



南京大學

本科畢業設計

院 系 _____ 软件学院

专 业 _____ 软件工程

题 目 Launchpad 系统搜索模块的设计

与实现

年 级 _____ 2012 学 号 _____ 12125002

学生姓名 _____ 卑一佳

指导老师 _____ 贝佳 职 称 _____ 副教授

论文提交日期 _____ 2016 年 5 月 25 日

南京大学本科生毕业论文（设计）中文摘要

毕业论文题目： Launchpad 系统搜索模块的设计与实现

软件学院 院系 软件工程 专业 2012 级本科生姓名：

卑一佳

指导教师（姓名、职称）： 贝佳 副教授

摘要：

SAP 公司，作为全球最大的企业管理和协同化电子商务解决方案供应商，客户（企业）通过不同的部署方案（包括 on-premise, cloud 和 hybrid）来获得产品和服务。为在技术支持方面给客户带来更好的体验，ONE Support Launchpad 应运而生，它是任务驱动的、客户定制化的、高度统一的一系列服务和支持应用的平台。

ONE Support Launchpad 采用了 Fiori 的架构，顺应了公司未来产品的发展路线，对不同产品线上的支持提供单一的服务入口，统一的对接平台和整齐划一的操作模式。

ONE Support Launchpad 的整个项目实际上是一个以旧换新的包装过程，底层与后台沿用老系统的逻辑和数据。系统基于灵活的 Fiori 架构进行设计开发，整个项目部署在 HANA Cloud Platform，作为前端项目使用 SAPUI5 框架（以实现 Fiori 的风格），整合了 ADOBE CQ5 技术，使用 Java Script、CSS、HTML5 编写完成。后台延续了旧系统，部署在 SAP HANA 的 WEB 服务器和数据库（存放了业务逻辑和数据源），由于后台开发不在 Launchpad 项目核心之中，在这里不作赘述。项目主要采用迭代式开发，从而满足不断变化的需求，并确保稳定性。

ONE Support Launchpad 系统包含了八大模块，分别管理支持体验的各个应用。其中，搜索模块用来协调其他模块，实现系统最关键的智能搜索功能，是项目的核心。目前，该项目已经成功上线，对 Success Factor 的客户开放使用，并不断受到好评。

关键词：技术支持，Launchpad，Fiori，HANA

南京大学本科生毕业论文（设计）英文摘要

THESIS: The Design and Implementation of Search Module in Launchpad System

DEPARTMENT: Software Institution

SPECIALIZATION: Software Engineering

UNDERGRADUATE: BEI YIJIA

MENTOR: BEI JIA

ABSTRACT:

Customers (enterprises) obtain products and service from SAP Company that is the world's largest provider of enterprise management and collaborative e-commerce solutions through different deployment plans (including on-premise, cloud, and hybrid). ONE Support Launchpad provides a series of mission-driven, customized, and highly unified services and support applications, and simplifies customer experience with SAP technology support. ONE Support Launchpad adopts Fiori framework, complies with the development route of future products of the company, provides single service entrance for online support of different products, and has unified docking platform and uniform operation mode.

In fact, the whole project of ONE Support Launchpad is a packaging process of replacing a new one with an old one. The bottom and background continue to use the logic and data of the old system. The whole project is deployed in the HANA Cloud Platform. As a front project, it uses SAPUI5 framework (to achieve Fiori style) and integrates DOBE CQ5 technology. It is compiled through Java Script, CSS, and HTML5. The background continues to use the old system. The web server and database (storing business logic and data source) deployed in SAP HANA is not repeated here; after all, background development is not in the Launchpad core project. The project focuses on iterative development, thus meets the changing needs, and ensures stability.

ONE Support Launchpad system includes eight major modules, which respectively manage various applications of support experience. Among them, search module coordinates other modules, achieves the most critical intelligent search function, and is the core of the project. At present, the project has been successfully launched online. It is open to Success Factor customers and obtains constant praises.

KEY WORDS: Technical Support, Launchpad, Fiori, HANA

目 录

图目录.....	II
表目录.....	III
第一章 引言	1
1.1 项目背景	1
1.2 国内(外)研究综述.....	2
1.3 本文的主要工作	4
1.4 论文的组织	5
第二章 相关技术概述.....	6
2.1 Fiori 框架	6
2.1.1 Apache Cordova	6
2.1.2 SAP UI5	7
2.2 HANA Cloud Platform.....	8
2.2.1 SAP HANA.....	8
2.2.2 HANA Cloud Platform	9
2.2.3 XS Server	10
2.3 其他技术	11
2.3.1 Github	11
2.3.2 UML 建模	11
2.4 本章小结	12
第三章 系统分析与设计	13
3.1 系统的需求分析	13
3.1.1 功能需求.....	13
3.1.2 流程需求.....	26
3.1.3 非功能性需求.....	27
3.2 系统总体设计	28
3.3 本章小结	30
第四章 搜索模块的详细设计与实现.....	32
4.1 搜索模块概述.....	32
4.2 搜索模块的详细设计	32
4.2.1 展示与应用层设计	32
4.2.2 数据存储层设计	36
4.3 搜索模块的实现	40
4.4 本章小结	48
第五章 总结与展望.....	49
5.1 总结	49
5.2 展望	49
参考文献	50
致谢	51

图目录

图 2.1 Apache Cordova 调用机制图	7
图 3.1 系统用例图	15
图 3.2 搜索子模块用例图	16
图 3.3 页板管理子模块用例图	20
图 3.4 软件下载中心子模块用例图	21
图 3.5 用户管理子模块用例图	22
图 3.6 系统数据管理子模块用例图	23
图 3.7 Notes 和 KBA 管理子模块用例图	24
图 3.8 许可密钥管理子模块用例图	25
图 3.9 Incident 管理子模块用例图	26
图 3.10 系统活动图	27
图 3.11 系统技术架构图	29
图 3.12 XS Server 示意图	30
图 4.1 standardsearch 界面原型图	34
图 4.2 suggestion 界面原型图	35
图 4.3 allrepositorysearch 界面原型图	36
图 4.4 数据存储架构图	37
图 4.5 主页 search 模块界面	40

表目录

表 3.1 系统主要功能表.....	13
表 3.2 一站式搜索用例描述表	17
表 3.3 跳转搜索结果用例描述表	17
表 3.4 自动补全结果用例描述表	18
表 3.5 自适应搜索用例描述表	19
表 3.6 非功能性需求表.....	27
表 4.1 item 内容格式一	34
表 4.2 item 内容格式二	35
表 4.3 JSON 对 value 的定义表.....	37
表 4.4 Attribute 数据库表	38
表 4.5 USER_PROFILE 类定义.....	39
表 4.6 USER_PROFILE 类方法介绍	39
表 4.7 LandingPage.js 代码	41
表 4.8 suggestionmanager 核心代码.....	43
表 4.9 处理 searchcenter 核心代码	44
表 4.10 注册一个应用示例代码	46
表 4.12 默认 repository 示例代码.....	47
表 4.13 xmlFragement 示例代码.....	47

第一章 引言

1.1 项目背景

SAP 公司，以创造和创新企业的商业解决方案为核心价值，是企业应用软件的市场领头羊，作为全球第三大独立软件供应商，SAP 公司不断寻找新的发展机遇来面对不同客户的全方位挑战，满足不同的商业需求。从行业来看，SAP 提供的解决方案已涵盖了国防、零售、汽车、制造、保险、银行、公用事业等各个行业；从业务线来看，财务管理、制造、人力资源、供应链管理、采购、成本、销售等等不同的业务路线都能使用 SAP 提供的各套方案，各有侧重；从产品来看，有提供企业管理的产品，有数据库管理的产品，有技术管理的产品等等^[1]。总之，从 SAP 进军中国以来，其发展之势如破竹，不但迅速占领了大型企业市场，更将发展的眼光锁定在中小型企业，力争做到企业定制化，满足各类企业的需求。

随着市场的高速变化，科技的日新月异，技术成本的变迁，SAP 不仅在产品和服务上寻求创新，更是要在技术支持方面寻找突破。一方面，随着产品和解决方案的扩张，原有的技术支持平台无法满足巨大的接口量的增加同时保证支持平台的统一和稳定；另一方面，原有的平台在性能和功能上不能适应新兴技术的发展要求；并且，平台在用户体验方面不能满足更高层次的要求^[2]。因此，SAP 技术支持团队决定在原有的 Support Portal 的基础上开发这样一个集中、统一的 ONE Support Launchpad。

从以信息驱动为核心的支持信息集成平台，到任务驱动为核心的支持系统，再到以用户为核心，可定制、个性化，SAP 的支持服务注定要走这样一条改革之路，而这条路的终点就是——用户。一切源于用户而终于用户，这才是真正的宗旨。这个系统被命名为 ONE Support Launchpad，即“一个”支持平台，用户无需翻阅公司的帮助网站去检阅帮助文档，无需联系特定的人员取得远程连接，无需登录复杂的系统申请权限，即所有的支持体验都由单一的平台获得，这样，大大降低了用户获得支持的复杂程度，提高了用户体验。

在开发前，SAP 向客户以多种形式开展了需求工作，构建这样一个新型支

持平台顺应了市场需要，能够给公司在技术支持方面带来飞跃性突破，有非常乐观的发展前景。

1.2 国内(外)研究综述

所谓 ERP（Enterprise Resource Plan）即企业资源计划，是以信息技术为手段，在先进的企业管理经验和知识的指导下，全方位地集成企业的资源，有效调动，为企业提供一个集计划、经营、决策、评估等为一体的企业资源管理平台^[3]。

随着经济发展的日新月异，大型企业的不断涌现，各个行业公司规模的不扩大，企业越来越需要一个独立的第三方资源协调者和企业管理模式，ERP 的市场在不断扩大，需求不断加强。近年来，国内外涌现出了一批做 ERP 软件的先驱。根据市场份额的占有情况和公司规模、认可度等，大致可以分为三层：第一层由 SAP 和 ORACLE 并驾齐驱；第二层有 IFS、BAAN 等等；第三层有在国内耳熟能详的金蝶、用友、浪潮等。

不同的公司对他们的 ERP 产品所提供的服务通过不同的渠道、不同的方式进行。有的创建线上支持平台，同步即时地与客户保持协调畅通；有的成立专门的支持部门，派专业技术人员驻扎在客户公司时刻维护。不同的支持服务带来不同的用户体验，所得到的产品评价、产品使用的持久度有了明显的差异。

下面我们针对国内外 ERP 行业内的主要企业产品以及它们提供支持服务的情况做简要的分析和对比：

（1）SAP ERP & Support

SAP，作为 ERP 产品根据市值排名第一的厂商，SAP 既指公司名称，又是该公司主打产品企业管理解决方案的名称。据调查，在世界五百强中，有超过百分之 86 的企业在从 SAP 提供的解决方案中受益，未来还将更多。SAP 主要有三条产品线：适应于超级大公司群组（世界五百强）的 R3；适应于中大型企业的 Business All-in-One；适应于我国国内中小企业的 Business One。对比于其他的 ERP 产品，SAP 最明显的优势是严谨，或许在灵活度、在性能上未来都有可能存在超越其产品的创造，但在整体的延续性、整体性和稳定性来看，SAP 在未来很长的一段时间都不至于掉队。然而缺点是价格太高，门槛高，对于小企业来说压力较大，且需要专业人士维护。

SAP 始终注意提升自己的服务质量,技术支持覆盖到每一个产品和解决方案。公司专门成立的 Service & Support 部门,作为 SAP 与客户之间的桥梁,不仅能帮助客户科学制定战略目标和业务目标,准备地做出关键决策,并建立了一种长期高效稳定的互动模式,有助于 SAP 产品的推行和客户公司应用的便捷。此番建立 ONE Support Launchpad 将成为最大且唯一的提供技术支持的平台,以便于公司转型。除此之外,SAP 有多渠道的丰富的支持社区,供客户和合作伙伴们交流互动。用户还可以通过成立 SAP 战略合作伙伴关系去获得最高优先级的支持体验。

(2) ORACLE ERP & Support

作为全球最大的信息管理软件和服务的供应商,在 ERP 方面也是锲而不舍,是目前世界排名第二的做 ERP 产品的厂商。ORACLE 的闪光点在于为其创造巨大商业价值的关系型数据库和数据库语言,因而受到许多中大型企业的欢迎。且对比于 SAP 的 ABAP,ORACLE 提供的 ERP 产品后台使用 JAVA 语言,学习门槛较低,相应地降低了实施成本。且其可移植性、再开发性都比 SAP 灵活的多。然而,功能上还不能完全覆盖到各类企业流程,加上对中文的支持较差,ORACLE 的 ERP 产品在中国的 ERP 市场上竞争不占优势^[4]。

Oracle Support 包括标准支持服务和高级客户支持服务^[5]。与其他的 ERP 软件公司相比,Oracle 明显在支持上花费的精力甚少,支持平台略显单薄,对 ERP 产品这一块也没有成熟的支持模式和单独的技术交流社区。尤其是近几年,屡受诟病。然而鉴于其以数据库系统为主要赢利点的优势,客户流量并没有因为支持体验的退步受到影响。

(3) 用友 ERP & Support

用户的 ERP 系统在中国 ERP 行业的最大竞争力,尤其在近几年通过自助创新,发展态势凶猛,占领了国内许多中小型企业市场。用友的 ERP 与 SAP 相比,在技术实力和服务质量上都不占优势,但由于对本土传统行业的了解,加上较快的市场应变速度,很适合国内的传统型企业,未来之势不可小觑。

用友公司对其 ERP 产品创建三千多人的支持团队,在全国有上千家网点,对于本土的中小型企业来说,获得支持十分便捷,问题解决及时高效^[6]。还成立了客户俱乐部,提供交流和获取咨询的平台。支持服务的进度、人员都做到透明化,

成为将 ERP 产品本土化的极好范本。然而，与成熟的 ERP 国际龙头相比，用友缺少完善迅速的线上平台，不能顺应未来支持服务的发展趋势。

（4）第三方支持系统

另外，随着 ERP 产品的日趋成熟，市场需求的响应，近年来涌现出一大批第三方咨询公司，作为第三方力量，站在客观科学的角度，计划和监督、辅助 ERP 产品的实施和维护。这样的技术支持通常建立在深厚的行业背景和对行业底层业务的深层了解之上，不但需要对 ERP 供应商系统的熟练掌握，更需要对行业衔接的把握。通常来说，中小型的不具备高级技术部门的企业，更加青睐于通过第三方支持服务缩小 ERP 维护成本，提高运营效率。然而这种第三方的支持也存在一定的弊端，例如无法及时掌握 ERP 系统的更新情况，多数的第三方公司只掌握老版本的技术，使用最新版本的客户企业依然需要到供应商处寻求支持。这就促使着 ERP 产品的供应商不断推陈出新，挖掘用户痛点，时刻了解和掌握用户的需求，不断创造和创新更先进的支持系统，带来更快捷的支持体验。

1.3 本文的主要工作

本文详细描述了 ONE Support Launchpad 系统的核心 Search 模块的设计与实现过程。

首先，本文从系统需求分析开始，按照用户的需求度罗列，确定了系统需要满足的主要功能需求和非功能需求。提出了基于 Fiori 开发风格，满足平台可移植性要求，涵盖支持板块系统主要功能点的概要设计方案。确定了整个前端系统使用 Eclipse 的 IDE 开发，后台延用 HANA DB 的数据库和在应用服务器的逻辑体系，再集中部署在 HANA Cloud Platform 以满足用户可移植、可云端创建应用的具体要求。

其次，本文分析了系统的架构设计，使用 B/S 架构将系统划分为三层：表示层（Web 浏览器）、应用层（应用服务器）和数据存储层（数据服务器）。其中，在表示层系统使用 SAP UI5 的框架融合 ADOBE CQ5 产生动静结合、实现 Fiori 风格的表示效果；应用层中的逻辑存放在 XS Server，数据存储层中的数据存放在 HANA DB，两者都由 SAP HANA 提供，共同实现了系统的功能需求。

