



西安电子科技大学

国家大学生创新创业训练计划

项目总结报告

项目级别：	<input checked="" type="checkbox"/> 国家级 <input type="checkbox"/> 省级
项目编号	S202210701007X
项目名称：	基于区块链的文创设计策展社区和版权确权流通平台
项目类别：	<input type="checkbox"/> 创新训练 <input checked="" type="checkbox"/> 创业训练 <input type="checkbox"/> 创业实践
作品形式：	<input type="checkbox"/> 硬件 <input checked="" type="checkbox"/> 软件 <input type="checkbox"/> 研究报告 <input checked="" type="checkbox"/> 计划书 <input type="checkbox"/> 其它
项目组成员：	洪铭锋、刘睿康、王渝普、王立文、刘博涵
指导教师：	崔江涛
项目申请单位：	西安电子科技大学
项目起止时间：	2022 年 9 月 至 2023 年 6 月

二〇二三年六月

西安电子科技大学教务处制

西安电子科技大学国家大学生创新创业训练计划项目

原创性声明

本人郑重声明：所呈交的项目总结报告以及所完成的作品实物等相关成果，是本人和队友独立进行研究工作所取得的成果。除文中已经注明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的作品成果，不侵犯任何第三方的知识产权或其他权利。本人完全意识到本声明的法律结果由本人承担。

参赛队员签名：井铭峰 刘睿康 王渝普 王立文 刘博强

日期：23 年 5 月 18 日

项目指导教师审核签名：

日期： 年 月 日

总结报告参照提纲及要求:

- 一、原创性声明**
 - 二、中英文对照题目**
 - 三、中英文对照摘要（中文摘要限 200 字）和关键字**
 - 四、项目概述**
 - 1) 项目成员基本情况（人数、院系、专业、年级）、指导教师基本情况（职称、专业领域）
 - 2) 项目的选题背景、目的与意义
 - 3) 项目实施过程的人员工作分配和完成情况
 - 4) 项目实施过程收获和体会
 - 五、项目预期成果完成情况和创新点**
 - 六、项目说明（实验方法设计及方案、系统测试方案、数据分析处理、原因分析说明、解决办法、最终实现功能、特色等）**
 - 七、项目总结**
 - 八、附录（含源代码和程序清单，扩展应用系统电路图，应用资料与参考文献目录等）**
 - 九、附件（包括经费使用说明、项目发表论文、专利等成果支撑材料、项目进展日志、相关照片说明资料等）**
-
- 另外，项目总结报告用 A4 纸激光打印（宋体，小 4 号字，单倍行距，默认页边距，页码标在右下角），一式叁份，含电子版一套。
 - 结题验收作品简介要求中英文对照，中文 500 字以内，用 A4 纸激光打印（宋体，小 4 号字，单倍行距），一式叁份，含电子版一套，作品简介和总结报告分开装订，一同递交。
 - 总结报告相关章节、图表等内容格式要求请参考本科生毕业设计（论文）工作手册。

二、 中英文对照题目

基于区块链的文创设计策展社区和版权确权流通平台

Blockchain-based Cultural and Creative Design Curatorial Community and Copyright confirmation Circulation Platform

三、 中英文对照摘要

项目为基于区块链的文创设计策展社区和版权确权流通平台。项目依托一个社交化的文创交流社区，该社交化平台搭建完成后，文创产品的设计者可以上传自己的原创设计进行展示，社区用户可以对每个文创的设计进行打分，根据社交化平台上面的热度、评分、分享量等数据，设计一个对产权的估价算法。与此同时，项目还希望实现基于区块链的文创产品设计版权的确权流通系统，实现版权的数字化、上链并生成 NFT、交易支付等功能。估价的结果会作为版权数字化流通时的参考交易价格。

The project is a blockchain-based design curation community and copyright distribution platform. The project relies on a social creative exchange community. After the social platform is built, designers of creative products can upload their original designs for display, and community users can score each creative design. At the same time, the project also hopes to realise a blockchain-based copyright distribution system for the design of cultural and creative products, realising the functions of digitising copyright, uploading it onto the chain and generating NFT, and making payments for transactions. The result of the valuation will be used as a reference transaction price when the copyright is digitised and circulated.

四、项目概述

1) 项目成员基本情况

姓名	学号	学院	专业	年级
洪铭锋	20069100191	经济与管理学院	市场营销	2020
刘睿康	20009200584	计算机科学与技术学院	计算机科学与技术	2020
王渝普	20012100012	通信工程学院	通信工程	2020
王立文	20009200043	计算机科学与技术学院	软件工程	2020
刘博涵	20009201455	计算机科学与技术学院	计算机科学与技术	2020

指导教师

姓名	崔江涛	职称	教授
研究领域	数据库管理与内核技术、复杂数据查询与索引、数据安全与隐私保护、区块链应用技术等	所属学院	计算机科学与技术学院

2) 项目的选题背景、目的与意义

本项目实施的目的和意义在于激励文创设计创作，推动其创作在社区展示与优化，同时拉近文创的消费者与创作者之间的距离，促进文创产品设计供需对接。项目的实施将促进文创产业的发展，加快经济创新的同时促进文化传播，在促进经济创新、推动文化创新、加深交流合作、提升国际形象等方面有着重要的意义。

该项目需要开发相应的应用程序和网站，以支持用户上传文创设计作品、进行版权确权和交易等操作。这需要计算机科学与技术领域的专业知识，如数据库设计、网络编程、用户界面设计等。还需要开发和维护复杂的软件系统，包括前端网站、后端服务和区块链节点等。因此，软件工程的最佳实践，如系统设计、需求分析、软件测试和代码管理等，都是必要的。此外，该项目基于区块链技术实现版权确权和交易功能，因此需要深入了解区块链技术的特点和应用。例如，如何设计智能合约、如何实现分布式账本、如何确保数据安全等问题都需要考虑。

实现这个项目，大学生团队需要掌握计算机科学与技术、软件工程和区块链技术等多方面的知识。在计算机科学与技术方面，团队需要学习如何开发应用程序和网站，以支持用户上传作品、进行版权确权和交易等操作。在软件

工程方面，团队需要掌握系统设计、需求分析、软件测试和代码管理等最佳实践，以确保软件系统的高质量和可靠性。在区块链技术方面，团队需要深入了解智能合约、分布式账本和数据安全等问题，以保证版权确权和交易的可信度和安全性。这个项目也是一个创新创业的机会，可以团队带来实践和创新的机会，提高我们的技能和创业能力。

3) 项目实施过程的人员工作分配和完成情况

洪铭锋：研究市场需求，了解文创设计师和版权交易市场的需求和趋势。收集用户需求和反馈，提出产品需求和功能规划。根据市场需求和用户反馈，制定产品的特性和用户界面设计。确定产品的价值主张和定位，制定产品营销策略。确认产品的功能和特性，维护产品路线图，并制定产品优先级。作为负责人组织和协调团队开发工作，包括前端开发、后端开发、智能合约开发和测试等。确定项目的范围、进度和质量目标，制定项目计划和工作分配。监督项目进展，并及时调整计划和资源分配，确保项目按时交付。跟进团队成员的工作进展，解决团队成员之间的协作问题。管理项目的风险和变更，制定应对方案和风险管理计划。（完成情况：撰写产品需求文档、用户操作手册、申报软件著作权）

刘睿康：确定后端技术架构，在多个技术选项中选择最适合项目的技术，例如使用哪种编程语言、使用哪种数据库管理系统等。设计和开发数据库，确保其能够支持高并发和高可靠性的数据存储和访问。设计和开发系统的 API，确保其能够支持各种类型的请求和响应。（完成情况：撰写用户操作手册、申报软件著作权）

王渝普：设计和开发智能合约，确保其能够实现版权确权和交易等功能。此外，还需要考虑智能合约的安全性和可靠性。设计和开发分布式账本，确保其能够实现版权交易的可追溯性和安全性。（完成情况：链上部署、智能合约部署）

王立文：收集大量的文创设计市场数据，包括作品的类型、风格、时代背景等方面的信息，以及市场交易数据、价格等信息。在多个算法选项中选择最

适合项目的算法，例如基于机器学习、基于统计分析等算法。基于收集到的市场数据和算法设计，算法工程师需要设计和开发估价算法。算法需要考虑作品的特征和市场需求，计算出作品的价值。（完成情况：算法开发）

刘博涵：设计和开发用户界面，确保其能够支持用户上传文创设计作品、进行版权确权和交易等操作。此外，还需要考虑用户界面的易用性和用户体验。设计和开发用户交互功能，确保其能够支持用户上传作品、进行版权确权和交易等操作。此外，还需要考虑用户交互功能的易用性。（完成情况：前端开发）

3) 项目实施过程收获和体会

在这个项目实施的过程中，团队成员掌握多个技术领域的知识和技能，具备了开发和实现类似区块链项目的能力。团队成员将会学习如何开展创业，团队成员需要协同工作，共同完成项目的开发和实施。这使我们体会到团队协作的重要性，学习如何有效地沟通、分工、协作，以实现项目的目标。

首先，在技术方面，团队成员掌握了区块链、智能合约、分布式账本、数据库、API、前端、后端等多个技术领域的知识和技能，以实现项目的目标。在这个过程中，他们需要不断学习和探索最新的技术，选择最适合项目的技术架构，并不断优化和改进技术方案，以提高系统的可靠性和稳定性。在实践和创新方面，团队成员需要不断地实践和创新，以寻找最佳解决方案。在这个过程中，他们需要不断优化和改进系统和算法方案，以提高项目的竞争力和用户体验。

其次，在创业方面，团队成员学习了如何开展创业，包括市场调研、产品设计、团队协作、资源管理、风险管理、营销等方面的知识和经验。在这个过程中，他们需要深入了解用户需求和市场趋势，设计出具有竞争力的产品和服务，并不断优化和改进创业方案，以提高项目的成功率和可持续性。

最后，团队协作方面，团队成员需要协同工作，共同完成项目的开发和实施。在这个过程中，他们需要建立起高效的沟通和协作机制，分工明确，任务清晰，以提高工作效率和项目成功率。团队成员需要与其他技术人员、文创设计师、投资人等进行交流，学习他们的经验、观点和建议。在这个过程中，他们需要建立起良好的人际关系和合作伙伴关系，不断学习和探索最新的技术和市场动态，以提高项目的成功率和可持续性。

五、项目预期成果完成情况和创新点

完成平台搭建，提供源码编译方式，云端 API 接口；

(已完成，源码、软件需求文档、用户操作手册见附件，平台已申请软件著作权 2 项)

进行模拟企业运营，在项目开发结束后完成详尽的可行性分析和小规模的试点测试，并作出报告；

(已完成，项目小规模试点已获技术应用证明与入驻协议，项目先后获得：西咸新区沣西新城陕西微软创新中心、西安高新技术产业风险投资有限责任公司、西安曲江文化产业投资（集团）有限公司、远诺科技前研科投（西安）科技有限公司、黄陵工业园区陕西省高新技术产业开发区共 5 家企业/孵化器/产业园区的投资意向，以及上海零数科技有限公司、华为技术有限公司共 2 家企业的合作对接意向，商业计划书见附件)

参加创新创业相关大赛并取得成果。

(已完成，项目先后获得：西安市大学生创新创业成长计划优秀创新创业项目、第 8 届中国国际“互联网+”大学生创新创业大赛陕西省银奖、第 21 届微软 Imagine Cup 复赛赛区优秀奖、第 13 届全国大学生电子商务“创新、创意及创业”挑战赛陕西省一等奖、陕西省第 14 届工业工程改善创意竞赛校一等奖（未完赛，已推省赛），相关证明材料见附件)

项目创新点如下：

该项目采用区块链技术来实现版权交易和确权，这是一种全新的技术手段，具有去中心化、不可篡改、安全可靠等优点。该平台利用智能合约实现自动化处理、快速结算和透明管理，使得版权交易和确权变得更加高效和便捷。与此同时采用区块链技术实现数据的去中心化存储和管理，同时保护用户的隐私数据。所有数据都被加密存储，只有用户本人持有私钥才能访问和控制数据，从而保障了用户的数据隐私和安全。

该项目将文创设计师和投资人、收藏家等文创产业链的各个环节紧密连接起来，实现了版权交易、文创设计策展、投资和收藏等多种功能，从而促进了文创产业的良性发展。该平台还提供了个性化的服务和定制化的用户体验，满足了用户对个性化和定制化服务的需求。不仅实现了文创设计作品的版权交易和确权，还实现了投资和收藏的功能。投资人可以通过该平台获得高品质的文创设计作品，同时可以获得作品增值的收益。收藏家可以通过该平台方便地收藏

和管理自己喜爱的文创设计作品，同时可以获得作品的增值和流通价值。还将社交功能与版权交易和文创设计策展功能相结合，实现了社区化的文创设计平台。用户可以通过该平台展示自己的作品、与其他用户交流和合作、获得更多的文创设计资源和信息等，从而促进了文创设计行业的交流和合作。

六、项目说明

在 PC 端采用 B/S 模式并进行 Wap 适配，地址可展示在后期在微信公众平台上开通的服务号中。同时在移动端开发小程序。由于考虑个体用户使用移动端的便捷性，在完成项目主体搭建后应立即筹备基于 iOS 和 Android 系统的移动端 App 的开发。

项目所开发的平台的用户主要有以下几类：

序号	角色名	角色描述
1	游客	未登录的用户，只能进行浏览，所有数据只会存入 cookies
2	普通用户	普通的注册用户，可进行信息浏览、搜索、评论、收藏、个人主页操作，也可参与文创产品的众筹或购买文创产品等操作
3	创作者	上传了作品的注册用户，包括机构用户和个人创作者两类用户，可进行作品发布、收藏、分享等操作。
4	文创企业	需进行企业认证，主要为文创 IP 或设计的版权购买者，可进行版权购买、授权、转让等操作。

项目根据 KANO 模型分析了用户需求。KANO 模型是一种常用的用户需求分析工具，其主要思想是将用户需求分为必须满足的需求、期望满足的需求和引起惊喜的需求三个层次。在基于区块链的文创设计策展社区和版权确权流通平台中，根据 KANO 模型分析用户需求，具体如下：

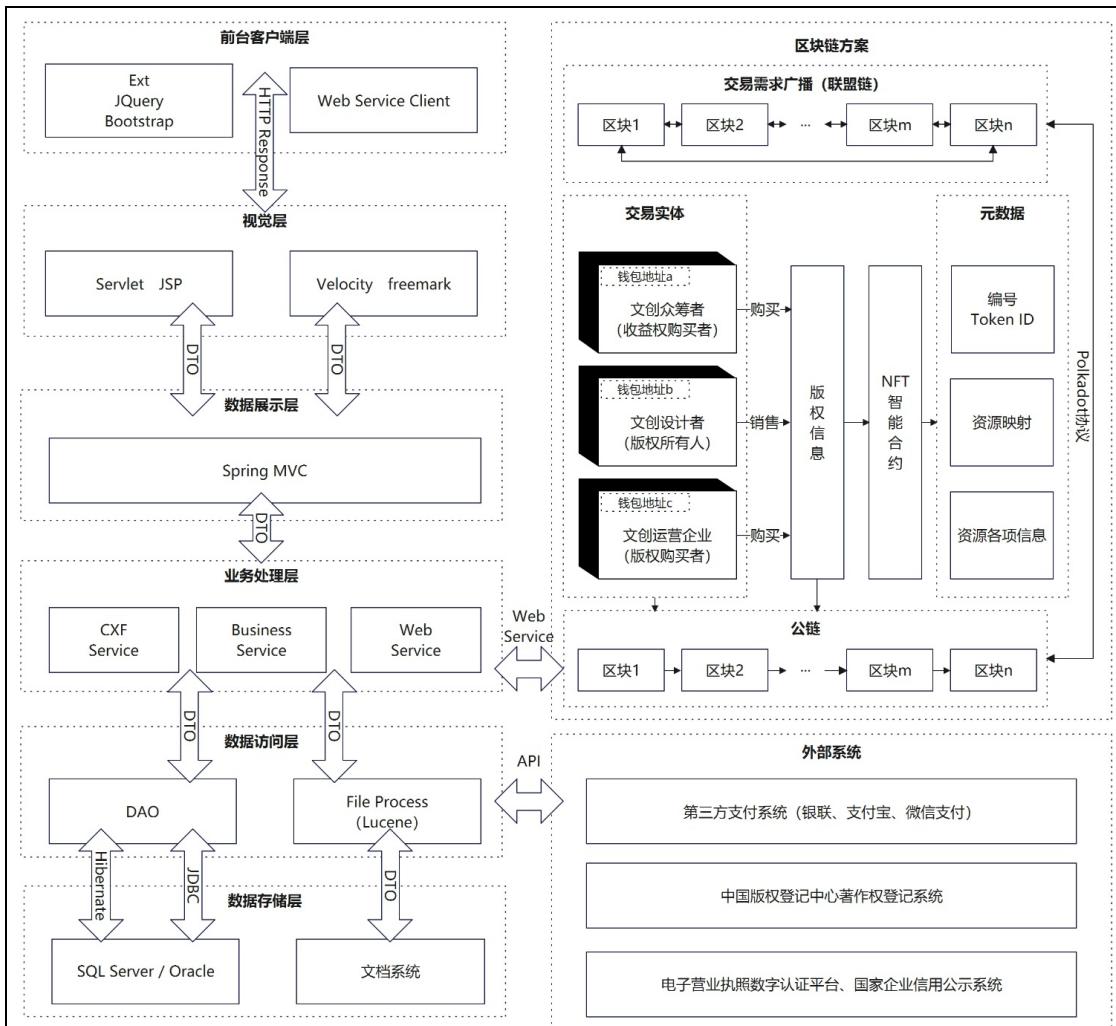
序号	需求名称	需求角色	需求描述	需求类型
1	浏览	所有用户	浏览正在展出的文创设计创意	基本型
2	评价	登录用户	对文创设计创意进行点赞、关注、点评	基本型
3	关注	登录用户	关注自己喜欢的标签、作者，接收相关推送	基本型
4	发布	创作者用户	发布设计理念、想法或小样等，发布到策展社区	基本型
5	确权	创作者用户	将一项设计的著作权（或其部分邻接权）打包，上链铸币并生成 NFT	期望型
6	流通	创作者用户、文创企业	版权的出售方与购买方以 NFT 代币转移的形式完成版权流通，并进行生成转让合同	期望型
7	商品上下	文创企业	在平台内嵌的电商上下架文创产品	期望型

	架			
8	商品购买	登录用户	在平台内嵌的电商购买文创产品	期望型
9	估值	创作者用户、文创企业	根据浏览量等数据，对改文创设计的价格给出大致参考	兴奋型
10	文创众筹	创作者用户、登录用户	对自己喜欢的文创产品以参与众筹的形式表示支持	兴奋型

项目完成了运营所依赖的平台的开发。项目首先完成文创社区的搭建，搭建该项目所开发的平台为 B/S 三层结构，它的结构从下至上分别为：数据访问层、业务逻辑层（又或成为领域层）、表示层。而平台实现的功能见下表：

序号	功能名称	功能需求标识	开发优先级	简要解释
1	平台门户	L1	高	用户操作的入口。
2	用户管理	L2	高	是本系统用户进行后续操作所需的前置子系，负责收集和管理用户的注册信息、区分个人用户与机构用户。
3	设计展示	L3	高	是本系统的核芯子系统，负责将用户所发表的文创设计展示在该子系统的用户界面。所有用户均可在该子系统中对展示的文创创意与文创设计进行打分和评论。
4	版权确权与版权估值	L4	中	提供设计者的版权确权服务，同时根据设计展示模块产生的数据进行估价。
5	合作撮合与版权流通	L5	中	提供创作版权流转、使用权授权等服务。同时运营、设计与生产可在此达成合作。
6	文创电商平台	L6	中	提供文创产品的售卖。
7	创意市集	L7	低	所有用户可在本系统内进行任意创意的分享，设计者可在此获得创作灵感与创作创意。
8	众筹管理	L8	低	用户可在本系统内发起设计与生产的众筹。众筹模式包括捐赠众筹与预售众筹。
9	正版检测与盗版举报	L9	低	根据版权确权流通和授权的信息，向用户提供正版检测与盗版举报服务

项目的技术架构如下：



商业模式方面，提供可信交易与防伪方案的 B 端服务，提供交易平台、提供 IP 供需对接与 IP 咨询服务、提供防伪方案、提供交易纠纷取证服务。在交易前环节，提供 APP 付费会员，会员权限包括 IP 推荐系统的“IP 调性”分析权限与“受众画像”分析权限，定价为 68 元/月，后期数据完善后将采取按月使用次数分梯度定价，并开放单一权限购买；在交易中后环节，向交易双方收取交易佣金，交易佣金为双方各收取成交额 17%，出现交易纠纷后链上数据取证服务为 280 元/次，申请者支付，防伪卡制作费为 1.2 元/件，含数字藏品支持。本阶段向各大应用商店提交上架申请，向社会公众开发平台注册，进行项目实际运营阶段；完成 B 端版权交易功能的全部测试，获取相关资质，著作权认证登记接入，正式入驻前期已完成洽谈的孵化器；开发防欺诈算法、估价算法开发，完善平台功能，比并完善资质，稳固市场地位。

七、项目总结

基于区块链的文创设计社区和版权交易平台项目的实施过程中，我们遇到了很多挑战和机遇，也获得了很多经验和收获。

首先，我们在技术方面采用了区块链、智能合约、分布式账本、数据库、API、前端、后端等多个技术领域的知识和技能，实现了版权交易和确权的安全、快速、便捷。在这个过程中，我们不断学习和探索最新的技术，选择最适合项目的技术架构，并不断优化和改进技术方案，以提高系统的可靠性和稳定性。

其次，我们在创业方面学习了市场调研、产品设计、团队协作、资源管理、风险管理、营销等方面的知识和经验。我们深入了解用户需求和市场趋势，设计出具有竞争力的产品和服务，并不断优化和改进创业方案，以提高项目的成功率和可持续性。

第三，在团队协作方面，我们建立起高效的沟通和协作机制，分工明确，任务清晰，以提高工作效率和项目成功率。我们通过协同工作，共同完成了项目的开发和实施。

第四，在价值链分析与产业调研方面，我们深入了解文创设计行业的发展趋势和特点，设计出支持版权保护和价值实现的系统和算法方案，以实现文创设计师的利益最大化。

第五，在交流学习方面，我们与其他技术人员、文创设计师、投资人等进行交流，学习他们的经验、观点和建议。我们建立起良好的人际关系和合作伙伴关系，不断学习和探索最新的技术和市场动态，以提高项目的成功率和可持续性。

第六，在实践和创新方面，我们不断地实践和创新，以验证和改进我们的想法和方案。我们积极探索新的业务领域和模式，借鉴和应用最新的技术和方法，以推动项目的发展和创新。

总的来说，基于区块链的文创设计社区和版权交易平台项目是一个具有挑战性和机遇的项目。在这个过程中，我们不断学习和探索，不断优化和改进，以实现项目的成功和可持续性。我们相信，这个项目将为文创设计行业的发展和创新提供新的思路和途径。

八、附录（含源代码和程序清单，扩展应用系统电路图，应用资料与参考文献目录等）

附录清单如下：

- 1、软件开发需求文档
- 2、商业计划书
- 3、软件源代码
- 4、用户操作手册
- 5、软件著作权
- 6、相关获奖证明
- 7、相关应用证明与协议

附件 1：

国家大学生创新创业训练计划项目经费使用情况表

项目编号:

项目名称	基于区块链的文创设计策展社区和版权确权流通平台	项目负责人	洪铭锋	联系方式	15332330887
所在学院	经济与管理学院		指导教师	崔江涛	
材料费 (器件注明 名称型号, 可 另附页)	明细	单价 (元)	数量	总金额 (元)	用途
	腾讯云轻量应用服务器(1000GB 流量包 (MED4-60GB-1000GB), 2 核 CPU、4GB 内存 (MED4-60GB-1000GB), 36 个月)	540	1	540	平台服务器
其他费用	创赛圈-创赛材料费用	98	1	98	参赛
	《IP 授权商业化: 从入门到精通》电子 工业出版社	43.12	1	43.12	项目参考
	交通费	642.2	1	642.2	项目外出调研
总计:					
1323.32 元					

基于区块链的文创设计策展社区和版权确权流通平台

软件开发需求文档

文件标识:

文件状态:	文件编号:	V1.0-SRS-2
【 】正在编辑	文档作者:	洪铭锋
【 】正在修改	创建日期:	2022年5月1日
【√】正式发布	修改日期:	2022年5月11日

修订记录:

更新时间	编号	变更内容	变更人	备注
2022.5.1	V1.0-SRS-1	创建本文档	洪铭锋	初始版本
2022.5.11	V1.0-SRS-2	添加业务架构图, 细化功能描述	洪铭锋	当前版本

一、引言

1.1. 编写目的

本节描述文档的目的是定义软件总体要求，作为用户和软件开发人员之间相互了解的基础；提供性能要求、初步设计和对用户影响的信息，作为软件人员进行软件结构设计和编码的基础；作为软件总体测试的依据。

本阶段已在系统的商业模式与业务构想的基础上，对项目的具体开发做需求描述。在下一阶段的详细设计中，开发者可参考此文档，在概要设计的基础上，对系统进行详细设计。在以后的软件测试以及软件维护阶段也可参考此说明书，以便于了解在概要设计过程中所完成的各模块设计结构，或在修改时找出在本阶段设计的不足或错误。

1.2. 相关文档

以下列出在概要设计过程中所使用到的有关资料：

- [1] 区块链的文创设计策展社区和版权确权流通平台 • 商业计划书, 2021.12.12.

