**ДОКУМЕНТАЦИЯ НА ПРОЕКТ**

**E-GradeBook**

**АВТОРИ:**

**Стефан Чолаков**

**Ивайло Тотев**

**Калоян Михайлов**

**Ангел Христов**

**МОДУЛ VII „РАЗРАБОТКА НА СОФТУЕР“ (РС)**

**ЗА 11 КЛАС**

**ПО НП „ОБУЧЕНИЕ ЗА ИТ КАРИЕРА“, ЦЕНТЪР ПЛОВДИВ**

**Място на провеждане на обучението:**

**ППМГ „Гео Милев“, гр. Стара Загора**

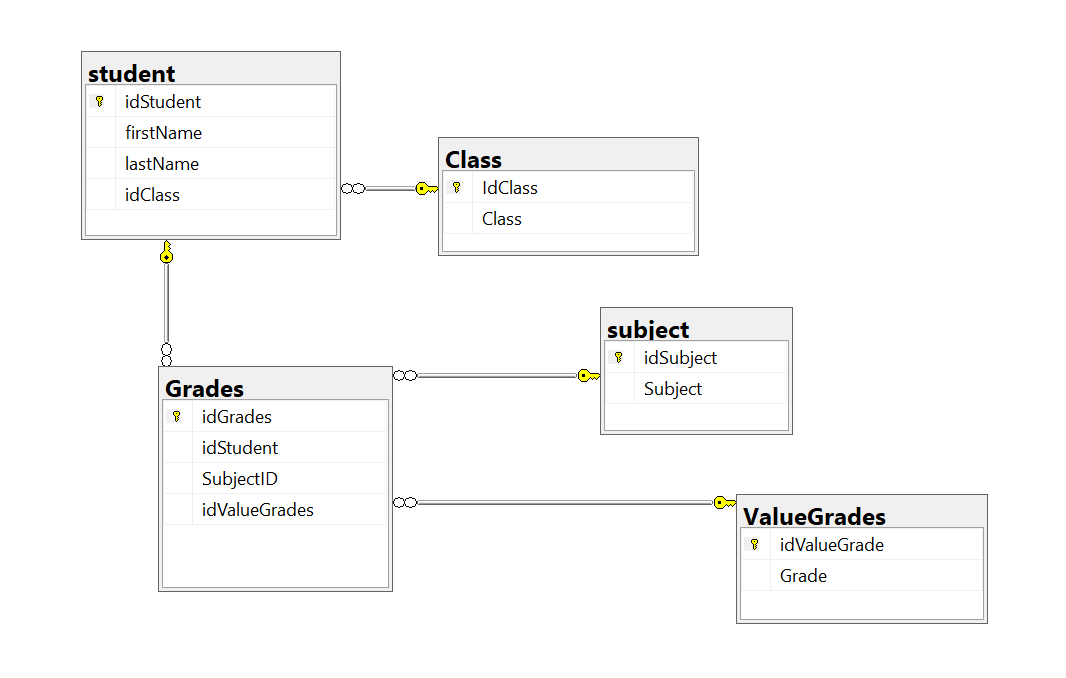
**РЕЗЮМЕ НА ПРОЕКТА**

Целта ни е да създадем прототип на електронен учителски бележник, с конзолен интерфейс, чрез който да се осъществява добавяне на ученици, оценки по предмети и достъп до тях.

Създали сме локална база данни E-GradeBook, като за целта използвахме приложение SQL Server Management Studio, а за кодовата част Visual Studio. Нашата база се състои от следните таблици:

* Class – съдържа полета: Id на класа и класа;
* Student – съдържа поле Id на ученика, имена и клас;
* Subject – таблица с Id и учебни предмети;
* ValueGrades – таблица с Id и оценки;
* Grades – таблица съдържаща Id на записа, Id на ученика, Id на предмета и Id на оценката.

Таблиците са свързани в релации по следния начин:



Създадената база използваме в реализацията на CRUD приложение без ORM в проект на Visual Studio, с език за програмиране C#.

Структурата на проекта е следната:

* Съдържа следната трислойна архитектура:
  + Слой за данни;
  + Бизнес слой;
  + Презентационен слой;

Посочената архитектура е реализирана чрез класове, поместени в следните папки:

* Папка **Common** – реализира общ (споделен) слой, който работи с всички останали. Тя съдържа следните класове:
  + **Grades.cs** - клас, който визуализира в проекта таблица Grades от базата данни;
  + **GradesQ.cs** – клас, който заменя идентификационните номера със значещите им съответстващи данни;
  + **Student.cs** - клас, който визуализира в проекта таблица student от базата данни;
  + **Subject.cs** – клас, който визуализира в проекта таблица subject от базата данни;
* Папка **Data** – реализира слоя за данни и включва следните класове:
  + **Database.cs** – статичен клас за връзка с базата данни;
  + **GradesData.cs** – клас, в който изпълняваме заявки към БД свързани с таблица Grades;
  + **StudentData.cs** – клас, в който ще изпълняваме заявки към БД свързани с таблица student;
* Папка **Education** – реализира бизнес слой, съдържащ следните класове:
  + **GradesManageent.cs** – описва бизнес логиката свързана с клас Grades;
  + **StudentEducation.cs** – описва бизнес логиката свързана с клас Student;
* Папка Pressentation – реализира презентационен слой и включва следните класове:
  + **DisplayStudents** – клас, в който реализираме конзолно управление на учениците (добавяне, актуализиране, изтриване, достъп до ученик(ци));
  + **DisplayGrades.cs** – клас, в който реализираме конзолно управление на оценките на учениците по учебни предмети;
  + **Display.cs** – клас, в който съчетаваме конзолните управления на учениците и оценките им (съчетаваме класовете DisplayStudents и DisplayGrades);
* **Program.cs** – Място, в което създаваме интерфейс на приложението, чрез обект от класа Display.

Създадохме приложение с лесен конзолен интерфейс, чрез което да могат да се реализират основните функции: добавяне на ученици, учебни предмети и оценки по предметите, като с това се опитахме да му придадем завършеност и функционалност.