接着，本文深入探讨了 Launchpad 系统的核心模块——搜索模块的详细设计。

一方面是对搜索模块的应用层的主要功能和结构设计，另一方面是对搜索模块数据存储层的数据库设计的详细阐述。

最后，本文展示了搜索模块的重要实现，叙述了包括搜索模块的主要前端代码，注册一个应用的逻辑代码和其他主要的辅助功能的实现过程。

1.4 论文的组织

第一章：引言，介绍了项目背景，当前国内外开发 ERP 产品的主要软件公司提供 ERP 产品技术支持的现状。并描述了该论文的主要结构和工作。

第二章：技术阐述，介绍了 ONE Support Launchpad 在设计与实现过程中设计到的主要技术点，包括 Fiori、HANA Platform 等等。

第三章：罗列系统的功能性和非功能性需求，给出系统的概要设计和体系结构设计。

第四章：具体叙述搜索模块的主要功能的设计与实现，包括核心的前端代码和核心功能的实现过程。

第五章：总结系统已经实现的功能和上线得到的反馈，探讨不足，并提出对未来发展的计划和建议。

第二章 相关技术概述

2.1 Fiori 框架

SAP Fiori 是一种全新的 UI 框架，是一整套全新的 SAP 用户交互系统，一种在前端进行交互的规范。能够做到简洁化、个性化、统一化、平面化，以用户为中心，灵活自适应多种终端，最终目的是最简化用户体验，让用户一目了然^[7]。

Fiori 有多种开发环境，有在 SAP 网页的 IDE，有在云端开发，也有 Fiori 的应用程序客户端。其中，Fiori 客户端使用了标准的 Apache Cordova 的命令行接口，和 SAP 开发的一些增值的 Cordova 插件。通过与微软的合作，企业用户可以用 Visual Studio 在 SAP 移动平台(SMP)通过 Apache Cordova 开发自定义的 Fiori 应用程序。这样，IT 开发人员的注意力将都集中在具体的解决方案的业务模型，而不用过多花费在技术手段和基础建设。

作为荣获过红点设计大奖的 Fiori 新型 UI，SAP 一直致力于将未来上线的产品和平台都建立在该设计独特的架构之上，为未来的转型打下坚实基础，在生产和支持维护都取得一致性，方便用户。因而，ONE Support Launchpad 也顺应了该趋势，以 Fiori 为依托，不仅在 Web 端使用，其链接的各个辅助应用程序也以 Fiori 技术的模型为基础。Fiori 的设计理念在不牺牲灵活性的前提下提供了一个可拓展的清晰的架构，不仅着力于移动用户体验上，还具备了实时计算和迅捷互动等功能。

在 SAP 设计团队的理念中，卓越的用户体验就意味着简化一切，所谓大道至简的企业文化，Fiori 的诞生在 UI、交互和功能上都极大简化的用户的使用，并能够完美满足任务在起码未来较短时间内可能诞生的业务和功能上的需求。在 Fiori 的帮助下，ONE Support Launchpad 能够实现它的最终目标——了解用户，指导用户，帮助用户，即能认知用户的想法、档案和优先级，在支持体验中指导用户的每一步操作，在用户遇到难题时提供援助。

2.1.1 Apache Cordova

Fiori 主要利用 Apache Cordova 这套设备 API，移动应用的开发人员通过它来

用 JavaScript 访问本地设备的一些功能如摄像头^[8]。联合了其他的 UI 框架，调用 CSS、Java Script、Html 开发智能的在多终端使用的应用程序。调用机制可见图 2.2。

使用 Apache Cordova 的 API 时候，不需要调用本地的 Java 或者 C 代码，而是直接使用 Web 技术，由于所遵循的 Web 标准是一定的，因而在多个平台是一致的，可移植性很强^[9]。从此，开发者只需要编写一次应用程序，就可以在包括 ios、android、blackberry、symbian、webOS 等多个平台发布。

目前许多巨头公司都在使用该 UI 框架，例如 Facebook 使用它来开发移动 SDK，Logitech 使用它来开发软硬件的 controller 应用程序。如今也作为 SAP Fiori 的核心技术，Apache Cordova 的多平台可移植满足了 Fiori 的灵活性需求。

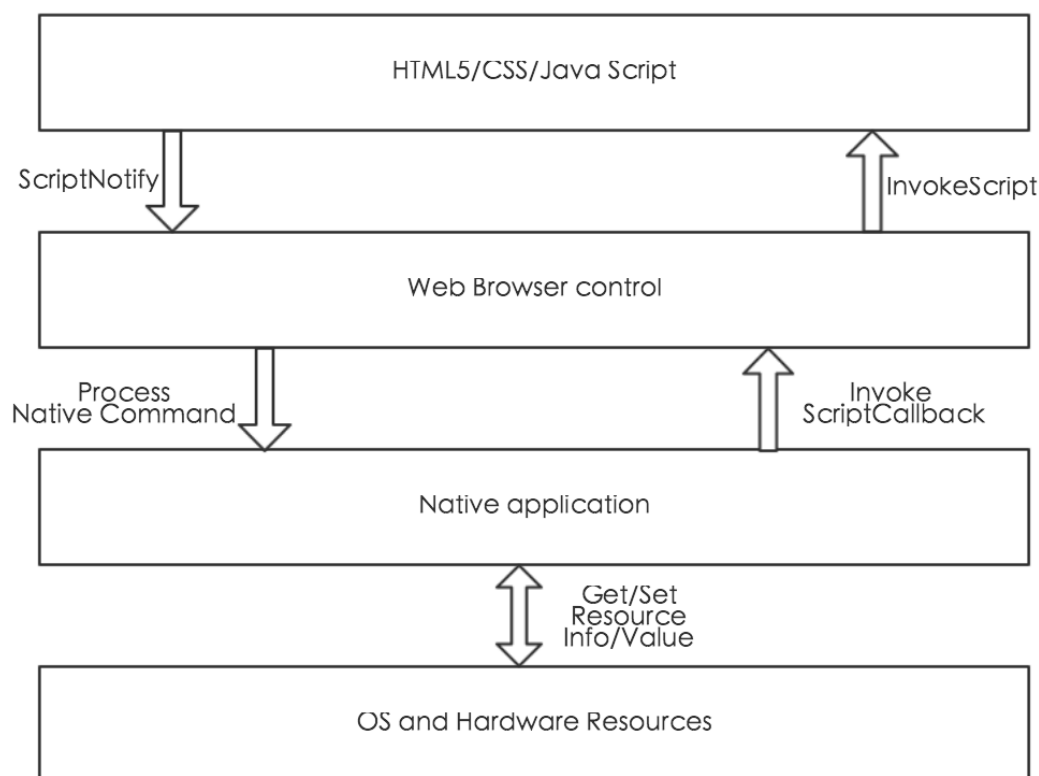


图 2.1 Apache Cordova 调用机制图

2.1.2 SAP UI5

SAP UI5 是 SAP UI Development Toolkit for HTML5 的简称，它是一套基于 HTML5 的 UI 开发组件，是 SAP 为了快速开发 Fiori 风格的 UI、构建企业级产品

的 JavaScript 框架，拥有丰富的界面控制、CSS 页面展示模板、上百个工业图标，同时支持控件的拓展^[10]。UI5 是一个具体的前端框架，能够充分满足 Fiori 的前端交互标准，因而大部分 Fiori Apps 或网页都使用了该框架。

作为一个 JavaScript 框架，SAP UI5 大概满足了以下几个特性：

- (1) 能够支持变成框架，例如 MVC，代码整合难度较低。
- (2) 能够用最简洁的代码来做到多终端开发，同时支持手机、电脑和平板的 UI。

在开发 Launchpad 的过程中，所有的应用都被开发成一个 component 而不是一个 html，这些应用组件一起被包含在一个容器里面，可以是 html div。Launchpad 本身是支持应用组件的容器，这个容器按照用户的权限显示用户能够看到的应用组件，并且按照用户的习惯和偏好对容器的排序、分组和位置做相关的调整。如此，才构成了移动 Portal 的雏形。

2.2 HANA Cloud Platform

2000 年，SAP 开始了 HANA 的部件开发，并在 2011 的 SAP 全球技术研发者大会上开始推广 SAP HANA。2013，HANA 技术平台已趋向成熟，SAP 开始了 HANA Cloud Platform 战略，用 HANA 云平台为所有云解决方案提供有力的技术和存储支持^[11]。如今，以 SAP HANA 为核心的云存储云计算布局已然初步形成，该内存计算技术为创想一个新的世界成为了可能。在云计算和大数据分析如此盛行的当下，奠定了 SAP 基于不败之地的基石。

2.2.1 SAP HANA

SAP HANA 首先是一项内存计算技术，是技术衍生的软硬件结合体。在不需要创造、处理和分析数据的情况下，用户就能对大量的实时数据进行高性能查询。用它来存储、分析、重构数据，处理实时事务非常便捷，且能够快如闪电地构建新应用。硬件是指专门的 HANA 服务器——与传统的服务器相比，HANA 服务器的优势在于将传统的行存储改变为列存储，且作为内存服务器，无论是存储还是读取都有超快的速度——其他服务器是不可用的，软件则是 SAP 的数据库 Sybase。用户可以通过云服务使用 HANA 服务器，而不需要购买硬件。除此之外，SAP HANA 也指基于该内存数据计算技术的高性能的实时数据计算平

台,是一个能够辅助企业开展实时业务的内存数据和应用平台。用户通过该平台,优化内存计算技术,思考和工作方式都得到了转变。

与传统的数据存储分析平台相比, SAP HANA 主要有以下优势:

(1) 快速。经过与 Intel 公司,使用 Intel CPU 处理器的集中扫描验证, SAP HANA 可以达到大于 30 亿次/秒/核,接近 1300 万次的聚合/秒/核,150 万次的插入/秒,处理千兆级别的数据不在话下。

(2) 高效。HANA 内存数据库中的数据不会仅在内存上,也会不停写到硬盘上,使用包括 Trigger-based、ETL-based 和 Log-based 的复制服务器,利用 Sybase 的 Replication Server。另外,开发团队会根据大量级的财会数据集来模拟客户产生数据,这样一来,数据足迹大大减少。

(3) 低成本。根据 Forrester 有关数据表明,从一开始使用 SAP Business Warehouse 的用户,迁移至 SAP ERP,再到使用 SAP HANA,大体上成本能得到百分之 30 至 40 的降低。

(4) 灵活。灵活与强大来自于对数据源的包容性。不同的数据类型、各种不一样的结构和非结构化数据都可以直接家到 HANA 平台进行分析处理,以得到完整的业务视图。

(5) 多重选择。SAP HANA 支持多重部署方式包括预置型部署(On-premise)、云部署(Cloud)和混合部署,所提供的云产品和服务也是分层的,用户可以根据自己的需求来选择基础的入门级产品,还是能够运行 ERP 和其他业务关键服务的企业级别托管云服务。

(6) 开放。开放不仅表现在上游, HANA 的产品和服务托管在三大洲的多个合作伙伴的产品和服务上,也表现在下游,可以运行上千种第三方的应用,也可以在其他的主流硬件上运行。

2.2.2 HANA Cloud Platform

基于 SAP HANA 的云平台,为企业提功能强大的实时应用程序,作为一个内存计算的云平台,基于开放的标准,用户在此平台上云端能够便捷访问到不同的开发环境,全面支持集成服务、企业移动化、协作以及商业分析。包括如下的主要产品:

(1) HANA 云平台移动服务。从云端构建并运行本地应用或者混合应用，企业预置系统和云系统。

(2) SAP Mobile Platform 软件开发工具包。提供给开发者一些列应用程序接口 (API)、工具和组件，实现数据访问、数据建模、身份验证、消息提醒等丰富功能，方便开发移动应用。

(3) HANA Cloud Portal，提供标准框架用来创建商业网站，可移动访问。并拓展现有的业务流程，涵盖包括与供应商、合作伙伴、客户、分销商等的对外业务场景。

(4) HANA Cloud Integration，用来集成来自预置型企业系统和在云端的企业系统的所有业务数据，得到单一的数据源，整合统一业务流程，从而更好地分析建模。

(5) SAP Jam，将关键数据和业务纳入到日常事务处理和社交功能中，这样方便对这些业务和数据的获取和使用。

(6) Fiori 和 SAP Web IDE，提供崭新的 UI 框架，方便定制化构建多终端移动使用的应用程序和服务。

HANA Cloud Platform 所涵盖的主要功能有，帮助构建和部署丰富的轻量级的云端应用并实现多平台可共享，帮助扩展现有的企业预置型行用和云应用（主要有高级分析工具、移动服务、身份验证功能、社交机制等）。

2.2.3 XS Server

XS 是 Extended Application Service（扩展应用服务）的缩写，是 SAP HANA 的重要组件。本文叙述的 Launchpad 系统所需要的 SAP UI5 的数据源和许多 ABAP 类库，底层的业务逻辑都存储在 XS Server 上。

XS Server 专注于用户应用的实现，能够打造更好的 SAP 与客户之间人机交互的体验。通过内置的轻量级 web 服务器，让用户能够在 HANA 上实现 B/S 应用。其中内置了大量的 Java Script API，用于访问服务器会话、调用 SQL 语句、访问资源库、访问 HTTP 请求等。

XS Server 支持包括 OData（Open Data Protocol OASIS 行业标准）在内的多重数据访问方式，企业已有的业务系统能够通过这种接口方式，与 HANA 实时数据

平台进行整合。

SAP HANA 支持了行业标准数据库接口(如 ODBC、JDBC 等),在 B/S 架构中,架设在最终用户与 HANA 之间的应用层,开发人员使用 Java 或 .NET 开发,封装了数据访问接口,实现了具体的业务逻辑,为最终用户的展示层提供基础。这一切都离不开 XS Server 的支持。现今 SAP 只拥有一个 XS Server 的实例并投入使用,不过再不久的将来,会有更多的应用和服务从 XS Server 受益。

2.3 其他技术

2.3.1 Github

最初,为了管理 Linux 内核代码, Linus Torvalds 创造了叫做 Git 的版本控制工具,从此 Git 这个工具在 Ruby 社区迅速火了起来。依托 Git 成立的 GitHub 成为了编程社会化的基石。Git 成长为一个分布式版本控制系统。

与 SVN 之流的传统的版本管理工具不同的是,当获取一份文件,不是获取它最新的快照,而是获取了所有原始的代码仓库的镜像文件。有强大的可移植性和可恢复性,任何一处的服务器发生故障都不会影响到文件内容,因为可以在任何一个镜像的本地仓库进行恢复。

而 GitHub 作为 Git 仓库托管平台,不仅作为托管平台,更是创建了一个多元的、流行的代码社区,成为一个供优秀工程师使用的社交平台,从此代码的合作就像 Facebook 上结识好友一样简单,程序员们通过协作和交流创建更多优秀的开源项目^[12]。

2.3.2 UML 建模

UML (Unified Modeling Language) 是一种对软件密集系统进行可视化建模的语言,可以跨越软件开发的各个阶段和周期,对数据建模、业务建模、组件建模都能提供一种标准简明的规范。在规范的模型建立之后,通过工具可以转化成特定的程序语言代码。

UML 的作用之一,是统一开发团队中所有人员的设计语言,打破了沟通壁垒。UML 所提供的一套标准的建模符号,使设计人员能够阅读和交流系统架构和设计

规划，降低了合作编程的难度。它集合了其他建模语言的优点，摒弃了缺点，定义良好、易于表达、功能强大且普遍适用。它的作用域已经不再局限在面向对象的软件模型的分析与设计，而是支持从需求分析开始的软件开发全过程。

