面对一个大项目，我们首先根据该项目的不同功能分模块，一个模块一个模块的进行分析。要想完成每个模块的实现，我们组首先对各个模块的数据结构进行了抽象。

1. 首先是课程信息管理和查询模块，该模块包括增删改查等功能。对于课程活动类操作，我们考虑用event类和file类这两个核心组件组成classes类和其他一些辅助数据结构来描述课程事件。

* Event类即事件类，用来描述课程或课外信息。其基本属性包括当日事件（包括起始时间和结束时间）、名称和地点等等。
* File类即文件类，用来应对文件的操作。其基本属性包括存储路径、发布时间和发布人等。
* Classes类即课程类，其包含5个基本属性：

1. 上课

* 上课日期（包括周数和星期数之类）
* Event：用来存储课程信息

1. 进度

* 通过上课日期和系统时间来计算

1. 作业

* Task类：学生上传作业

1. 考试（和课程活动类似）

* 考试日期
* Event

1. 资料

* File

实现学生上传和更新课程资料

* 辅助数据属性
  1. Data:日期类
* 年
* 月
* 日
  1. Time时间类（为event服务）
* 起始时间
* 结束时间
  1. Time-line：时间轴
* 时间点数组（event事件名称）
* Clock数组（提醒）
  1. Clock：闹钟类
* Data类
* 时间点（0-24）
  1. Task作业类
* 作业名称
* 提交作业状态数组（0/1来区分）
* 作业集存储路径

1. 课外信息管理和查询模块，该末了也包含增删查改等功能。要实现该模块的操作，其实和课程模块的内容差不多，我们依旧选择用event类和file类这两个核心组件来进行数据抽象，再加上activity类和一些辅助数据结构来实现。

* Event类和file类不再赘述
* Activity类即活动类，其继承了event类还包含了一些额外的属性，包括：

1. 发起人、负责人
2. 个人/集体（若为集体就录入成员名单）
3. 闹钟的开关（函数实现）

* 辅助数据类型和课程相似

这两个模块要考虑到首先是操作者时学生还是管理员的一个问题，然后也融入了关于模拟系统时间的一些思考，时间管理和查询势必涉及到时间，二者不可分开，关于clock的一些数据结构抽象我们就是暂时想到的方法。

1. 课程导航模块

首先要先制作地图，将校园沙河或者本部进行抽象。我们考虑到将建筑出入口，道路分叉，拐弯的地方抽象为节点vertice,将道路抽象为边edge。

* Vertice：将数据与实际地图对应，可对不同节点进行编号，将文字地点转化为数字编号。其基本属性包括：

1. 校区（沙河/本部）
2. 横纵坐标
3. 名称

* Edge：赋予道路相应的属性。

1. 自行车道/人行道
2. 长度（直接得到两点之间距离，无需用点的左边计算
3. 拥挤度
4. 名称

* 图的实现方式：哈希链表(因为已经存储了节点的坐标，所以无需存储路径长度)。

1）头部:当前节点的编号

2）尾部:直接相连接的节点编号、对应路径的属性

* 信息导入来源：

手动导入信息入txt文件，每次开启程序进行导入。

* 辅助数据结构object：地图上的建筑

1. 楼层数
2. 出入口编号数组

导航模块除此之外还有一些其他需要考虑的方面，包括班车的车次信息，交通工具的选用，这些都用来计算导航的最短策略等。