DataWise

Product Requirements Document

<大闸蟹版>

|  |  |  |
| --- | --- | --- |
| Author: |  | Kimmy Chen |
| Contributors: |  | Jian Zhou; Wenyuan Cheng; Yunfeng Tao; Jun Zhang; Cong Fan |
| Document Revision: |  | September 09, 2013 |

Feisha Confidential. For Internal Use Only.

This document contains proprietary trade secrets of Feisha Inc. Receipt or possession does not convey any right to reproduce, disclose its contents, or to manufacture, use, or sell anything that it may describe. Reproduction, disclosure, or use without specific authorization of Feisha Inc. is strictly forbidden.

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Revision | Summary | Date | Author(s) |
| 1.0 | 数据日志分析需求简略描述 | 09/09/2013 | Kimmy Chen |
| 2.0 | 经过产品头脑风暴，我们会做的是big data as a service的产品，而不是某个细分领域内如日志分析的产品，对big data as a service的产品需求作出简略描述 | 09/10/2013 | Kimmy Chen |
| 3.0 | 根据游戏客户的客户访谈的反馈，更新需求细节，主要更新的部分为“数据源管理“子系统的功能和原型页面 | 09/16/2013 | Kimmy Chen |
| 4.0 | 增加对于SQL查询的需求和描述 | 09/18/2013 | Kimmy Chen |
| 5.0 | 1，修正datawise系统组成diagram  2，修正collector状态逻辑  3，修正常用查询的逻辑  4，描述展示结果页面展现  5，补充用户中心功能  6，补充数据收集器配置详细 | 09/22/2013 | Kimmy Chen |
| 6.0 | 1，修正数据收集器的操作逻辑，对已停止的只能删除不能编辑 | 09/25/2013 | Kimmy Chen |
| 7.0 | 1，产品从datawise重命名为whaleminer，规避版权风险  2，规划whaleminer二期需求，此版本增加 a) 数据来源支持文本文件，即非结构化数据导入 b)数据安全性和服务安全性要求 | 10/12/2013 | Kimmy Chen |

Contents

Introduction 4

Solution Overview 4

Customer Need 5

System Components 5

Product Roadmap 6

Core Functionality – 数据收集器 7

Core Functionality – 用户管理查询后台 8

Installation 17

Account Provisioning 17

Authorization and Authentication 17

Security 18

Performance 18

Reliability 18

User Documentation 18

Internationalization and Localization 18

Branding 18

Platform Requirements 18

Product Demonstration 19

Use Cases 19

# Introduction

大数据时代，企业面临两个挑战：

1，数据量变的庞大，已有的IT架构性能无法满足数据的存储/查询需求。对于运营的某些查询分析，因为怕影响线上系统性能或在现有架构下查询很不方便甚至不可执行，运营只得放弃不查询。另外因为现有架构如MYSQL的容量问题，企业只能定期删除数据或干脆不保存数据 。

2，Hadoop已成为大数据存储和分析解决方案的事实标准，而企业缺乏Hadoop搭建和运维的人才，因而搭建和维护一个Hadoop集群的成本对企业来说较为高昂。并且在数据分析领域，数据分析师或程序员需要掌握新的语言和技能，如mapreduce才能够对Hadoop上的数据进行挖掘和分析，这对企业应用大数据又是一个新的挑战或阻碍。

费沙信息科技将为企业提供big data as a service（BDAAS）的服务，整合用户的各种数据，并为用户提供基于Hadoop的BI分析能力，我们称之为云BI服务。费沙进一步利用云平台的resource on demand，向用户提供data processing performance on demand的特性。