1.3. 参考资料

- [1] 业务架构 • 应用架构 • 数据架构实战/温显著.-北京：电子工业出版社, 2021.4
- [2] 软件开发与测试文档编写指南/刘文红等编著.-北京：清华大学出版社, 2020.12
- [3] 软件设计与体系结构/董威, 文艳军, 陈振邦编著.-2 版-北京：高等教育出版社, 2017.12

二、综合描述

1、项目概述

项目为基于区块链的文创设计策展社区和版权确权流通平台。项目依托一个社交化的文创交流社区，该社交化平台搭建完成后，文创产品的设计者可以上传自己的原创设计进行展示，社区用户可以对每个文创的设计进行打分，根据社交化平台上面的热度、评分、分享量



等数据，设计一个对产权的估价算法。与此同时，项目还希望实现基于区块链的文创产品设计版权的确权流通系统，实现版权的数字化、上链并生成 NFT、交易支付等功能。估价的结果会作为版权数字化流通时的参考交易价格。

2、用户角色

序号	角色名	角色描述
1	游客	未登录的用户，只能进行浏览，所有数据只会存入 cookies
2	普通用户	普通的注册用户，可进行信息浏览、搜索、评论、收藏、个人主页管理、打赏等操作，也可参与文创产品的众筹或购买文创产品等操作
3	创作者	上传了作品的注册用户，包括机构用户和个人创作者两类用户，可进行作品管理等操作。
4	文创企业	需进行企业认证，主要为文创 IP 或设计的版权购买者，可进行版权购买、产品上架等。

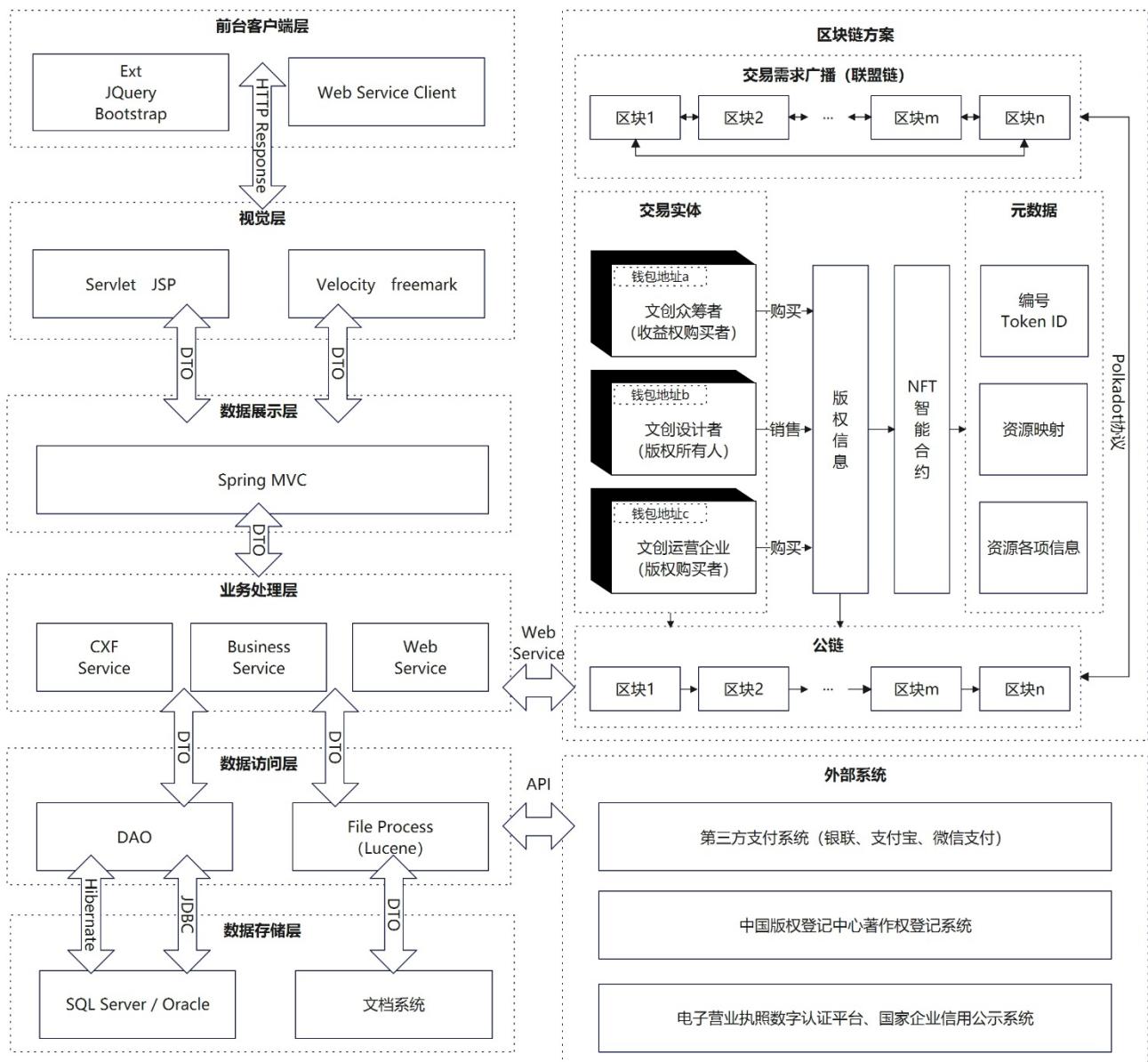
3、需求描述

序号	需求名称	需求角色	需求描述	需求类型
1	浏览	所有用户	浏览正在展出的文创设计创意	基本型
2	评价	登录用户	对文创设计创意进行点赞、关注、点评	基本型
3	关注	登录用户	关注自己喜欢的标签、作者，接收相关推送	基本型
4	发布	创作者用户	发布设计理念、想法或小样等，发布到策展社区	基本型
5	确权	创作者用户	将一项设计的著作权（或其部分邻接权）打包，上链铸币并生成 NFT	期望型
6	流通	创作者用户、文创企业	版权的出售方与购买方以 NFT 代币转移的形式完成版权流通，并进行生成转让合同	期望型
7	商品上下架	文创企业	在平台内嵌的电商上下架文创产品	期望型
8	商品购买	登录用户	在平台内嵌的电商购买文创产品	期望型

9	估值	创作者用户、文创企业	根据浏览量等数据,对改文创设计的价格给出大致参考	兴奋型
10	文创众筹	创作者用户、登录用户	对自己喜欢的文创产品以参与众筹的形式表示支持	兴奋型

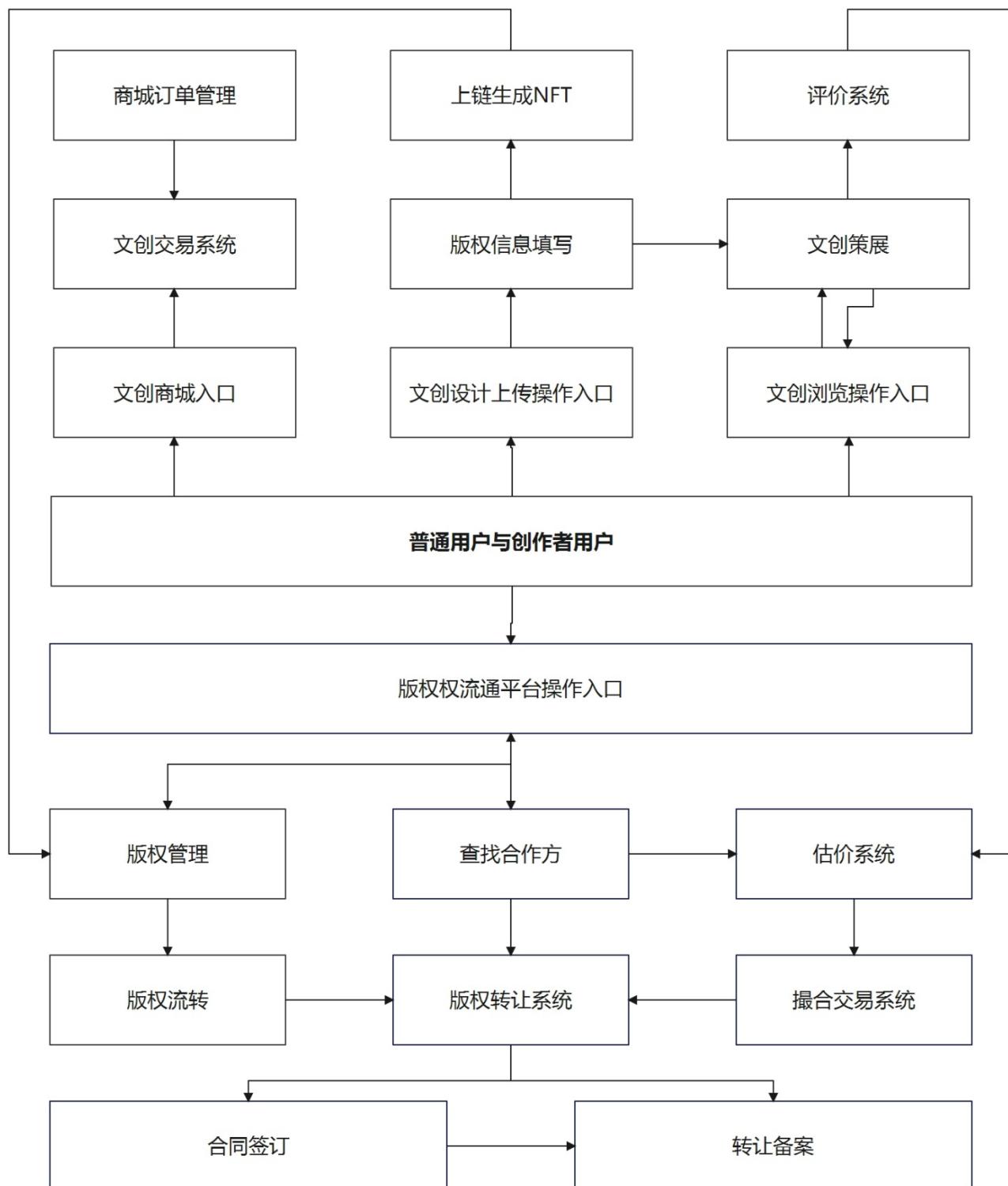
三、开发需求

1、技术架构

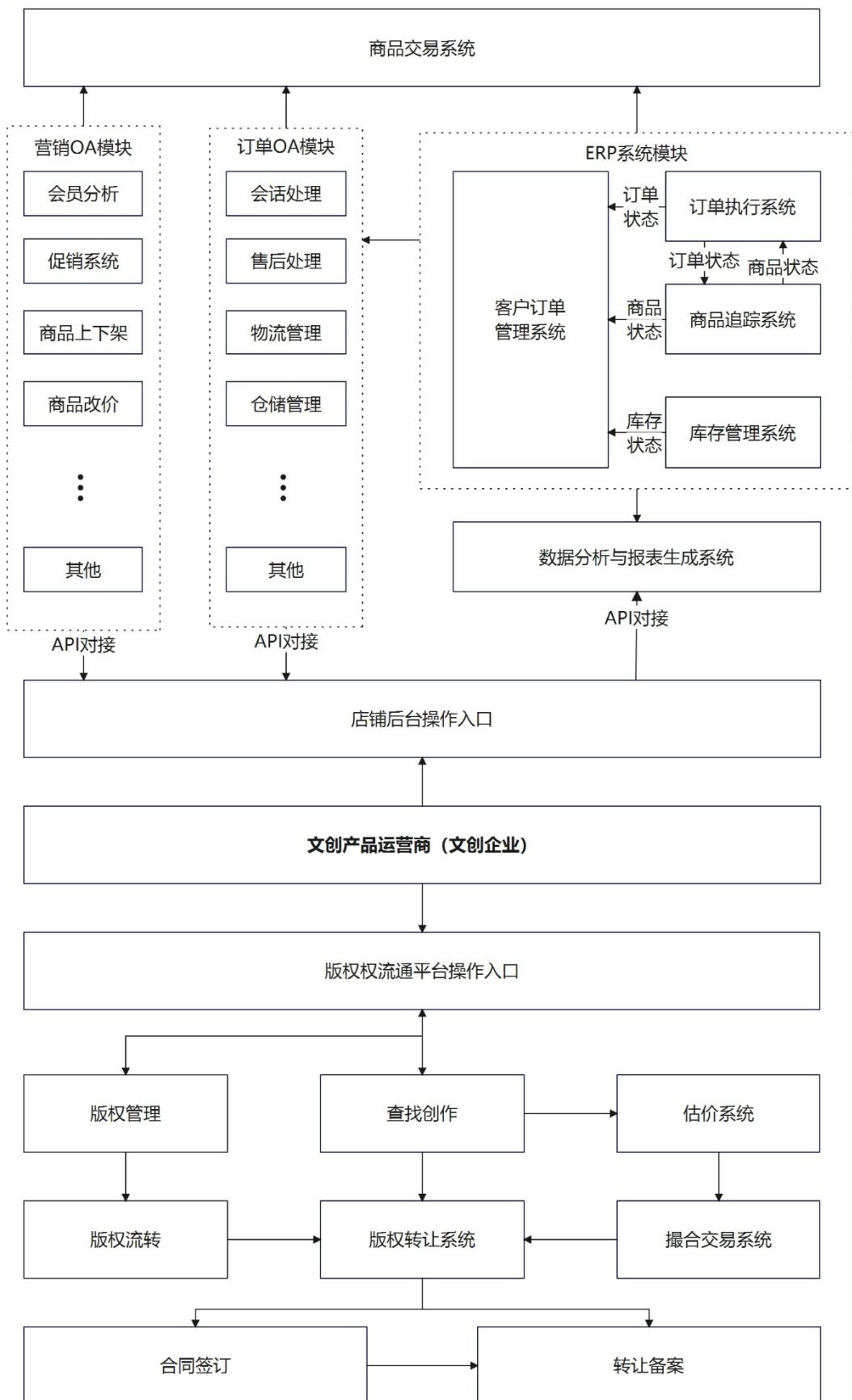


2、业务架构

个人用户（普通用户与创作者用户）业务架构



机构用户（文创企业）业务架构





3、功能清单

序号	功能名称	功能需求标识	开发优先级	简要解释
1	平台门户	L1	高	用户操作的入口。
2	用户管理	L2	高	是本系统用户进行后续操作所需的前置子系，负责收集和管理用户的注册信息。包含注册和登录（①注册：快速注册，完整注册，推荐注册等；②登陆：主站 ID 登录和 open-id 登录）、订单中心，服务中心、信息管理（包含基本信息，收款账户信息，收货地址等）。
3	平台导航	L3	中	是平台门户界面展示的推送内容的来源，包含频道导航和其他导航。频道导航由页头的导航标签页（如绘画、数字作品、文具等，具体展示的标签可由用户自定义）构成，其他导航有广告位、推荐位、当前热议、相关商品、曾经访问等构成。当前热议要求不人为编辑，曾经访问商品的数据来源为本 cookies 曾经访问过的商品。
4	站内搜索	L4	高	搜索规则由搜索范围（如名称、类型等）构成。搜索提示分为热门词和联想词，热门词将自动列出热门搜索，联想词将在输入搜索时在下拉菜单中显示相关的词语。搜索的结果页，包含搜索的成功页和搜索的失败页。搜索失败将显示缺省页，而是跳转到相似推荐导航。搜索结果页将按照一定的排列规则进行展示。
5	文创列表	L5	高	以卡片流（Web）信息流瀑（移动端）形式展示文创设计列表
6	发布入口	L6	高	用户发布自己文创设计的入口
7	设计展示	L7	高	将用户所发表的文创设计展示在该子系统的用户界面。所有登录用户均可在该子系统中对

				展示的文创创意与文创设计进行打分和评论。
8	版权确权	L8	高	将版权信息、设计小样等上链生成 NFT，以便后续进行流通。
9	版权流转	L9	高	提供创作版权流转、使用权授权等服务。同时运营、设计与生产可在此达成合作。
10	文创电商平台	L10	中	提供文创产品的售卖。
11	版权管理	L11	高	所有用户可在本系统内管理自己所拥有的版权，包括经版权登记后上链确权的原始取得版权与流转取得的版权。
12	众筹管理	L12	低	用户可在本系统内发起设计与生产的众筹。众筹模式包括捐赠众筹与预售众筹。
13	后台门户	L13	高	用户操作的入口。
14	客户关系管理系统	L14	高（外包）	即 CRM 管理，可使用 SaaS 套装软件，在其基础上进行定制化开发。需包含以下内容：会员管理（会员信息的增删改查）、用户关怀（生日祝福等）、定向营销（用户细分、根据用户数据进行交叉销售和推广销售）、客户管理（客户事件管理、购买评论回复、商品咨询回复、价格举报处理、账户信息管理、退换货等售后管理、投诉建议管理、客户黑名单等）、呼叫中心、统计报表浏览与下载等。
15	订单管理系统	L15	高（外包）	可使用 SaaS 套装软件，在其基础上进行定制化开发。需包含以下内容：订单录入（平台订单、外部订单接入、赠品等无金额订单录入）、订单审核、RMA 申请单与处理单等。
16	库存管理系统	L16	高（外包）	可使用 SaaS 套装软件，在其基础上进行定制化开发。需包含以下内容：入库管理、库存管理（库位管理、批次管理、条码打印、盈亏等）、出库管理（抓货管理、分拣管理、出



				库核检、指派运单、配送数据交换)、票据管理(自动发票、三联单)。
17	采购管理	L17	低(外包)	可使用 SaaS 套装软件，在其基础上进行定制化开发。需包含以下内容：供应商管理(供应商信息管理、供应商合同/发票管理)、采购单管理(PO 单和负 PO 单管理)、库存管理、缺货登记等。
18	财务管理系统	L18	低(外包)	可使用 SaaS 套装软件，在其基础上进行定制化开发。需包含以下内容：供应商结算(货款结算等)、渠道结算(CPA/CPS/CPC/CPM 结算)、配送结算(正向/负向物流快递结算、业务费用结算、保险费结算)、内部结算(奖金结算、内部费用报销、备用金管理)、佣金管理等。
19	报表管理系统	L19	低(外包)	可使用 SaaS 套装软件，在其基础上进行定制化开发。需包含以下内容：搜索报表、销售报表、财务报表、客服报表、仓储报表、物流报表。
20	WA 系统	L20	低	UE 分析、渠道分析、目标分析等。
21	系统设置与运维人员入口	L21	低	Session 会话过期时间设置、最大登录失败次数设置、购物车最大数量设置、未支付自动取消订单时间设置、权限设置、操作日志、帮助中心、运维入口等。

4、非功能性需求

软件结构

在 PC 端采用 B/S 模式并进行 Wap 适配，地址可展示在后期在微信公众平台上开通的服务号中。同时计划在移动端开发小程序。由于考虑个体用户使用移动端的便捷性，在完成项

目主体搭建后应立即筹备基于 iOS 和 Android 系统的移动端 App 的开发。

软硬件要求

开发者需考虑到用户群体的软硬件性能可能严重低于主流软硬件配置，故需做到较大的向下兼容，建议 PC 端配置如下：

项目	要求
处理器	1GHz 或者更高。
最小内存	运行完整功能需 128MB 的内存空间。由于部分数据存储在本地，因而在复杂查询的场景下对内存的需求量较高。
可用硬盘空间	至少 16MB 用于安装应用程序包与元数据存储。预留 10%以上的磁盘剩余空间用于数据存储。
网络	由于应用数据以上云数据为主，50 兆以上以太网或等速率无线网络（建议），完全支持 TCP/IP 协议。
系统	建议开发者在开发时应兼容 Windows XP 及以上的版本，同时适配当前最新版本的 Windows 系统。
开发工具	为保证软件的兼容性，开发者应选择比较通用的开发工具的较新版本进行开发。
本地 DBMS	Oracle 11g 以上版本或 SQL Server 2008 以上版本。
其他	CSS 样式和 JavaScript 脚本，要兼容 IE、Safari、火狐、谷歌等主流浏览器。

移动端本阶段暂不开发，故不详列其配置要求。

用户界面需求

需求名称	详细要求
首页显示	网站发布后首先浏览到的页面是服务门户。
页面风格	由前端和美工自行设计 UI。
页面字体显示	以清晰显示为原则，以美观为次要原则，建议以黑色为主。



性能需求

主要质量属性	详细要求
正确性	确保各项资料数据的准确有效，禁止数据遗漏，重复，丢失。
健壮性	数据异常捕获，灾难性恢复。
可靠性	数据校验，人机稽核，平衡检查。
易用性	操作简单，符合用户工作习惯
清晰性	流程清晰易记，分类管理
安全性	错误提示，数据验证
可扩展性	不同平台之间数据共享
兼容性	自适应各种系统环境
可移植性	自由选择不同类型的数据库

