

## “丢哪儿啦” 需求规格说明书

# 目录

1.	问题描述	3
1.1	业务背景	3
1.2	术语表	3
1.3	主要问题	3
1.4	本系统的使用场景（User Story）	4
2.	系统需求概述	4
2.1	简述	4
2.2	系统用例图	5
3.	核心用例描述	5
3.1	发布失物/拾获信息	5
4.	非功能性需求	6
4.1	可用性	6
4.2	可移植性	6
4.3	安全性	6
5.	领域模型	7

# 1. 问题描述

## 1.1 业务背景

（注：以下背景描述仅针对广东省中山大学）

在学校的日常生活中，同学们遗失物品的现象较为普遍，在这种情况下，通常的做法是，通过人际关系网在学校的各个 QQ 群或者微信群求助；另一方面，捡到物品的同学也是通过这种方式去寻找失主；这种方式存在明显的缺点：寻找的效率和成功率依赖于人际关系网的覆盖面，而且这种求助信息在 QQ 微信群中经常被其他消息淹没，导致拾获者和失主不能及时注意到。另外一种我们所了解的方式是在微博上找公众号求助，比如拥有较高声望的@中大 Din，这种方式也存在明显缺陷：一个前提是失主和拾获者需要同时关注该微博账号，而这并不是专门用于发布失物信息的公众号，要实现这个前提是不现实的，其次据调查发现@中大 Din 并不保证会响应所有的失物求助信息，因此不是一个可靠的途径。综上所述，目前校园内并没有提供合理的失物发布/招领的解决方案。

## 1.2 术语表

1. 失物：遗失的物品
2. 相关人：失物发布/招领情景中涉及的人群
3. 失主：遗失物品的相关人
4. 拾获者：拾获失物的相关人
5. 用户：失主或拾获者
6. 失物信息：一次物品遗失中，有助于找到失物的相关信息
7. 拾获信息：一次拾获物品中，有助于找到失主的相关信息
8. 本系统：本项目的最终产品，包括服务端和各个客户端（PC Web 端，移动端，微信公众号）

## 1.3 主要问题

要提供一个合理的失物发布/招领的解决方案，至少需要解决以下几个问题：

1. 对于拾获者，要提供尽可能友好的拾获信息发布环境，提高其发布拾获信息的主动性，同时引导其提供充足的拾获信息，增大失主找回失物的成功率。
2. 对于失主，要引导其提供充足的失物信息，目的同上。

3. 对于拾获信息和失物信息，需要对其进行必要的分类、分析，尽可能做出匹配，向失主主动推送有用的信息

## 1.4 本系统的使用场景（User Story）

当失主需要寻找失物时，可以通过本系统的三个客户端入口之一（PC Web 端，移动端，微信公众号）检索已有的拾获信息，若已有匹配的拾获信息，可以根据拾获信息中的联系方式自行联系拾获者，在成功认领后，由拾获者关闭该拾获信息。

当失主没有检索到需要的拾获信息，需要发布失物信息，可以打开本系统的移动端（可能需要先安装，安装移动端并不是一个友好的解决方案，但通过移动端能最及时地发布失物信息），配合 GPS 定位和内置的校园地图，在系统的引导下提供失物信息（包括但不限于时间、地点、物品描述和联系方式等），之后系统会自动在数据库中寻找可能匹配的拾获信息，若有相关信息则直接提供给失主，否则在取得拾获信息之时通过可能的联系方式（本系统的客户端、短信等）推送给失主。此外，失主也可以通过 PC Web 端发布信息（这种轻量方式不需要用户安装软件，拥有更好的输入体验，缺点是只能在 PC 使用）。介于以上两种方式，系统也提供了一种折中的入口：通过微信公众号来进入 Web 端（这是最便捷的方式，但是受限于微信公众号和移动浏览器，用户体验也最差）