# Solution Overview

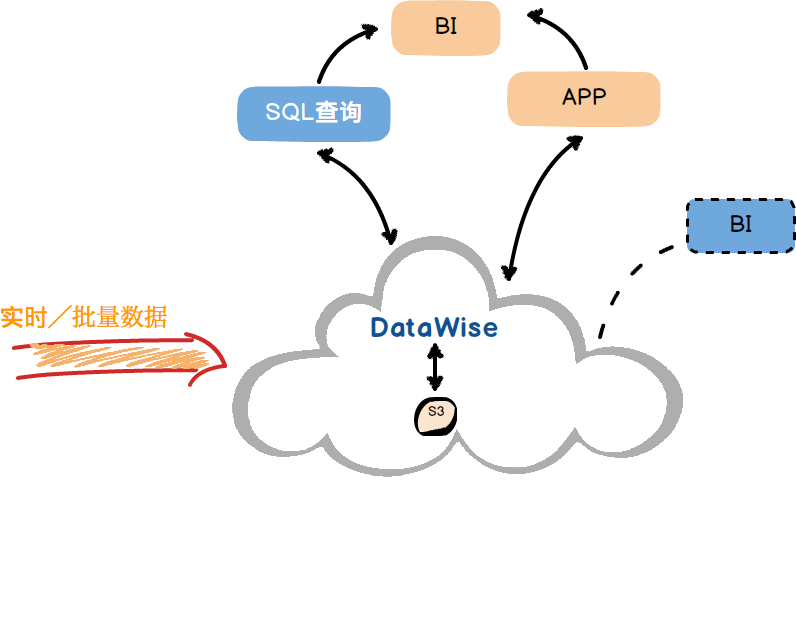


diagram 1: Solution Overview

DataWise是一个基于阿里云（目前只支持阿里云，未来将扩展到其他IAAS）的云上的big data as a service服务，对于已经在阿里云上的客户，直接提取客户的S3文件地址或数据库地址即可对获取客户的数据；而对于自建数据中心的客户，DataWise需要额外为客户提供数据实时或批量备份到云上的服务。客户数据提取或备份好之后，通过DataWise引擎的处理，在用户指定的时间内完成，用户的数据分析师或应用，通过DataWise的SQL查询界面或API接口，可实现对大数据的分析和挖掘。

在数据分析和挖掘方面，除了支持对大数据的SQL查询之外，还将提供高级算法对原始数据进行挖掘，如机器学习算法，即图中用虚线表示的部分。

## Customer Need

客户分为两种：

1，已经在使用阿里云等IAAS服务商服务的客户

2，自建数据中心的客户

对于＃2，需要额外在云端给客户提供数据存储和备份的服务，并且需要用户安装一个数据收集器用于收集用户数据。另外，对于数据的传输，数据收集器需要同时支持“推”和“拉”两种模式，“推”模式即收集器主动的将数据同步给云，“拉”模式即云主机将主动连到用户数据中心，将数据拉回。

现在市场上有一些专为特定行业服务的IAAS，比如腾讯云，专为游戏客户服务，不允许其他行业的客户入驻。在使用这些IAAS的服务的客户对于DataWise来说，相当于自建数据中心的客户，因为DataWise需要为这些客户提供数据备份服务。

## System Components

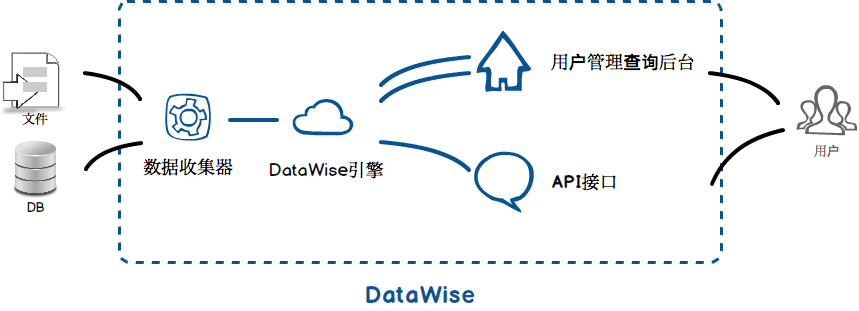


Diagram 2: 主要系统部件

主要系统部件功能

1. 数据收集器：安装在客户的服务器上，将用户需要查询的数据（如文本文件或数据库内容），按照用户指定的时间间隔进行备份，发送到云端或等待云端来取数据。数据收集器的配置是通过部署在客户服务器上的配置文件完成的。
2. DataWise引擎：这是部署在阿里云上的大数据存储分析系统，负责接收并存储数据，对数据建立索引，对外提供查询功能，等等，这是费沙Big Data As A Service服务的大脑。这里是DataWise引擎的详细设计文档：《注：*链接到文档*》。
3. 用户管理查询后台：这是客户的使用界面，客户的系统运维人员和数据分析师对数据收集器进行管理，并对已经提取完毕的数据进行SQL查询和获取查询结果的WEB控制台。另外，对费沙的系统管理员，提供一个后台界面，管理集群和系统中的用户。
4. API接口：客户的程序通过API接口调用和DataWise引擎进行交互，我们将提供如数据备份，数据查询，查询结果取回等API。