安全及保密要求

支持常见网络攻击类型的防护，包括防爬、防暴力破解、防注入、防跨站脚本攻击等。

敏感信息传输及保存需加密。

需进行访问控制及权限控制。

支持系统数据及操作日志审查。

敏感信息接口调用需要安全认证。

需要对当前功能需求中涉及到的每个接口及服务器进行监控，若发生故障需及时发送短信提醒。

备份与恢复要求

当系统出现故障时，需要在部门要求的时间范围内让系统能正常运行。

日志要求

要求系统对旅客的每一步操作都要有系统日志记录；

要求系统对每一个接口调用的情况都有系统日志记录；（含调用接口时，结果返回所耗时间、接口返回的结果，接口接入参数，接口返回的地址）

```
package com.hmkcode.android.signin;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        // present.
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
package com.hmkcode.android;
import java.util.List;
import com.hmkcode.android.model.Book;
import com.hmkcode.android.sqlite.MySQLiteHelper;
import android.os.Bundle;
import android.app.Activity;
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        MySQLiteHelper db = new MySQLiteHelper(this);

        /**
         * CRUD Operations
         */
        // add Books
        db.addBook(new Book("Android Application Development Cookbook", "Wei
Meng Lee"));
        db.addBook(new Book("Android Programming: The Big Nerd Ranch Guide",
"Bill Phillips and Brian Hardy"));
        db.addBook(new Book("Learn Android App Development", "Wallace
```

```
Jackson"));
```

```
// get all books
List<Book> list = db.getAllBooks();
```

```
// delete one book
db.deleteBook(list.get(0));
```

```
// get all books
db.getAllBooks();
```

```
}
```

```
}
```

```
package com.hmkcode.android;
import android.os.Bundle;
import android.app.Activity;
import android.content.Intent;
import android.view.Menu;
import android.view.View;
import android.view.View.OnClickListener;
import android.widget.Button;
public class MainActivity extends Activity implements OnClickListener {
```

```
    Button btnStartAnotherActivity;
```

```
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        btnStartAnotherActivity = (Button)
```

```
findViewById(R.id.btnStartAnotherActivity);
    btnStartAnotherActivity.setOnClickListener(this);
}
@Override
public void onClick(View view) {
    Intent intent = new Intent(this, AnotherActivity.class);
    // calling an activity using <intent-filter> action name
    // Intent intent = new Intent("com.hmkcode.android.ANOTHER_ACTIVITY");
    startActivity(intent);
}
}
package com.hmkcode;
import android.support.design.widget.TabLayout;
import android.support.v4.view.ViewPager;
import android.os.Bundle;
import android.support.v7.app.AppCompatActivity;
import com.hmkcode.adapters.MyFragmentPagerAdapter;
public class MainActivity extends AppCompatActivity {
    MyFragmentPagerAdapter myFragmentPagerAdapter;
    ViewPager viewPager;
    TabLayout tabLayout;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        viewPager = (ViewPager) findViewById(R.id.viewPager);
        tabLayout = (TabLayout) findViewById(R.id.tabLayout);
        setPagerAdapter();
        setTabLayout();
    }
    private void setPagerAdapter() {
        myFragmentPagerAdapter = new
```

```
MyFragmentPagerAdapter (getSupportFragmentManager ());
    viewPager.setAdapter (myFragmentPagerAdapter);
}
private void setTabLayout () {
    tabLayout.setupWithViewPager (viewPager);
    tabLayout.getTabAt (0).setText ("First");
    tabLayout.getTabAt (1).setText ("Second");
}
}
package com.hmkcode.android;
public class Model {
    private int icon;
    private String title;
    private String counter;
    private boolean isGroupHeader = false;
    public Model (String title) {
        this (-1, title, null);
        isGroupHeader = true;
    }
    public Model (int icon, String title, String counter) {
        super ();
        this .icon = icon;
        this .title = title;
        this .counter = counter;
    }
    public int getIcon () {
        return icon;
    }
    public void setIcon (int icon) {
        this .icon = icon;
    }
    public String getTitle () {
        return title;
    }
    public void setTitle (String title) {
        this .title = title;
    }
    public String getCounter () {
        return counter;
    }
    public void setCounter (String counter) {
        this .counter = counter;
    }
    public boolean isGroupHeader () {
        return isGroupHeader;
    }
    public void setGroupHeader (boolean isGroupHeader) {
        this .isGroupHeader = isGroupHeader;
    }
}
```

```
package com.hmkcode.android;
import java.util.ArrayList;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.TextView;
public class MyAdapter extends ArrayAdapter<Item> {
    private final Context context;
    private final ArrayList<Item> itemsArrayList;
    public MyAdapter (Context context, ArrayList<Item> itemsArrayList) {
        super (context, R.layout.row, itemsArrayList);
        this.context = context;
        this.itemsArrayList = itemsArrayList;
    }
    @Override
    public View getView (int position, View convertView, ViewGroup parent) {
        // 1. Create inflater
        LayoutInflater inflater = (LayoutInflater) context
            .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

        // 2. Get rowView from inflater
        View rowView = inflater.inflate(R.layout.row, parent, false);

        // 3. Get the two text view from the rowView
        TextView labelView = (TextView) rowView.findViewById(R.id.label);
        TextView valueView = (TextView) rowView.findViewById(R.id.value);

        // 4. Set the text for textView
        labelView.setText(itemsArrayList.get(position).getTitle());
        valueView.setText(itemsArrayList.get(position).getDescription());

        // 5. retrn rowView
        return rowView;
    }
}
package com.hmkcode.android;
import java.util.ArrayList;
import com.hmkcode.android.R;
import android.content.Context;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ArrayAdapter;
import android.widget.ImageView;
import android.widget.TextView;
public class MyAdapter extends ArrayAdapter<Model> {
    private final Context context;
```

```
private final ArrayList<Model> modelsArrayList;
public MyAdapter(Context context, ArrayList<Model> modelsArrayList) {

    super(context, R.layout.target_item, modelsArrayList);
    this.context = context;
    this.modelsArrayList = modelsArrayList;
}

@Override
public View getView(int position, View convertView, ViewGroup parent) {

    // 1. Create inflater
    LayoutInflator inflater = (LayoutInflator) context
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);

    // 2. Get rowView from inflater
    View rowView = null;
    if(!modelsArrayList.get(position).isGroupHeader()) {
        rowView = inflater.inflate(R.layout.target_item, parent,
false);
        // 3. Get icon,title & counter views from the rowView
        ImageView imgView = (ImageView)
rowView.findViewById(R.id.item_icon);
        TextView titleView = (TextView)
rowView.findViewById(R.id.item_title);
        TextView counterView = (TextView)
rowView.findViewById(R.id.item_counter);

        // 4. Set the text for textView

        imgView.setImageResource(modelsArrayList.get(position).getIcon());
        titleView.setText(modelsArrayList.get(position).getTitle());

        counterView.setText(modelsArrayList.get(position).getCounter());
    }
    else{
        rowView = inflater.inflate(R.layout.group_header_item,
parent, false);
        TextView titleView = (TextView)
rowView.findViewById(R.id.header);
    }
}
```

```
titleView.setText(modelsArrayList.get(position).getTitle());
}

// 5. retrn rowView
return rowView;
}

}

package com.hmkcode.adapters;
import android.support.annotation.NonNull;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.hmkcode.R;
import com.hmkcode.model.Link;
import java.util.List;

public class MyAdapter extends RecyclerView.Adapter<MyAdapter.MyViewHolder> {
    private List<Link> links;
    // constructor
    public MyAdapter(List<Link> links) {
        this.links = links;
    }
    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
```

```
viewType) {
    // inflate item_layout
    View itemLayoutView = LayoutInflater.from(parent.getContext())
        .inflate(R.layout.item_layout, null);
    MyViewHolder vh = new MyViewHolder(itemLayoutView);
    return vh;
}
@Override
public void onBindViewHolder(@NonNull MyViewHolder holder, int position) {
    holder.itemIcon.setImageResource(links.get(position).getIcon());
    holder.itemTitle.setText(links.get(position).getTitle());
    holder.itemUrl.setText(links.get(position).getUrl());
}
@Override
public int getItemCount() {
    if(links != null)
        return links.size();
    else
        return 0;
}
// inner static class
public static class MyViewHolder extends RecyclerView.ViewHolder {
    public TextView itemTitle;
    public TextView itemUrl;
    public ImageView itemIcon;
    public MyViewHolder(View itemLayoutView) {
        super(itemLayoutView);
        itemTitle = itemLayoutView.findViewById(R.id.item_title);
        itemUrl = itemLayoutView.findViewById(R.id.item_url);
        itemIcon = itemLayoutView.findViewById(R.id.item_icon);
    }
}
}
package com.hmkcode.browser.bookmark;
import java.util.Date;
import android.content.Context;
import android.database.Cursor;
import android.graphics.Bitmap;
import android.graphics.BitmapFactory;
import android.provider.Browser;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.CursorAdapter;
import android.widget.ImageView;
import android.widget.TextView;
public class MyCursorAdapter extends CursorAdapter {
    private LayoutInflater cursorInflater;

    // Default constructor
```

```
public MyCursorAdapter(Context context, Cursor cursor, int flags) {
    super(context, cursor, flags);
    cursorInflater = (LayoutInflater) context.getSystemService(
        Context.LAYOUT_INFLATER_SERVICE);
}
@Override
public void bindView(View view, Context context, Cursor cursor) {
    TextView tvDate = (TextView) view.findViewById(R.id.date);
    TextView tvTitle = (TextView) view.findViewById(R.id.title);
    TextView tvBoV = (TextView) view.findViewById(R.id.bov);
    TextView tvURL = (TextView) view.findViewById(R.id.url);
    ImageView ivFavicon = (ImageView) view.findViewById(R.id.favicon);
    // long date = cursor.getLong(Browser.SEARCHES_PROJECTION_DATE_INDEX);
    // String title =
    cursor.getString(Browser.SEARCHES_PROJECTION_SEARCH_INDEX);
    long date = cursor.getLong(Browser.HISTORY_PROJECTION_DATE_INDEX);
    String title =
    cursor.getString(Browser.HISTORY_PROJECTION_TITLE_INDEX);
    String url = cursor.getString(Browser.HISTORY_PROJECTION_URL_INDEX);
    byte[] icon = cursor.getBlob(Browser.HISTORY_PROJECTION_FAVICON_INDEX);
    String bokmrk =
    cursor.getString(Browser.HISTORY_PROJECTION_BOOKMARK_INDEX);
    String visits =
```

```
cursor.getString(Browser.HISTORY_PROJECTION_VISITS_INDEX);
String id = cursor.getString(Browser.HISTORY_PROJECTION_ID_INDEX);

// String names[] = cursor.getColumnNames();

tvDate.setText(new Date(date).toString());
tvBoV.setText(bokmrk.equals("1")?"Bookmarked":"Visited");
tvTitle.setText(title);
tvURL.setText(url);
if(icon != null){
    Bitmap bm = BitmapFactory.decodeByteArray(icon, 0, icon.length);
    ivFavicon.setImageBitmap(bm);
}
}

@Override
public View newView(Context context, Cursor cursor, ViewGroup parent) {
    return cursorInflater.inflate(R.layout.link, parent, false);
}

}

package com.hmkcode.adapters;
import android.support.v4.app.Fragment;
import android.support.v4.app.FragmentManager;
import android.support.v4.app.FragmentPagerAdapter;
import com.hmkcode.fragments.FirstFragment;
import com.hmkcode.fragments.SecondFragment;
public class MyFragmentPagerAdapter extends FragmentPagerAdapter {
    public MyFragmentPagerAdapter(FragmentManager fm) {
        super(fm);
    }
    @Override
    public Fragment getItem(int position) {
        switch (position) {
            case 0:
                return new FirstFragment();
            case 1:
                return new SecondFragment();
            default:
                return null;
        }
    }
    @Override
    public int getCount() {
        return 2;
    }
}
package com.hmkcode.listeners;
import com.hmkcode.models.Link;
@FunctionalInterface
public interface MyItemClickListener {
```

```
    public void onClick(Link link);
}
package com.hmkcode.adapters;
import android.support.annotation.NonNull;
import android.support.v7.recyclerview.extensions.ListAdapter;
import android.support.v7.util.DiffUtil;
import android.support.v7.widget.RecyclerView;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.ImageView;
import android.widget.TextView;
import com.hmkcode.R;
import com.hmkcode.listeners.MyItemClickListener;
import com.hmkcode.models.Link;
public class MyListAdapter extends ListAdapter<Link,
MyListAdapter.MyViewHolder> {
    MyItemClickListener myItemClickListener;
    public MyListAdapter(@NonNull DiffUtil.ItemCallback diffCallback,
                         MyItemClickListener myItemClickListener) {
        super(diffCallback);
        this.myItemClickListener = myItemClickListener;
    }
    @NonNull
    @Override
    public MyViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int
viewType) {
        View itemLayoutView = LayoutInflater.from(parent.getContext())
            .inflate(R.layout.item_layout, null);
        return new MyViewHolder(itemLayoutView);
    }
    @Override
    public void onBindViewHolder(@NonNull MyViewHolder myViewHolder, int
```

```
position) {
    myViewHolder.bind(getItem(position));
}
// inner class
public class MyViewHolder extends RecyclerView.ViewHolder {
    private View itemLayoutView;
    private TextView itemTitle;
    private TextView itemUrl;
    private ImageView itemIcon;
    public MyViewHolder(View itemLayoutView) {
        super(itemLayoutView);
        this.itemLayoutView = itemLayoutView;
        this.itemTitle = itemLayoutView.findViewById(R.id.item_title);
        this.itemUrl = itemLayoutView.findViewById(R.id.item_url);
        this.itemIcon = itemLayoutView.findViewById(R.id.item_icon);
    }
    public void bind(Link link) {
        this.itemIcon.setImageResource(link.getIcon());
        this.itemTitle.setText(link.getTitle());
        this.itemUrl.setText(link.getUrl());
        this.itemLayoutView.setOnClickListener(
            v -> myItemClickListener.onClick(link));
    }
}
package com.hmkcode.android.image;
import android.annotation.SuppressLint;
import android.content.Context;
import android.content.CursorLoader;
import android.database.Cursor;
import android.net.Uri;
import android.provider.DocumentsContract;
import android.provider.MediaStore;
public class RealPathUtil {
    @SuppressLint("NewApi")
    public static String getRealPathFromURI_API19(Context context, Uri uri) {
        String filePath = "";
        String wholeID = DocumentsContract.getDocumentId(uri);
        // Split at colon, use second item in the array
        String id = wholeID.split(":")[1];
        String[] column = { MediaStore.Images.Media.DATA };
        // where id is equal to
        String sel = MediaStore.Images.Media._ID + "=?";
        Cursor cursor =
context.getContentResolver().query(MediaStore.Images.Media.EXTERNAL_CONTENT_URI
```

```
'  
        column, sel, new String[]{ id }, null);  
  
    int columnIndex = cursor.getColumnIndex(column[0]);  
    if (cursor.moveToFirst()) {  
        filePath = cursor.getString(columnIndex);  
    }  
    cursor.close();  
  
    return filePath;  
}  
@SuppressLint("NewApi")  
public static String getRealPathFromURI_API11to18(Context context, Uri  
contentUri) {  
    String[] proj = { MediaStore.Images.Media.DATA };  
    String result = null;  
  
    CursorLoader cursorLoader = new CursorLoader(  
        context,  
        contentUri, proj, null, null, null);  
    Cursor cursor = cursorLoader.loadInBackground();  
  
    if(cursor != null){  
        int column_index =  
            cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);  
        cursor.moveToFirst();  
        result = cursor.getString(column_index);  
    }  
  
    return result;  
}  
public static String getRealPathFromURI_BelowAPI11(Context context, Uri  
contentUri) {  
    String[] proj = { MediaStore.Images.Media.DATA };  
    Cursor cursor = context.getContentResolver().query(contentUri,
```

```
proj, null, null, null);
        int column_index
    = cursor.getColumnIndexOrThrow(MediaStore.Images.Media.DATA);
        cursor.moveToFirst();
        return cursor.getString(column_index);
    }
}

package com.hmkcode.fragments;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import com.hmkcode.R;
public class SecondFragment extends Fragment {
    public SecondFragment() {}
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater,
                           ViewGroup container,
                           Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_second, container, false);
    }
}
package com.hmkcode.android.sign;
import android.os.Bundle;
import android.app.Activity;
public class SignInActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_in_screen);
    }
}
package com.hmkcode.android.sign;
import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
public class SignUpActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_sign_up_screen);
    }
}
```

```
package com.hmkcode.views;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Canvas;
import android.graphics.CornerPathEffect;
import android.graphics.Paint;
import android.util.AttributeSet;
import android.view.View;
public class SimpleView extends View {
    float dim;
    int shape;
    Paint paint;
    public static final int CIRCLE = 0;
    public static final int SQUARE = 1;
    public SimpleView(Context context, AttributeSet attrs) {
        super(context, attrs);
        TypedArray a = context.getTheme().obtainStyledAttributes(
            attrs,
            R.styleable.SimpleView,
            0, 0
        );
        try {
            dim = a.getDimension(R.styleable.SimpleView_dim, 20f);
            shape = a.getInteger(R.styleable.SimpleView_shape, 0);
        } finally {
            a.recycle();
        }
        paint = new Paint();
        paint.setColor(0xffffed325); // yellow
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        // draw circle
        switch(shape) {
            case CIRCLE:
                canvas.drawCircle(dim, dim, dim, paint);
                break;
            case SQUARE:
                canvas.drawRect(0, 0, dim, dim, paint);
                break;
        }
    }
}
/**
 * ownCloud Android client application
 *
 * @ David A. Velasco
 * @ Christian Schabesberger
 * 0 ownCloud GmbH.
```

```
* <p>
* .
* <p>
*.
* <p>
* .
*/
package com.owncloud.android.ui.preview;
import android.accounts.Account;
import android.os.Bundle;
import android.view.LayoutInflater;
import android.view.View;
import android.view.View.OnClickListener;
import android.view.ViewGroup;
import android.widget.ProgressBar;
import android.widget.TextView;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentStatePagerAdapter;
import com.owncloud.android.R;
import com.owncloud.android.datamodel.FileDataStorageManager;
import com.owncloud.android.datamodel.OCFile;
import com.owncloud.android.ui.controller.TransferProgressController;
import com.owncloud.android.ui.fragment.FileFragment;
import com.owncloud.android.utils.PreferenceUtils;
import timber.log.Timber;
/**
 * This Fragment is used to monitor the progress of a file downloading.
 */
public class FileDownloadFragment extends FileFragment implements
OnClickListener {
    public static final String EXTRA_FILE = "FILE";
    public static final String EXTRA_ACCOUNT = "ACCOUNT";
    private static final String EXTRA_ERROR = "ERROR";
    private static final String ARG_FILE = "FILE";
    private static final String ARG_IGNORE_FIRST = "IGNORE_FIRST";
    private static final String ARG_ACCOUNT = "ACCOUNT";
    private Account mAccount;
    public TransferProgressController mProgressController;
    private boolean mIgnoreFirstSavedState;
    private boolean mError;
    private ProgressBar mProgressBar;
    /**
     * Public factory method to create a new fragment that shows the progress
     * of a file download.
     *
     * Android strongly recommends keep the empty constructor of fragments as
     * the only public constructor, and
     * use {@link #setArguments(Bundle)} to set the needed arguments.
     *
     * This method hides to client objects the need of doing the construction
}
```

```
in two steps.  
*  
* When 'file' is null creates a dummy layout (useful when a file wasn't  
tapped before).  
*  
* @param file An {@link OCFFile} to show in the  
fragment  
* @param account An OC account; needed to start  
downloads  
* @param ignoreFirstSavedState Flag to work around an unexpected  
behaviour of {@link FragmentStatePagerAdapter}  
* TODO better solution  
*/  
public static Fragment newInstance(OCFile file, Account account, boolean  
ignoreFirstSavedState) {  
    FileDownloadFragment frag = new FileDownloadFragment();  
    Bundle args = new Bundle();  
    args.putParcelable(ARG_FILE, file);  
    args.putParcelable(ARG_ACCOUNT, account);  
    args.putBoolean(ARG_IGNORE_FIRST, ignoreFirstSavedState);  
    frag.setArguments(args);  
    return frag;  
}  
/**  
 * Creates an empty details fragment.  
 *  
 * It's necessary to keep a public constructor without parameters; the  
system uses it when tries to  
* reinstantiate a fragment automatically.  
*/  
public FileDownloadFragment() {  
    super();  
    mAccount = null;  
    mProgressController = null;  
    mIgnoreFirstSavedState = false;  
    mError = false;  
}  
@Override  
public void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    Bundle args = getArguments();  
    setFile((OCFile) args.getParcelable(ARG_FILE));  
    // TODO better in super, but needs to check ALL the class extending
```

```
FileFragment; not right now
    mIgnoreFirstSavedState = args.getBoolean(ARG_IGNORE_FIRST);
    mAccount = args.getParcelable(ARG_ACCOUNT);
}
@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    super.onCreateView(inflater, container, savedInstanceState);
    if (savedInstanceState != null) {
        if (!mIgnoreFirstSavedState) {
            setFile((OCFile)
                savedInstanceState.getParcelable(FileDownloadFragment.EXTRA_FILE));
            mAccount =
                savedInstanceState.getParcelable(FileDownloadFragment.EXTRA_ACCOUNT);
            mError =
                savedInstanceState.getBoolean(FileDownloadFragment.EXTRA_ERROR);
        } else {
            mIgnoreFirstSavedState = false;
        }
    }
    View rootView = inflater.inflate(R.layout.file_download_fragment,
        container, false);
    mProgressBar = rootView.findViewById(R.id.progressBar);
    (rootView.findViewById(R.id.cancelBtn)).setOnClickListener(this);
    rootView.setFilterTouchesWhenObscured(
        PreferenceUtils.shouldDisallowTouchesWithOtherVisibleWindows(getContext())
    );
    if (mError) {
        setButtonsForRemote(rootView);
    } else {
        setButtonsForTransferring(rootView);
    }
    return rootView;
}
@Override
public void onActivityCreated(Bundle savedInstanceState) {
    super.onActivityCreated(savedInstanceState);
    mProgressController = new
```

```
TransferProgressController(mContainerActivity);
    mProgressController.setProgressBar(mProgressBar);
}

@Override
public void onSaveInstanceState(Bundle outState) {
    super.onSaveInstanceState(outState);
    outState.putParcelable(FileDownloadFragment.EXTRA_FILE, getFile());
    outState.putParcelable(FileDownloadFragment.EXTRA_ACCOUNT, mAccount);
    outState.putBoolean(FileDownloadFragment.EXTRA_ERROR, mError);
}

@Override
public void onStart() {
    super.onStart();
    listenForTransferProgress();
}

@Override
public void onStop() {
    leaveTransferProgress();
    super.onStop();
}

@Override
public void onClick(View v) {
    switch (v.getId()) {
        case R.id.cancelBtn: {

mContainerActivity.getFileOperationsHelper().cancelTransference(getFile());
            getActivity().finish();
            break;
        }
        default:
            Timber.e("Incorrect view clicked!");
    }
}

/**
 * Enables buttons for a file being downloaded
 */
private void setButtonsForTransferring(View rootView) {
    if (rootView != null) {
        rootView.findViewById(R.id.cancelBtn).setVisibility(View.VISIBLE);
        // show the progress bar for the transfer
    }
}

rootView.findViewById(R.id.progressBar).setVisibility(View.VISIBLE);
    TextView progressText = rootView.findViewById(R.id.progressText);
```

```
progressText.setText(R.string.downloader_download_in_progress_ticker);
    progressText.setVisibility(View.VISIBLE);
    // hides the error icon
    rootView.findViewById(R.id.errorText).setVisibility(View.GONE);
    rootView.findViewById(R.id.error_image).setVisibility(View.GONE);
}
*/
* Enables or disables buttons for a file locally available
*/
private void setButtonsForDown(View rootView) {
    if (rootView != null) {
        rootView.findViewById(R.id.cancelBtn).setVisibility(View.GONE);
        // hides the progress bar
        rootView.findViewById(R.id.progressBar).setVisibility(View.GONE);
        // updates the text message
        TextView progressText = rootView.findViewById(R.id.progressText);
        progressText.setText(R.string.common_loading);
        progressText.setVisibility(View.VISIBLE);
        // hides the error icon
        rootView.findViewById(R.id.errorText).setVisibility(View.GONE);
        rootView.findViewById(R.id.error_image).setVisibility(View.GONE);
    }
}
/*
* Enables or disables buttons for a file not locally available
* <p/>
* Currently, this is only used when a download was failed
*/
private void setButtonsForRemote(View rootView) {
    if (rootView != null) {
        rootView.findViewById(R.id.cancelBtn).setVisibility(View.GONE);
        // hides the progress bar and message
        rootView.findViewById(R.id.progressBar).setVisibility(View.GONE);
        rootView.findViewById(R.id.progressText).setVisibility(View.GONE);
        // shows the error icon and message
        rootView.findViewById(R.id.errorText).setVisibility(View.VISIBLE);
    }
}
```

```
rootView.findViewById(R.id.error_image).setVisibility(View.VISIBLE);
    }
}
@Override
public void onTransferServiceConnected() {
    listenForTransferProgress();
}
@Override
public void onFileMetadataChanged(OCFile updatedFile) {
    if (updatedFile != null) {
        setFile(updatedFile);
    }
    // view does not need any update
}
@Override
public void onFileMetadataChanged() {
    FileDataStorageManager storageManager =
mContainerActivity.getStorageManager();
    if (storageManager != null) {
        setFile(storageManager.getFileByPath(getFile().getRemotePath()));
    }
    // view does not need any update
}
@Override
public void onFileContentChanged() {
    // view does not need any update, parent activity will replace this
```

```
fragment
}
@Override
public void updateViewForSyncInProgress() {
    setButtonsForTransferring(getView());
}
@Override
public void updateViewForSyncOff() {
    if (getFile().isDown()) {
        setButtonsForDown(getView());
    } else {
        setButtonsForRemote(getView());
    }
}
private void listenForTransferProgress() {
    if (mProgressController != null) {
        mProgressController.startListeningProgressFor(getFile(), mAccount);
        setButtonsForTransferring(getView());
    }
}
private void leaveTransferProgress() {
    if (mProgressController != null) {
        mProgressController.stopListeningProgressFor(getFile(), mAccount);
    }
}
public void setError(boolean error) {
    mError = error;
}
}
/**
 * ownCloud Android client application
 *
 * @ David A. Velasco
 * @ Christian Schabesberger
 * @ David González Verdugo
 * 0 ownCloud GmbH.
 * <p>
 * .
 * <p>
 * .
 * <p>
 * .
 */
package com.owncloud.android.ui.fragment;
import android.content.Context;
import androidx.fragment.app.Fragment;
import com.owncloud.android.datamodel.OCFile;
import com.owncloud.android.files.services.FileDownloader;
import com.owncloud.android.files.services.FileUploader;
import com.owncloud.android.ui.activity.ComponentsGetter;
```

```
/*
 * Common methods for {@link Fragment}s containing {@link OCFile}s
 */
public abstract class FileFragment extends Fragment {
    private OCFile mFile;
    protected ContainerActivity mContainerActivity;
    /**
     * Creates an empty fragment.
     *
     * It's necessary to keep a public constructor without parameters; the
     * system uses it when
     * tries to reinstantiate a fragment automatically.
     */
    public FileFragment() {
        mFile = null;
    }
    /**
     * Getter for the hold {@link OCFile}
     *
     * @return The {@link OCFile} hold
     */
    public OCFile getFile() {
        return mFile;
    }
    protected void setFile(OCFile file) {
        mFile = file;
    }
    /**
     * {@inheritDoc}
     */
    @Override
    public void onAttach(Context context) {
        super.onAttach(context);
        try {
            mContainerActivity = (ContainerActivity) context;
        } catch (ClassCastException e) {
            throw new ClassCastException(context.toString() + " must implement "
                    +
                    ContainerActivity.class.getSimpleName());
        }
    }
    /**
     * {@inheritDoc}
     */
    @Override
    public void onDetach() {
        mContainerActivity = null;
        super.onDetach();
    }
    public void onSyncEvent(String syncEvent, boolean success, OCFile
```

```

updatedFile) {
    if (syncEvent.equals(FileUploader.getUploadStartMessage())) {
        updateViewForSyncInProgress();
    } else if (syncEvent.equals(FileUploader.getUploadFinishMessage())) {
        if (success) {
            if (updatedFile != null) {
                onFileMetadataChanged(updatedFile);
            } else {
                onFileMetadataChanged();
            }
        }
        updateViewForSyncOff();
    } else if (syncEvent.equals(FileDownloader.getDownloadAddedMessage()))
    {
        updateViewForSyncInProgress();
    } else if
(syncEvent.equals(FileDownloader.getDownloadFinishMessage())) {
        if (success) {
            if (updatedFile != null) {
                onFileMetadataChanged(updatedFile);
            } else {
                onFileMetadataChanged();
            }
            onFileContentChanged();
        }
        updateViewForSyncOff();
    }
}
public abstract void updateViewForSyncInProgress();
public abstract void updateViewForSyncOff();
public abstract void onTransferServiceConnected();
public abstract void onFileMetadataChanged(OCFile updatedFile);
public abstract void onFileMetadataChanged();
public abstract void onFileContentChanged();
/**
 * Interface to implement by any Activity that includes some instance of
FileListFragment
 * Interface to implement by any Activity that includes some instance of
FileFragment
 */
public interface ContainerActivity extends ComponentsGetter {
    /**
     * Request the parent activity to show the details of an {@link
OCFile}.
     *
     * @param file      File to show details
     */
    void showDetails(OCFile file);
    //////////////////////////////////////////////////////////////////// TO UNIFY IN A SINGLE CALLBACK METHOD - EVENT NOTIFICATIONS ->

