccount(QString, QString): bool
+ isAdmin(QString, QString): bool
+ addAccount(QString, QString, bool) : void
- on_LoginButton_clicked(): void
- on_pushButton_viewData_clicked() : void
0..*
user
+ name: QString
+ password: QString
+ okAdmin: bool
Database
1
- created: static bool
- globalBCInstance: static Database*
+ db: QSqlDatabase
+ getInstance(): static Database*
+ SetDBPath(QString): void
+ Autoload(): void
+ isCreated(): bool
<<Interface>>
MainWindow
0..1
- database: Database*
- stadiumList: QVector<Stadium>
- customTripNameList: QVector<QString>
- completeTripNameList: QVector<QString>
- stadiumGraph: Graph
- shop: souvenirwindow
- Login: adminLogin*
- numStadiums: int
+ LoadStadiumList(): void
+ LoadStadiumGraph(): void
+ PopulateTable(): void
+ UpdateTableConference(): void
+ UpdateTableRoofTypes(): void
+ UpdateCustomTripTable(QString, bool): void
+ UpdateCompleteTripTable(QString): void
+ GetConferenceList(): QVector<Stadium>
+ GetRoofTypeList(): QVector<Stadium>
+ GetColumns(): int
+ on_editTeamSearch_editingFinished(): void
<<Interface>>
CheckoutWindow
0..1
+ invalidCVV(QString): bool
+ invalidCardNumber(QString): bool
+ checkNum(QChar): bool
+ setUsernameLabel(): void
- on_pushButton_clicked(): void
- on_pushButton_2_clicked(): void
<<Interface>>
AdminWindow
0..1
- m_db: Database*
- priceChangeCount: int
- addNewSouvBCount: int
- delSouvBCount: int
- modCount: int
- stadiumName: QString
- columnName: QString
- on_sucSailorButton_clicked(): void
- on_priceChangeButton_clicked(): void
- on_done_clicked(): void
- on_priceGoButton_clicked(): void
- on_addNewSouvButton_clicked(): void
- on_addNewSouv_clicked(): void
- on_delSouvButton_clicked(): void
- on_delSouv_clicked(): void
- on_modifyInfoButton_clicked(): void
- on_modGoButton_clicked(): void
- on_modModButton_clicked(): void
- on_backLogin_clicked(): void
<<Interface>>
souvenirwindow
0..1
- displayList: QVector<QString>
- souvenirs: QVector<souvenir>
- cartSouvenirs: QVector<souvenir>
- database: Database*
- currentStadium: int
- cart: QMap<cartNames, double>
- checkout: CheckoutWindow
+ LoadDisplayList(): void
+ LoadWindow(QVector<QString>): void
- on_nextStadium_clicked(): void
- on_previousStadium_clicked(): void
- on_addToCart_clicked(): void
- on_deleteFromCart_clicked(): void
- on_checkOut_clicked(): void
0..*
Stadium
+ teamName: QString
+ stadiumName: QString
+ seatingCapacity: int
+ location: QString
+ surfaceType: QString
+ league: QString
+ date: QString
+ dist: QString
+ typology: QString
+ roofType: QString
0..*
0..*
cartNames
+ itemName: QString
+ teamName: QString
0..*
souvenir
+ teamName: QString
+ stadiumName: QString
+ itemName: QString
+ price: int