## Product Roadmap

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 时间节点 | 数据收集器 | 用户后台 | API接口 | 引擎 |
| 2013/11/30 | * 支持拉模式 * 支持从MYSQL数据库增量备份数据 * 支持用户配置的参数，如对数据scan间隔等 * 安全模式运行，即不影响用户应用性能 | * 数据收集器管理 * 数据SQL查询和结果导出 * 用户注册登陆 | 不提供 | 搭建基础架构：Hadoop上的Shark |
| 2014/01/29 | * 支持推模式 * 支持从文本文件增量备份数据 | * 数据收集器管理升级，支持推模式和文本文件的处理 * SQL查询scheduler：即支持周期性查询 * dashboard：数据可视化 * 计费系统［待定］ * 用户权限系统［待定］ | 不提供 | * 集群可动态伸缩 * 集群的监控和性能［待定］ |

第一期以满足第一个客户（杭州的游戏客户）的应用场景为主，因而先实现拉模式以及从MYSQL数据库增量备份数据的需求，后端以搭建好平台的各项基本功能和跑通主干流程为目标。

二期的规划以接进第二个客户为目标，完善数据备份的功能，如支持推模式，市场上数据的推模式是更为普遍的需求；增加支持的数据源，支持备份和导入从机器生成的文本日志。另外在前端提供给客户周期性查询的功能，使得客户可以使用我们的系统搭建data pipeline，增强客户的数据分析和挖掘能力，或者客户也可以将结果和自己的应用系统接起来，另外在前端界面上提供给客户一些基本常用的数据可视化报表。后端以提供集群动态伸缩功能为目标，为将来提供弹性性能以及计费做准备。

## Core Functionality – 数据收集器

数据收集器是部署在客户机器上，按照客户的配置，增量备份客户数据，并将数据发到云端或等待云端拉取的客户端程序。它同时需满足易部署安装，故障自动重启，和运行安全即不影响客户应用运行性能等特点。

第一期数据收集器的应用场景为客户提供MYSQL数据库的必要连接信息，如MYSQL数据库IP和用户名密码，以及对数据库表的备份需求如备份时间间隔等参数，增量备份MYSQL数据库内容到制定文件路径，等待服务器拉取数据，并将必要的meta信息如数据表类型等信息同步给云端服务器。

数据收集器的配置文件如下：

{

version:”1.0”,

    app:"appname",

    appversion:"appversion",

    database:[

        {

        dbconnstr:"",

        dbuser:"admin",

        dbpassword:"1234",

        tables:[

            {

                tablename:"t1",

                hasTimestamp:"true",

                timestampCol:"timestamp"

            },

            {

                tablename:"t2",

                hasTimestamp:"false"

            }]

        }

    ]

}

|  |  |  |  |
| --- | --- | --- | --- |
| 配置项 | 描述 | 数值类型 | 默认值 |
| version | 此配置文件的版本 | 文本，10个字符 | 无 |
| app | 定义此台机器收集的数据隶属于哪个产品或应用，如游戏客户app的名称为“魔兽世界”，如电商客户app的名称为“母婴之家”。 | 文本，25个字符 | 无 |
| appversion | 定义此台机器收集的数据隶属的产品版本，如“燃烧的远征”是“魔兽世界”的一个版本 | 文本，20个字符 | 无 |
| dbconnstr | 数据库的连接字串 | 文本 | 无 |
| dbuser | 数据库用户名 | 文本 | 无 |
| dbpassword | 数据库密码 | 文本 | 无 |
| tablename | 需要备份的目标数据库表名称 | 文本 | 无 |
| hasTimestamp | 此张表格是否含有一列用来做增量同步的依据，如时间戳，或自增ID | 布尔型 | true |
| timestampCol | 如果hasTimestamp为true，则需要指定该列的列名，否则此项可不需指定。 | 文本 | 无 |

* 一个配置文件下，可配置多个数据库，一个数据库下面可配置多个数据库表
* 配置文件必须为utf-8编码

## Core Functionality – 用户管理查询后台

用户通过此WEB界面来管理数据收集器，并对收集的数据进行SQL查询，获得查询结果。 同时提供一个供费沙的系统管理员登陆的后台，对集群的运行情况和客户进行管理。

第一期的主要应用场景为，客户（一般为客户的运维人员）登陆WEB，手工添加数据收集器并配置好部署数据收集器的远程服务器访问地址，然后客户将数据收集器启动起来。或者用户先启动数据收集器，然后完成在WEB端的手工添加操作，这两个步骤先后次序无关紧要。云主机定期（一般为30分钟）轮询系统中已经添加的数据收集器，按照数据收集器传回来的meta信息，将数据收集备份到云端。