```

```
something happened
    // inside the fragment, MAYBE activity is interested --> unify in
notification method
    /**
     * Callback method invoked when a the user browsed into a different
```

```
        folder through the
        * list of files
        *
        * @param folder
        */
        void onBrowsedDownTo(OCFile folder);
    }
}

/**
 * ownCloud Android client application
 *
 * @ Bartek Przybylski
 * @ Tobias Kaminsky
 * @ David A. Velasco
 * @ masensio
 * @ Christian Schabesberger
 * @ David González Verdugo
 * @ Shashvat Kedia
 * @ Abel García de Prada
 * 1 Bartek Przybylski
 * 0 ownCloud GmbH.
 * <p>
 * .
 * <p>
 * .
 * <p>
 * .
 */

package com.owncloud.android.ui.adapter;
import android.accounts.Account;
import android.content.Context;
import android.graphics.Bitmap;
import android.graphics.Color;
import android.util.SparseBooleanArray;
import android.view.LayoutInflater;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AbsListView;
import android.widget.BaseAdapter;
import android.widget.GridView;
import android.widget.ImageView;
import android.widget.LinearLayout;
import android.widget.ListAdapter;
import android.widget.TextView;
import androidx.constraintlayout.widget.ConstraintLayout;
import com.owncloud.android.R;
import com.owncloud.android.authentication.AccountUtils;
import com.owncloud.android.datamodel.FileDataStorageManager;
import com.owncloud.android.datamodel.OCFile;
import com.owncloud.android.datamodel.ThumbnailsCacheManager;
```

```
import com.owncloud.android.db.PreferenceManager;
import com.owncloud.android.files.services.FileDownloader.FileDownloaderBinder;
import com.owncloud.android.files.services.FileUploader.FileUploaderBinder;
import com.owncloud.android.services.OperationsService.OperationsServiceBinder;
import com.owncloud.android.ui.activity.ComponentsGetter;
import com.owncloud.android.utils.DisplayUtils;
import com.owncloud.android.utils.FileStorageUtils;
import com.owncloud.android.utils.MimetypeIconUtil;
import com.owncloud.android.utils.PreferenceUtils;
import java.util.ArrayList;
import java.util.List;
import java.util.Vector;
/**
 * This Adapter populates a ListView with all files and folders in an ownCloud
 * instance.
 */
public class FileListListAdapter extends BaseAdapter implements ListAdapter {
    private Context mContext;
    private Vector<OCFile> mImmutableFilesList = null; // List containing the
database files, doesn't change with search
    private Vector<OCFile> mFiles = null; // List that can be changed when
using search
    private boolean mJustFolders;
    private boolean mOnlyAvailableOffline;
    private boolean mSharedByLinkFiles;
    private boolean mFolderPicker;
    private FileDataStorageManager mStorageManager;
    private Account mAccount;
    private ComponentsGetter mTransferServiceGetter;
    private enum viewType {LIST_ITEM, GRID_IMAGE, GRID_ITEM}
    public FileListListAdapter(
        boolean justFolders,
        boolean onlyAvailableOffline,
        boolean sharedByLinkFiles,
        boolean folderPicker,
        Context context,
        ComponentsGetter transferServiceGetter
    ) {
        mJustFolders = justFolders;
        mOnlyAvailableOffline = onlyAvailableOffline;
        mSharedByLinkFiles = sharedByLinkFiles;
        mFolderPicker = folderPicker;
        mContext = context;
        mAccount = AccountUtils.getCurrentOwnCloudAccount(mContext);
        mTransferServiceGetter = transferServiceGetter;
        // Read sorting order, default to sort by name ascending
        FileStorageUtils.mSortOrderFileDisp =
PreferenceManager.getSortOrder(mContext,
            FileStorageUtils.FILE_DISPLAY_SORT);
        FileStorageUtils.mSortAscendingFileDisp =
```

```
PreferenceManager.getSortAscending(mContext,
        FileStorageUtils.FILE_DISPLAY_SORT);
// initialise thumbnails cache on background thread
new ThumbnailsCacheManager().InitDiskCacheTask().execute();
}

@Override
public boolean areAllItemsEnabled() {
    return true;
}

@Override
public boolean isEnabled(int position) {
    // Disable click for files when selecting a folder in copying and
```

```
moving operations
    return !mFolderPicker || mFiles.get(position).isFolder();
}
@Override
public int getCount() {
    return mFiles != null ? mFiles.size() : 0;
}
@Override
public Object getItem(int position) {
    if (mFiles == null || mFiles.size() <= position) {
        return null;
    }
    return mFiles.get(position);
}
@Override
public long getItemId(int position) {
    if (mFiles == null || mFiles.size() <= position) {
        return 0;
    }
    return mFiles.get(position).get fileId();
}
@Override
public int getItemViewType(int position) {
    return 0;
}
@Override
public View getView(int position, View convertView, ViewGroup parent) {
    View view = convertView;
    OCFfile file = null;
    LayoutInflator inflator = (LayoutInflator) mContext
        .getSystemService(Context.LAYOUT_INFLATER_SERVICE);
    if (mFiles != null && mFiles.size() > position) {
        file = mFiles.get(position);
    }
    // Find out which layout should be displayed
    final viewType viewType;
    if (parent instanceof GridView) {
        if (file != null && file.isImage()) {
            viewType = viewType.GRID_IMAGE;
        } else {
            viewType = viewType.GRID_ITEM;
        }
    } else {
        viewType = viewType.LIST_ITEM;
    }
    // Create view only if differs, otherwise reuse
    if (convertView == null || convertView.getTag() != viewType) {
        switch (viewType) {
            case GRID_IMAGE:
                view = inflator.inflate(R.layout.grid_image, parent,
```

```
false);
        view.setTag(ViewType.GRID_IMAGE);
        break;
    case GRID_ITEM:
        view = inflator.inflate(R.layout.grid_item, parent,
false);
        view.setTag(ViewType.GRID_ITEM);
        break;
    case LIST_ITEM:
        view = inflator.inflate(R.layout.item_file_list, parent,
false);
        view.setTag(ViewType.LIST_ITEM);
        // Allow or disallow touches with other visible windows
        view.setFilterTouchesWhenObscured(
PreferenceUtils.shouldDisallowTouchesWithOtherVisibleWindows(mContext)
);
        break;
    }
}
if (file != null) {
    final ImageView localStateView =
view.findViewById(R.id.localFileIndicator);
    final ImageView fileIcon = view.findViewById(R.id.thumbnail);
    fileIcon.setTag(file.getFileId());
    TextView fileName;
    String name = file.getFileName();
    final LinearLayout linearLayout =
view.findViewById(R.id.ListItemLayout);
    if (linearLayout != null) {
        linearLayout.setContentDescription("LinearLayout-" + name);
        // Allow or disallow touches with other visible windows
        linearLayout.setFilterTouchesWhenObscured(
PreferenceUtils.shouldDisallowTouchesWithOtherVisibleWindows(mContext)
);
    }
    switch (viewType) {
        case LIST_ITEM:
            ConstraintLayout constraintLayout =
view.findViewById(R.id.file_list_constraint_layout);
            // Allow or disallow touches with other visible windows
            constraintLayout.setFilterTouchesWhenObscured(
PreferenceUtils.shouldDisallowTouchesWithOtherVisibleWindows(mContext));
            TextView fileSizeTV =
view.findViewById(R.id.file_list_size);
            TextView lastModTV =
view.findViewById(R.id.file_list_last_mod);
```

```
fileSizeTV.setText(DisplayUtils.bytesToHumanReadable(file.getFileLength(),
mContext));
lastModTV.setText(DisplayUtils.getRelativeTimestamp(mContext,
file.getModificationTimestamp()));
    if (mOnlyAvailableOffline || mSharedByLinkFiles) {
        TextView filePath =
view.findViewById(R.id.file_list_path);
        filePath.setVisibility(View.VISIBLE);
        filePath.setText(file.getRemotePath());
    }
    case GRID_ITEM:
        // filename
        fileName = view.findViewById(R.id.filename);
        name = file.getFileName();
        fileName.setText(name);
    case GRID_IMAGE:
        // sharedIcon
        ImageView sharedIconV = view.findViewById(R.id.sharedIcon);
        if (file.isSharedViaLink()) {

sharedIconV.setImageResource(R.drawable.ic_shared_by_link);
        sharedIconV.setVisibility(View.VISIBLE);
        sharedIconV.bringToFront();
    } else if (file.isSharedWithSharee() ||
file.isSharedWithMe()) {

sharedIconV.setImageResource(R.drawable.shared_via_users);
        sharedIconV.setVisibility(View.VISIBLE);
        sharedIconV.bringToFront();
    } else {
        sharedIconV.setVisibility(View.GONE);
    }
    break;
}
// For all Views
setIconPinAccordingToFileLocalState(localStateView, file);
final ImageView checkBoxV =
```

```

view.findViewById(R.id.custom_checkbox);
    checkBoxV.setVisibility(View.GONE);
    view.setBackgroundColor(Color.WHITE);
    AbsListView parentList = (AbsListView) parent;
    if (parentList.getChoiceMode() != AbsListView.CHOICE_MODE_NONE &&
        parentList.getCheckedItemCount() > 0
    ) {
        if (parentList.isItemChecked(position)) {
            view.setBackgroundColor(mContext.getResources().getColor(
                R.color.selected_item_background));
            checkBoxV.setImageResource(
                R.drawable.ic_checkbox_marked);
        } else {
            view.setBackgroundColor(Color.WHITE);
            checkBoxV.setImageResource(
                R.drawable.ic_checkbox_blank_outline);
        }
        checkBoxV.setVisibility(View.VISIBLE);
    }
    if (file.isDirectory()) {
        // Folder
        fileIcon.setImageResource(
            MimetypesIconUtil.getFolderTypeIconId(
                file.isSharedWithMe() ||
                file.isSharedWithSharee(),
                file.isSharedViaLink()));
    } else {
        if (file.isImage() && file.getRemoteId() != null) {
            // Thumbnail in Cache?
            Bitmap thumbnail =
                ThumbnailsCacheManager.getBitmapFromDiskCache(file.getRemoteId());
            if (thumbnail != null && !file.needsUpdateThumbnail()) {
                fileIcon.setImageBitmap(thumbnail);
            } else {
                // new Thumbnail
                if
                    (ThumbnailsCacheManager.cancelPotentialThumbnailWork(file, fileIcon)) {
                        final
                ThumbnailsCacheManager.ThumbnailGenerationTask task =
                    new
                ThumbnailsCacheManager.ThumbnailGenerationTask(
                    fileIcon, mStorageManager, mAccount
                );
                if (thumbnail == null) {
                    thumbnail = ThumbnailsCacheManager.mDefaultImg;
                }
                final ThumbnailsCacheManager.AsyncThumbnailDrawable
            asyncDrawable =
                new
            
```

```
ThumbnailCacheManager.AsyncThumbnailDrawable(
    mContext.getResources(),
    thumbnail,
    task
);
fileIcon.setImageDrawable(asyncDrawable);
task.execute(file);
}
}
if (file.getMimetype().equalsIgnoreCase("image/png")) {
    fileIcon.setBackgroundDrawable(mContext.getResources()
        .getColor(R.color.background_color));
}
} else {
    fileIcon.setImageResource(MimetypeIconUtil.getFileTypeIconId(file.getMimetype()
        ,
        file.getFileName()));
}
}
}
return view;
}
private void setIconPinAccordingToFileLocalState(ImageView localStateView,
OCFile file) {
    // local state
    localStateView.bringToFront();
    final FileDownloaderBinder downloaderBinder =
        mTransferServiceGetter.getFileDownloaderBinder();
    final FileUploaderBinder uploaderBinder =
        mTransferServiceGetter.getFileUploaderBinder();
    final OperationsServiceBinder opsBinder =
        mTransferServiceGetter.getOperationsServiceBinder();
    localStateView.setVisibility(View.INVISIBLE); // default first
    if (opsBinder != null && opsBinder.isSynchronizing(mAccount, file)) {
        //syncing
        localStateView.setImageResource(R.drawable.sync_pin);
        localStateView.setVisibility(View.VISIBLE);
    } else if (downloaderBinder != null &&
downloaderBinder.isDownloading(mAccount, file)) {
        // downloading
        localStateView.setImageResource(R.drawable.sync_pin);
        localStateView.setVisibility(View.VISIBLE);
    } else if (uploaderBinder != null &&
```

```
uploaderBinder.isUploading(mAccount, file)) {
    // uploading
    localStateView.setImageResource(R.drawable.sync_pin);
    localStateView.setVisibility(View.VISIBLE);
} else if (file.getEtagInConflict() != null) {
    // conflict
    localStateView.setImageResource(R.drawable.error_pin);
    localStateView.setVisibility(View.VISIBLE);
} else {
    if (file.isDown()) {
        localStateView.setVisibility(View.VISIBLE);
        localStateView.setImageResource(R.drawable.downloaded_pin);
    }
    if (file.isAvailableOffline()) {
        localStateView.setVisibility(View.VISIBLE);

        localStateView.setImageResource(R.drawable.offline_available_pin);
    }
}
@Override
public int getViewTypeCount() {
    return 1;
}
@Override
public boolean hasStableIds() {
    return true;
}
@Override
public boolean isEmpty() {
    return (mFiles == null || mFiles.isEmpty());
}
 * When 'file' is null creates a dummy layout (useful when a file wasn't
tapped before).
 *
 * @param file
 * @param account
 * @param ignoreFirstSavedState
 * @param
 */
public static Fragment newInstance(OCFile file, Account account, boolean
```

```

ignoreFirstSavedState) {
    FileDownloadFragment frag = new FileDownloadFragment();
    Bundle args = new Bundle();
    args.putParcelable(ARG_FILE, file);
    args.putParcelable(ARG_ACCOUNT, account);
    args.putBoolean(ARG_IGNORE_FIRST, ignoreFirstSavedState);
    frag.setArguments(args);
    return frag;
}
/***
 * Creates an empty details fragment.
***/
 * Change the adapted directory for a new one
 *
 * @param folder           New folder to adapt. Can be NULL, meaning
 *                        "no content to adapt".
 * @param updatedStorageManager Optional updated storage manager; used to
 * replace
 *                               mStorageManager if is different (and not
NULL)
 */
public void swapDirectory(OCFile folder, FileDataStorageManager
updatedStorageManager) {
    if (updatedStorageManager != null && updatedStorageManager !=
mStorageManager) {
        mStorageManager = updatedStorageManager;
        mAccount = AccountUtils.getCurrentOwnCloudAccount(mContext);
    }
    boolean isRootFolder =
folder.equals(updatedStorageManager.getFileByPath(OCFile.ROOT_PATH));
    if (mStorageManager != null) {
        if (mOnlyAvailableOffline && (isRootFolder
|| !folder.isAvailableOffline())) {
            mImmutableFilesList =
updatedStorageManager.getAvailableOfflineFilesFromCurrentAccount();
        } else if (mSharedByLinkFiles && isRootFolder) {
            mImmutableFilesList =
updatedStorageManager.getSharedByLinkFilesFromCurrentAccount();
        } else {
            mImmutableFilesList = mStorageManager.getFolderContent(folder);
        }
        mFiles = mImmutableFilesList;
        if (mJustFolders) {
            mFiles = getFolders(mFiles);
        }
    } else {
        mFiles = null;
    }
    mFiles = FileStorageUtils.sortFolder(mFiles,

