# GitMining分层模型概略设计文档V0.1

By 徐江河 2016/3/1

1. 系统抽象层次及内部模块职责协作说明（由于作图网站暂时无法服务，故没插图）

GitMining系统从抽象层次上分为四层：

* 1. Presentation展示层
     1. 内部模块：单一模块。
     2. 职责：构建用户界面的窗体、按钮、图表等以展示数据；接受用户界面的搜索、详细查看、更新数据等输入请求，从Logic获取相关统计数据、图表对象与计算结果。
     3. 协作者：Logic层各模块。
  2. Logic逻辑层
     1. 内部模块：logic.data数据整理模块，logic.chart图表生成模块，logic.calc数据计算模块。
     2. 职责：
        1. logic.data：从Data层获取Repository、User等信息并整理，请求Data层更新数据；
        2. logic.chart：利用logic.data的数据接口与（第三方）图表生成类制作图表对象；
        3. logic.calc：利用logic.data的数据接口与数学计算公式计算所需统计数据。
     3. 协作者：
        1. logic.data：Data层的data.manage模块；
        2. logic.chart：logic.data模块；
        3. logic.calc：logic.data模块。
  3. Data数据层
     1. 内部模块：data.manage数据管理模块，data.storage数据存储模块
     2. 职责：
        1. data.manage：向Network层请求新数据，与data.storage交互以存储或读取Git相关数据，对数据进行管理、筛选；
        2. data.storage：对来自data.manage模块的Git相关数据进行格式化与解格式并进行存储、读取。
     3. 协作者：
        1. data.manage：Network层的network.data模块，data.storage模块；
        2. data.storage：data.manage模块
  4. Network网络层
     1. 内部模块：network.api API管理模块，network.connection 链接模块，network.data数据处理模块
     2. 职责：
        1. network.api：保存信息更新所需的API数据；
        2. network.connection：利用network.api提供的API与服务器链接取得数据；
        3. network.data：将从network.connection获取的数据进行处理，格式化为系统使用的Java对象。
     3. 协作者：
        1. network.api：无；
        2. network.connection：network.api模块；
        3. network.data：network.connection模块。

1. 各层次主要接口说明
   1. Logic层：
      1. （数据整理模块）按序获取信息，e.g. getRepository (page: int, numPerPage: int): List<Repository>, getUser (page: int, numPerPage: int): List<User>, etc..
      2. （数据整理模块）搜索信息，e.g. searchRepository (keyword: String[], page: int, numPerPage: int): List<Repository>, searchUser (keyword: String[], page: int, numPerPage: int): List<User>, etc..
      3. （图表生成模块、数据计算模块）获取统计对象，e.g. getXXChart (columns: ColEnumType[]):　XXChart, getTotalNumberOfRepository(): int, etc..
      4. （数据整理模块）部分与统计无关的功能，如请求更新系统数据 refreshData (): void，获取当前版本数据版本获取时间 getDataStatus (): DataStatus, etc..
   2. Data层：
      1. （数据管理模块对上层提供）同Logic层的i, ii, iv；
      2. （数据存储模块）存储对象，e.g. saveRepo (p: Pipeline): void, saveUser (p: Pipeline): void, etc.. ；读取对象，e.g. getNameList(): String[], loadRepo (): Repository, etc..
   3. Network层：
      1. （数据处理模块对上层提供）获取Repository或User信息，e.g. getRepositories (): Pipeline, getUsers (): Pipeline, etc.. （相比链接模块，Pipeline内包含的是系统定义的Java对象而不是JSON）
      2. （链接模块）获取Repository或User信息，e.g. getRepositories (): Pipeline, getUsers (): Pipeline, etc.. （Pipeline中传递的数据为JSON）
      3. （API管理模块）获取各种API地址信息，e.g. getNameListURL (): String, makeRepositoryInfoURL (repoName: String, owner: String): String, etc..
2. 项目架构风险预估及解决方案
   1. （老师上课威胁的）未来网站API提供的信息内可能有噪音：

在network.data中两个Pipeline中间添加Filter进行筛选，滤除错误信息。

* 1. （迭代三会发生的）数据存储方式由自定义文件系统变为数据库：

替换data.storage模块为数据库系统实现。

* 1. 需要替换第三方图表绘制JAR包：

构建时为每种图表建立Wrapper类，为每套第三方JAR包的对象设计一套子类，利用方法委托隐藏内部实现，以方便地替换logic.chart模块实现。