客户（一般为客户的数据分析师）登陆系统，对已经备份好的数据进行SQL查询。他可以查看现在系统中的数据表和数据，提交新的查询并将查询结果导出为CSV文件，可以查看并下载历史查询和查询结果，或保存查询为常用查询。

### 管理后台的主要组成部分：



说明：

1，数据收集器管理：配置和管理数据收集器

2，数据分析：SQL查询和结果展现，历史查询和数据表展现

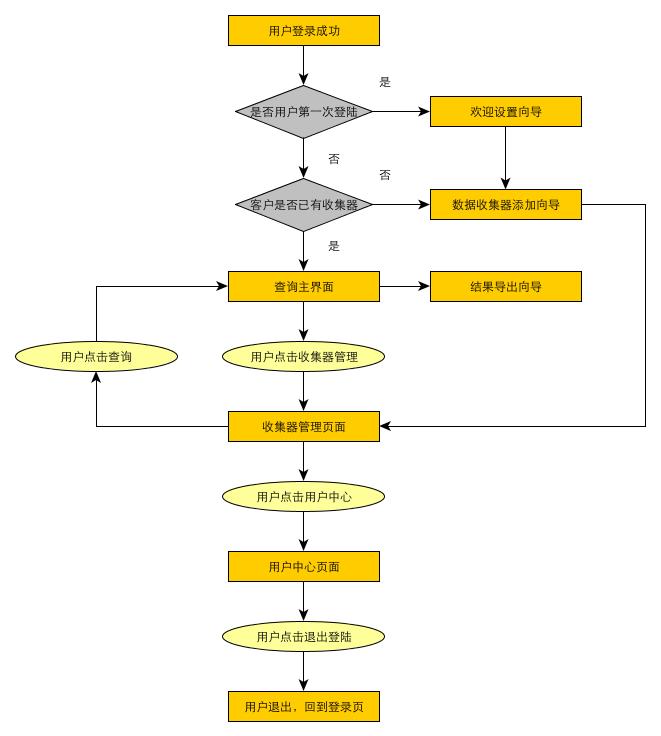
3，用户管理

4，计费系统<暂不做>

5，系统管理员后台：费沙内部界面，用来管理和监控阿里云机器，及其他系统管理员的工具和功能

其中1，2，3，4为客户的界面，5为内部界面。

### 主页面跳转逻辑：

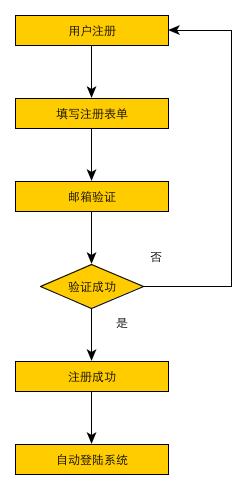


补充说明：

1. 上图不包含
   1. 网站首页／介绍页等，即未登录状态下访问的页面
   2. 用户注册流程所涉及的页面
   3. 出错页
2. 这个是主干跳转逻辑，细节在此节中详细展开

### 用户流程

1，用户注册／登陆流程



用户注册需要提供：

1. 用户邮箱
2. 用户昵称：文本，25个字符
3. 密码：英文和字母，最少6个字符
4. 公司名称：文本，50个字符
5. 用户QQ，数字，20个字符
6. 用户手机或电话，数字，手机就是11位，电话，区号＋号码

邮箱未验证通过之前，用户不能登陆系统。

激活邮件发件人：DataWise机器人 <no-reply@datawise.me>

标题：请激活您的邮箱以完成注册

正文内容：

您好！您刚刚注册了DataWise，请点击以下的链接完成注册。

http://…

（如果上述链接无法点击，请将地址拷贝粘贴至浏览器地址栏，然后按回车键访问)