```

```
FileStorageUtils.mSortOrderFileDisp,
        FileStorageUtils.mSortAscendingFileDisp);
    notifyDataSetChanged();
}
/**
 * Filter for getting only the folders
 *
 * @param files Collection of files to filter
 * @return Folders in the input
 */
private Vector<OCFile> getFolders(Vector<OCFile> files) {
    Vector<OCFile> ret = new Vector<>();
    OCFile current;
    for (int i = 0; i < files.size(); i++) {
        current = files.get(i);
        if (current.isFolder()) {
            ret.add(current);
        }
    }
    return ret;
}
public void setSortOrder(Integer order, boolean ascending) {
    PreferenceManager.setSortOrder(order, mContext,
    FileStorageUtils.FILE_DISPLAY_SORT);
    PreferenceManager.setSortAscending(ascending, mContext,
    FileStorageUtils.FILE_DISPLAY_SORT);
    FileStorageUtils.mSortOrderFileDisp = order;
    FileStorageUtils.mSortAscendingFileDisp = ascending;
    mFiles = FileStorageUtils.sortFolder(mFiles,
    FileStorageUtils.mSortOrderFileDisp,
        FileStorageUtils.mSortAscendingFileDisp);
    notifyDataSetChanged();
}
public ArrayList<OCFile> getCheckedItems(AbsListView parentList) {
    SparseBooleanArray checkedPositions =
```

```
parentList.getCheckedItemPositions();
ArrayList<OCFile> files = new ArrayList<>();
Object item;
for (int i = 0; i < checkedPositions.size(); i++) {
    if (checkedPositions.valueAt(i)) {
        item = getItem(checkedPositions.keyAt(i));
        if (item != null) {
            files.add((OCFile) item);
        }
    }
}
return files;
}
public void filterBySearch(String query) {
    query = query.toLowerCase();
    clearFilterBySearch();
    List<OCFile> filteredList = new ArrayList<>();
    // Gather files matching the query
    for (OCFile fileToAdd : mFiles) {
        final String nameOfFileToAdd =
```

```
fileToAdd.getFileName().toLowerCase();
        if (nameOfFileToAdd.contains(query)) {
            filteredList.add(fileToAdd);
        }
    }
    // Remove not matching files from the filelist
    for (int i = mFiles.size() - 1; i >= 0; i--) {
        if (!filteredList.contains(mFiles.get(i))) {
            mFiles.remove(i);
        }
    }
    // Add matching files to the filelist
    for (int i = 0; i < filteredList.size(); i++) {
        if (!mFiles.contains(filteredList.get(i))) {
            mFiles.add(i, filteredList.get(i));
        }
    }
    notifyDataSetChanged();
}
public void clearFilterBySearch() {
    mFiles = (Vector<OCFile>) immutableFilesList.clone();
    notifyDataSetChanged();
}
/**
 * ownCloud Android client application
 *
 * @ David A. Velasco
 * @ Christian Schabesberger
 * @ Abel García de Prada
 * @ Shashvat Kedia
 * 0 ownCloud GmbH.
 * <p>
 * .
 * <p>
 * .
 * <p>
 * .
 */
package com.owncloud.android.files;
import android.accounts.Account;
import android.content.Context;
import android.view.Menu;
import android.view.MenuItem;
import androidx.fragment.app.Fragment;
import androidx.fragment.app.FragmentActivity;
import com.owncloud.android.R;
import com.owncloud.android.datamodel.OCFile;
import com.owncloud.android.domain.capabilities.model.OCCapability;
import com.owncloud.android.files.services.FileDownloader.FileDownloaderBinder;
```

```
import com.owncloud.android.files.services.FileUploader.FileUploaderBinder;
import com.owncloud.android.services.OperationsService.OperationsServiceBinder;
import com.owncloud.android.ui.activity.ComponentsGetter;
import com.owncloud.android.ui.preview.PreviewVideoFragment;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
/**
 * Filters out the file actions available in a given {@link Menu} for a given
{@link OCFfile}
 * according to the current state of the latest.
 */
public class FileMenuFilter {
    private static final int SINGLE_SELECT_ITEMS = 1;
    private static final String TAG_SECOND_FRAGMENT = "SECOND_FRAGMENT";
    private List<OCFile> mFiles;
    private ComponentsGetter mComponentsGetter;
    private Account mAccount;
    private Context mContext;
    /**
     * Constructor
     *
     * @param targetFiles List of {@link OCFfile} file targets of the action to
filter in the {@link Menu}.
     * @param account ownCloud {@link Account} holding targetFile.
     * @param cg Accessor to app components, needed to access
synchronization services
     * @param context Android {@link Context}, needed to access build setup
resources.
     */
    public FileMenuFilter(List<OCFile> targetFiles, Account account,
ComponentsGetter cg,
                           Context context) {
        mFiles = targetFiles;
        mAccount = account;
        mComponentsGetter = cg;
        mContext = context;
    }
    /**
     * Constructor
     *
     * @param targetFile {@link OCFfile} target of the action to filter in the
{@link Menu}.
     * @param cg Accessor to app components, needed to access
synchronization services
     * @param context Android {@link Context}, needed to access build setup
resources.
     */
    public FileMenuFilter(OCFile targetFile, Account account, ComponentsGetter
```

```

cg,
        Context context) {
    this(Arrays.asList(targetFile), account, cg, context);
}
/**
 * Filters out the file actions available in the passed {@link Menu} taken
into account
 * the state of the {@link OCFfile} held by the filter.
 *
 * @param menu Options or context menu to filter.
 */
public void filter(Menu menu, boolean displaySelectAll, boolean
displaySelectInverse,
                    boolean onlyAvailableOffline, boolean sharedByLinkFiles)
{
    if (mFiles == null || mFiles.size() <= 0) {
        hideAll(menu);
    } else {
        List<Integer> toShow = new ArrayList<>();
        List<Integer> toHide = new ArrayList<>();
        filter(toShow, toHide, displaySelectAll, displaySelectInverse,
onlyAvailableOffline, sharedByLinkFiles);
        MenuItem item;
        for (int i : toShow) {
            item = menu.findItem(i);
            if (item != null) {
                item.setVisible(true);
                item.setEnabled(true);
            }
        }
        for (int i : toHide) {
            item = menu.findItem(i);
            if (item != null) {
                item.setVisible(false);
                item.setEnabled(false);
            }
        }
    }
}

private void hideAll(Menu menu) {
    MenuItem item;
    for (int i = 0; i < menu.size(); i++) {
        item = menu.getItem(i);
        item.setVisible(false);
        item.setEnabled(false);
    }
}
/**
 * Performs the real filtering, to be applied in the {@link Menu} by the

```

```
caller methods.  
 * <p>  
 * Decides what actions must be shown and hidden.  
 *  
 * @param toShow List to save the options that must be shown in the menu.  
 * @param toHide List to save the options that must be shown in the menu.  
 */  
 private void filter(List<Integer> toShow, List<Integer> toHide, boolean  
 displaySelectAll,  
                     boolean displaySelectInverse, boolean  
 onlyAvailableOffline, boolean sharedByLinkFiles) {  
     boolean synchronizing = anyFileSynchronizing();  
     boolean videoPreviewing = anyFileVideoPreviewing();  
     boolean videoStreaming = !anyFileDown() && anyFileVideoPreviewing();  
     // decision is taken for each possible action on a file in the menu  
     if (displaySelectAll) {  
         toShow.add(R.id.file_action_select_all);  
     } else {  
         toHide.add(R.id.file_action_select_all);  
     }  
     if (displaySelectInverse) {  
         toShow.add(R.id.action_select_inverse);  
     } else {  
         toHide.add(R.id.action_select_inverse);  
     }  
     // DOWNLOAD  
     if (mFiles.isEmpty() || containsFolder() || anyFileDown() ||  
     synchronizing || videoPreviewing ||  
         onlyAvailableOffline || sharedByLinkFiles) {  
         toHide.add(R.id.action_download_file);  
     } else {  
         toShow.add(R.id.action_download_file);  
     }  
     // RENAME  
     if (!isSingleSelection() || synchronizing || videoPreviewing ||  
     onlyAvailableOffline || sharedByLinkFiles) {  
         toHide.add(R.id.action_rename_file);  
     } else {  
         toShow.add(R.id.action_rename_file);  
     }  
     // MOVE & COPY  
     if (mFiles.isEmpty() || synchronizing || videoPreviewing ||
```

```

onlyAvailableOffline || sharedByLinkFiles) {
    toHide.add(R.id.action_move);
    toHide.add(R.id.action_copy);
} else {
    toShow.add(R.id.action_move);
    toShow.add(R.id.action_copy);
}
// REMOVE
if (mFiles.isEmpty() || synchronizing || onlyAvailableOffline || sharedByLinkFiles) {
    toHide.add(R.id.action_remove_file);
} else {
    toShow.add(R.id.action_remove_file);
}
// OPEN WITH (different to preview!)
if (!isSingleFile() || !anyFileDown() || synchronizing) {
    toHide.add(R.id.action_open_file_with);
} else {
    toShow.add(R.id.action_open_file_with);
}
// CANCEL SYNCHRONIZATION
if (mFiles.isEmpty() || !synchronizing || anyFavorite() || onlyAvailableOffline || sharedByLinkFiles) {
    toHide.add(R.id.action_cancel_sync);
} else {
    toShow.add(R.id.action_cancel_sync);
}
// SYNC CONTENTS (BOTH FILE AND FOLDER)
if (mFiles.isEmpty() || (!anyFileDown() && !containsFolder()) || synchronizing || onlyAvailableOffline || sharedByLinkFiles) {
    toHide.add(R.id.action_sync_file);
} else {
    toShow.add(R.id.action_sync_file);
}
// SHARE FILE
boolean shareViaLinkAllowed = (mContext != null &&
mContext.getResources().getBoolean(R.bool.share_via_link_feature));
boolean shareWithUsersAllowed = (mContext != null
&&mContext.getResources().getBoolean(R.bool.share_with_users_feature));
OCCapability capability =
mComponentsGetter.getStorageManager().getCapability(mAccount.name);
boolean notAllowResharng = anyFileSharedWithMe() &&
capability != null &&
capability.GetFilesSharingResharng().isFalse();
if ((!shareViaLinkAllowed && !shareWithUsersAllowed)

```

```
|| !isSingleSelection() ||
    notAllowResharang || onlyAvailableOffline) {
    toHide.add(R.id.action_share_file);
} else {
    toShow.add(R.id.action_share_file);
}
onlyAvailableOffline || sharedByLinkFiles) {
    toHide.add(R.id.action_cancel_sync);
} else {
    toShow.add(R.id.action_cancel_sync);
}
// SYNC CONTENTS (BOTH FILE AND FOLDER)
if (mFiles.isEmpty() || (!anyFileDown() && !containsFolder()) ||
synchronizing || onlyAvailableOffline || sharedByLinkFiles)
{
    toHide.add(R.id.action_sync_file);
} else {
    toShow.add(R.id.action_sync_file);
}
// SHARE FILE
boolean shareViaLinkAllowed = (mContext != null
&&mContext.getResources().getBoolean(R.bool.share_via_link_feature));
boolean shareWithUsersAllowed = (mContext != null &&
// SEE DETAILS
if (!isSingleFile()) {
    toHide.add(R.id.action_see_details);
} else {
    toShow.add(R.id.action_see_details);
}
// SEND
boolean sendAllowed = (mContext != null &&

mContext.getString(R.string.send_files_to_other_apps).equalsIgnoreCase("on"));
    if (!isSingleFile() || !sendAllowed || synchronizing || videoStreaming
```

```

    || onlyAvailableOffline) {
        toHide.add(R.id.action_send_file);
    } else {
        toShow.add(R.id.action_send_file);
    }
    // SET AS AVAILABLE OFFLINE
    if (synchronizing || !anyUnfavorite() || videoStreaming) {
        toHide.add(R.id.action_set_available_offline);
    } else {
        toShow.add(R.id.action_set_available_offline);
    }
    // UNSET AS AVAILABLE OFFLINE
    if (!anyFavorite() || videoStreaming) {
        toHide.add(R.id.action_unset_available_offline);
    } else {
        toShow.add(R.id.action_unset_available_offline);
    }
}
private boolean anyFileSynchronizing() {
    boolean synchronizing = false;
    if (mComponentsGetter != null && !mFiles.isEmpty() && mAccount != null) {
        OperationsServiceBinder opsBinder =
mComponentsGetter.getOperationsServiceBinder();
        FileUploaderBinder uploaderBinder =
mComponentsGetter.getFileUploaderBinder();
        FileDownloaderBinder downloaderBinder =
mComponentsGetter.getFileDownloaderBinder();
        synchronizing = (
            anyFileSynchronizing(opsBinder) ||           // comparing local
and remote
            anyFileDownloading(downloaderBinder) ||
            anyFileUploading(uploaderBinder)
        );
    }
    return synchronizing;
}
private boolean anyFileSynchronizing(OperationsServiceBinder opsBinder) {
    boolean synchronizing = false;
    if (opsBinder != null) {
        for (int i = 0; !synchronizing && i < mFiles.size(); i++) {
            synchronizing = opsBinder.isSynchronizing(mAccount,

```

```
mFiles.get(i));
        }
    }
    return synchronizing;
}
private boolean anyFileDownloading(FileDownloaderBinder downloaderBinder) {
    boolean downloading = false;
    if (downloaderBinder != null) {
        for (int i = 0; !downloading && i < mFiles.size(); i++) {
            downloading = downloaderBinder.isDownloading(mAccount,
mFiles.get(i));
        }
    }
    return downloading;
}
private boolean anyFileUploading(FileUploaderBinder uploaderBinder) {
    boolean uploading = false;
    if (uploaderBinder != null) {
        for (int i = 0; !uploading && i < mFiles.size(); i++) {
            uploading = uploaderBinder.isUploading(mAccount,
mFiles.get(i));
        }
    }
    return uploading;
}
private boolean anyFileVideoPreviewing() {
    final FragmentActivity activity = (FragmentActivity) mContext;
    Fragment secondFragment =

```

```
activity.getSupportFragmentManager().findFragmentByTag(
    TAG_SECOND_FRAGMENT);
boolean videoPreviewing = false;
if (secondFragment instanceof PreviewVideoFragment) {
    for (int i = 0; !videoPreviewing && i < mFiles.size(); i++) {
        videoPreviewing = ((PreviewVideoFragment) secondFragment).
            getFile().equals(mFiles.get(i));
    }
}
return videoPreviewing;
}
private boolean isSingleSelection() {
    return mFiles.size() == SINGLE_SELECT_ITEMS;
}
private boolean isSingleFile() {
    return isSingleSelection() && !mFiles.get(0).isFolder();
}
private boolean containsFolder() {
    for (OCFile file : mFiles) {
        if (file.isFolder()) {
            return true;
        }
    }
    return false;
}
private boolean anyFileDown() {
    for (OCFile file : mFiles) {
        if (file.isDown()) {
            return true;
        }
    }
    return false;
}
private boolean anyFavorite() {
    for (OCFile file : mFiles) {
        if (file.getAvailableOfflineStatus() ==
OCFile.AvailableOfflineStatus.AVAILABLE_OFFLINE) {
            return true;
        }
    }
    return false;
}
private boolean anyUnfavorite() {
    for (OCFile file : mFiles) {
        if (file.getAvailableOfflineStatus() ==
```

```
OCFile.AvailableOfflineStatus.NOT_AVAILABLE_OFFLINE) {
        return true;
    }
}
return false;
}
private boolean anyFileSharedWithMe() {
    for (OCFile file : mFiles) {
        if (file.isSharedWithMe()) {
            return true;
        }
    }
    return false;
}
/**
 * ownCloud Android client application
 *
 * @ masensio
 * @ David A. Velasco
 * @ Juan Carlos González Cabrero
 * @ David González Verdugo
 * @ Shashvat Kedia
 * 0 ownCloud GmbH.
 * <p>
 * .
 * <p>
 * .
 * <p>
 * .
 */
package com.owncloud.android.ui.helpers;
import android.accounts.Account;
import android.content.ActivityNotFoundException;
import android.content.Intent;
import android.content.pm.PackageManager;
import android.content.pm.ResolveInfo;
import android.webkit.MimeTypeMap;
import androidx.fragment.app.DialogFragment;
import com.owncloud.android.R;
import com.owncloud.android.authentication.AccountUtils;
import com.owncloud.android.datamodel.OCFile;
import com.owncloud.android.domain.sharing.shares.model.OCShare;
import com.owncloud.android.files.services.AvailableOfflineHandler;
import com.owncloud.android.files.services.FileDownloader.FileDownloaderBinder;
import com.owncloud.android.files.services.FileUploader.FileUploaderBinder;
import com.owncloud.android.lib.resources.status.OwnCloud;
import com.owncloud.android.presentation.ui.sharing.ShareActivity;
import com.owncloud.android.services.OperationsService;
import com.owncloud.android.ui.activity.FileActivity;
```

```
import com.owncloud.android.ui.dialog.ShareLinkToDialog;
import timber.log.Timber;
import java.util.Collection;
import java.util.List;
public class FileOperationsHelper {
    private static final String FTAG_CHOOSER_DIALOG = "CHOOSER_DIALOG";
    private FileActivity mFileActivity;
    /// Identifier of operation in progress which result shouldn't be lost
    private long mWaitingForOpId = Long.MAX_VALUE;
    public FileOperationsHelper(FileActivity fileActivity) {
        mFileActivity = fileActivity;
    }
    public void openFile(OCFile file) {
        if (file != null) {
            String storagePath = file.getStoragePath();
            Intent intentForSavedMimeType = new Intent(Intent.ACTION_VIEW);
            intentForSavedMimeType.setDataAndType(
                file.getExposedFileUri(mFileActivity),
                file.getMimetype()
            );
            intentForSavedMimeType.setFlags(
                Intent.FLAG_GRANT_READ_URI_PERMISSION |
                Intent.FLAG_GRANT_WRITE_URI_PERMISSION
            );
            Intent intentForGuessedMimeType = null;
            if (storagePath.lastIndexOf('.') >= 0) {
                String guessedMimeType =
                   MimeTypeMap.getSingleton().getMimeTypeFromExtension(
                        storagePath.substring(storagePath.lastIndexOf('.') + 1)
                    );
                if (guessedMimeType != null)
```

```
&& !guessedMimeType.equals(file.getMimetype())) {
    intentForGuessedMimeType = new Intent(Intent.ACTION_VIEW);
    intentForGuessedMimeType.setDataAndType(
        file.getExposedFileUri(mFileActivity),
        guessedMimeType
    );
    intentForGuessedMimeType.setFlags(
        Intent.FLAG_GRANT_READ_URI_PERMISSION |
        Intent.FLAG_GRANT_WRITE_URI_PERMISSION
    );
}
Intent openFileWithIntent;
if (intentForGuessedMimeType != null) {
    openFileWithIntent = intentForGuessedMimeType;
} else {
    openFileWithIntent = intentForSavedMimeType;
}
List<ResolveInfo> launchables = mFileActivity.getPackageManager().queryIntentActivities(openFileWithIntent,
PackageManager.MATCH_DEFAULT_ONLY);
if (launchables != null && launchables.size() > 0) {
    try {
        mFileActivity.startActivity(
            Intent.createChooser(
                openFileWithIntent,
                mFileActivity.getString(R.string.actionbar_open_with)
            )
        );
    } catch (ActivityNotFoundException anfe) {
        mFileActivity.showSnackBar(
            mFileActivity.getString(
                R.string.file_list_no_app_for_file_type
            )
        );
    }
} else {
    mFileActivity.showSnackBar(
        mFileActivity.getString(R.string.file_list_no_app_for_file_type)
    );
}
} else {
    Timber.e("Trying to open a NULL OCFile");
}
}
/***
 * Show dialog to allow the user to choose an app to send the private link