除此之外，UML 还应用于开发非软件系统，如机械系统、企业结构、信息系统、工业系统等。

如今涌现了一大批 UML 建模软件，有 IBM 的 Rational ROSE，有 ArgoUML，对于一般的学习途径，可以选择开源的 UML 建模工具，而企业应用则推荐使用商业的 UML 工具软件。

2.4 本章小结

本章主要介绍了开发 ONE Support Launchpad 需要使用的相关技术。首先，Web 前端采用了 Fiori 的风格，它主要使用 Apache Cordova 的 API，和满足 Fiori 的 SAP UI5 框架。然后介绍了项目部署的平台 HANA Cloud Platform，详细解释了 HANA 技术，系统后台的数据来源于 HANA 的数据库，后台的业务逻辑来源于 HANA 的应用服务器 XS Server，这样，为后文的体系结构和详细设计介绍作了铺垫。最后补充介绍了系统开发过程使用的版本控制工具 Git 和设计计划使用的 UML 模型。

第三章 系统分析与设计

3.1 系统的需求分析

3.1.1 功能需求

ONE Support Launchpad首先应该是用户获得支持服务的主要平台和唯一入口，智能地根据用户的档案以组为单位安排陈列用户所需要的支持应用程序的接口。所有的分组、排列的生成都是以用户为中心的，并且，用户可以自行定义组的划分和陈列方式。

通过调查了解客户对 SAP 的 Support 部门寻求技术支持的体验流程，结合了流程中所涉及到的角色，本文划分出以下主要功能点模块。

如图 3.1，是系统总的用例图，描述了系统的主要功能需求，具体的细节需求没有展开，下文将对各大模块的具体用例做简要阐述，并给出系统的核心——搜索模块的具体用例列表和解释。

表 3.1 系统主要功能表

搜索	用户在搜索模块通过关键词查询到与关键字相关的全部内容（包括 Note、KBA、软件、用户、远程系统、安装信息等），并能进入到该内容涉及到的模块目录下。
页板管理	用户在页板管理模块根据自己的档案（信息、权限和身份）获得不同布局的应用组，可以直接得到每个应用的实时状态和关键信息，可以直接进入每个应用。用户自行定义获得哪些应用以及应用的分组布局。
软件下载中心	用户在软件下载中心模块搜索目标软件的下载文件/更新补丁包/数据库的相关信息，用户得到目标软件经过甄选后适合下载的版本链接和配套补丁。用户可以管理自己的下载篮。
用户管理	用户在用户管理模块可以查询所有自己的账户列表并管理每

	个账户的访问权限。用户可以申请并监管新的账户。
系统数据管理	用户在系统数据管理可以查询和修改自己所有系统的信息，可以删除已有系统，可以标志最常用的和最关注的系统，可以给系统分组和排序。
Notes 和 KBA 管理	用户在 Notes 和 KBA 管理模块可以查询并下载相关的 Notes 或 KBA。用户可以标记/移除标记 Notes 和 KBA 为最关注。用户可以给 KBA 打分和点评。用户可以获取关注的所有 Notes 和 KBA 的最新更新消息。
许可密钥管理	用户可以在许可密钥管理模块申请、查询、管理访问所有系统的密钥。
Incident 管理	用户可以在 Incident 管理模块查询、跟进关注的 Incident 的信息，添加回复，对 Incident 操作等。用户可以智能标记和给 Incident 分类。

ONE Support Launchpad首先应该是用户获得支持服务的主要平台和唯一入口，智能地根据用户的档案以组为单位安排陈列用户所需要的支持应用程序的接口。所有的分组、排列的生成都是以用户为中心的，并且，用户可以自行定义组的划分和陈列方式。

通过调查了解客户对 SAP 的 Support 部门寻求技术支持的体验流程，结合了流程中所涉及到的角色，本文划分出以下主要功能点模块。

如图 3.1，是系统总的用例图，描述了系统的主要功能需求，具体的细节需求没有展开，下文将对各大模块的具体用例做简要阐述，并给出系统的核心——搜索模块的具体用例列表和解释。

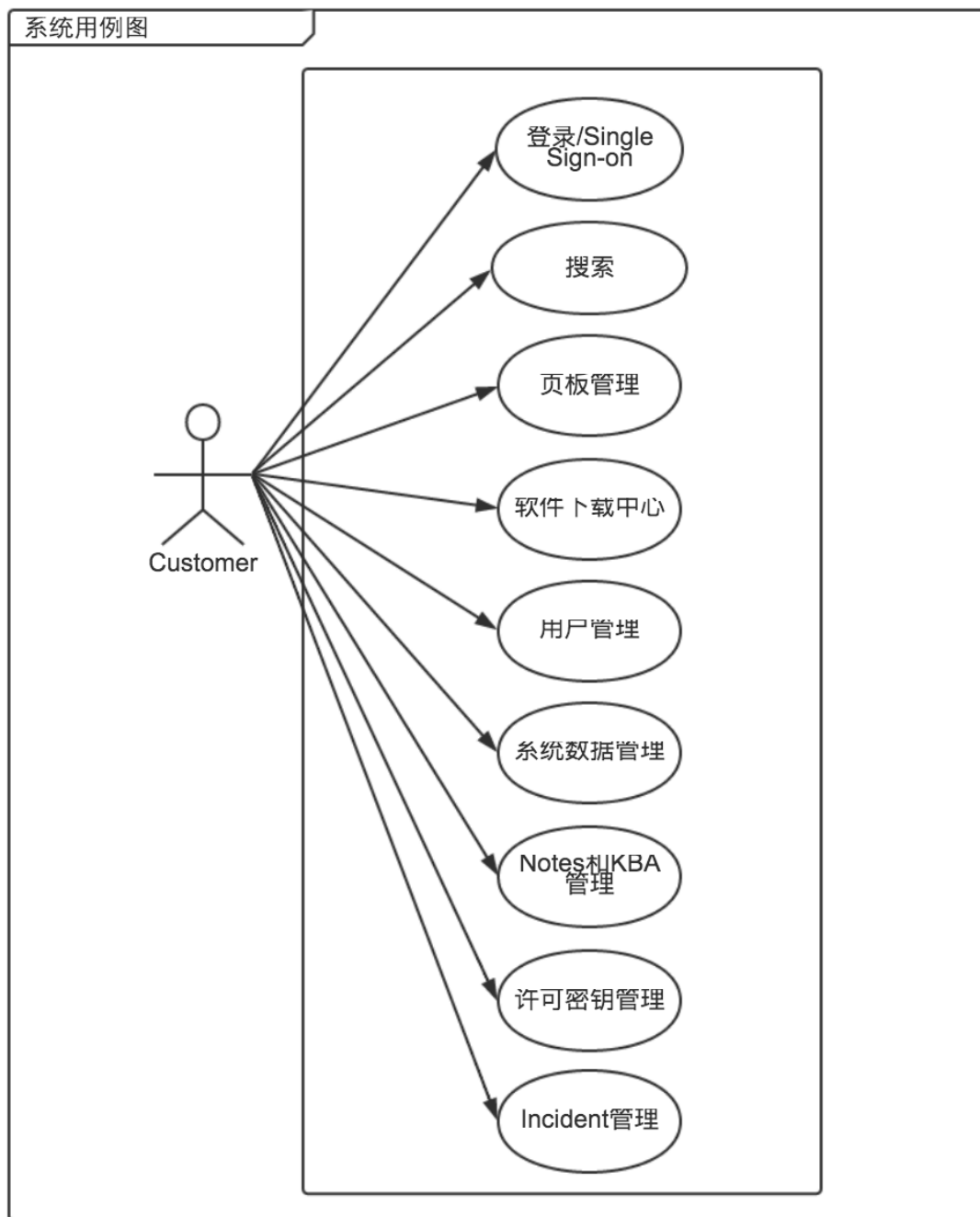


图 3.1 系统用例图

(1) 登录

用户通过账户和密码的验证，成功后方可登录到 Launchpad 平台上，当然，使用 SAP 系统的终端可以直接通过 Single Sign-On 连接到账户。系统要设置一定的账号安全机制，例如，规定时间内无有效操作会自动登出账户，异地登录的账户异常检测等等，使得系统的安全性得到保障。

(2) 搜索

下面给出搜索模块的具体需求描述。

如图 3.2 所示，当用户进入到搜索模块后，可以根据输入关键字进行一站式搜索，是一种全对象搜索包括 KBA & Notes、软件、系统、用户、版本更新、客户 Message 等。系统能够智能地给出用户最近的搜索记录和全站的热门搜索关键字来提示用户。用户可以选择感兴趣的结果，直接跳转到结果所在的模块页面。用户可以给出不完整的关键字，系统能根据关键字智能联想与之相关的搜索结果。系统能根据用户的系统、权限等信息自动给出相关性高的结果优先显示。

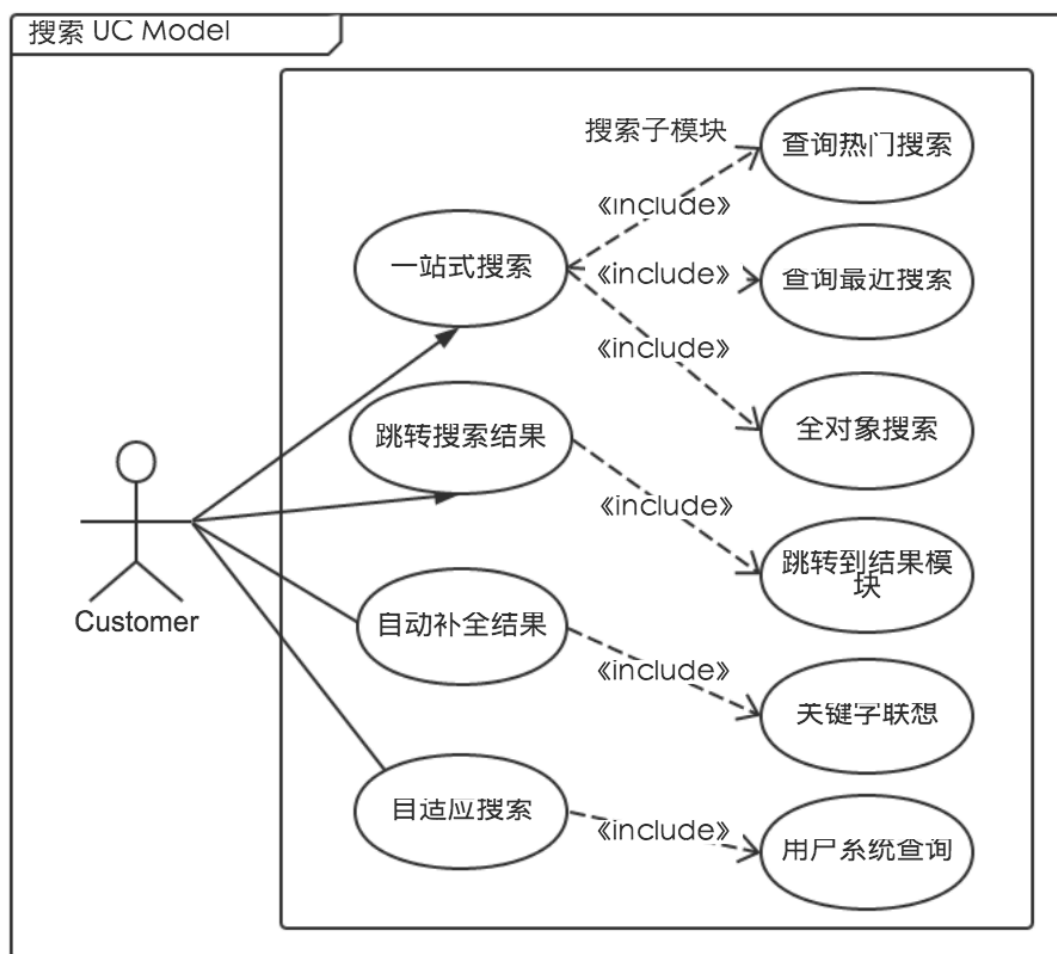


图 3.2 搜索子模块用例图

对图 3.2 子模块中的用例依次进行描述，其具体的用例描述表如下所示。

如表 3.2 所示，用户在始终悬留在页面最上方的搜索栏，根据关键字可以进行一站式的全对象搜索。用户首先要选择搜索的范围 repository，如果选择了 solutions，则可以进行全对象搜索，可能搜索到各种类型的结果。如果选择了其他非解决方案的 repository，具体如 Notes 如 KBA 如 User 如 System，则只对该

范围内的数据进行相关搜索。前提是用户输入的关键字必须有效，用户不能输入空的关键字。不同的搜索范围呈现不同的搜索结果页面，**solutions** 范围的结果列表用复合的界面展示，要能来回切换不同的结果类型。

表 3.2 一站式搜索用例描述表

ID	UC1
名称	一站式搜索
用例属性	创建者：卑一佳 创建日期：2016/04/07 更新日期：无
描述	用户输入关键字，在搜索子模块中进行全对象搜索，结果根据不同种类（Note/KBA、S-User、Customer、System、Installation、Customer Message）进行分组显示。搜索栏提示热门搜索和最近搜索记录。
优先级	高
触发条件	用户输入关键字，点击搜索。
前置条件	用户已登录，且输入了有效关键字。
后置条件	显示搜索结果。
正常流程	移动到搜索栏； 输入关键字，点击搜索按钮； 获得搜索结果。

如表 3.3 所示，用户在选择某搜索结果后，可以直接跳转到该结果的子模块，显示详细信息，系统的 **shell** 要能与结果对应的模块无缝对接。要实现，搜索和结果本身的应用 **tile** 是进入该应用模块的两个入口，两个入口的方式不一样但是能够带来一致的结果。从搜索入口进入到的模块，能够做该模块实现的若干操作，不仅仅是显示结果详情。

表 3.3 跳转搜索结果用例描述表

ID	UC2
名称	跳转搜索结果

用例属性	创建者：卑一佳 创建日期：2016/04/07 更新日期：无
描述	用户选择某搜索结果，直接跳转进入到该结果的子模块中，显示搜索结果的详细信息。
优先级	高
触发条件	用户点击选择某搜索结果。
前置条件	用户选择了某搜索结果。
后置条件	跳转到搜索结果的子模块，显示结果的详细信息。
正常流程	用户选择某搜索结果； 页面跳转到结果的子模块； 显示详情。

如表 3.4 所示，用户输入部分关键字，系统能智能联想到搜索结果并给出提示，根据用户的角色能够自动排序提示，将相关性高的优先显示，且能给出符合范围的提示（例如用户输入数字，系统联想到 Note 编号或者用户）。这样，能够帮助用户最快地得到搜索结果，且能在不知道完整搜索特征的情况下找到想要的结果。

表 3.4 自动补全结果用例描述表

ID	UC3
名称	自动补全结果
用例属性	创建者：卑一佳 创建日期：2016/04/07 更新日期：无
描述	用户输入部分关键字，系统能自动补全，智能联想到与关键字有关的所有结果。
优先级	高
触发条件	用户输入不完整的关键字，点击搜索。
前置条件	用户已登录，且输入了不完整的关键字。

后置条件	显示联想到的搜索结果。
正常流程	用户输入不完整的关键字，点击搜索； 获得搜索结果。

如表 3.5 所示，系统在根据搜索条件搜索到的结果列表中，能够根据用户的系统/版本等信息，自动排序结果中与用户相关性高的优先显示。

表 3.5 自适应搜索用例描述表

ID	UC4
名称	自适应搜索
用例属性	创建者：卑一佳 创建日期：2016/04/07 更新日期：无
描述	用户输入关键字，系统自动检测用户所在的系统和版本，根据角色筛选结果，与用户相关性高的结果优先显示。
优先级	高
触发条件	用户输入关键字，点击搜索。
前置条件	用户已登录，且输入了有效关键字。用户已设置了角色信息。
后置条件	搜索结果列表按照优先级序列显示，相关度高的结果在前排突出显示。
正常流程	用户输入关键字，点击搜索按钮； 获得关联强度高的搜索结果。