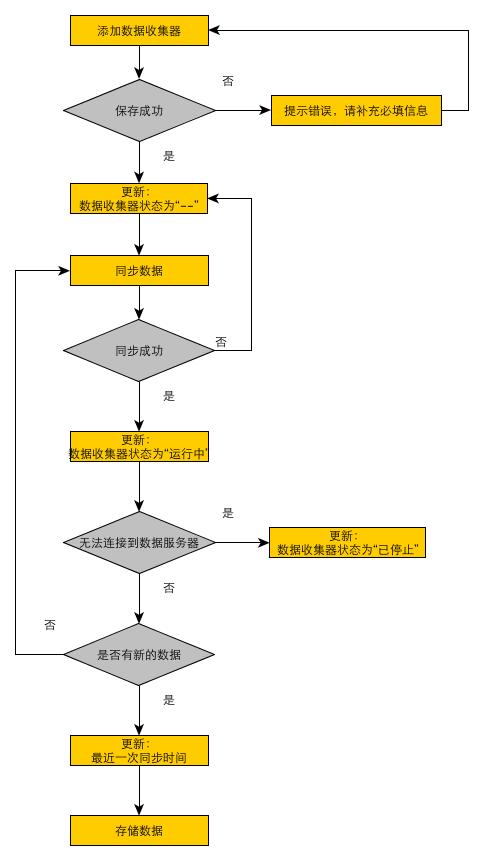
用户登录：

1. 用户使用邮箱和密码登陆系统，连续输错5次及以上密码，需要额外输入验证码
2. 用户可通过邮箱找回密码，即用户点“忘记密码”，将发送密码重置链接至用户邮箱，用户通过此链接重置密码。重置成功后，跳转到登录页。
3. 错误信息：
   1. 用户名不存在
   2. 密码错误
   3. 用户邮箱尚未激活

2，用户添加收集器和管理流程

用户在自己机器上装配好数据采集器之后，对于拉模式的客户，需要手工在云端添加这些收集器。对于已经添加的数据收集器，用户可以查看数据收集器状态。

添加和管理数据收集器的主干流程如下图：



* 添加数据收集器需要填写的内容：
  + 数据收集器名称：文本，最长50个字符，不可重复。用户可能会一次性添加多个（3个以上）数据收集器，数据收集器的名称可以默认帮用户填写为“收集器1，收集器2，。。。收集器n”。
  + 远程访问路径：文本，最长255个字符。
* 数据收集器的同步：
  + 用户添加完数据收集器之后，服务器应立即尝试连接数据收集器并开始同步数据。
  + 数据收集器两次同步的时间间隔最长不超过30分钟。
* 管理数据收集器：

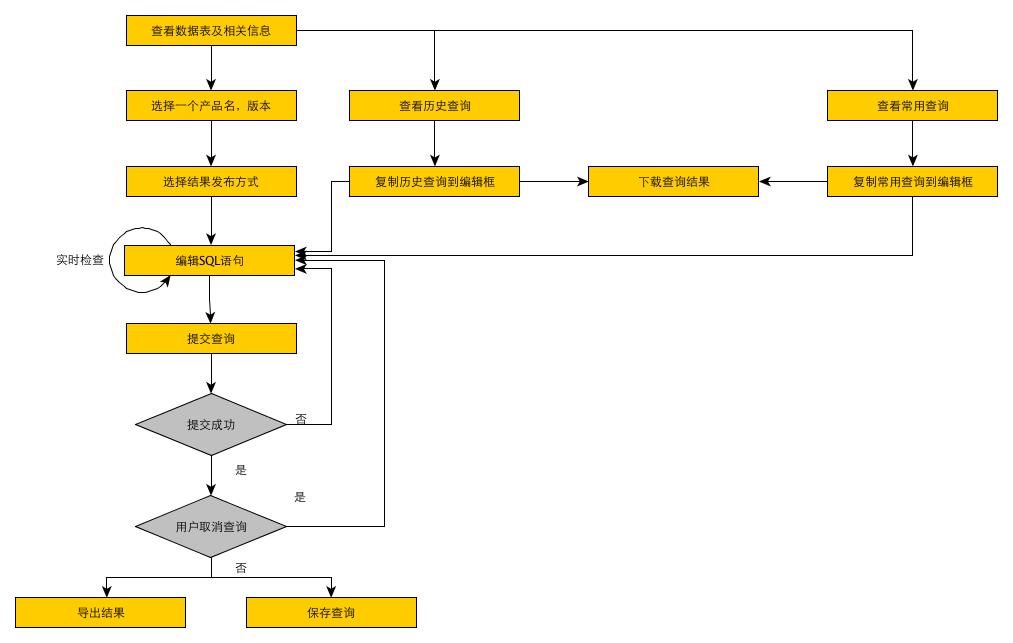
1. 数据收集器一览，列表内容：
   1. 数据收集器名称：文本，不可重复，可排序，按字母序升序排列。
   2. 最后一次同步的时间：云主机最近一次成功从此服务器取回数据的时间，精确到分钟，可排序，默认按时间降序排列。如果从未从此收集器同步过数据，显示：“--”。
   3. 数据传输总量：MB为单位，可排序，无默认排序，初始状态为“--”。
   4. 状态：运行中／已停止／“--”，此状态最长每30分钟刷新一次，列表默认按此列排序，顺序为：已停止》运行中》“—”，已停止的数据收集器在最上面。初始状态为“—”状态。
      1. 发生数据收集器停止工作的情况，可能有以下的原因：
         1. 数据收集器本身出现问题，需要重启
         2. 客户服务器正常下线
         3. 客户服务器出现问题，需要重启或恢复
      2. 对于＃2，即客户服务器正常下线的情况，需要可以允许用户将此收集器删除。
      3. 对于其他情况，需要客户的系统管理员进行相应的操作，恢复数据收集器。
   5. 操作：编辑/删除，客户可以对“已停止”的数据收集器进行删除操作，也可对“－－”的数据收集器编辑路径（名称不可编辑）。
2. 可在此页面添加数据收集器
3. 可查看历史已经被删除的数据收集器
4. 页面顶端消息栏，仅当有“已停止”的收集器时出现，告知用户有数据收集器意外停止工作了，请用户采取后续行动。消息内容：“您的<数据收集器名称>（最多显示3个）等N个数据收集器已停止工作，部分数据没有同步，请通知您的系统管理员帮助重新开启数据收集器。”。用户点“我知道了”将隐藏此消息，用户下次登陆时将重新提醒。