```

```
of an {@link OCFfile},
 * or copy it to clipboard.
 *
 * @param file @param file {@link OCFfile} which will be shared with
internal users
 */
public void copyOrSendPrivateLink(OCFfile file) {
    // Parse remotefile
    String privateLink = file.getPrivateLink();
    if (privateLink == null || privateLink.isEmpty()) {
        mFileActivity.showSnackBar(
            mFileActivity.getString(R.string.file_private_link_error)
        );
        return;
    }
    shareLink(privateLink);
}
/***
 * Show dialog to allow the user to choose an app to send the link of an
{@link OCShare},
 * or copy it to clipboard.
 *
 * @param share {@link OCShare} which link will be sent to the app chosen
by the user.
 */
public void copyOrSendPublicLink(OCShare share) {
    String link = share.getShareLink();
    if (link.length() <= 0) {
        mFileActivity.showSnackBar(
            mFileActivity.getString(R.string.share_no_link_in_this_share)
        );
        return;
    }
    shareLink(link);
}
/***
 * Show an instance of {@link
com.owncloud.android.domain.sharing.shares.model.ShareType} for sharing or
unsharing
 * the {@link OCFfile} received as parameter.
 *
 * @param file File to share or unshare.
 */
public void showShareFile(OCFfile file) {
    Intent intent = new Intent(mFileActivity, ShareActivity.class);
    intent.putExtra(FileActivity.EXTRA_FILE, file);
    intent.putExtra(FileActivity.EXTRA_ACCOUNT,
```

```
mFileActivity.getAccount();
    mFileActivity.startActivity(intent);
}
public void sendDownloadedFile(OCFile file) {
    if (file != null) {
        Intent sendIntent = new Intent(android.content.Intent.ACTION_SEND);
        // set MimeType
        sendIntent.setType(file.getMimetype());
        sendIntent.putExtra(
            Intent.EXTRA_STREAM,
            file.getExposedFileUri(mFileActivity)
        );
        sendIntent.putExtra(Intent.ACTION_SEND, true);           // Send Action
        // Show dialog, without the own app
        String[] packagesToExclude = new
String[] {mFileActivity.getPackageName()};
        DialogFragment chooserDialog =
ShareLinkToDialog.newInstance(sendIntent, packagesToExclude);
        chooserDialog.show(mFileActivity.getSupportFragmentManager(),
FTAG_CHOOSER_DIALOG);
    } else {
        Timber.e("Trying to send a NULL OCFile");
    }
}
public void syncFiles(Collection<OCFile> files) {
    for (OCFile file : files) {
        syncFile(file);
    }
}
/**
 * Request the synchronization of a file or folder with the OC server,
including its contents.
 *
 * @param file The file or folder to synchronize
 */
public void syncFile(OCFile file) {
    if (!file.isFolder()) {
        Intent intent = new Intent(mFileActivity, OperationsService.class);
        intent.setAction(OperationsService.ACTION_SYNC_FILE);
        intent.putExtra(OperationsService.EXTRA_ACCOUNT,
mFileActivity.getAccount());
        intent.putExtra(OperationsService.EXTRA_REMOTE_PATH,
file.getRemotePath());
        mWaitingForOpId =
mFileActivity.getOperationsServiceBinder().queueNewOperation(intent);
    } else {
        Intent intent = new Intent(mFileActivity, OperationsService.class);
        intent.setAction(OperationsService.ACTION_SYNC_FOLDER);
        intent.putExtra(OperationsService.EXTRA_ACCOUNT,
```

```
mFileActivity.getAccount();
    intent.putExtra(OperationsService.EXTRA_REMOTE_PATH,
file.getRemotePath());
    intent.putExtra(
        OperationsService.EXTRA_SYNC_REGULAR_FILES,
        true
    );
    mFileActivity.startService(intent);
}
}
public void toggleAvailableOffline(Collection<OCFile> files, boolean
isAvailableOffline) {
    for (OCFile file : files) {
        toggleAvailableOffline(file, isAvailableOffline);
    }
}
public void toggleAvailableOffline(OCFile file, boolean isAvailableOffline)
{
    if (OCFile.AvailableOfflineStatus.AVAILABLE_OFFLINE_PARENT ==
file.getAvailableOfflineStatus()) {
        // files descending of an av-offline folder can't be toggled
        mFileActivity.showSnackBar(
mFileActivity.getString(R.string.available_offline_inherited_msg)
        );
    } else {
        // update local property, for file and all its descendants (if
        folder)
        OCFile.AvailableOfflineStatus targetAvailableOfflineStatus =
isAvailableOffline ?
            OCFile.AvailableOfflineStatus.AVAILABLE_OFFLINE :
            OCFile.AvailableOfflineStatus.NOT_AVAILABLE_OFFLINE;
        file.setAvailableOfflineStatus(targetAvailableOfflineStatus);
        boolean success =
mFileActivity.getStorageManager().saveLocalAvailableOfflineStatus(file);
        if (success) {
            // Schedule job to check to watch for local changes in
available offline files and sync them
            AvailableOfflineHandler availableOfflineHandler =
                new AvailableOfflineHandler(mFileActivity,
mFileActivity.getAccount().name);

            availableOfflineHandler.scheduleAvailableOfflineJob(mFileActivity);
            /// immediate content synchronization
            if (OCFile.AvailableOfflineStatus.AVAILABLE_OFFLINE ==
```

```
file.getAvailableOfflineStatus() {
    syncFile(file);
} else {
    cancelTransference(file);
}
} else {
    // unexpected error
    mFileActivity.showSnackBar(
        mFileActivity.getString(R.string.common_error_unknown)
    );
}
}

public void renameFile(OCFile file, String newfilename) {
    // RenameFile
    Intent service = new Intent(mFileActivity, OperationsService.class);
    service.setAction(OperationsService.ACTION_RENAME);
    service.putExtra(OperationsService.EXTRA_ACCOUNT,
mFileActivity.getAccount());
    service.putExtra(OperationsService.EXTRA_REMOTE_PATH,
file.getRemotePath());
    service.putExtra(OperationsService.EXTRA_NEWNAME, newfilename);
    mWaitingForOpId =
mFileActivity.getOperationsServiceBinder().queueNewOperation(service);
    mFileActivity.showLoadingDialog(R.string.wait_a_moment);
}

/**
 * Start operations to delete one or several files
 *
 * @param files      Files to delete
 * @param onlyLocalCopy When 'true' only local copy of the files is
removed; otherwise files are also deleted
 *                      in the server.
 */
public void removeFiles(Collection<OCFile> files, boolean onlyLocalCopy) {
    int countOfFilesToRemove = 0;
    boolean isLastFileToRemove = false;
    for (OCFile file : files) {
        countOfFilesToRemove++;
        // RemoveFile
        Intent service = new Intent(mFileActivity,
OperationsService.class);
        service.setAction(OperationsService.ACTION_REMOVE);
        service.putExtra(OperationsService.EXTRA_ACCOUNT,
mFileActivity.getAccount());
        service.putExtra(OperationsService.EXTRA_REMOTE_PATH,
file.getRemotePath());
        service.putExtra(OperationsService.EXTRA_REMOVE_ONLY_LOCAL,
```

```
onlyLocalCopy);
        if (countOfFilesToRemove == files.size()) {
            isLastFileToRemove = true;
        }
        service.putExtra(OperationsService.EXTRA_IS_LAST_FILE_TO_REMOVE,
isLastFileToRemove); mWaitingForOpId =
mFileActivity.getOperationsServiceBinder().queueNewOperation(service);
    }
    mFileActivity.showLoadingDialog(R.string.wait_a_moment);
}
public void createFolder(String remotePath, boolean createFullPath) {
    // Create Folder
    Intent service = new Intent(mFileActivity, OperationsService.class);
    service.setAction(OperationsService.ACTION_CREATE_FOLDER);
    service.putExtra(OperationsService.EXTRA_ACCOUNT,
mFileActivity.getAccount());
    service.putExtra(OperationsService.EXTRA_REMOTE_PATH, remotePath);
    service.putExtra(OperationsService.EXTRA_CREATE_FULL_PATH,
createFullPath);
    mWaitingForOpId =
mFileActivity.getOperationsServiceBinder().queueNewOperation(service);
    mFileActivity.showLoadingDialog(R.string.wait_a_moment);
}
/**
 * Cancel the transference in downloads (files/folders) and file uploads
 *
 * @param file OCFfile
 */
public void cancelTransference(OCFfile file) {
    Account account = mFileActivity.getAccount();
    if (file.isFolder()) {
        OperationsService.OperationsServiceBinder opsBinder =
mFileActivity.getOperationsServiceBinder();
        if (opsBinder != null) {
            opsBinder.cancel(account, file);
        }
    }
    // for both files and folders
    FileDownloaderBinder downloaderBinder =
mFileActivity.getFileDownloaderBinder();
    if (downloaderBinder != null &&
downloaderBinder.isDownloading(account, file)) {
        downloaderBinder.cancel(account, file);
    }
    FileUploaderBinder uploaderBinder =
mFileActivity.getFileUploaderBinder();
    if (uploaderBinder != null && uploaderBinder.isUploading(account,
```

```
file)) {
        uploaderBinder.cancel(account, file);
    }
}
/***
 * Start operations to move one or several files
 *
 * @param files      Files to move
 * @param targetFolder Folder where the files will be moved into
 */
public void moveFiles(Collection<OCFile> files, OCFile targetFolder) {
    for (OCFile file : files) {
        Intent service = new Intent(mFileActivity,
OperationsService.class);
        service.setAction(OperationsService.ACTION_MOVE_FILE);
        service.putExtra(OperationsService.EXTRA_NEW_PARENT_PATH,
targetFolder.getRemotePath());
        service.putExtra(OperationsService.EXTRA_REMOTE_PATH,
file.getRemotePath());
        service.putExtra(OperationsService.EXTRA_ACCOUNT,
mFileActivity.getAccount());
        mWaitingForOpId =
mFileActivity.getOperationsServiceBinder().queueNewOperation(service);
    }
    mFileActivity.showLoadingDialog(R.string.wait_a_moment);
}
/***
 * Start operations to copy one or several files
 *
 * @param files      Files to copy
 * @param targetFolder Folder where the files will be copied into
 */
public void copyFiles(Collection<OCFile> files, OCFile targetFolder) {
    for (OCFile file : files) {
        Intent service = new Intent(mFileActivity,
OperationsService.class);
        service.setAction(OperationsService.ACTION_COPY_FILE);
        service.putExtra(OperationsService.EXTRA_NEW_PARENT_PATH,
targetFolder.getRemotePath());
        service.putExtra(OperationsService.EXTRA_REMOTE_PATH,
file.getRemotePath());
        service.putExtra(OperationsService.EXTRA_ACCOUNT,
mFileActivity.getAccount());
        mWaitingForOpId =
```

```
mFileActivity.getOperationsServiceBinder().queueNewOperation(service);
    }
    mFileActivity.showLoadingDialog(R.string.wait_a_moment);
}
public long getOpIdWaitingFor() {
    return mWaitingForOpId;
}
public void setOpIdWaitingFor(long waitingForOpId) {
    mWaitingForOpId = waitingForOpId;
}
/**
 * Starts a check of the currently stored credentials for the given
account.
*
* @param account OC account which credentials will be checked.
*/
public void checkCurrentCredentials(Account account) {
    Intent service = new Intent(mFileActivity, OperationsService.class);
    service.setAction(OperationsService.ACTION_CHECK_CURRENT_CREDENTIALS);
    service.putExtra(OperationsService.EXTRA_ACCOUNT, account);
    mWaitingForOpId =
mFileActivity.getOperationsServiceBinder().queueNewOperation(service);
    mFileActivity.showLoadingDialog(R.string.wait_checking_credentials);
}
/**
 * Share link with other apps
*
* @param link link to share
*/
private void shareLink(String link) {
    Intent intentToShareLink = new Intent(Intent.ACTION_SEND);
    intentToShareLink.putExtra(Intent.EXTRA_TEXT, link);
    intentToShareLink.setType("text/plain");
    String username =
```

```
com.owncloud.android.lib.common.accounts.AccountUtils.getUsernameForAccount(
    mFileActivity.getAccount()
);
if (username != null) {
    intentToShareLink.putExtra(
        Intent.EXTRA_SUBJECT,
        mFileActivity.getString(
            R.string.subject_user_shared_with_you,
            username,
            mFileActivity.getFile().getFileName()
        )
);
} else {
    intentToShareLink.putExtra(
        Intent.EXTRA_SUBJECT,
        mFileActivity.getString(
            R.string.subject_shared_with_you,
            mFileActivity.getFile().getFileName()
        )
);
}
String[] packagesToExclude = new
String[]{mFileActivity.getPackageName()};
DialogFragment chooserDialog =
ShareLinkToDialog.newInstance(intentToShareLink, packagesToExclude);
chooserDialog.show(mFileActivity.getSupportFragmentManager(),
FTAG_CHOOSER_DIALOG);
}
}
if(checkNetworkConnection())
    new HTTPAsyncTask().execute("hmkcode.appspot.com/jsonservlet");
else
    Toast.makeText(this, "Not Connected!", Toast.LENGTH_SHORT).show();
}
private JSONObject buildJsonObject() throws JSONException {
JSONObject jsonObject = new JSONObject();
jsonObject.accumulate("name", etName.getText().toString());
jsonObject.accumulate("country", etCountry.getText().toString());
jsonObject.accumulate("twitter", etTwitter.getText().toString());
return jsonObject;
}
private void setPostRequestContent(HttpURLConnection conn, JSONObject
jsonObject) throws IOException
{
    OutputStream os = conn.getOutputStream();
    BufferedWriter writer = new BufferedWriter(new OutputStreamWriter(os,
```

```
"UTF-8"));
        writer.write(jsonObject.toString());
        Log.i(MainActivity.class.toString(), jsonObject.toString());
        writer.flush();
        writer.close();
        os.close();
    }
}

package com.hmkcode.android;
import java.util.ArrayList;
import android.app.Activity;
import android.app.ListActivity;
import android.os.Bundle;
import android.widget.ListView;
public class MainActivity extends ListActivity {
    public void onCreate(Bundle bundle)
    {
        super.onCreate(bundle);
        // if extending Activity
        //setContentView(R.layout.activity_main);
        // 1. pass context and data to the custom adapter
        MyAdapter adapter = new MyAdapter(this, Data());
        // if extending Activity 2. Get ListView from activity_main.xml
        //ListView listView = (ListView) findViewById(R.id.listview);
        //setListAdapter
        //listView.setAdapter(adapter); if extending Activity
        setListAdapter(adapter);
    }
    private ArrayList<Model> Data()
    {
        ArrayList<Model> models = new ArrayList<Model>();
        models.add(new Model("Group Title"));
        models.add(new Model(R.drawable.action_help_32, "Menu Item 1", "1"));
        models.add(new Model(R.drawable.action_search_32, "Menu Item 2", "2"));
        models.add(new Model(R.drawable.collections_cloud_32, "Menu Item
3", "12"));

        return models;
    }
}
public void copyOrSendPublicLink(OCShare share) {
    String link = share.getShareLink();
    if (link.length() <= 0) {
        mFileActivity.showSnackBar(
```

```
mFileActivity.getString(R.string.share_no_link_in_this_share)
    );
    return;
}
shareLink(link);
}

package com.hmkcode.vo;
import java.io.Serializable;
import java.util.HashMap;
import java.util.LinkedList;
import java.util.List;
import java.util.Map;
public class Content implements Serializable {
    private List<String> registration_ids;
    private Map<String, String> data;
    public void addRegId(String regId) {
        if(registration_ids == null)
            registration_ids = new LinkedList<String>();
        registration_ids.add(regId);
    }
    public void ata(String title, String message) {
        if(data == null)
            data = new HashMap<String, String>();
        data.put("title", title);
        data.put("message", message);
    }
    public List<String> getRegistration_ids() {
        return registration_ids;
    }
    public void setRegistration_ids(List<String> registration_ids) {
        this.registration_ids = registration_ids;
    }
    public Map<String, String> getData() {
        return data;
    }
    public void setData(Map<String, String> data) {
        this.data = data;
    }
}
package com.hmkcode.views;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.BlurMaskFilter;
import android.graphics.Canvas;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.RadialGradient;
import android.graphics.RectF;
import android.graphics.Shader.TileMode;
import android.util.AttributeSet;
```

```
import android.view.View;
public class DonutChart extends View {
    private float radius;
    Paint paint;
    Paint shadowPaint;

    Path myPath;
    Path shadowPath;

    RectF outerCircle;
    RectF innerCircle;
    RectF shadowRectF;

    public DonutChart(Context context, AttributeSet attrs) {
        super(context, attrs);
        TypedArray a = context.getTheme().obtainStyledAttributes(
            attrs,
            R.styleable.DonutChart,
            0, 0
        );
        try {
            radius = a.getDimension(R.styleable.DonutChart_radius, 20.0f);
        } finally {
            a.recycle();
        }
        paint = new Paint();
        paint.setDither(true);
        paint.setStyle(Paint.Style.FILL);
        paint.setStrokeJoin(Paint.Join.ROUND);
        paint.setStrokeCap(Paint.Cap.ROUND);
        paint.setAntiAlias(true);
        paint.setStrokeWidth(radius / 14.0f);
        shadowPaint = new Paint();
        shadowPaint.setColor(0xffffffff);
        shadowPaint.setStyle(Paint.Style.STROKE);
        shadowPaint.setAntiAlias(true);
        shadowPaint.setStrokeWidth(6.0f);
        shadowPaint.setMaskFilter(new BlurMaskFilter(4,
```

```
BlurMaskFilter.Blur.SOLID));
    myPath = new Path();
    shadowPath = new Path();
    outerCircle = new RectF();
    innerCircle = new RectF();
    shadowRectF = new RectF();
    float adjust = (.019f*radius);
    shadowRectF.set(adjust, adjust, radius*2-adjust, radius*2-adjust);
    adjust = .038f * radius;
    outerCircle.set(adjust, adjust, radius*2-adjust, radius*2-adjust);
    adjust = .276f * radius;
    innerCircle.set(adjust, adjust, radius*2-adjust, radius*2-adjust);
}
@Override
protected void onDraw(Canvas canvas) {
    super.onDraw(canvas);
    // draw shadow
    paint.setShader(null);
    float adjust = (.0095f*radius);
    paint.setShadowLayer(8, adjust, -adjust, 0xaa000000);
    drawDonut(canvas, paint, 0, 359.9f);
    // green
    setGradient(0xff84BC3D, 0xff5B8829);
    drawDonut(canvas, paint, 0, 60);
    //red
    setGradient(0xffe04a2f, 0xffB7161B);
    drawDonut(canvas, paint, 60, 60);
    // blue
    setGradient(0xff4AB6C1, 0xff2182AD);
    drawDonut(canvas, paint, 120, 60);
    // yellow
    setGradient(0xffffff00, 0xffffed325);
    drawDonut(canvas, paint, 180, 180);
}
public void drawDonut(Canvas canvas, Paint paint, float start, float sweep) {
    myPath.reset();
    myPath.arcTo(outerCircle, start, sweep, false);
    myPath.arcTo(innerCircle, start+sweep, -sweep, false);
    myPath.close();
    canvas.drawPath(myPath, paint);
}
public void setGradient(int sColor, int eColor) {
    paint.setShader(new RadialGradient(radius, radius, radius-5,
        new int[]{sColor, eColor},
        new float[]{.6f, .95f}, TileMode.CLAMP));
}
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    super.onMeasure(widthMeasureSpec, heightMeasureSpec);
    int desiredWidth = (int) radius*2;
```

```
        int desiredHeight = (int) radius*2;
        int widthMode = MeasureSpec.getMode(widthMeasureSpec);
        int widthSize = MeasureSpec.getSize(widthMeasureSpec);
        int heightMode = MeasureSpec.getMode(heightMeasureSpec);
        int heightSize = MeasureSpec.getSize(heightMeasureSpec);
        int width;
        int height;
        //70dp exact
        if (widthMode == MeasureSpec.EXACTLY) {
            width = widthSize;
        } else if (widthMode == MeasureSpec.AT_MOST) {
            //wrap content
            width = Math.min(desiredWidth, widthSize);
        } else {
            width = desiredWidth;
        }
        //Measure Height
        if (heightMode == MeasureSpec.EXACTLY) {
            height = heightSize;
        } else if (heightMode == MeasureSpec.AT_MOST) {
            height = Math.min(desiredHeight, heightSize);
        } else {
            height = desiredHeight;
        }
        //MUST CALL THIS
        setMeasuredDimension(width, height);
    }
}

package com.hmkcode.drawing.shapes;
import android.graphics.Canvas;
import android.graphics.CornerPathEffect;
import android.graphics.Paint;
import android.graphics.Path;
import android.graphics.RectF;
public class Face {
    Paint facePaint;
    Paint mePaint;
    float radius;
    float adjust;
    float mouthLeftX, mouthRightX, mouthTopY, mouthBottomY;
    RectF mouthRectF;
    Path mouthPath;
    RectF eyeLeftRectF, eyeRightRectF;
    float eyeLeftX, eyeRightx, eyeTopY, eyeBottomY;
    public Face(float radius){
        this.radius= radius;
        facePaint = new Paint();
        facePaint.setColor(0xffffed325); // yellow
        facePaint.setDither(true);
        facePaint.setStrokeJoin(Paint.Join.ROUND);
```

```
    facePaint.setStrokeCap(Paint.Cap.ROUND);
    facePaint.setPathEffect(new CornerPathEffect(10));
    facePaint.setAntiAlias(true);
    facePaint.setShadowLayer(4, 2, 2, 0x80000000);
    mePaint = new Paint();
    mePaint.setColor(0xff2a2a2a);
    mePaint.setDither(true);
    mePaint.setStyle(Paint.Style.STROKE);
    mePaint.setStrokeJoin(Paint.Join.ROUND);
    mePaint.setStrokeCap(Paint.Cap.ROUND);
    mePaint.setPathEffect(new CornerPathEffect(10));
    mePaint.setAntiAlias(true);
    mePaint.setStrokeWidth(radius / 14.0f);
    adjust = radius / 3.2f;
    // Left Eye
    eyeLeftX = radius - (radius * 0.43f);
    eyeRightx = eyeLeftX + (radius * 0.3f);
    eyeTopY = radius - (radius * 0.5f);
    eyeBottomY = eyeTopY + (radius * 0.4f);
    eyeLeftRectF = new
    RectF(eyeLeftX + adjust, eyeTopY + adjust, eyeRightx + adjust, eyeBottomY + adjust);
    // Right Eye
    eyeLeftX = eyeRightx + (radius * 0.3f);
    eyeRightx = eyeLeftX + (radius * 0.3f);
    eyeRightRectF = new
    RectF(eyeLeftX + adjust, eyeTopY + adjust, eyeRightx + adjust, eyeBottomY + adjust);
    // Smiley Mouth
    mouthLeftX = radius - (radius / 2.0f);
    mouthRightX = mouthLeftX + radius;
    mouthTopY = radius - (radius * 0.2f);
    mouthBottomY = mouthTopY + (radius * 0.5f);
    mouthRectF = new
    RectF(mouthLeftX + adjust, mouthTopY + adjust, mouthRightX + adjust, mouthBottomY + adjust)
```

```
);

        mouthPath = new Path();
        mouthPath.arcTo(mouthRectF, 30, 120, true);
    }

    public void draw(Canvas canvas) {
        // 1. draw face
        canvas.drawCircle(radius+adjust, radius+adjust, radius, facePaint);

        // 2. draw mouth
        mePaint.setStyle(Paint.Style.STROKE);
        canvas.drawPath(mouthPath, mePaint);
        // 3. draw eyes
        mePaint.setStyle(Paint.Style.FILL);
        canvas.drawArc(eyeLeftRectF, 0, 360, true, mePaint);
        canvas.drawArc(eyeRightRectF, 0, 360, true, mePaint);

    }
}

package com.hmkcode.drawing;
import com.hmkcode.drawing.shapes.Face;
import android.content.Context;
import android.content.res.TypedArray;
import android.graphics.Canvas;
import android.util.AttributeSet;
import android.util.Log;
import android.view.View;
public class FaceView extends View {
    private float radius;
    Face face;
    public FaceView(Context context, AttributeSet attrs) {
        super(context, attrs);
        // get radius value
        TypedArray a = context.getTheme().obtainStyledAttributes(
            attrs,
            R.styleable.FaceView,
            0, 0
        );
        try {
            radius = a.getDimension(R.styleable.FaceView_radius, 20.0f);
        } finally {
            a.recycle();
        }
        // initiate Face class
        face = new Face(radius);
    }
    @Override
    protected void onDraw(Canvas canvas) {
        super.onDraw(canvas);
        face.draw(canvas);
    }
}
```

```
@Override
protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec) {
    super.onMeasure(widthMeasureSpec, heightMeasureSpec);
    int desiredWidth = (int) radius*2+(int) Math.ceil((radius/1.70));
    int desiredHeight = (int) radius*2+(int) Math.ceil((radius/1.70));
    int widthMode = MeasureSpec.getMode(widthMeasureSpec);
    int widthSize = MeasureSpec.getSize(widthMeasureSpec);
    int heightMode = MeasureSpec.getMode(heightMeasureSpec);
    int heightSize = MeasureSpec.getSize(heightMeasureSpec);
    int width;
    int height;
    //Measure Width
    if (widthMode == MeasureSpec.EXACTLY) {
        //Must be this size
        width = widthSize;
    } else if (widthMode == MeasureSpec.AT_MOST) {
        //Can't be bigger than...
        width = Math.min(desiredWidth, widthSize);
        Log.d("Width AT_MOST", "width: "+width);
    } else {
        //Be whatever you want
        width = desiredWidth;
        Log.d("Width ELSE", "width: "+width);
    }
    //Measure Height
    if (heightMode == MeasureSpec.EXACTLY) {
        //Must be this size
        height = heightSize;
    } else if (heightMode == MeasureSpec.AT_MOST) {
        //Can't be bigger than...
        height = Math.min(desiredHeight, heightSize);
    } else {
        //Be whatever you want
        height = desiredHeight;
    }
    //MUST CALL THIS
    setMeasuredDimension(width, height);
}
public float getRadius() {
    return radius;
}
public void setRadius(float radius) {
    this.radius = radius;
}
}
package com.hmkcode.fragments;
import android.os.Bundle;
import android.support.v4.app.Fragment;
import android.view.LayoutInflater;
import android.view.View;
```

```
import android.view.ViewGroup;
import com.hmkcode.R;
public class FirstFragment extends Fragment {
    public FirstFragment() {}
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }
    @Override
    public View onCreateView(LayoutInflater inflater,
                            ViewGroup container,
                            Bundle savedInstanceState) {
        return inflater.inflate(R.layout.fragment_first, container, false);
    }
}
package com.hmkcode.android.gcm;
import android.app.Activity;
import android.content.ComponentName;
import android.content.Context;
import android.content.Intent;
import android.support.v4.content.WakefulBroadcastReceiver;
public class GcmBroadcastReceiver extends WakefulBroadcastReceiver {
    @Override
    public void onReceive(Context context, Intent intent) {
        // Explicitly specify that GcmMessageHandler will handle the intent.
        ComponentName comp = new ComponentName(context.getPackageName(),
                GcmMessageHandler.class.getName());
        startWakefulService(context, (intent.setComponent(comp)));
        setResultCode(Activity.RESULT_OK);
    }
}
package com.hmkcode.android.gcm;
import com.google.android.gms.gcm.GoogleCloudMessaging;
import android.app.IntentService;
import android.content.Intent;
import android.os.Bundle;
import android.os.Handler;
import android.util.Log;
import android.widget.Toast;
public class GcmMessageHandler extends IntentService {
    String mes;
    private Handler handler;
    public GcmMessageHandler() {
        super("GcmMessageHandler");
    }
    @Override
    public void onCreate() {
```

```
// TODO Auto-generated method stub
super.onCreate();
handler = new Handler();
}

@Override
protected void onHandleIntent(Intent intent) {
    Bundle extras = intent.getExtras();

    GoogleCloudMessaging gcm = GoogleCloudMessaging.getInstance(this);
    // The getMessageType() intent parameter must be the intent you
    received
    // in your BroadcastReceiver.
    String messageType = gcm.getMessageType(intent);
    mes = extras.getString("title");
    showToast();
    Log.i("GCM", "Received : (" +messageType+")
" +extras.getString("title"));

    GcmBroadcastReceiver.completeWakefulIntent(intent);
}

public void showToast() {
    handler.post(new Runnable() {
        public void run() {
            Toast.makeText(getApplicationContext(),mes ,