(3) 页板管理

页板管理模块是实现用户个性化服务、定制主页的核心。在该模块下，所有的应用对应于界面的一个磁贴（tile），每个磁贴提供到该应用模块的入口。同样性质的磁贴罗列在一起组成一个组（group）。在 Launchpad 的 shell 下，用户要能够自由移动磁贴的位置，更改磁贴的分组。由用户的角色和购买的产品来决定显示哪些磁贴和分组。

如图 3.3 所示，用户进入页板管理子模块后，可以对页面的排版布局进行管

理。可以对根据用户个人档案和系统信息给出的每个组（group）的磁贴（tile）进行管理，包括在组内添加磁铁、删除磁贴、将该磁贴移动到别的组。可以进行常用区的管理（My Home）包括添加磁贴、删除磁贴。可以进行磁贴的页面拖拽。可以根据关键字搜索未出现的磁贴。

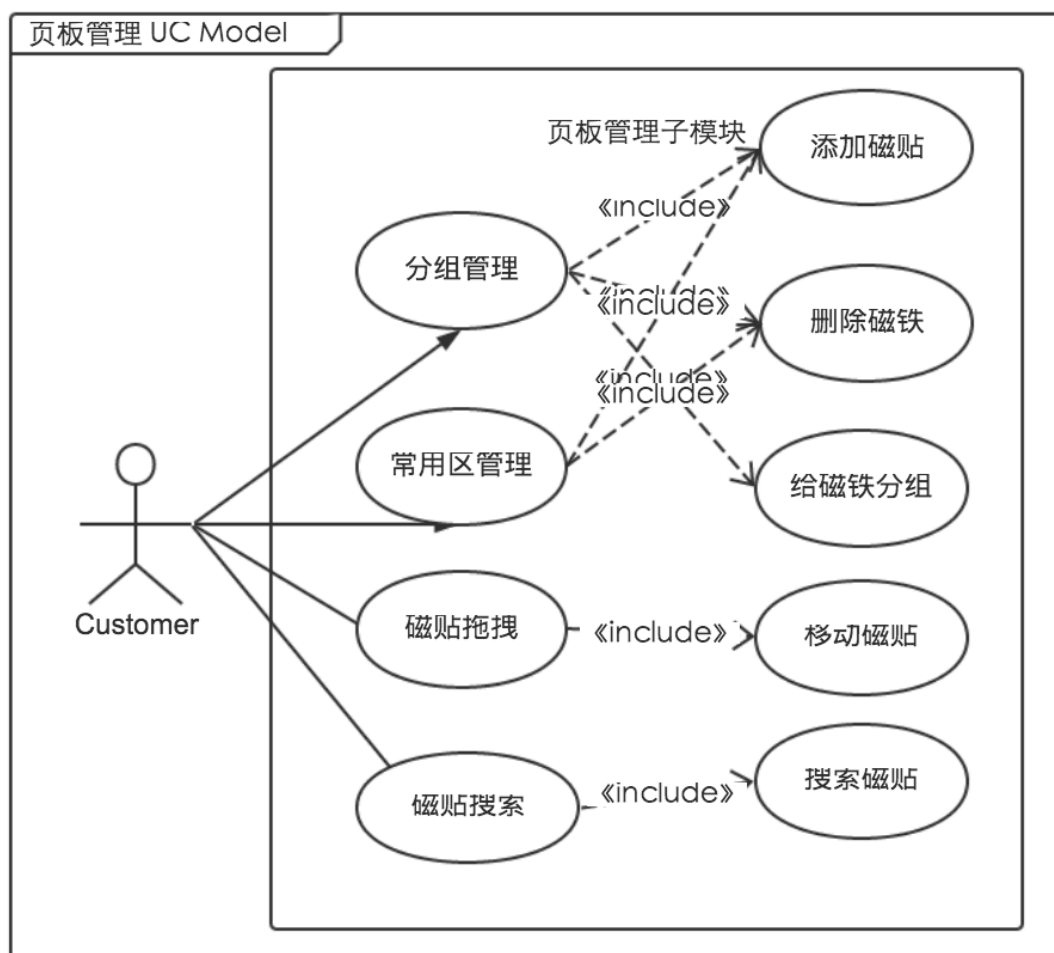


图 3.3 页板管理子模块用例图

其中，首先，用户能够进行磁贴的分组管理包括添加未出现的磁铁进页面，删除页面中已有的磁贴让其不再显示，将某磁铁添加进某分组，将某磁贴从某分组内移除，或者将某分组下的磁贴移动到另一个分组。其次，用户在页板管理模块能够设置自己的关注区，可以将特定的磁贴添加如常用区，设置常用区优先显示在最前，这样能够不用每次寻找应用都先找分组，而是直接从常用区进入应用。用户还要能够对磁贴直接拖拽能够改变磁贴的分组和布局。用户在磁贴管理模块的搜索栏输入查找的磁贴的名称关键字，可以查找到相应的磁贴，并进行操作。

（4）软件下载中心

如图 3.4 所示，用户进入软件下载中心子模块后，可以通过关键字搜索软件，获得所在公司获得权限的软件列表，能够查看软件的具体信息（包括版本、下载要求、辅助应用），能够对软件列表进行分组、筛选和排序，能够选择想要下载的软件某版本加入下载篮，能够管理下载篮包括能够一键下载和查看下载记录。下载中心模块能让用户在 Launchpad 能够自由灵活地找到想要下载的软件的相关信息和本。

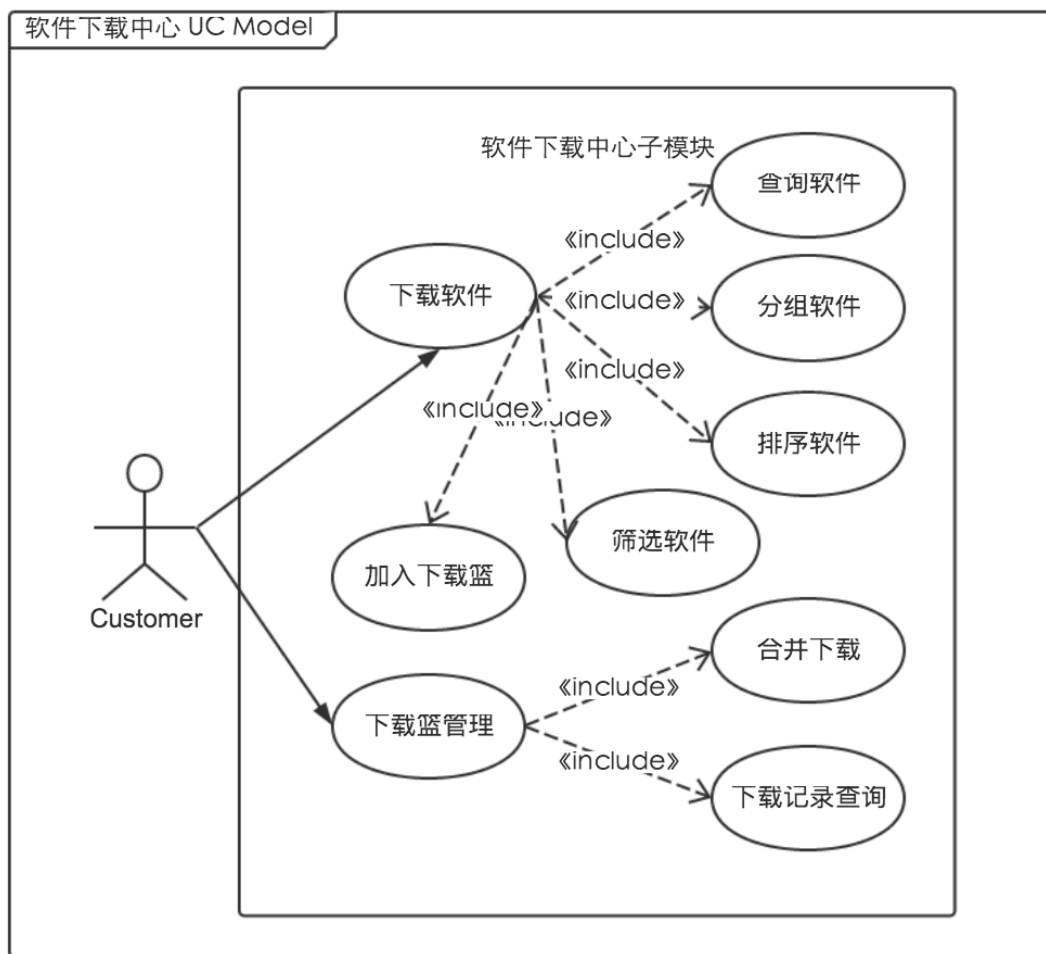


图 3.4 软件下载中心子模块用例图

用户进入软件下载中心后，可以通过关键字来查询想要下载的软件，并且对查询结果进行筛选、分组和排序，以便更快地检索到结果。用户可以点选某个软件，查看其具体信息包括软件版本、大小、功能介绍、适用的系统和其对应的下载文件等。如若用户选择下载该文件，可将其加入下载篮。用户能够对下载篮进行单独管理，可以选择篮中的下载文件进行一键合并下载，也可以删除下载篮中的文件，也可以查询历史的下载记录并对记录中的文件进行重新操作（删除记录

或者将文件添加回下载篮重新下载)。

(5) 用户管理

如图 3.5 所示，用户进入用户管理子模块后，可以查看完整的该账号拥有的用户列表，可以分开查询被请求的用户列表，被删除的用户列表和重要的联系人列表；可以通过用户的 first name/last name/user info 等各种信息来检索用户；可以查看用户的具体信息和删除用户；可以对用户进行权限管理（添加权限或删除权限）；可以请求新的用户。

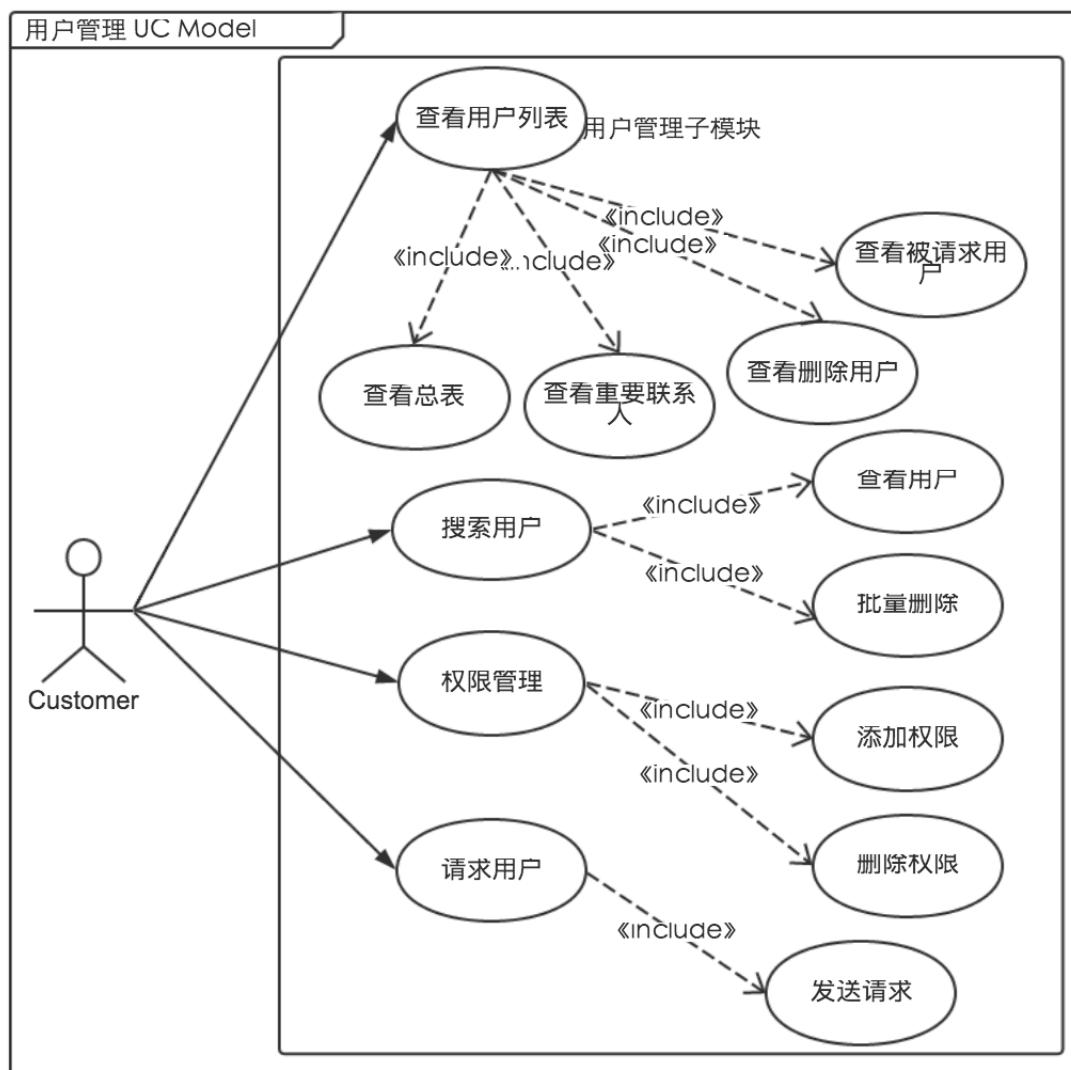


图 3.5 用户管理子模块用例图

用户进入了用户管理模块，可以对查看权限内能够管理的所有用户列表，该列表要以复合的形式展示，分别能够显示总的用户列表，被标记为重要关注人的用户列表，已经删除过的用户记录，已经请求过的用户记录。能够对用户列表进

行相关操作，例如能够点选相关的用户进行合并删除。能够用关键字（可以是用户编号、名称等）来搜索具体的用户，并对其进行操作。能够对用户进行权限管理，方式是在列表中点选用户，添加或者删除权限。也能够请求新的用户，发送请求等待响应，在已请求的用户列表中就能出现发送过的请求以及它们的状态。

（6）系统数据管理

如图 3.6 所示，用户进入系统数据管理子模块后，可以选择查看所有的系统列表，也可以切换查看所有关注的系统列表；可以对列表进行分组、筛选或排序。可以标记系统为关注或者取消原来关注的系统，可以通过勾选列表合并删除系统，可以编辑系统的具体信息。