3，用户数据查询处理流程

客户在此界面可以：

1. 浏览所有数据表
2. 浏览历史查询
   1. 已保存的查询
   2. 系统自动保存最近未保存的20条查询
   3. 正在执行的查询
3. 输入查询，执行查询，查询可中途取消
   1. SQL编辑器
   2. 查询结果显示
      1. 界面显示
      2. 查询结果以邮件发送给客户
      3. 查询结果以CSV文件的形式导出

用户提交查询的主干流程：



说明：

* 浏览所有数据表，数据表是按照树状结构组织起来的，树状结构的配置由客户运维人员在配置文件中指定，由数据收集器将配置发送给服务器，在服务器端照此配置组织数据。树状结构如下：
* 产品名，25个字符，如“魔兽争霸”，
  + 版本，20个字符，如“燃烧的远征”
    - 数据表名，最多显示20个字符，如“user\_profile”
      * 数据库列名，最多显示20个字符，如“username”，另外，显示此数据列的数据类型，如“timestamp”,“string”
* 产品名和目标版本的选择，是级联选择，单选。即先选择目前系统中可供选择的产品；接下来选择版本，单选。（备注：如果想实现跨产品或跨版本查询，则需要用户自己在编辑框中手写app.version.table）
* SQL编辑器提供如下的功能：
  + 语法高亮
  + 自动提示和自动补全，如SQL的关键词，产品名，版本名，表名，列名。
  + 支持快捷键，包括：
    - ctrl – c
    - ctrl ＋ v
    - ctrl ＋ x
    - ctrl ＋ r : 执行全部查询
    - ctrl ＋ shift + r ： 执行选中查询
    - tab －显示自动补全选项，上下键回车选择
  + 用户可同时打开多个SQL编辑器，分TAB的形式。
* 查询结果：
  + 页面显示
    - 以数据列表的形式，显示结果的前500条数据，翻页显示。
    - 数据列表提供按列排序的功能。
  + CSV导出
    - 用户可将结果导出为CSV文件保存到本地，也可将CSV文件的下载链接复制出来，使用专门下载工具（如迅雷）下载。
  + 邮件发送
    - 用户可选择将数据CSV文件下载url用邮件的方式发送给指定邮箱。
* 对于历史查询：
  + 用户可将当前编辑的SQL保存下来，以便日后重复使用。用户点击“保存查询”，输入名称（25个字符），即可保存查询到“常用查询”。保存的内容：
    - 查询名称
    - 查询产品名，版本名
    - 查询保存时间，精确到分钟
    - SQL内容
  + 查询历史，系统将为客户保存最新的20条查询和查询结果，列表中的展示内容：
    - SQL正文：最多显示SQL内容的前100个字符
    - 查询产品名（如有多个，仅显示第一个，用“…”隐藏详细）
    - 查询开始执行的时间
    - 查询结果下载
    - 查看详细：用户点击查看详细或直接点击SQL正文，将在新TAB页中展开查询详细，用户可基于此查询编辑SQL并执行SQL，注意此后的改动将不会影响到被编辑的历史查询，这是一个全新的查询。
  + 系统当前正在执行的查询，在列表模式下，显示：
    - SQL正文：最多显示SQL内容的前100个字符
    - 查询的进度，是一个百分比的进度
    - 用户点击某个查询或点击查看详细，新开一个tab页面，回到查询执行现场，此时查询不可被编辑。
* SQL查询提交失败后服务器给的错误信息，在结果展示区域显示给客户。
* SQL查询可以中途取消，结果展示区域里显示错误提示：“任务中途取消”。
* 用户可以开多个tab同时提交多个查询，对于正在执行的查询，在当前tab页面显示查询的进度（百分比）。
  + 用户点击历史查询或已保存的查询，都将新开一个tab页面
* 异常处理：
  + 用户中途退出系统或关闭浏览器或关闭浏览器标签
    - 尚未执行的SQL将不会被保存，因而在用户关闭浏览器或关闭浏览器标签的时候，提示用户：“请先保存在DataWise里的工作”
    - 用户下次登陆系统时，对于正在执行的SQL，用户可以直接回到SQL执行现场。