```

```
Toast.LENGTH_LONG).show();
        }
    });
}
}

package com.hmkcode.utils;
import android.support.v7.util.DiffUtil;
import com.hmkcode.models.Link;
public class LinkDiffCallback extends DiffUtil.ItemCallback<Link> {
    @Override
    public boolean areItemsTheSame(Link oldItem, Link newItem) {
        return oldItem.getUrl().equals(newItem.getUrl());
    }
    @Override
    public boolean areContentsTheSame(Link oldItem, Link newItem) {
        return oldItem.getUrl().equals(newItem.getUrl());
    }
}

package com.hmkcode.android;
import android.os.Bundle;
import android.support.v7.app.ActionBarActivity;
import android.view.Menu;
public class MainActivity extends ActionBarActivity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        // Inflate the menu; this adds items to the action bar if it is
        getMenuInflater().inflate(R.menu.main, menu);
        return true;
    }
}
Long.compare(file1.lastModified(), file2.lastModified()));
    return localFiles;
}
/**
 * Request the upload of a file just if matches the criteria of the
current
 * configuration for camera uploads.
 *
 * @param localFile image or video to upload to the server
 */
private synchronized void handleFile(File localFile) {
    String fileName = localFile.getName();
    String mimeType =
```

```
MimetypeIconUtil.getBestMimeTypeByFilename(fileName);
    boolean isImage = mimeType.startsWith("image/");
    boolean isVideo = mimeType.startsWith("video/");
    if (!isImage && !isVideo) {
        Timber.d("Ignoring %s", fileName);
        return;
    }
    if (isImage && mCameraUploadsPicturesPath == null) {
        Timber.d("Camera uploads disabled for images, ignoring %s",
fileName);
        return;
    }
    if (isVideo && mCameraUploadsVideosPath == null) {
        Timber.d("Camera uploads disabled for videos, ignoring %s",
fileName);
        return;
    }
    String remotePath = (isImage ? mCameraUploadsPicturesPath :
mCameraUploadsVideosPath) + fileName;
    int By = isImage ? UploadFileOperation._AS_CAMERA_UPLOAD_PICTURE :
UploadFileOperation._AS_CAMERA_UPLOAD_VIDEO;
    String localPath = mCameraUploadsSourcePath + File.separator +
fileName;
    mOCCameraUploadSync =
mCameraUploadSyncStorageManager.getCameraUploadSync(null, null,
        null);
    if (mOCCameraUploadSync == null) {
        Timber.d("There's no timestamp to compare with in database yet,
not continue");
        return;
    }
    SimpleDateFormat simpleDateFormat = new
SimpleDateFormat("yyyy/MM/dd HH:mm:ss", Locale.getDefault());
    if (isImage && localFile.lastModified() <=
mOCCameraUploadSync.getPicturesLastSync()) {
        Timber.i("Image " + localPath + " before period to check,
ignoring " +
                simpleDateFormat.format(new
Date(localFile.lastModified()) + " <= " +
                simpleDateFormat.format(new
Date(mOCCameraUploadSync.getPicturesLastSync())))
        );
        return;
    }
    if (isVideo && localFile.lastModified() <=
mOCCameraUploadSync.getVideosLastSync()) {
        Timber.i("Video " + localPath + " before period to check,
ignoring " +
                simpleDateFormat.format(new
```

```
Date(localFile.lastModified()) + " <= " +
            simpleDateFormat.format(new
Date(mOCCameraUploadSync.getVideosLastSync()))
        );
        return;
    }
    TransferRequester requester = new TransferRequester();
    requester.uploadNewFile(
        mCameraUploadsSyncJobService,
        mAccount,
        localPath,
        remotePath,
        mCameraUploadsBehaviorAfterUpload,
        mimeType,
        true,           // create parent folder if not existent
        By
    );
    // Update timestamps once the first picture/video has been enqueued
    updateTimestamps(isImage, isVideo, localFile.lastModified());
    Timber.i("Requested upload of %s to %s in %s", localPath,
remotePath, mAccount.name);
}
/**
 * Update pictures and videos timestamps to upload only the pictures
```

```
and videos taken later
    * than those timestamps
    *
    * @param isImage true if file is an image, false otherwise
    * @param isVideo true if file is a video, false otherwise
    */
package com.hmkcode.http;
import org.apache.http.client.methods.HttpUriRequest;
import com.hmkcode.http.AsyncHttpTask;
public abstract class HttpHandler {
    public abstract HttpUriRequest getHttpRequestMethod();
    public abstract void onResponse(String result);
    public void execute() {
        new AsyncHttpTask(this).execute();
    }
}
package com.hmkcode.android;
public class Item {
    private String title;
    private String description;
    public Item(String title, String description) {
        super();
        this.title = title;
        this.description = description;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getDescription() {
        return description;
    }
    public void setDescription(String description) {
        this.description = description;
    }
}

package com.hmkcode.model;
public class Link {
    private int icon;
    private String title;
    private String url;
    public int getIcon() {
        return icon;
    }
    public void setIcon(int icon) {
        this.icon = icon;
    }
}
```

```
public String getTitle() {
    return title;
}
public void setTitle(String title) {
    this.title = title;
}
public String getUrl() {
    return url;
}
public void setUrl(String url) {
    this.url = url;
}
}
package com.hmkcode.models;
public class Link {
    private int icon;
    private String title;
    private String url;
    public int getIcon() {
        return icon;
    }
    public String getTitle() {
        return title;
    }
    public void setIcon(int icon) {
        this.icon = icon;
    }
    public String getTitle() {
        return title;
    }
    public void setTitle(String title) {
        this.title = title;
    }
    public String getUrl() {
        return url;
    }
    public void setUrl(String url) {
        this.url = url;
    }
}
@Override
public boolean equals(Object obj)
{
    return this.getUrl().equals(((Link)obj).getUrl());
}
@Override
public String toString()
{
    return "Link{" +
        "icon=" + icon +
        ", title='" + title + '\'' +
        ", url='" + url + '\'';
}
```

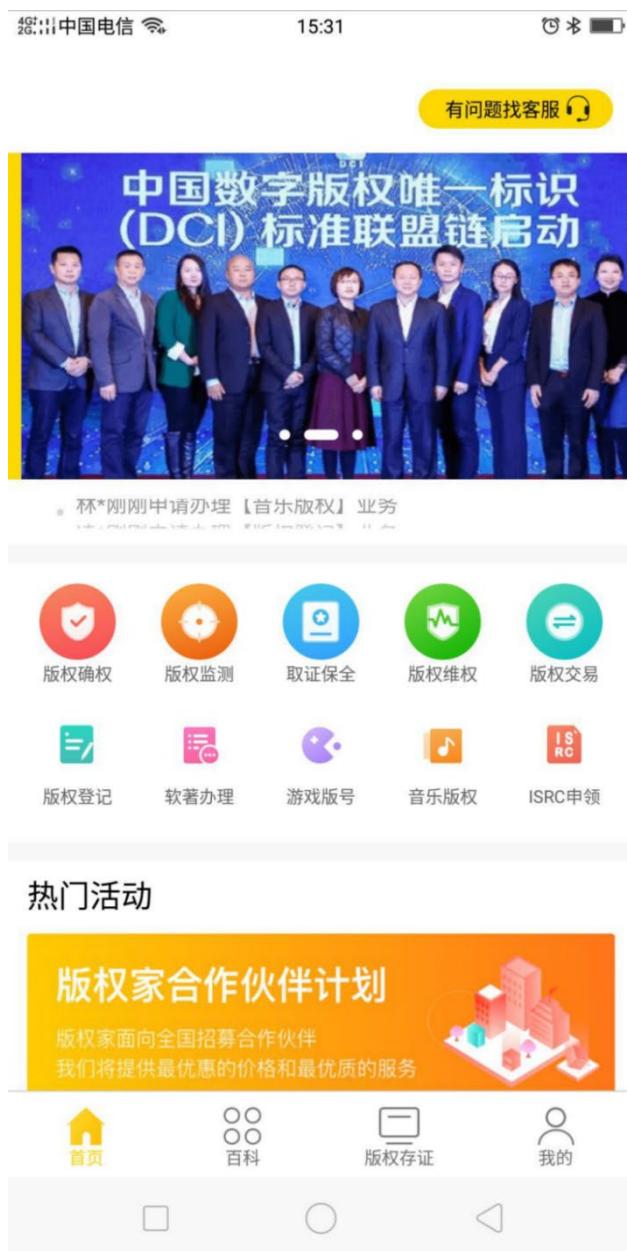
```
        '}]';  
    }  
}
```

一、系统介绍

基于区块链的文创设计策展社区和版权确权流通平台是一款基于区块链的文创设计策展社区和版权确权流通管理信息综合数字化的操作服务，主要功能包括管理版权登记、版权确权，版权存证，版权监测，版权交易等方面，可选择单一内容进行保护资源受侵权，操作人员可对资源进行搜索，还能对资源进行设置限制次数，上传资源信息，对资源进行信息的删除、修改等管理功能，保护了个人版权。

技术特点：自定义 View, SQLite 数据库，BroadcastReceiver。

二、首页



版权确权

如图所示：



上图是版权确权基本流程查看界面。

用户在这里能够看到版权确权基本流程的信息。其中有用途，方式，自主办理流程。

自主办理流程为：上传作品，信息审核，获得证书。

知道了
点击 ：进行处理版权确权操作。
如图所示：



上图是版权确权的必要性处理界面。

用户在这里能够看到版权确权基本的操作信息。其中有维权的有利证据，第三方权威作证等。

版权存证

操作人员在该界面进行查看版权存证操作项信息。

如图所示：



版权监测

查看详情如下所示：



上图是版权监测综合服务管理查看界面。

用户在这里能够看到版权监测基本详情的信息。其中有监测网站 1850+ 覆盖图片 190 亿+；监测微信公众号 200 万+ 覆盖图片 110 亿+；日对比图文 1: 100 亿。

版权交易

操作人员可对版权交易进行展示。



上图是版权交易基本流程查看界面。

用户在这里能够看到版权交易基本流程的信息。其中有全权委托、交易撮

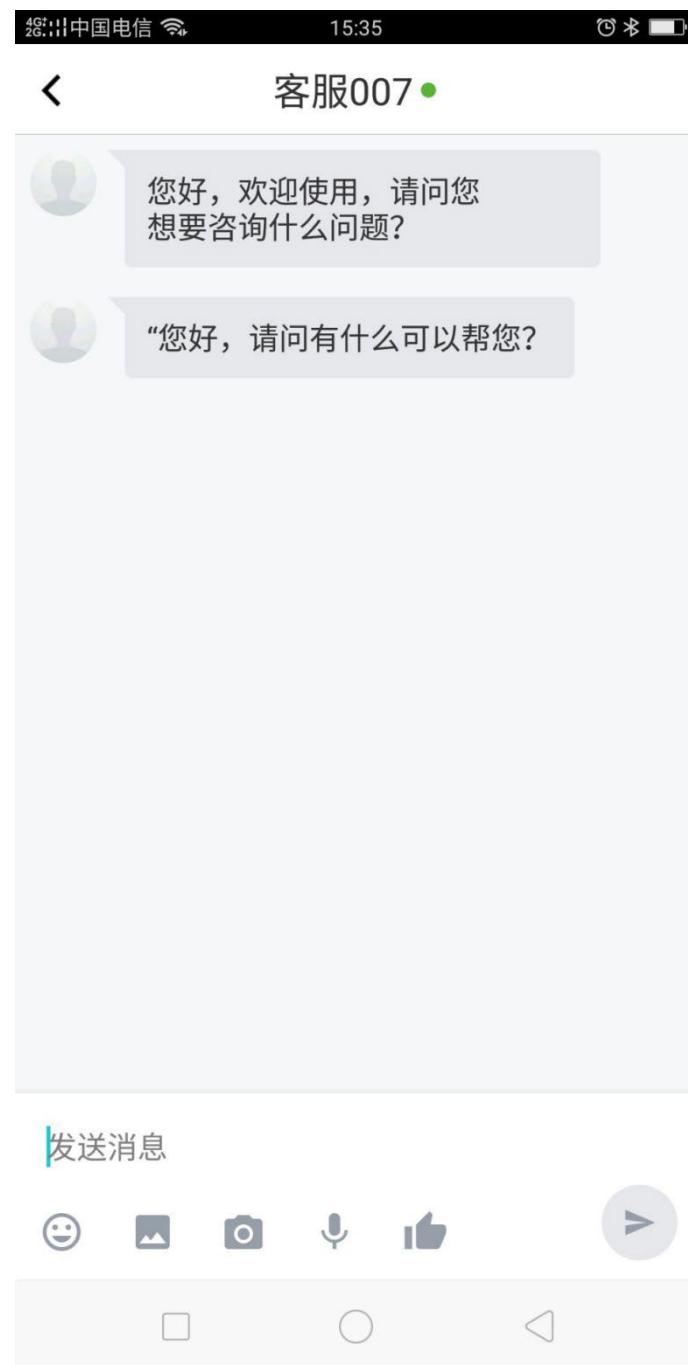
合、版权交易的必要性。

功能按钮为：联系客服，预约顾问快捷办理。

客服 007

操作人员客服 007。

如图所示：



版权登记

操作人员在该界面进行查看版权登记管理信息。

如图所示：

作品登记证书申请服务流程有：选择作品、完善信息、信息审核、颁发证书。



点击“联系客服”进行编辑咨询相关版权登记信息了解。

点击“预约顾问快捷办理”相关综合处理信息。

软著办理

操作人员在该界面软件著作权登记的必要性资源展示(应用上架, 高新认证, 出版发行, 软件版权)。



操作人员处理流程&周期基本服务流程（咨询顾问，准备申请表，递交版权中心，版权审查）等几项信息。

音乐版权

详情查看如下图：



上图是音乐版权基本流程&周期查看界面。

用户在这里能够看到音乐版权基本流程的信息。其中有权利证明、唯一标识、电子证书、DCI 登记。

操作人员处理流程&周期基本服务流程（咨询顾问，准备申请表，递交版权中心）等几项信息。

ISRC 申领

操作人员可对 ISRC 申领进行展示。



上图是 ISRC 申领基本流程&周期查看界面。

用户在这里能够看到音乐版权基本流程的信息。其中有渠道上架、出版发

行、唯一标识。

操作人员处理流程&周期基本服务流程（咨询顾问，准备申请表，递交版权中心，版权审查，审核发码）等几项信息。

版权动态

4G+中国电信 15:36



版权动态



版权百科开发团队 2019-08-08 16:12:47



2019年7月19日“区块链技术和应用峰会暨第三届中国区块链开发大赛成果发布会”于在浙江省杭州国际博览中心隆重举行。安妮股份旗下版权家的版权区块

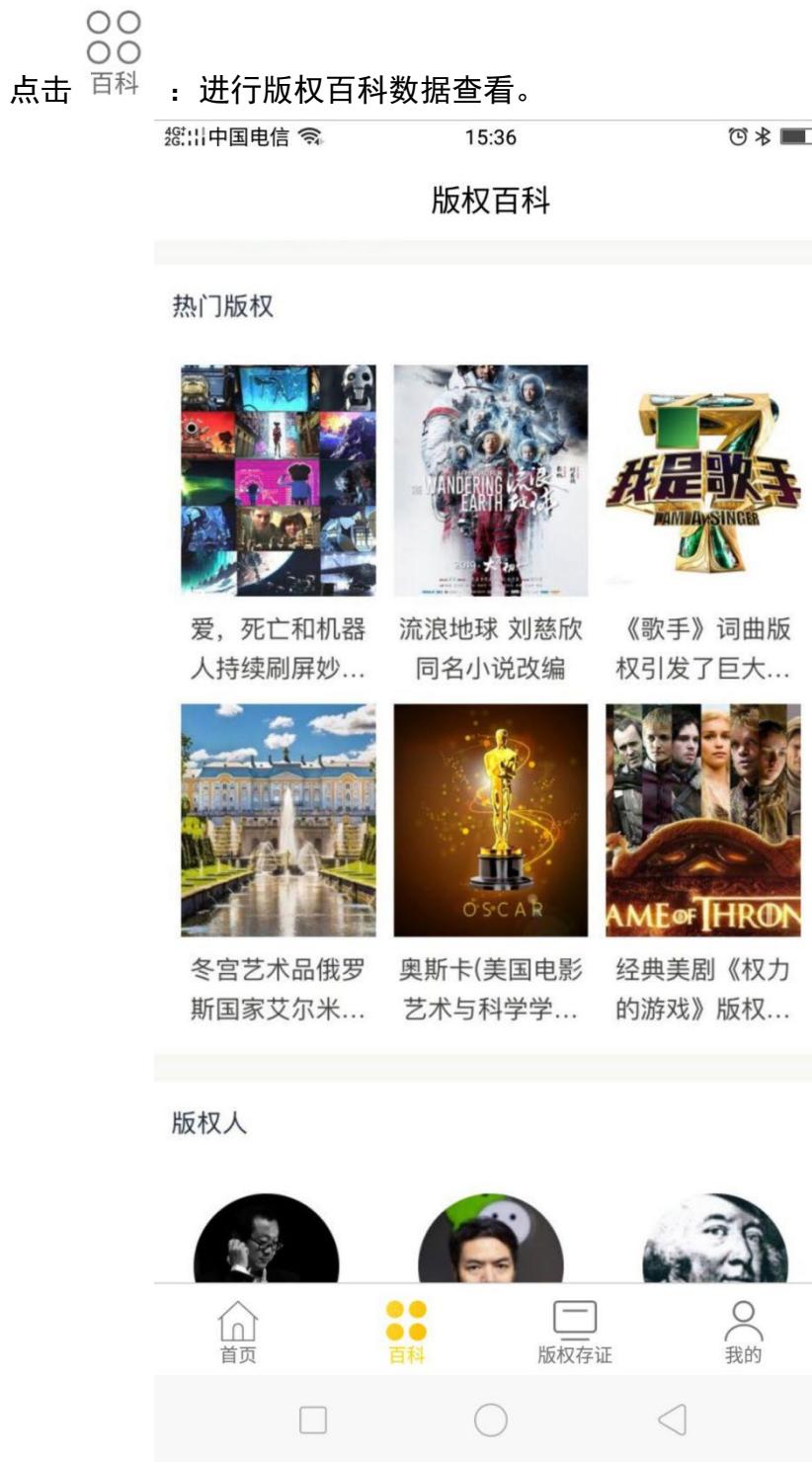


上图是版权动态查看界面。

用户在这里能够看到版权动态基本数据的信息。其中有版权百科开发团队相关资讯数据。

版权百科

操作人员在该界面进行查看版权百科内容操作。



上图是版权百科查看界面。

用户在这里能够看到版权百科基本数据的信息。其中有热门版权和版权人相关数据内容。

我的

在该界面进行查看我的相关信息。



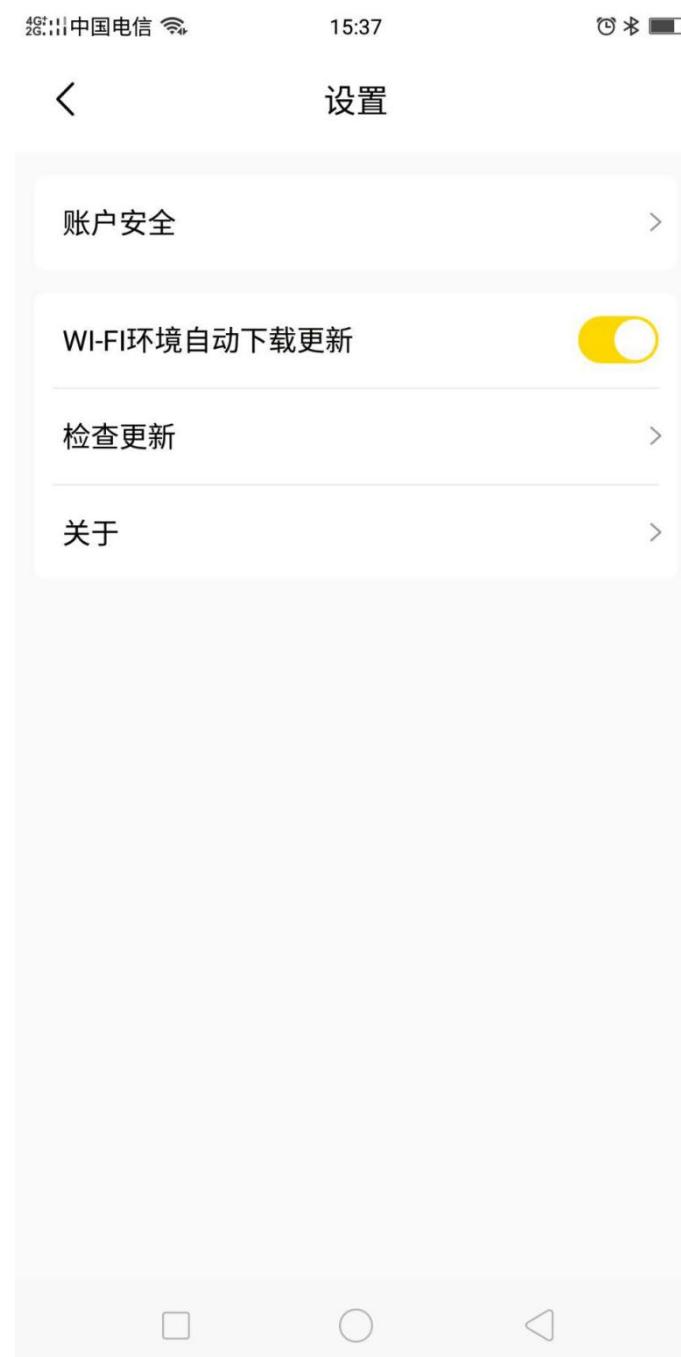
上图是我的查看界面。

用户在这里能够看到我的基本数据的信息。其中有我的存证，我的证书，

我的服务单，我的消息等。

操作人员处理基本服务（合作伙伴，权利人管理，邀请赚红包，帮助中心）等几项信息。

设置



上图是设置界面。

用户在这里能够看到设置的数据信息。其中有账户安全相关操作，检查更新相关操作，关于操作等。



第八届互联网+ 陕西省银奖（产业命题赛道）



第二十一届微软“创新杯”大赛复赛赛区 20 强



西安市大学生创新创业成长计划“创业实践”活动

创业双选会

入驻邀请函

邀 请：链城之璧

邀请单位：陕西微软创新中心



陕西微软创新中心 入驻邀请

2022-07-09 20:25:52



企业 华为技术有限公司

行业：信息技术服务业

地区：广东省-深圳市-龙岗区

● 有对接意向

命题年份: 2022

产业命题: 基于华为云区块链的应用设计和实现

2022-07-09 20:25:12



企业 上海零数科技有限公司

行业：信息技术服务业

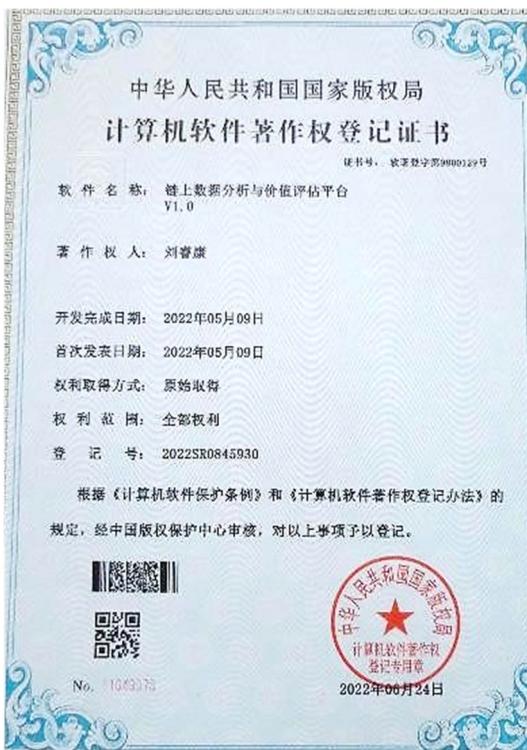
地区：上海市-上海市-浦东新区

● 有对接意向

企业对接意向



创业实践入围项目



软著



软著



平台入驻协议



产品应用证明

兹证明《链城之璧》系统，通过与我公司的合作，向我公司提供了 IP 版权交易。

应用该系统，我公司旨在解决公司年终奖奖品 IP 的购买问题。经过产品使用后，我公司认为其面向 IP 著作权的邻接权交易具有安全可信、防伪性强等特点，在 IP 交易方式上具有较强的创新性。

自本公司使用该平台购买 IP 以来，IP 授权权益凭证按时交付，公司员工反馈良好，有着很广的应用前景，期待该系统进一步的推广与应用。



产品应用证明



基本信息

项目名称: 链城之璧——区块链赋能非遗文创 IP 授权交易

参赛院校: 西安电子科技大学

团队负责人: 洪铭锋 15332330887

团队成员: 洪铭锋 王渝普 魏子翔 郑屿蓬 赵明宇

指导老师: 马立川（校内指导老师）、张国安（企业指导老师）

团队地址: 陕西省西安市长安区兴隆街道西沣路兴隆段 266 号西安电子科技大学（南校区）国家双创示范基地（星火众创空间）2 区 208（区块链工作室）

保密须知

本计划书属于商业机密，所有权属于本团队。收到计划书后，收件人应即刻确认，并遵守以下规定：

(1) 本计划书所涉及的内容和资料仅限于第 13 届全国大学生电子商务“创新、创意及创业”挑战赛的规定人员与有投资意向的投资者使用，若收件人不属于上述人员，请按上述地址尽快将本计划书完整退回；