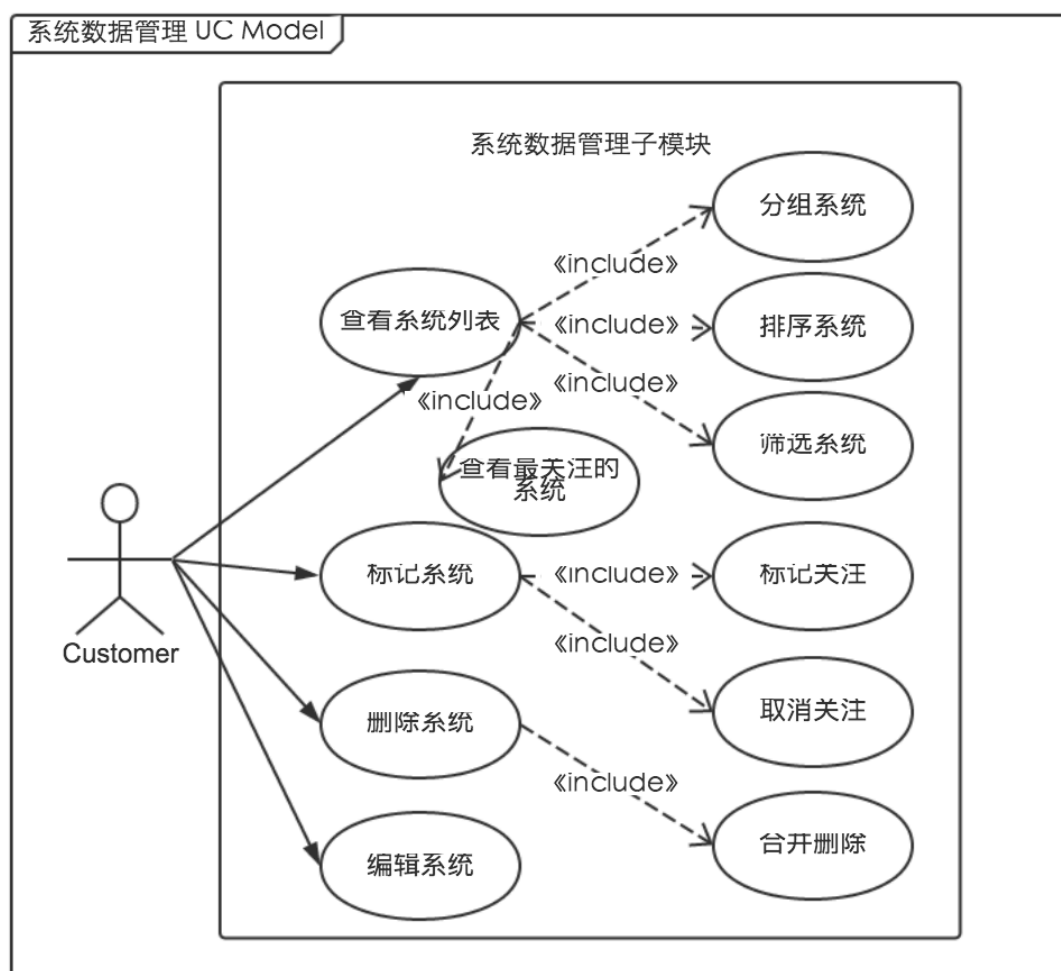


图 3.6 系统数据管理子模块用例图

用户在系统数据管理子模块能够对拥有权限管理的系统进行维护和管理。包括能够查看所有的系统列表，能对系统列表进行筛选、分组和排序，能够查看自己标记为最关注的系统列表。能够选择具体的系统展示其具体信息，能够标记或

者取消标记系统为最关注，能够点选相关系统进行合并删除，能够在具体信息页面编辑系统的信息并保存修改。

(7) Notes 和 KBA 管理

如图 3.7 所示，用户进入 Notes 和 KBA 管理子模块，可以查看 Note/KBA 列表包括查看自上次以来新建的 Note/KBA，包括查看自上次以来更新过的 Note/KBA，包括查看自己关注的 Note/KBA。可以通过关键字搜索 Note/KBA，可以根据自己关注的系统来检查 Note/KBA 与系统的相关性。可以查看具体的 Note/KBA，将其进行关注/取消关注，可以将其下载用作 SNote，可以将其以 PDF 格式打开。

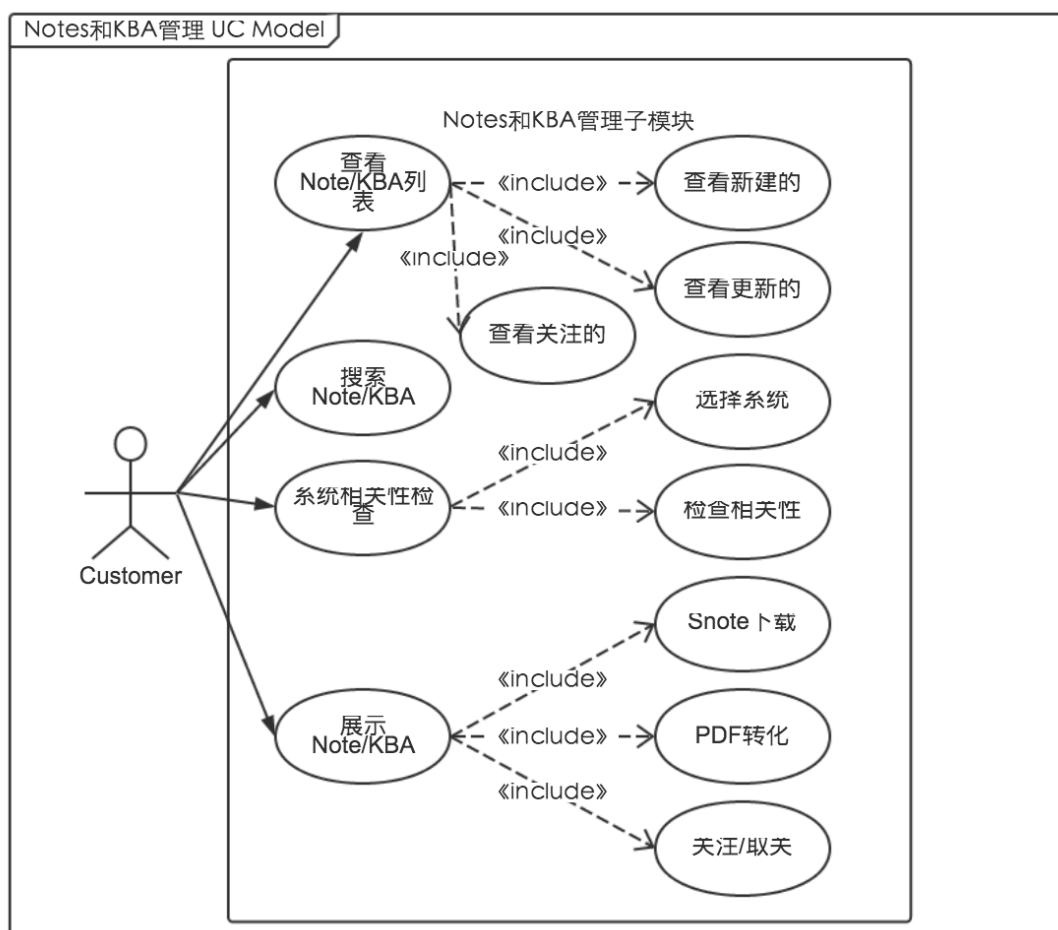


图 3.7 Notes 和 KBA 管理子模块用例图

Notes 和 KBA 是用户获得问题解决方案的主要途径，系统需要完善的 Notes 和 KBA 管理模块，才能深入落实 SWI（Support without Incident）的路线，能够极大地减轻 Support 部门的任务，让用户学会自己引导自己。

用户在该模块能够查看 Notes/KBA 列表，界面要以复合的形式展示，包括刚

刚获得更新的 Notes/KBA 列表，刚刚新建的 Notes/KBA 列表，已被标记为最关注的 Notes/KBA 列表。用户可以将重要的最关注的 Notes/KBA 放进最关注列表以便随时查看。用户能够根据关键字包括 Notes/KBA 编号、内容或标题中的关键词来进行搜索。当用户查看某 Note 或者 KBA 的具体情况时，要能够将其进行系统的相关度检查，首先选择用户权限内的某个系统，然后检查相关度，以决定是否应用。还能将 Note 或 KBA 作 Snote 下载使用，也能转化成 PDF 进行保存。

(8) 许可密钥管理

如图 3.8 所示，用户在许可密钥管理子模块中可以请求新的 license，包括选择 installation、system、detail 的相关信息，可以查看请求过的 license 记录，可以查看关注的 license 的信息并进行编辑。

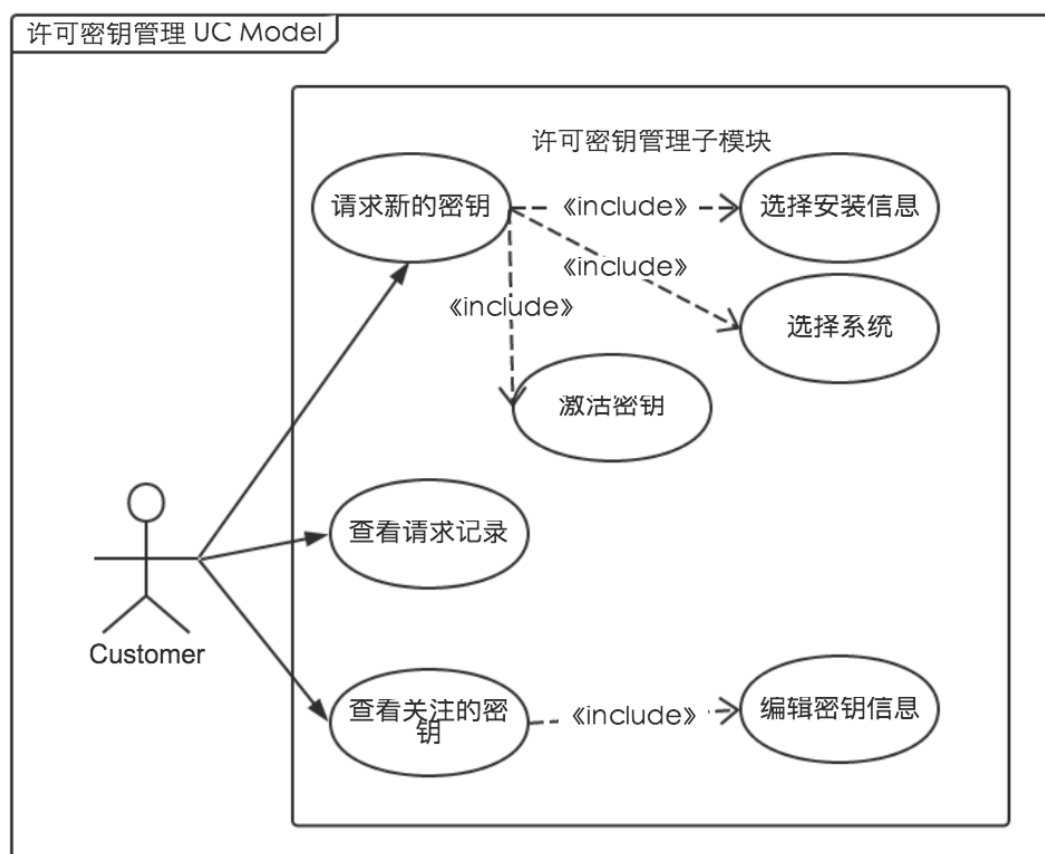


图 3.8 许可密钥管理子模块用例图

其中，用户能够请求新的密钥，请求过程中选择安装信息，选择密钥应用的系统，然后手动激活该密钥。以往的密钥请求用户要等到系统的邮件回复，如此自动激活密钥很大程度降低了时间成本，避免了等待邮件的麻烦。用户还能查看请求过的密钥记录，能够将密钥标记为最关注，并查看最关注列表，能够查看密

钥的具体信息，编辑并保存。

(9) Incident 管理

如图 3.9 所示，用户在 Incident 管理子模块中可以查看所有 Incident 或者查看我的 Incident（所有是指该用户所在的公司提交过的全部该用户能有权限挂你了的 Incident，我的是指该用户提交过的所有 Incident），可以在系统中搜索关键字寻找解决方案，如果得不到解决，就提交新的 Incident。

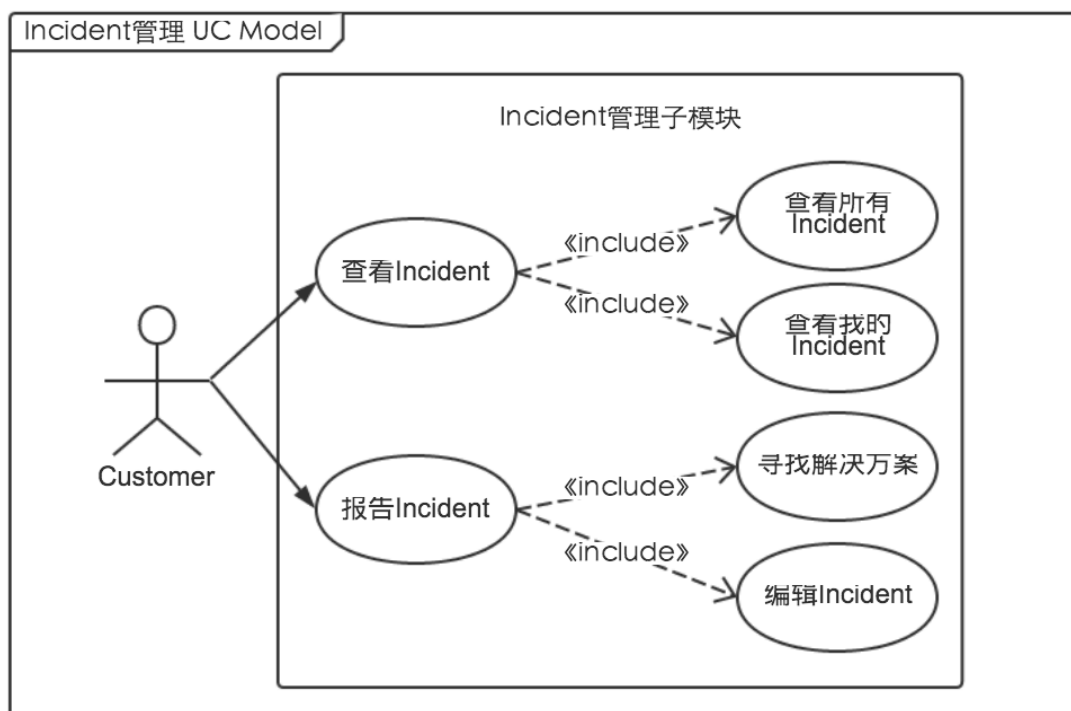


图 3.9 Incident 管理子模块用例图

其中，用户在查看 Incident 的过程中，首先进入 Incident 的 Inbox，可以分别查看 Open Incident，或者 Solution Proposed 的 Incident，或者 No Updated 的 Incident，或者 High Priority 的 Incident，或者 Draft Incident。这应该是一个复合包含 Tab 的页面，可以分别查看所有的 Incident 和我的 Incident。用户在报告 Incident 的过程中，系统要首先引导客户去自己寻早解决方案，用关键字搜索 Note 或者 KBA，查询无果，再编辑 Incident 并发送报告给 SAP Support。

3.1.2 流程需求

如图 3.10 所示，用户需要登录账号经过验证（或者通过 single sign-on）才能进入主页，进入主页后可以访问各个子模块的应用主页包括软件下载中心、许可

密钥管理中心、磁贴页面管理中心、系统数据管理中心、用户管理中心、Notes/KBA 管理中心、Incident 管理中心。也可以在主页的搜索模块进行全对象搜索，通过搜索结果可以访问系统信息、Notes/KBA、用户信息、Incident 等。

