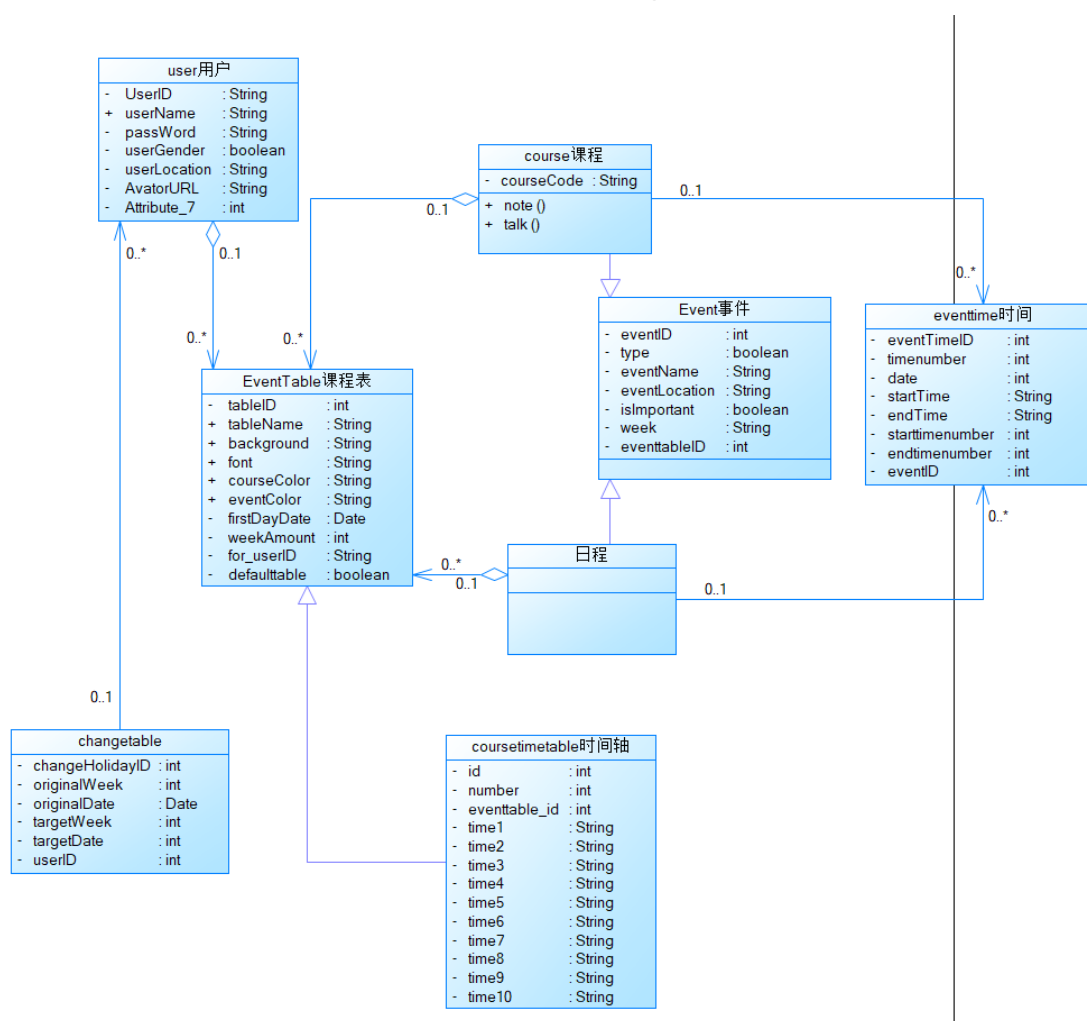


## NSD设计模式

- **template method模式**

- 首先我们小组采用了 template method 模式，我们首先创建了一个 event 类，就是事件类，它是所有课程类和 schedule 日程类的父类。其中包含其通用信息（时间，地点，是否重要等等）。
- 实现一个event模板类,在其中编码了事件基本的策略、规则、流程。在针对具体问题进行精化时，我们用课程类和日程类继承event模板类，来扩展这个Template中的一些细节，比如课程编码，访问笔记和论坛。
- 之后我们创建了 user 用户类，以及 EventTable 类来完善该模式的设计，一个用户对应多张课表，所以每张课表中有一个 for\_userID 来对应用户。
- 为了调休的实现方便，我们专门加入了一个类 changetable 来为用户提供调休的方式



- **迭代器模式**

我们采用了迭代器模式，具体表现在如下所示，

- 在软件构建过程中，class类结构复杂，对于这些聚集对象，我们希望在暴露其内部结构的同时，可以让外部客户代码透明地访问其中包含的元素。故采用了iterater模式，便利他所有的数据。
- 利用迭代器模式很好的实现课程的找寻，在用户查找课表中的课程时，迭代器会快速搜索到course

- 我们设计将数据结构和遍历算法分离，使得它们可以独立变化，在之后优化时可以更好的改变
- 我们小组 NSD APP 可以隐藏数据结构背后的复杂性，将课程等数据隐藏于背后，使得后端同学对数据集的遍历变得简单而统一，实现了分布式开发。