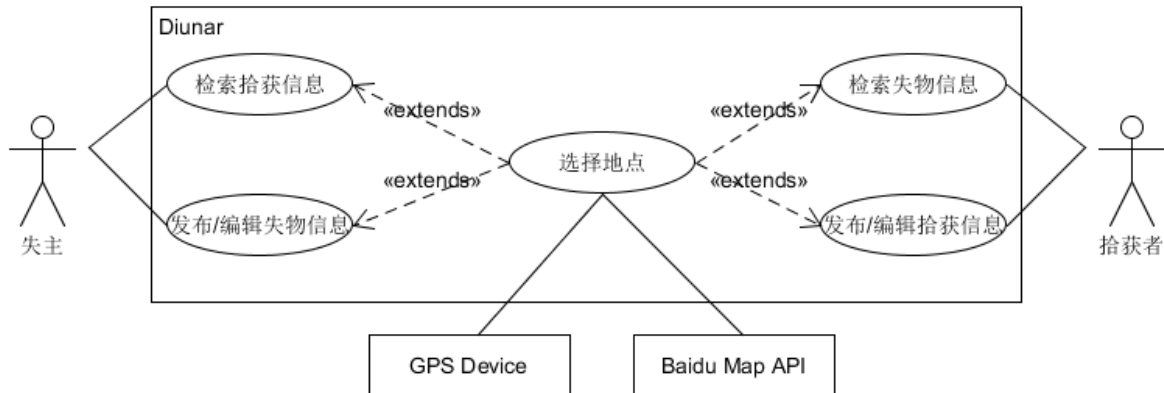
当拾获者需要发布失物信息时，同样可以通过本系统的客户端检索失物信息，或是发布拾获信息。

## 2. 系统需求概述

### 2.1 简述

本系统是一个针对中山大学的失物招领平台，基于失物信息和拾获信息的发布，通过多客户端支持，自动化的信息匹配及推送，提供便捷高效的失物招领体验。

## 2.2 系统用例图



2.2 系统用例图 1

## 3. 核心用例描述

### 3.1 检索&发布失物信息

#### 3.1.1 用例描述

用户在系统中检索/发布失物信息

#### 3.1.2 活动流

1. 用户在系统中检索已有的拾获信息（按分类，地点，时间检索）
2. 若用户找到需要的信息（失物已被拾获），则自行联系拾获者，结束
3. 若失物未被拾获，则用户发布失物信息（提供丢失的地点、物品分类、描述以及联系信息）
4. 用户等待拾获者

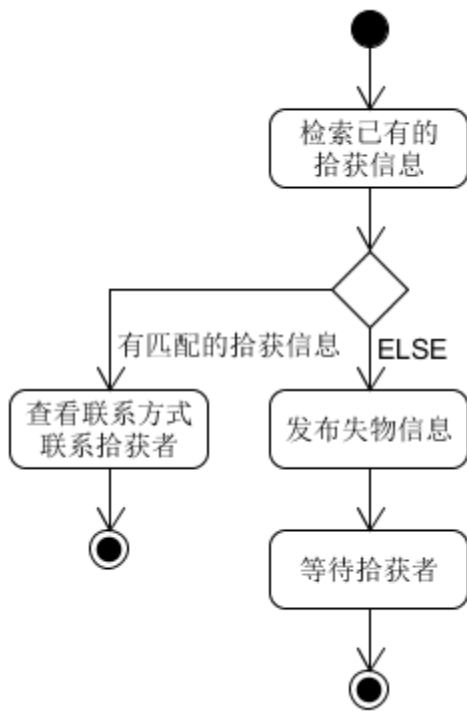
#### 3.1.4 前置条件

无

#### 3.1.5 后置条件

若用户发布失物信息在系统中生成一条失物/拾获信息记录

#### 3.1.6 活动图



3.1 活动图 1

## 4. 非功能性需求

### 4.1 可用性

1. 移动端应至少兼容 Android 4.0 及以上版本
2. PC Web 端应至少兼容 Google Chrome 浏览器
3. 移动 Web 端应至少兼容 UC 浏览器及 Google Chrome 浏览器

### 4.2 可移植性

1. 服务端应可以方便地移植到各种平台（Linux/Windows 服务器，SAE 等云平台）

### 4.3 安全性

1. 服务端应有基本的 Web 攻击抵御能力

## 5. 领域模型