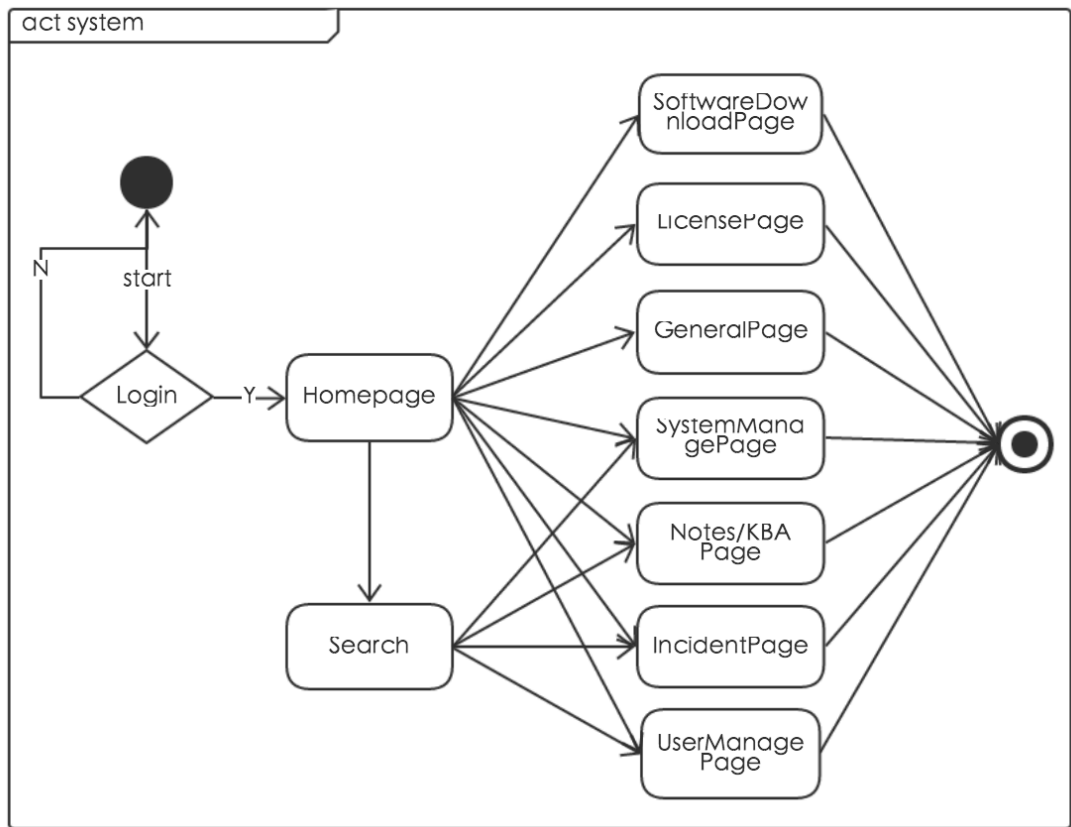


图 3.10 系统活动图

3.1.3 非功能性需求

系统的非功能性需求描述如表 3.6 所示：

表 3.6 非功能性需求表

时间特性	系统对用户操作的响应时间不大于 1 秒
	系统页面刷新载入的时间不大于 5 秒
吞吐量	20 个应用服务器 60 个并发进程/服务器
负载	预期用户数量：225000 用户次/月现在用户数量：8000 用户现在请求数：400 并发请求
灵活性	系统在不同类型机器上（能够使用 browser）均可使用
	系统在不同浏览器如 IE（10 或以后）/Chrome/Firefox/Safari 上均

	可使用。
	<p>如果使用禁用了 TLS1.0/SSL2.0/SSL3.0 的 IE 浏览器，你将不能打开客户支持系统。需要确保浏览器支持 TLS1.0 连接并且不对不安全的密码（<128bit）设限。在 IE 浏览器中，在 Tools > Internet Options > Advanced tab > Scroll 移动到 75%并且勾选所有选项。如果主页按钮和三条横线（汉堡）按钮没有显示，确保在 Tools > Internet Options > Security tab > Button Custom Level... > Scroll 到 40%，找到下载区域并且选择允许。</p>

3.2 系统总体设计

系统总体采用了 B/S 的架构模式，根据功能不同可分为三层，分别是表示层、应用层、数据存储层。如图 3.11 的系统高层架构图，其中，web browser 所展示的界面即为展示层，HANA 中的 XS Server 作为一个轻量级应用服务器向展示层提供服务 and 业务逻辑资源，HANA 的数据库作为数据存储层供业务层的数据调用和保存。

Web 表示层提供了用户的操作界面，主要用 Java Script 和 CSS 语言编写，用成熟的 SAP UI5 框架融合了 HTML5，处理应用层返回的数据，将其绑定在相关的页面控件上，生成完整的页面。在 Launchpad 的 shell 中，所有的应用都被打包成一个 component，在页面展示为一个磁贴（tile），Launchpad 作为盛放这些 component 的容器，能够根据用户的 Role 和 Preference 决定磁贴的展示与布局。使用 ADOBE CQ5 让页面能够与后台即时通讯，显示即时的数据指标。将前端的 Java Script API 全部集成在 HANA Cloud Platform，实现用户可以自由注册应用，获得接口，定制属于自己的应用界面。

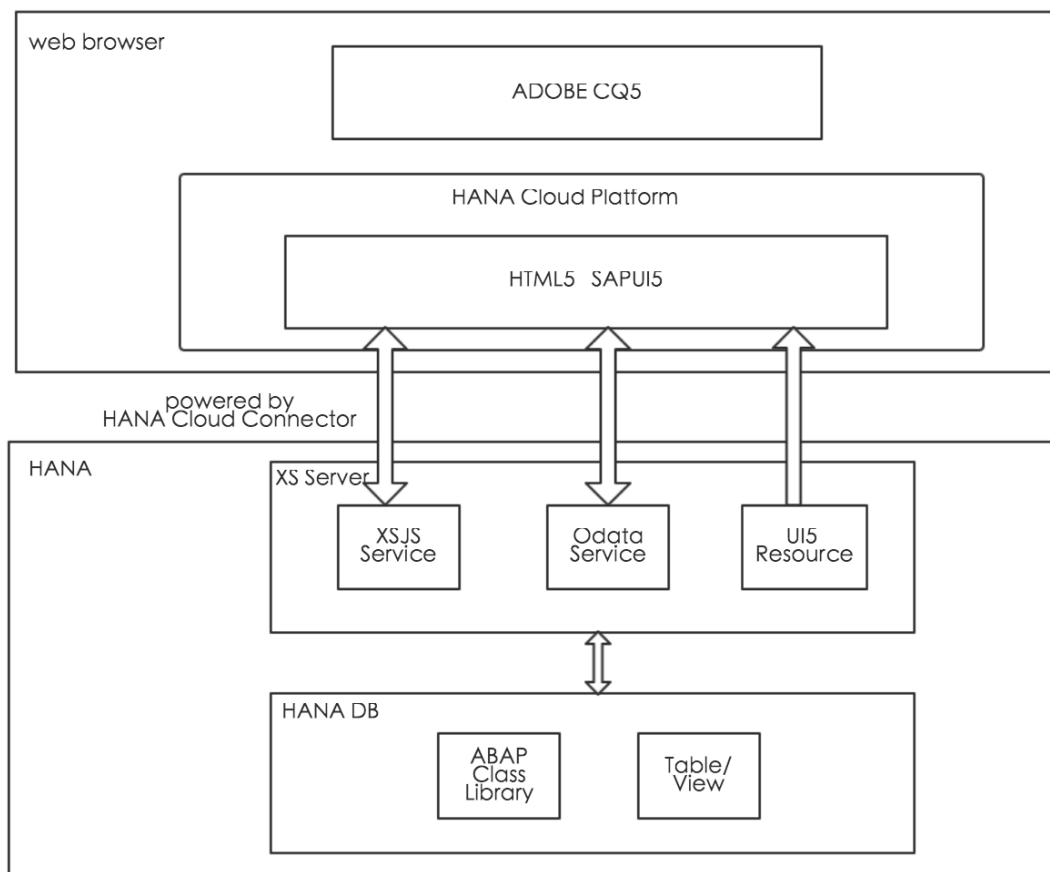


图 3.11 系统技术架构图

SAP HANA 作为应用层的应用服务器和数据存储层的数据库。在应用层，展示层的页面加载时，HTTP 向 XS Server 请求 UI5 的资源文件，系统通过 OData 或者 Ajax 协议向 XS Server 请求数据，应用层再将对数据的请求转化为对数据库的查询，从 HANA DB 内部使用 ABAP 类库里封装的接口来获取和更新 DB 里的数据，数据一般以 JSON 的方式返回给应用层，最后返回到 UI5 框架或其他 JS 代码来处理。

不言而喻，XS Server 是前端获得支撑的核心，它内置了一个 Web 应用服务器，不需要依托其他任何软件即可在 SAP HANA 的助力下实现一个 B/S 架构的应用。下面，我们对 XS Server 的内部架构实现进行分析。

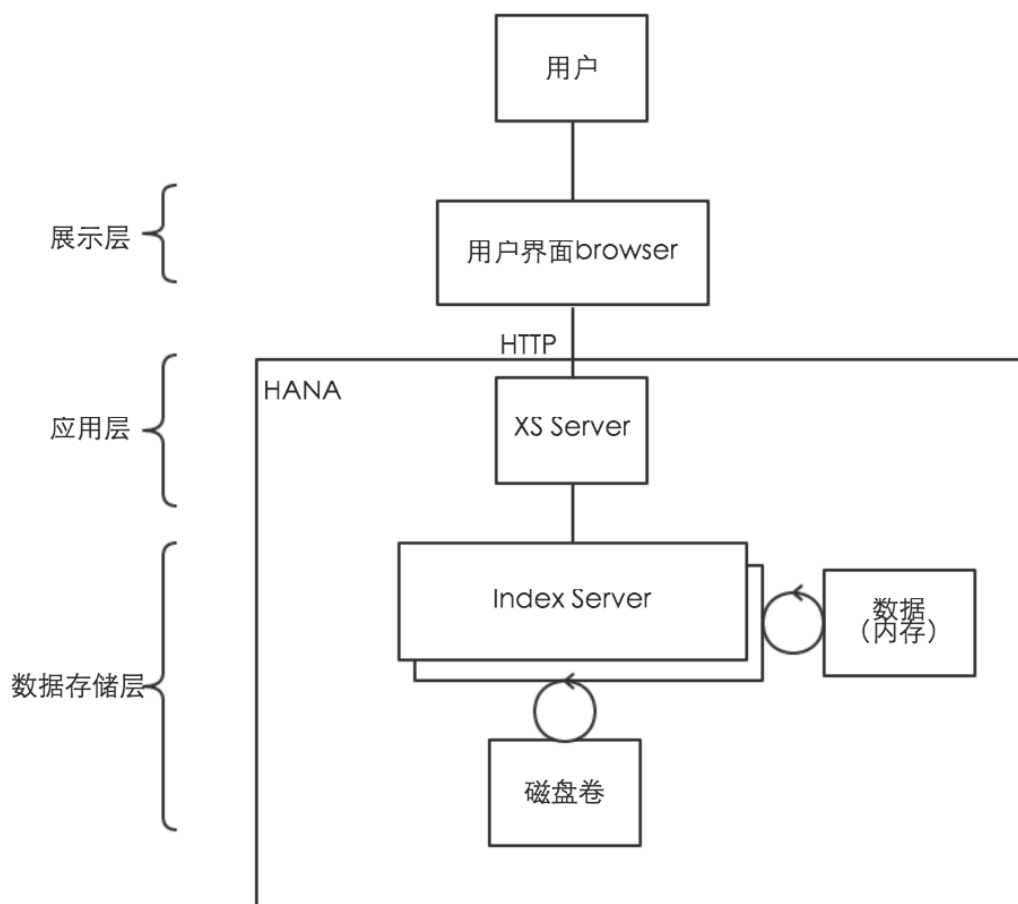


图 3.12 XS Server 示意图

如图 3.12 所示，数据存储层的 Index Server 负责处理应用执行过程中的所有数据查询及更新操作，内置 OLAP、计算引擎，可以对大量级的数据进行系统的密集型操作。应用层 XS Server 中，开发人员使用多重方式将用户和 HANA 中的业务数据真正关联起来。

3.3 本章小结

本章主要介绍了系统的需求分析和总体设计。

在需求分析中，首先根据功能划分给出了系统的八大模块，画出了整体的系统用例图，阐明了用户角色和系统的主要功能。然后按照每个子模块依次展开，画出子模块的用例图，介绍说明每个用例实现的任务。在核心搜索模块详细说明了具体用例，给出了用例列表。接着用系统活动图展现了一般用户的操作流程和具体的业务逻辑顺序。最后给出了非功能性需求列表。

在系统总体设计中，首先指出系统使用的 B/S 架构模式，划分为三个层次：展示层、应用层和数据存储层。然后具体介绍了展示层所使用的 UI 框架和 JS 接口，应用层使用的 HANA 的应用服务器 XS Server 的工作模式和数据存储层 HANA DB 如何获取和更新数据。

第四章 搜索模块的详细设计与实现

4.1 搜索模块概述

搜索模块虽然功能简单，任务单一，却是整个 Launchpad 功能的核心模块。用户在搜索模块输入关键字进行全对象搜索，可以根据搜索结果直接跳转进入结果对象包括 Note、KBA、软件、用户、远程系统、安装信息等对应的模块；用户能够获得搜索提示（包括最近的搜索记录和热门搜索）；系统能够对输入的关键字进行智能联想，自动补全和纠正，以搜索到更多符合条件的结果；系统能够检测结果与用户系统的相关度，提高相关度高的结果的优先级。

搜索模块始终位于 Shell 的 Header，不随界面的移动和更改和改变；搜索模块同其他的模块紧密地集成在一起，能够灵活的相互跳转。

4.2 搜索模块的详细设计

4.2.1 展示与应用层设计

用户在成功登录平台后，系统会自动调用 User Personalization 的 Profile（通过访问 ABAP 的 API）查询 User 被派署的 Role（Role 是由相关的 business 和 profile 的 area 决定的），根据 Role 的结果和语言的 preference 来设置界面显示的 group 和静态、动态的 tile。由此形成了个性化定制的 shell body。搜索模块的 input field 在 shell 的 header 中，不随客户 Role 和页面的转移而变化，以支持客户随时随地的检索和导航。登录后，搜索的 scope 的下拉框默认选择为 solution 的全对象搜索。

用户首先在下拉框选择 scope，再在 input field 输入关键字来进行搜索。搜索模块允许用户通过输入与 application 相关的内容来链接进入特定的 application，从而实现 shell 与 application 的交互。一般，有以下三种情况，通过搜索模块实现与 application 的交互：标准搜索，用户选择了具体的 repository，输入了具体的该 repository 下有效的关键字；提示下搜索，用户输入了关键字，系统提示合

适的选项，要避免搜索无结果的选项出现，从而提高检索效率（例如，用户输入一串数字，系统只能进行数字相关的 Incident 或者 User 的联想）；全对象搜索，用户没有选择具体的 repository，系统会给出所有相关的结果，且提示一共匹配的结果数量（用计数器统计），会有一个框架页面展示所有的符合结果。所有的交互都是以 central SearchField 这个组件为核心的，下面对这三种方式进行详细描述。

（1）Standard search

用户选择了具体的 repository，触发了系统通过 shellRouter 进行导航，这同在界面里点击进入某 tile 对应的 application 的原理一样，不同的是，前者传入了一个 searchterm 作为 parameter。应用层的 UIComponent 需要具体化一个路径，用来对 searchterm 的值进行 retrieve。这样的路径可以写作 rest-style 的形式："incidents/mysearchterm"，或者写作一个 querystring 的 parameter："incidents/?parameter1=value1 &searchtermparametername=mysearchterm"。所有参数必须被正确地声明和识别，正确地传递给 central SearchField 组件，从而正确地导航到相关的应用。由于在同一个 application 的 session 下，searchterm 和 application parameter 可能同时被修改，已被调用的 application 需要在不丢失 context 的前提下接受 search route 的调用。Context 的建立和保存需要在组件 start up 的时候就确保。对于空的 searchterm，组件可以选择不进行 application 的调用，也可以交给 Filter-Field 让 application 来决定是否在该应用下包含空的搜索值。

如图 4.1 给出了 standard search 的界面原型设计，用户在搜索过程中要能够获得系统给出的最近搜索和最热门搜索的提示。

?
close

Top Search

Hana

Sales order

Cloud platform

Recent Search

Dump

Master Data

Sap note

图 4.1 standardsearch 界面原型图

(2) Suggestion

提供 suggestion 的 service 应该在注册一个 application 的时候具体化，因为它是在用户选择了有效的 repository 和关键字后调用一个 application 的时候激活的。且提供的 suggestion 应该是同被调用的 application 内容相关的。例如：在 user management application 中建议的是 user name。使用 JSON 通过命名属性的键值对或者数组来提供内容。一项 item 可以有如下的内容格式：