4，用户中心流程

用户中心提供：

1. 更改密码
2. 更改注册邮箱，新邮箱需要重新验证，验证成功后下一次登陆生效，否则不生效。
3. 更改昵称
4. 退出当前登陆，提示用户，当前SQL编辑器内未保存的信息将会被丢弃

5，系统管理员后台

费沙内部界面，用来管理和监控阿里云机器，以及管理用户。自己搞一个吧，程序猿们，丑点儿二点儿能跑起来就行。

## Installation

This section includes requirements involving the software installation and un-installation by our customer and/or the end consumer. This class of requirements is most relevant for packaged products. For hosted applications and other business services like VideoTracker, this section can be deleted.

## Account Provisioning

This section includes requirements involve tools and processes required to establish and populate accounts for new customers and their employees, enabling them to receive one or more Vobile services. In some cases, this may involve the Vobile CRM system. In other cases, it may involve Vobile employees using home grown tools to populate product-specific databases (e.g., VDDB). In yet other cases, it may involve delivering to customer software functionality that enables them to create new accounts, create new users, manage user permissions, etc. This class of requirements is most relevant for services like VideoTracker. For packaged software products this section can be deleted.

## Authorization and Authentication

This section includes requirements involving user accounts, access control, identity management, and granularity of permissions. For example, what are the different roles? Which rights are assigned to individuals in each role? How are user accounts provisioned and managed?

## Security

This section includes requirements involving, for example, proactive defenses against hacking, cracking, and unauthorized system access. These might involve encryption, password expiration, specific business processes, etc.

## Performance

Depending on the product, this section might include requirements involving, for example: playability, effectiveness, capacity, throughput, scalability, latency, responsiveness, and the like.

## Reliability

Depending on the product, this section might include requirements involving, for example: maximum scheduled downtime, maximum unscheduled downtime, redundancy, quality of service guarantees, and the like.

## User Documentation

This section summarizes requirements for online and offline documentation for external users. Typical examples include: context-sensitive online help, release notes, a user manual, an installation guide, certification procedures, a quick reference card. The exact set of publications depends up on the product.

## Internationalization and Localization

This section summarizes requirements for internationalization (colloquially, “I18N”) and/or localization (colloquially, “L10N”). The former typically refers to up-front architecture and infrastructural investment to provide a generalized capability to support non-Western characters, currencies, laws, cultures, and geographies. The latter typically involves implementing specific applications to support specific locales by translating user interfaces to the local language, validating the product on a corresponding foreign language version of the operating system, etc.

## Branding

This section summarizes the extent to which Vobile corporate, product, and feature branding must (not) appears throughout the product itself, including the user interface, file and directory names, user documentation, support web sites, and interactive end user license agreements (EULAs). Some products will be overtly branded Vobile solutions. Other products will be private label, OEM solutions branded by our customers or mere ingredient technologies embedded within a customer’s product.