表 4.1 item 内容格式一

```
{
  "items": [
    {"name": "suggestion1", "value": "eins"},
    {"name": "suggestion2", "value": "zwei"},
    {"name": "suggestion3", "value": "drei"}
  ]
}
```


表 4.2 item 内容格式二

```
[
["suggestion1", "eins"],
["suggestion2", "zwei"],
["suggestion3", "drei"]
]
```

两种item的形式供suggestion中的PathToItemAgregation来调用。如图4.2所示，给出了suggestion的界面设计，当search item全部为数字时，系统要能够捕捉并给出在user、note等范围内的suggestion，并且在相应的suggestion item边上要提示item对应的repository。

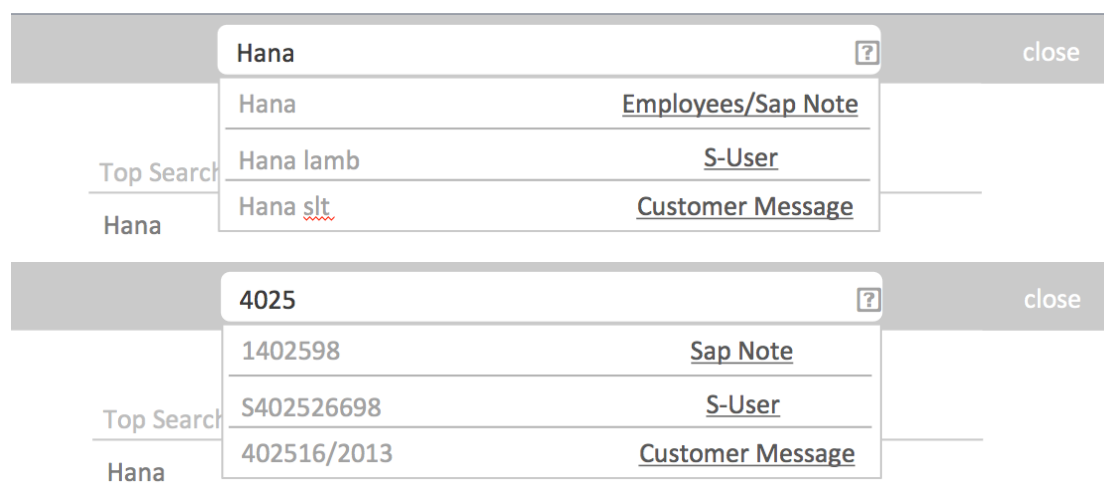


图 4.2 suggestion 界面原型图

(3) All repository search

这种类型的搜索可以被理解为一个独立的可搜索的 application，同其他的 user management 和 incident management 的应用是平行的，唯一不同的是，它是由 SearchCenter 这个组件实施的。但它并不能自己检索内容，而是作为一个 wrapper application 将来自不同应用源的结果统一在一个复合界面。结果提供者是注册过 standard search 的应用。

如图 4.3，给出了 all repository search 的界面原型设计，其中根据 repository 的不同分为若干个 tab 包括 Note、Customer、S-User、Customer Message、Installation、System，可以在搜索时来回切换。

当触发了搜索范围为All repository的search request，Search Center组件加载注册过standard search的应用的result module，将它们集成到一个result page。系统为了实现这个功能点，需要满足以下三个要求：使用JQuery的方式声明模块([jQuery.sap.declare('sap.support.<modulename>');函数getFragement返回的UI5 Control实现；函数updateModel(searchterm)处理search的调用。

值得注意的是，当一个搜索请求调用时，被调用的是应用对应的 result module 而非应用本身，即应用本身并未加载或启动，只有相关模块被激活。对于 result module 使用了 MVC 的概念，view 层可以是 xmlFragement，通过 UI5 的 Model 与其建立绑定，controller 层是 module 本身的实现。这样，其他类型的 fragement 也可以使用，通过使用 fragement 提高了代码复用率，降低了编码成本。

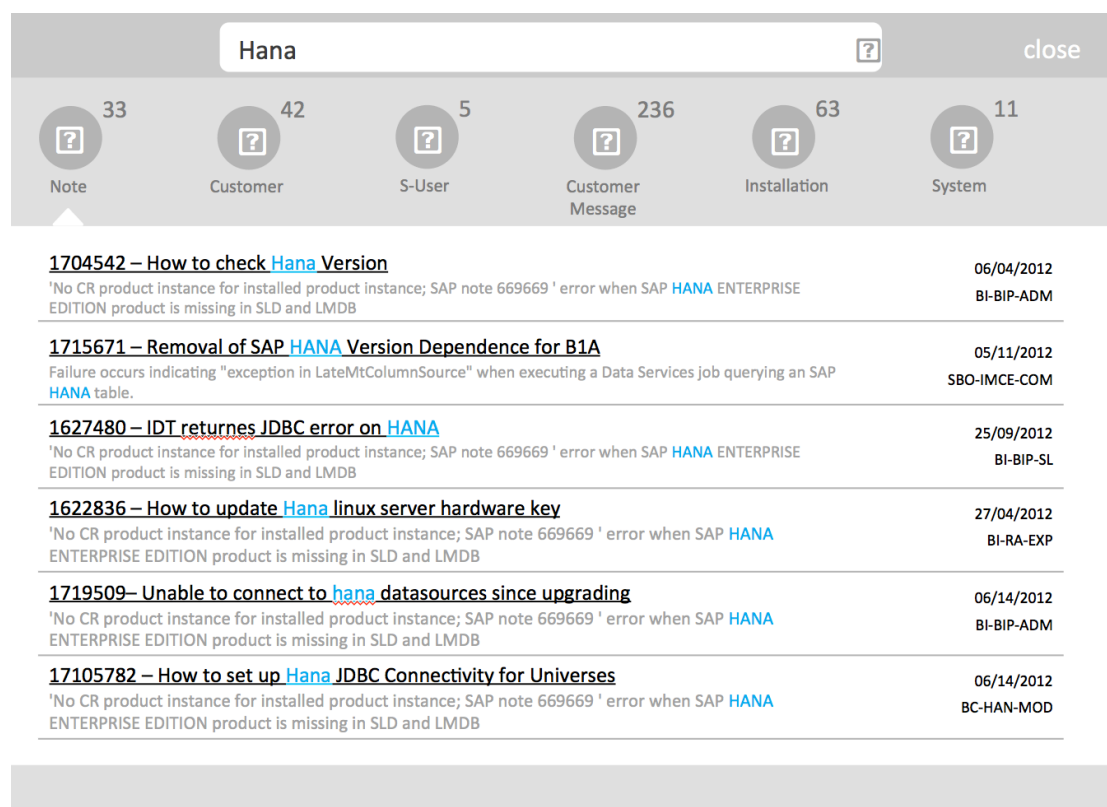


图 4.3 allrepositorysearch 界面原型图

4.2.2 数据存储层设计

系统在数据存储层使用了 SAP HANA 提供的 HANA DB 来存放有关的用户数据、产品数据和授权数据，整成 view 供 ABAP 类库调用。其在数据存储层的架构大致如下图所示：

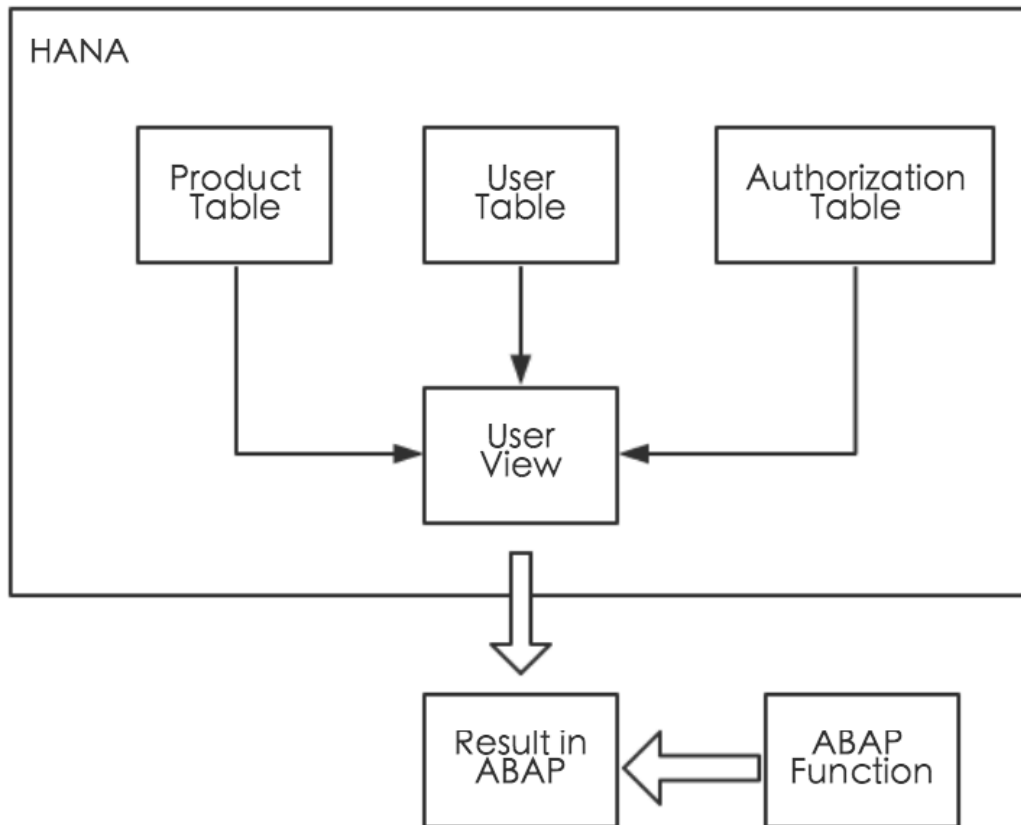


图 4.4 数据存储架构图

下面将对 HANA DB 中搜索模块涉及到的主要数据的 schema 进行说明。由于在搜索模块中对不同的 repository，所用类别的数据模型定义过程是相似的，在此为了避免重复，仅对 User/User Profile 这一个 repository 的数据模型进行详细介绍。

首先是 Attribute，Attribute 主要由三个属性构成：Name，Group 和 Value。其中 Name 是长度不超过 32 个字符的字符串（由大写字母和下划线组成）。Group 是用来形容 Name 隶属的属性，长度不超过 16 个字符，由大写字母组成。Value 是复合 JSON 定义规范的若干数值，具体定义的类别如下表 4.3 所示：

表 4.3 JSON 对 value 的定义表

STRING	<p>A <i>string</i> is a sequence of zero or more Unicode characters. A character is represented as a single character string. A string is very much like a C or Java string.</p> <p>A <i>number</i> is very much like a C or Java number, except that the octal and hexadecimal formats are not used.</p>
--------	---

NUMBER	A <i>number</i> is very much like a C or Java number, except that the octal and hexadecimal formats are not used.
OBJECT	An <i>object</i> is an unordered set of name/value pairs. An object begins with { (left brace) and ends with } (right brace). Each name is followed by: (colon) and the name/value pairs are separated by , (comma).
ARRAY	An <i>array</i> is an ordered collection of values. An array begins with [(left bracket) and ends with] (right bracket). Values are separated by , (comma).
TRUE/FALSE	

如下表 4.4 给出 Attribute 的数据库表设计：

表 4.4 Attribute 数据库表

“Attribute”	Description
NAME(PK): CHAR(32)	Attribute 的名称
GROUP:CHAR(16)	Attribute 所分派的组，例如 Role/Authorization
TYPE:CHAR(3)	Attribute 的类型，例如 STR/NUM/OBJ/ARR/TOF
DATSRG:CHAR(4)	Attribute 的数据来源
STOR_PROG:CHAR(40)	提交 Attribute 时的说明
PULL_INTVL:INT2	每次 Push 调用之间的时间差
SQL_STMT:STRING	要调用的 SQL 语句（如果是 SQL 的 Attribute）
SQL_PROC:CHAR(30)	要调用的 SQL 的程序片段名称（如果是 PROC 的 Attribute）
ABAP_FUNC:CHAR(30)	要调用的函数模块（如果是 ABAP 的 Attribute）
RULE_OP:CHAR(4)	这条 RULE 要使用的操作符，例如 NOT/AND/OR/XOR/NAND/NOR/XNOR

对于表中的属性 DATSRG，表示的 Attribute 值的来源，可以由大约四种描述。

Customized Attribute 是用户可写的需要用到 ID Service；Stored Attribute 是存储在自己数据库结构中，可读可写修改可删除，适用于 PULL/PUSH 操作；Calculated Attribute 不是存在于自己数据库结构中，只能读，可以是 SQL 语句，可以是 HANA

数据库存储的程序，可以是 ABAP 计算程序；Combined Attribute 是已经定义的 Attribute 的连结。

在上表 4.4 的基础上。给出在 ABAP Class Library 结合 Attribute 的数据库表的具体的 UserProfile 类定义。

表 4.5 USER_PROFILE 类定义

CL_USER_PROFILE	
+	ATTRIBUTES : Table of "Attribute"
-	CONSTRUCTOR(iv_user : CHAR(12) default SY-UNAME)
-	GET(iv_name : CHAR(32)) : er_data : Ref to DATA
-	FIND(iv_name : CHAR(32)) : Table of name : CHAR(32), data : Ref to DATA
-	GROUP(iv_group : CHAR(16)) : Table of name : CHAR(32), data : Ref to DATA
-	SET (iv_name : CHAR(32), iv_value : STRING)
-	ADD(iv_name : CHAR(32), iv_index : INT4 OPTIONAL, iv_value : STRING)
-	REMOVE(iv_name : CHAR(32), iv_index : INT4)
-	PUT(iv_name : CHAR(32), iv_param : CHAR(32), iv_value : STRING)
-	DELETE(iv_name : CHAR(32))

表 4.6 USER_PROFILE 类方法介绍

CL_USER_PROFILE
CONSTRUCTOR: 初始化 USER_PROFILE 对象（对于一个给定的 USER）并说明要获得该对象引用所需要的权限，对该对象进行权限管理。
GET: 在 USER 的属性集中返回特定的属性。被返回的属性必须在 ATTRIBUTE 数据库表中是存在且 active 的。在 GET 方法中的操作视属性值的 DATSRC 即数据来源所示，对于不同来源的采取不同的操作。

<p>CUST: 读取相应的数据供应来源 (IDS)</p> <p>PULL / PUSH: 读取 STORAGE 数据库表</p> <p>SQL: 执行 SQL 操作 <i>SQL_STMT</i></p> <p>PROC: 执行 SQL 程序 <i>SQL_PROC</i></p> <p>ABAP: 执行 ABAP 程序 <i>ABAP_PROC</i></p> <p>RULE: 用 RULE ENGINE 根据顺序分析 RULE</p>
<p>FIND: 通过在 IV_NAME 中指定任何通配字符串, 任意的属性会返回 FIND 方法。在属性表中的所有匹配活动元素执行 GET 方法。</p>
<p>GROUP: 在组内所有匹配的属性实行 GET 方法。</p>
<p>SET: 重置 PULL/PUSH 的属性值。</p>
<p>ADD: 在 PULL/PUSH Attribute 的 array 中添加元素。当 IV_INDEX 存在且不为 0, 所添加的元素对应于一个具体的 index。</p>
<p>REMOVE: 在 PULL/PUSH Attribute 的 array 中删除元素。</p>
<p>PUT: 重置一个 PULL/PUSH Attribute 的类型对象的元素值。</p>
<p>DELETE: 从 USER STORAGE 删除 PULL/PUSH Attribute。</p>