## Platform Requirements

This section includes requirements specifying the gamut of hardware and software environments in which the product must function properly. This may require specification, alone or in combination, of minimum supported browser versions, OS versions, RAM, disc space, and/or screen resolution. For example,

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Win98** | **Win NT** | **Win 2K** | **Win ME** | **Win XP** | **Mac 8.X** | **Max 9.X** | **OS X** |
| **IE 5.5** |  |  |  |  |  |  |  |  |
| **IE 5.5 SP1** |  |  |  |  |  |  |  |  |
| **IE 5.5 SP2** |  |  |  |  |  |  |  |  |
| **IE 6.0** |  |  |  |  |  |  |  |  |
| **IE 6.0 SP1** |  |  |  |  |  |  |  |  |
| **Safari 1.0** |  |  |  |  |  |  |  |  |

Figure : Browser and OS Compatibility

## Product Demonstration

In some cases, to facilitate the sales process, a product must be built in such a way as to enable live product demonstrations by Vobile employees and other non-customers. This section lists requirements specifically designed to address this need.

# Use Cases

*Use cases* are stories about how users will, under typical circumstances, take advantage of the capabilities of the product to satisfy one or more basic needs. Time permitting, they can be very casual and abstract, written at a high level, or they are more formal and more detailed. They provide a more granular walkthrough of one path of the system, from the perspective of the user.

1. <Use case goes here>
2. <Use case goes here>
3. <Use case goes here>

PRD Checklist (Delete Prior to Publication)

* **Adhere to This Template** – If this template does not meet your needs and you feel the urge to deviate substantively from it--update the template. That way, all current and future products and product managers benefit from your insight, and the PRD template becomes more comprehensive, more general, and more robust.
* **Don’t Specify Implementation** – Requirements should be written at the highest possible level of abstraction, subject to the constraint that they are specific enough to guarantee that satisfying the requirement necessarily addresses the customer need. In general, a PRD must refrain from attempting to dictate the implementation, architecture, algorithms, data structures, or internal interfaces. These are the responsibility of Engineering.
* **Write Objective Requirements** – Each numbered requirement must be worded objectively. That is, it must be written in such a way as to guarantee that there cannot be any disagreement, in retrospect, about whether the product meets a specific requirement. For example, avoid the use of vague, subjective, and qualitative adjectives such as “fast” data transfer, “responsive” user interface, “many” files, “everything”, “all”, etc.
* **Use Numbered Requirements** – Requirements and use cases are numbered sequentially and grouped hierarchically. This is necessary in order to map them to corresponding, downstream, internal requirements, ensuring that none are overlooked and that unimplemented requirements are captured for later development. Once requirement and use cases have been assigned a specific number, these numbers should not change. Stability is required to enable end-to-end traceability of requirements through QA and acceptance testing. Add new requirements only at the end of each section. There should be a quantum increase in requirement number at the beginning of each section.
* **Use Conventional Terminology** – By convention, formal requirements are written using **shall**, and optional or recommended requirements are written using **may**. For example,

R23 The dimples on the knob of the Framowitz panel shall be octagonal.

* **Work Closely with Engineering** – Engage engineering from the very beginning. Interact closely with the development lead and, later in the process, the QA and Ops leads. When the PRD is finalized and frozen, it must represent a meeting of the minds among all parties, and a commitment by Engineering.
* **Solicit Peer Review** – When authoring and vetting PRDs, solicit feedback from the other product managers. Find the time to read each other’s PRDs and provide thoughtful written feedback. This has many benefits. It makes our requirements stronger, which improves downstream quality and schedule. Sometimes a fresh, novice pair of eyes can ask good, basic questions from a fresh perspective. And this will be a good way for all of us to learn from each other, while developing a holistic sense of all of the Vobile product lines and the interfaces and synergies among them.
* **Legal Review** – Consider whether this product has any legal risk, which could include anything from civil and criminal liability to patent, copyright, and trademark issues. When in doubt, ask legal to review it.
* **Modularity** – Rather than delivering to Engineering a monolithic laundry list of requirements, organize requirements into modular, encapsulated bundles corresponding related functionality. This has many advantages. It provides liquidity to the feature prioritization process, and it enables us to add/remove functionality from a release in a less disruptive way. If the PRD defines a wide range of disparate functionality, consider breaking apart the PRD into two or more smaller PRDs.
* **Formalize the Freeze** - This is a working document, until it is frozen. At that point, the “Draft” watermark should be removed.