4.3 搜索模块的实现

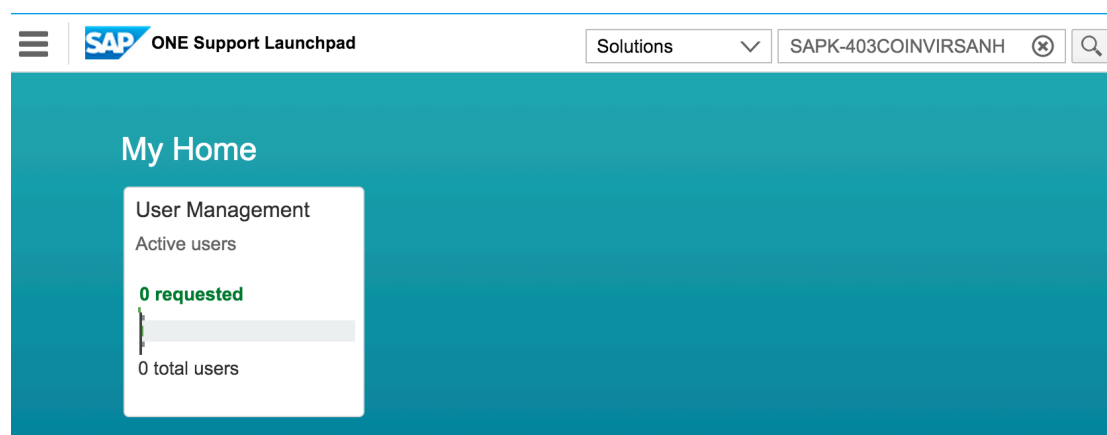


图 4.5 主页 search 模块界面

如图 4.4 所示的主页界面, 其在 shellHeader 的搜索模块的关键代码如表 4.3 所示:

表 4.7 LandingPage.js 代码

```

var bSwaReady, fnSwaReady = function() { bSwaReady = true; };

sap.ui.getCore().attachInit(function() {

// Save certain elements not to be replaced

var aSaved = jQuery.map(['#systemNews','#sessionExpiryNotification'],
function(sId) {

return jQuery(sId).get();

});

// Load component and place at content

new sap.ui.core.ComponentContainer({

name: 'sap.support.shell.component',

settings: { id: 'shell' }

}).placeAt('content');

// Prepend the saved elements to content again

jQuery.sap.delayedCall(0, this, function() {

jQuery.each(aSaved, function(iIndex, oElement) {

jQuery('#content').prepend(oElement);

});

});

// Require additional includes (if not already loaded)

jQuery.sap.require('sap.ui.core.routing.HashChanger');

// Attach to browser / SAPUI5 events, to fire WARP tracker (as soon as SWA is
ready)

fnSwaReady = function() {

// Attach to the SAPUI5 hash changer, to record navigation / searches etc.

var oHashChanger = sap.ui.core.routing.HashChanger.getInstance(),

```

```

fnHashChanged,

bInitialized = !!oHashChanger._initialized, bFirst = true;

oHashChanger.attachEvent('hashChanged',
fnHashChanged=function(oEvent) {

var sHash =
oEvent?oEvent.getParameter('newHash'):oHashChanger.getHash(),
sOldHash = oEvent?oEvent.getParameter('oldHash'):null, aHash,
sApplication;

jQuery.sap.delayedCall(0, this, function() {

// Trigger track load event on page / hash change
if(bFirst) bFirst = false; else swa.trackLoad();

// Custom search event (triggered on every search)
if((aHash=/(?:^(?searchV(?:[^\V]*V)?|^\V?solutionsV.*(?:[?&q=|VsearchV)([^\&]*)/.
exec(sHash))&&aHash.length)

swa.trackCustomEvent('search',aHash[1].substring(0,255).toLowerCase());

});

}); if(bInitialized) // in case HashChanger is already initialized, execute the
event for the first time

fnHashChanged();

// Attach to the 'load' event, to execute when the DOM is fully loaded (complete)
jQuery(window).load(function() {

// Performance / dom content load time event (triggered on page load)
jQuery.sap.log.trace("Window load event");
jQuery.sap.delayedCall(0,null,function() {

swaPerformance();

jQuery.sap.intervalCall(10*1000,null,swaResourcePerformance);

```



```
});
});
}; if(bSwaReady) // in case SWA is ready already, immediately register to the
event
fnSwaReady();
});
```

主要内容模块由 UI5 的核心部件容器完成 shell 的布局,用 UI5 的 hash changer 记录每一次的 navigation/search。使用 DOM (Document Object Model) 即时 load event 保证每一个 tile 显示的都是更新后的即时数据。

表 4.8 suggestionmanager 核心代码

```
function(h,p,q,l,d){d=sap.support.search.searchcenter.util.TermAnalyzer.cleanTerm(d);
if(this.m_currentTerm===d){var
r=sap.ui.getCore().byId("search--searchCenterInputfield");
r?r.suggest(true):null;
return
}this.m_currentTerm=d;
this.m_selectedAppKey=l;
this.m_responseItems=[];
var j=this;
var m=new sap.ui.model.json.JSONModel();
var e={q:d,responsesize:5,linguistic:"false"};
var
b=h.getMetadata().getConfig("suggestion.parameters").servicePrefix+"suggestedterms.xsjs";
if(this.m_currentTerm.length==0){b=h.getMetadata().getConfig("suggestion.parameters").servicePrefix+"rec_search.xsjs"
}m.attachRequestCompleted({term:d},function(z,t){if(t.term===j.m_currentTerm){var
```

```

v=(t.term.length>0?this.getData()[encodeURIComponent(t.term)]:this.getData()
);
var x=[];
for(var w in
v){if(t.term.length>0){x.push({group:"termsuggest",entry:v[w],category:""})
}else{x.push({icon:"sap-icon://past",group:"recentsearch",entry:v[w]})
}}if(t.term.length>0){var s=q.getData();
for(var w in
s){if(s[w].key!==j.m_selectedAppKey){x.push({group:"searchapp",entry:t.term,c
ategory:("in
"+sap.ui.getCore().getModel("i18n_searchcenter").getResourceBundle().getTe
xt(s[w].key)),target:s[w].route})
}}}var y=sap.ui.getCore().byId("SearchToolbarView").getModel("suggestions");
var u=y.getData();
if(x.length>0){j.m_responseItems=j.m_responseItems.concat(x);
y.setData({items:j.m_responseItems});

```

如表 4.4 给出了 suggestion manager 的关键代码，是对 suggestion 的实现，用户输入了有效的 search term，用 JSON model 去捕捉，然后调用 xsjs service 寻找 search term 的 xsjs 文件，然后相应地匹配，罗列出 response item 返回给展示层，出现在用户的 searchtoolbar。

如此构成搜索流程的第一步，输入 search term 并获得系统联想后的提示。然后系统对 search term 进行处理，核心代码如表 4.5 所示。

表 4.9 处理 searchcenter 核心代码

```

var A=sap.ui.getCore().byId("search--searchCenterInputfield");
if(A){A.suggest(true);
sap.ui.getCore().byId(A.getSuggestionItems()[0].getDomRef().parentNode.id).r
ender()
}}});







```

```

m.attachRequestFailed(function(){console.log("Suggestion Service not
available");
var
i=sap.ui.getCore().getModel("i18n_searchcenter").getResourceBundle().getTex
t("ServiceConnectFailedMSG");
sap.ui.getCore().byId("SearchToolbarView").getModel("suggestions").setData({
items:[{entry:i,category:""}]})
});
m.loadData(b,e,true,"GET");
var a=q.getData();
var o=[];
for(var f in
a){if(a[f].suggestions&&a[f].suggestions.serviceUrl&&I===a[f].key){var g=new
sap.ui.model.json.JSONModel();
var n=new
sap.ui.model.Binding(g,"",g.getContext(a[f].suggestions.pathToItemsAggregati
on+(a[f].suggestions.searchTermAsItemAggregationElementName?j.m_current
Term:"")));
n.attachChange({app:a[f]},function(t,s){var
v=t.getSource().getContext().getObject();






```

这样在 all repository search 完成时，进入如下的界面：不同 category 的结果可以用 tab 来回切换显示，对每一个 category 给出的 list 能够根据 component、type、priority、category、release date 进行 filter 和 sorting 的操作。

 131463	 469330	 17940	 7029	 10362	 0
SAP Notes	SCN Forums	SCN Wiki	Support Portal	SF Community	Sybase Solved Ca...

Component (All)Type (All)Priority (All)Category (All)Release Date (Last 3 Years)

42 New since yesterdayEnglish (131463)

<input type="checkbox"/>	2101278 - Release information and other system texts 	28.11.2014 IS-U SAP Note
<input type="checkbox"/>	2138530 - FKKINV: Corrections to the system documentation 	06.03.2015 XX-PROJ-FI-CA SAP Note
<input type="checkbox"/>	2019928 - DDF AVDH : System text editing for SP2 	14.08.2014 CA-DDF-INT-AVD-HDR SAP Note
<input type="checkbox"/>	1860844 - Corrections in system documentation required 	30.03.2016 SD-BF-PR SAP Note
<input type="checkbox"/>	1791570 - dbacockpit: System information -> large tables 	06.08.2013 BC-DB-LCA SAP Note

4.6Allrepositorysearch 界面

如果需要将某 application 加入到能够搜索的 repository 的下拉框中，则要在注册该 application 的时候说明。

如表，静态的参数可以通过提供 queryPathParameterValue 参数设置，比如设置 queryStringObjectName,或者 queryStringParameterName 参数，而且我们可以对一些 querypath 和一些 querystring 参数进行组合来设置静态参数。

表 4.10 注册一个应用示例代码

```
{
  "text" : <Display text> that will be shown in the searchfield selection e.g.
  "Users",
  "route" : <name of the application route for searching> e.g.
  "userManagement",
  "queryPathParameterName" : <parameter name receiving the search term>
  e.g. "searchterm"
}
```

从 shell 通过 search 调用交互到 application 的过程，当进入到了具体的某个 application 后，search field 依然在 header 中，然而默认的 repository 得到了改变，如图 4.3，开发人员添加了 searchRouteName 这个参数，参数值是 incidentFilterBySearchTerm，这个参数的意思就是，当一个应用从 Launchpad Shell 中启动的时候，搜索输入框的下拉菜单就会被预选为当前正在运行的应用。

表 4.12 默认 repository 示例代码

```
{
  "name" : "incident",
  "pattern" : "incident",
  "title" : "Incidents",
  "component" : "sap.support.incidentform.incidentlist",
  "feedbackTopic" : "INCIDENT",
  "registerModulePath" : {
    "moduleName" : "sap.support.incidentform.incidentlist",
    "urlPrefix" : "/applications/incidentform/WebContent/incidentlist/"    },
  "fallbackUrl" :
    "https://apps.support.sap.com/sap/bc/bsp/svt/ngmsg/inbox.htm",
  "searchRouteName" : "incidentFilterBySearchTerm"
}
```

当对 result list 进行访问时,调用 Controller 的 onEntryClick 函数,传递 oEvent 参数,这样当点击不同的 application 对象时进入特定的 application 模块。

表 4.13 xmlFragement 示例代码

```
jQuery.sap.declare('sap.support.search.searchdemoapp.views.resultsModule');
(function(){
  sap.support.search.searchdemoapp.views.resultsModule =
  sap.support.search.searchdemoapp.views.resultsModule
  || {
    _model:null,
    getFragment : function(){
      this._model = this._model || new sap.ui.model.json.JSONModel();
      var oFragment =
      sap.ui.xmlfragment("sap.support.search.searchdemoapp.views.results", this);
      oFragment.setModel(this._model);
    }
  }
})
```

```

return oFragment;
},
updateModel : function(searchTerm){
this._model.loadData(url, params, false, "GET", false, true);
},
onEntryClick : function(oEvent){
alert(oEvent.getParameter("listItem").getProperty("title"));
}
}
})();

```

4.4 本章小结

本章主要对系统的搜索模块的详细设计与实现进行了描述。

在描述过程中，先对模块进行简要概述，然后阐述详细设计的过程，最后将关键代码和实现难点的代码示例进行分析。

第五章 总结与展望

5.1 总结

本文描述了 ONE Support Launchpad 实现搜索模块的具体过程。从整个系统的需求分析和概要设计，到搜索模块的详细设计与实现。

从项目背景出发，解释了为什么 SAP 要推陈出新，将原始以信息交互为核心的 Support Portal 创新成以用户为核心的 Launchpad。分析了国内外 ERP 软件公司的现状以及它们提供支持服务的情况，从而知晓了客户在得到 ERP 软件支持体验上的痛点，也成为了 Launchpad 要解决的主要问题。

然后本文介绍了 Launchpad 涉及的主要知识和技术手段，包括 SAP 的 HANA 技术和获得红点设计大奖的 Fiori 架构。还有用于版本控制的工具 Git。这些技术点是完成 Launchpad 设计的基石。

接着本文阐述了需求和概要设计的整个过程，包括系统的功能模块划分，和主要功能模块的用例设计、非功能需求和流程需求。

针对本文的核心搜索模块进行了详细设计的说明和重要代码演示。

开发团队严格按照前期制定的时间表进行开发，在合理的版本控制管理下如期构建了整个系统。

5.2 展望

目前 ONE Support Launchpad 系统在 SuccessFactor 业务板块已经上线，从用户反馈的情况来看，还有一些不足的地方需要改进，例如，在系统响应时间和搜索时间上的不能满足客户的要求。还有一些小功能没有完全实现，例如 tile 的直接拖拽，这些在未来的开发中都将被实现。

此外，随着产品线的增加，新的辅助支持应用也将诞生，系统将对这些新的应用提供支持和接口，这样，所有的支持体验都能在同一平台进行。

参考文献

- [1] *SAP Overview*, <https://en.wikipedia.org/wiki/SAP>.
- [2] *SAP Support*, <https://support.sap.com/home.html>.
- [3] *ERP Enterprise Resource Planning*,
https://en.wikipedia.org/wiki/Enterprise_resource_planning.
- [4] 黄晓俭, 《我看用友 U9 PK SAP/Oracle》, CSDN, 2009
- [5] *ORACLE Overview*, <http://www.oracle.com/index.html>.
- [6] *Yongyou Product Overview*,
<http://www.yonyou.com/index.html?t=0.8617088438477367>.
- [7] *Fiori Overview*, <http://help.sap.com/fiori>.
- [8] *SAP Fiori Mobile Apps and Tools for Apache Cordova*,
<https://blogs.msdn.microsoft.com/visualstudio/2015/11/17/sap-fiori-mobile-apps-and-tools-for-apache-cordova/>.
- [9] *Target Multiple Platforms with one code base free and open source*,
<http://cordova.apache.org/>.
- [10] *SAP UI5*, <http://www.cnblogs.com/qianmarv/p/4681777.html>.
- [11] *SAP HANA Building Handbook*, <http://scn.sap.com/docs/DOC-30826>.
- [12] *Github Official Guidance*, <https://github.com/>.

致谢

在此论文完成之际，我想对所有帮助、指导过我的人表达最诚挚的感谢。

首先，我要感谢我的导师贝佳，感谢您的谆谆教导。从论文的选题、提纲的拟定，到撰写、修改以及最后的定稿，感谢您不厌其烦的指导和帮助，在我遇到困难的时候积极引导，遇到疑惑的时候同我一起商量克服，您无微不至的关怀是我的巨大动力。

其次，我要感谢南京大学软件学院，感谢学院四年的栽培，让我度过了一段充实又美好的大学时光。感谢学院提供给我良好的学习环境、一流的学习体验。感谢学院的老师们辛勤的付出。

我还要感谢给我提供实习机会的 SAP 上海研究院，作为一个风气严谨的外企，让我真正开阔了视野，接触到最前沿的技术和知识。感谢公司的每一位领导、同事对我的关怀和培养。

最后，感谢所有关心、帮助过我的同学，祝你们都学业顺利，幸福安康！