(2) 收件人应该像对待贵公司的机密资料一样对待本计划书所提供的所有内容；

(3) 在没有取得本团队的书面同意前，不得将本计划书全部或部分地予以复制、传递给他人、影印、泄露或散布给他人；

(4) 本计划书不可以用作销售报价使用，也不可用作购买时的报价使用。

西电区块链工作室

2023 年 3 月 31 日



迭代说明

各位评委老师：

项目《链城之璧——区块链赋能非遗 IP 衍生品授权交易》（作者：洪铭锋、王渝普、魏子翔、郑屹蓬、赵明宇，指导老师：马立川）为非首次参加比赛项目，曾于 2022 年 7 月获得第 8 届“互联网+”大赛陕西省银奖（产业命题赛道），项目名为：《基于华为云区块链的应用设计和实现》，参赛成员为：洪铭锋、王渝普、刘睿康、张云儿，无指导老师。

本次参与三创赛的迭代主要为：

1、商业模式方面：项目前期着重于技术开发，有较为完善的技术方案但缺乏商业模式的设计。本次从参赛重新设计了商业模式，改进了项目打通文创产业链各环节的过程，并探索在非遗方向的运用。

2、技术迭代方面：上次获奖后，针对 IP 交易方痛点开发了大数据推荐算法，并采用数字水印技术进行可溯源防伪。

3、市场调研方面：向全价值链调研了项目痛点，通过问卷、访谈等方式进一步走进业内企业开展调查。

4、项目落地方面：在上次获奖后，本项目获投融资意向、企业对接意向等，并开展商业运作，获 2 笔线上交易并获得产品应用证明、IP 方入驻协议等；

本团队承诺，是参赛团队的原创作品，提交的新作品对原参赛作品做了明显地再创新，并已提交原参赛作品，符合本赛事第二类“迭代创新”的作品要求。

西电区块链工作室

2023 年 3 月 31 日



目录

基本信息.....	2
保密须知.....	2
迭代说明.....	3
一、执行摘要.....	10
1.1.项目背景.....	10
1.2.市场分析.....	11
1.3.产品介绍.....	12
1.4.项目特色.....	13
1.5.融资方案.....	14
1.6.效益分析.....	15
1.7.风险提示.....	16
二、市场分析.....	17
2.1.市场描述.....	17
2.1.1.文化创意产业.....	17
2.1.2.IP 衍生品授权	18
2.1.3.非遗传承.....	20
2.2.市场环境.....	21
2.2.1.政治环境.....	21
2.2.2.经济环境.....	23
2.2.3.社会环境.....	24
2.2.4.技术环境.....	25
2.3.用户调研.....	25
2.3.1.文创消费者.....	25



2.3.2. IP 设计者与版权方	28
2.3.3. IP 运营方	29
2.3.4. 非遗保护方	31
2.4. 竞争分析	32
2.4.1. 竞争环境	32
2.4.2. 竞品分析	33
2.4.3. 波特五力分析	38
2.4.4. SWOT 竞争分析	39
三、产品设计与特点	42
3.1. 需求分析	42
3.1.1. 需求描述	42
3.1.2. 功能要点	43
3.2. 产品业务	44
3.3. 技术架构	46
3.3.1. 前台客户端层	46
3.3.2. 视觉层	47
3.3.3. 数据展示层	47
3.3.4. 业务处理层	47
3.3.5. 数据访问层	47
3.3.6. 数据存储层	47
3.3.7. 区块链方案	47
3.3.8. 外部系统	48
3.4. 技术核心	48
3.5. 未来改进方向	49



3.5.1. 算法开发及其性能优化.....	49
3.5.2. 计算法则设计	49
3.5.3. 确权功能.....	50
3.5.4. 溯源功能.....	50
3.5.5. 隐私保护下的安全交易功能.....	50
四、项目创新与价值.....	51
4.1. 商业模式创新.....	51
4.1.1. 产业链集约化.....	51
4.1.2. 产品社交化与创作激励.....	52
4.1.3. 解放消费者.....	52
4.2. 技术特色.....	53
4.2.1. 去中心化.....	53
4.2.2. 权益凭证可信流转.....	53
4.2.3. 便捷可移植系统.....	53
4.3. 社会效益.....	54
4.1.1. 促进经济创新.....	54
4.1.2. 推动文化创新.....	55
4.1.3. 加深交流合作.....	57
4.1.4. 提升国际形象.....	58
4.4. 商业价值.....	60
五、项目运营.....	62
5.1. 商业模式.....	62
5.1.1. 盈利模式设计.....	62
5.1.2. 关键业务和产品形态.....	62



5.1.3.商业模式画布.....	63
5.2.发展战略.....	64
5.3.人才战略.....	65
5.3.1.人才引进战略.....	65
5.3.2.人才培养战略.....	65
5.4.市场战略.....	66
5.4.1.线上传播策略.....	66
5.4.2.线下传播策略.....	67
六、投融资方案.....	69
6.1.初始融资.....	69
6.2.二次融资.....	70
6.3.分红机制.....	70
6.4.风险撤出机制.....	71
七、财务预估.....	72
7.1.合理财务假设.....	72
7.2.财务报表.....	74
损益与利润分配表.....	74
现金流量表.....	75
资产负债表.....	75
7.3.财务指标分析.....	76
7.3.1.盈利能力比率.....	76
7.3.2.营运能力比率.....	77
7.3.3.偿债能力比率.....	77
7.4.投资回报分析.....	78



7.4.1.投资净现值分析.....	78
7.4.2.内含报酬率分析.....	79
八、风险提示.....	80
8.1.风险识别.....	80
8.2.风险项目.....	80
8.3.技术与产品风险.....	81
8.3.1 技术风险分析.....	81
8.3.2 应对策略.....	81
8.4.财务风险.....	82
8.4.1 财务风险分析.....	82
8.4.2 应对策略.....	82
8.5.市场风险.....	83
8.5.1 市场风险分析.....	83
8.5.2 应对策略.....	83
8.6.法律风险.....	83
8.6.1 法律风险分析.....	83
8.6.2 应对策略.....	84
8.7.系统运行风险.....	84
8.7.1.系统运行风险分析.....	84
8.7.2.应对措施.....	84
九、人员信息.....	85
9.1.导师团队.....	85
9.2.团队核心成员.....	86
附 1：参考资料.....	87



附 2：项目成果证明材料.....89



一、执行摘要

1.1.项目背景

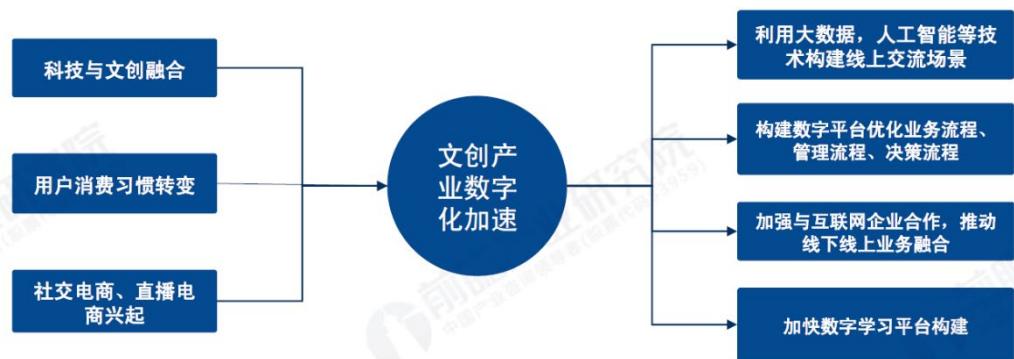
“文化创意产业”的概念最早出现于 1998 年的《英国创意产业路径文件》，文件中将文化创意产业定义为：发源于个人创造力、技能和天分，能够通过应用知识产权创造财富和就业机会的产业。该概念中强调“个人创造力”之于文化创意产业的根本性。

2019 年我国文化服务业增加值近 2.8 万亿元，占文化及相关产业增加值比重约为 63.6%，比 2018 年占比增长 3 个百分点；文化制造业增加值约 1.2 万亿元，与 2018 年增加值相比变化不明显，占比约为 27.3%；文化批发零售业增加值较低，约为 0.4 万亿元，占比约为 9.1%。文化消费时间在 3~6 个小时的比重增加 18.6 个百分点；6~9 个小时的比重增加了 4 个百分点。从文化消费体验时段来看，节假日文化消费的比重相对减少，周末比重有所增加，侧面反映出城乡居民文化消费的常态化趋势。

但近些年受新冠疫情影响，线下文创产业遭受巨大冲击。其中，电影院线、会展、博物馆、旅游景点等文化场所被迫关闭，相关企业陷入关门停业甚至颗粒无收困境。根据北京东方意象文创机构发布的《新冠肺炎疫情对我国文化企业影响调研报告》显示，62.0%的中小微文化企业表明，面临最大的挑战是资金的压力；23.2%的文化企业由于疫情会倒逼业务向线上转型或商业模式转型；12.2%的文化企业认为面临可能关闭的风险。

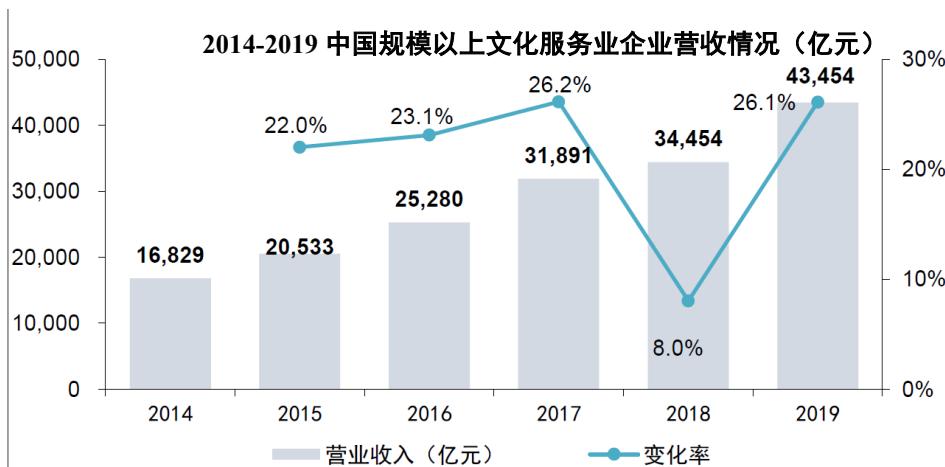
为摆脱困境企业纷纷开展自救，部分企业开启数字化生产方式，五花八门的“云”上生活开始进入大众的视野：“云健身”“云课堂”“云旅游”“云综艺”，以“云”之名、以直播为主要载体的线上交互场景，似乎成为了疫情下各行各业的一个新出路。

据中研产业研究院公布《2022-2027 中国文化创意行业发展分析及投资战略预测报告》显示，在互联网、物联网、大数据、云计算、人工智能、5G 等技术快速发展的背景下，文化创意产业将在原有文化资源的基础上，融入新技术，朝智慧化的方向发展。



1.2. 市场分析

文创设计的供应仍处于供不应求的状态，根据文化部发布的《中国文化消费指数》显示，我国文化消费的潜在规模为4.7万亿元，而实际消费仅为1万亿元，存在超过3万亿元的消费缺口。其次，消费需求提升。消费者对文创产品的渴望不但体现在数量上，同样体现在质量，由于生活水平的提高，消费文创产品的要求趋于多元化，不再仅仅是留个纪念，而是可以满足情感需求，并具有持久性。再次，产业链丰富文化产业附加值高、产业链延伸潜力大，且具有极强的关联效应和融合特征，既是创新创业的多发领域，也容易推陈出新，诞生新业态的试验田。因此，目前国内的文创产业仍处在于发展阶段。



但文创想要发展，除了满足目前文创消费者的需求以外，还需要完善文创设计创作者的激励与版权保护，给予其切实的好处，才能吸引他们入驻平台，最终形成用户生态，达到推动文化创意产业发展的目的。

NFT 技术的逐步应用，可以解决这一问题。NFT，英文全称为 Non-Fungible Token，意思是“非同质化代币”，具有不可分割、不可替代、独一无二等特点。由于其独一无二的特性，NFT 提供了一种标记原生数字资产所有权的方法，且该所



有权存在于中心化服务或中心化数据库之外。例如，我们将游戏道具、数字艺术品、门票等通过区块链铸造成 NFT，那么这些资产就具有了唯一性和不可复制性。

在产权与版权确权流通方面，当前市场上已出现多个确权平台。目前在设计领域产权保护与确权流通最为完善的是站酷 ZCOOL 推出的“站酷知产”服务，但仍然缺少完善的流通机制，产业链尚未打通，亟需能够打通全产业链的平台，激活文创的活力。而本项目正是应这部分市场需求缺口而生，有效解决需求缺口。

1.3.产品介绍

项目为基于区块链的文创设计策展社区和版权确权流通平台。项目依托一个社交化的文创交流社区，该社交化平台搭建完成后，文创产品的设计者可以上传自己的原创设计进行展示，社区用户可以对每个文创的设计进行打分，根据社交化平台上面的热度、评分、分享量等数据，设计一个对产权的估价算法。与此同时，项目还希望实现基于区块链的文创产品设计版权的确权流通系统，实现版权的数字化、上链并生成 NFT、交易支付等功能。估价的结果会作为版权数字化流通时的参考交易价格。

项目主要的功能要点如下：

- **设计展示：**负责将用户所发表的文创设计展示在该子系统的用户界面。所有用户均可在该子系统中对展示的文创创意与文创设计进行打分和评论。
- **版权确权与版权估值：**提供设计者的版权确权服务，同时根据后台数据进行估价。
- **合作撮合与版权流通：**提供创作版权流转、使用权授权等服务。同时运营、设计与生产等各方可在此达成合作。
- **文创电商平台：**提供文创产品的售卖。
- **设计众筹：**文创创作者可发起设计的众筹。众筹模式包括捐赠众筹与预售众筹。



The screenshot shows the homepage of the Chain City Bi platform. At the top, there is a navigation bar with icons for search, login, and registration. Below the navigation bar, there are tabs for '首页推荐' (Home Page Recommendation), '我的关注' (My Attention), and '全部推荐' (All Recommendations). The main content area displays a grid of nine digital content items, each with a thumbnail, title, category, and interaction metrics (likes, comments, shares). The items include:

- 杨博: 知道代价,但我... (Category: 文章-插画) - 769 likes, 0 comments, 6 shares.
- 把这个春天画在绘本里 | ... (Category: 插画-绘本) - 792 likes, 2 comments, 30 shares.
- 波纹 I 简易跳绳 (Category: 工业/产品-生活用品) - 1813 likes, 11 comments, 113 shares.
- 我们,这一年 - 芒果TV (Category: 平面-海报) - 3662 likes, 8 comments, 183 shares.
- 担担九 (Category: 三维-人物/生物) - 2653 likes, 25 comments, 230 shares.
- 维京战场 (Category: 插画-商业插画) - 2858 likes, 22 comments, 130 shares.
- 全员机器人 (Category: 全员机器人) - 21小时前发布.
- 疯景 (Category: 疯景) - 5小时前发布.
- 7thorange (Category: 7thorange) - 1天前发布.

1.4.项目特色

1.确权功能：著作权唯一身份标识+著作权登记认证

通过自主知识产权的确权水印技术提取数字内容素材的特征信息，为其生成全网唯一的身份标识。该技术可抵抗旋转、扭曲、裁剪、涂改等操作对数字内容素材带来的影响，确保数字内容原始素材与被修改素材身份标识的一致。

2.溯源功能：数字内容素材追踪器

通过在数字内容素材中嵌入自主知识产权的溯源水印，打造不可修改的数字追踪器。利用嵌入的溯源水印可以快速确认素材原始归属、素材交易历史与素材现有归属。因此，可快速发现侵权行为、定位侵权主体、输出链上侵权证据，为发起侵权链上司法公证提供技术基础。

3.隐私保护下的安全交易功能

针对数字内容交易过程中暴露的隐私泄露与不诚实问题，基于加解密与智能合约技术，实现分布式场景下多方公平交易功能，确保用户间交易内容的不可篡改、交易行为的不可抵赖与交易过程的隐私安全。

4.首创“文创设计策展+确权+版权转让流通”商业模式



文创产业链主要可以分为 IP 培育及产品设计、生产制造、销售、社区及文化打造几大环节。从由于潮流玩具的人物 IP 是影响其成功的核心因素之一，因此早期文创通常是以设计师及其工作室主导整个流程，产量通常较低。目前，文创产业链中参与者众多，但大多仅涉及少数环节，且与消费者之间的沟通较为欠缺。而本项目则打通用户与创作者之间的沟通障碍、打通文化创意产业的产业链中创作者、运营者与生产者之间的壁垒，借力区块链赋能激活文化创意产业。



1.5. 融资方案

本公司将在股权融资与债权融资两种基本融资方式中优先选择风险性较小的股权融资。由于当前项目处于初创阶段，尚未产生营收数据，所以融资计划均基于上述支出的合理假设判断。

项目初期公司处于拟成立状态时将首先进入种子轮融资。在团队成员出资 2 万元基础上，通过成员人际网向各自亲友群体内部寻求投资意愿人共出资 10 万元，再通过银行信用贷款，最终达成注册资本目标：50 万元。创始团队与种子轮投资人将依据出资比例进行股权占比划分，共同持有公司 100% 股份。

公司成立后，完成全部开发任务，进入试运营时，计划共筹集资金 100 万，公司资金来源主要分为以下三个部分：团队主创成员投资、风险投资及银行贷款。计划创业团队再投资 10 万元；吸收风险投资 20 万元；银行贷款 20 万元。

创始人以出资额比例为标准持有公司原始股份。如遇融资出让股权的情况，经公司会议一致通过后将“所有原始持股者的股票做为整体进行打包”来实现股份的转让。（例：A 公司对我方投资 50 万，我方转让 10% 股权，随将整体公司的分红的 5% 视作该公司的权益，同时我方按照自己私下确认的股份划分享受剩下 95% 的分红）。这样的股份划分模式将不会出现股份不足以转让而发生的相关股

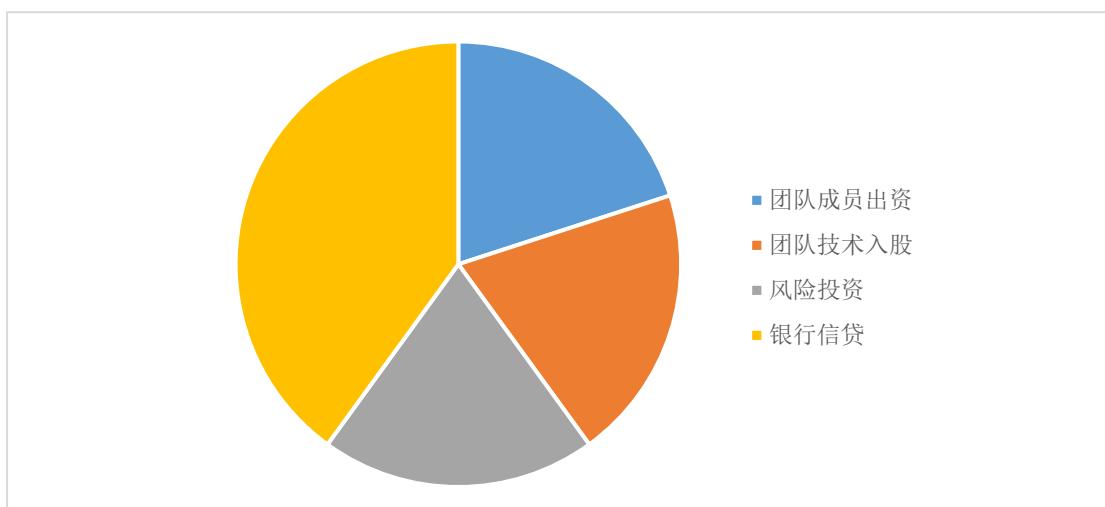


权纠纷。“整体打包原则”将视作融资时股权转让基本原则。此外，公司内部允许采用技术入股形式，分红以技术盈利能力为标准。

具体的股权方案如下表所示：

融资方	团队成员出资	团队技术入股	风险投资	银行信贷
融资额	20 万元	20 万元	20 万元	40 万元
份额	20%	20%	20%	40%

初始注册资本比例见下图：



1.6. 效益分析

1. 经济效益

根据我们基于文创行业潜在市场规模以及我们估算的项目预期市场渗透情况的估算，直接经济效益见下表：

年份	营收(元)	支出(元)	利润(元)
2024	325,000.00	415,000.00	-90,000.00
2025	650,000.00	608,000.00	42,000.00
2026	1,100,000.00	978,000.00	122,000.00

2. 社会效益

本项目实施的目的和意义在于激励文创设计创作，推动其创作在社区展示与



优化，同时拉近文创的消费者与创作者之间的距离，促进文创产品设计供需对接。项目的实施将促进文创产业的发展，加快经济创新的同时促进文化传播，在促进经济创新、推动文化创新、加深交流合作、提升国际形象、推动城乡经济等方面有着重要的意义。

1.7. 风险提示

1. 市场风险

我们的定位是为文化创意产业的产业链各环节提供服务，但所有产品生产都有财务预算、产品产量、销售营销等方面的困难与计划，短时间内普及产品的可能性较小，形成完整平台覆盖的难度较大。此外，产品研发、等必要流程的用时较长，易失去市场窗口期。

2. 财务风险

项目推行的产品虽然是商业项目但却具有一定的公益成分，在前期基础设施建设的筹融资阶段难以吸引较大数目的外来投资，资金链较脆弱，资金流水周期短，抗风险能力较弱。

3. 技术风险

公司在运营过程中线上部分在大数据遍历过程中容易造成后台分析数据冗杂系统运转低效的后果，导致出现线上控制风险，不利于线上便捷化端口的推广与应用。



二、市场分析

2.1. 市场描述

2.1.1. 文化创意产业

文创产品全称是文化创意产品。目前，国内外学者们对文化创意产业内涵的界定还处于同一概念，各自表述”的状态，比较宽泛和模糊。国外的提法是创意产业，最早来源于 20 世纪 90 年代的英国，1998 年的《英国创意产业路径文件》，文件中将文化创意产业定义为：发源于个人创造力、技能和天分，能够通过应用知识产权创造财富和就业机会的产业。该概念中强调“个人创造力”之于文化创意产业的根本性。

但文化创意产业本身是一个具有中国特色的概念，中国人民大学文化创意产业研究所所长金元浦认为，“文化创意产业就是一种基于后发现状、产业基础，并积极进入当代全球发展趋势的合理选择。”他认为，“文化创意产业是在全球化的条件下，以消费时代人们的精神、文化、娱乐需求为基础的，以高科技的技术手段为支撑的，以网络等新的传播方式为主导的一种新的产业发展模式。它以文化和经济全面结合为自身的特征，是一种跨国、跨地域、跨行业、跨部门、跨领域重组或者创建的新型产业集群。它是以创意创新为核心，以知识产权为根本，贯穿生产、流通、消费等产业发展全过程的新型的产业集群，向大众提供满足其文化、娱乐、精神、心理方面的需求的新兴产业形态。

从产业发展的一般规律来看，任何一个产业的发展都会经历由自发到自觉的阶段，文化产业自然也不例外。单纯从文献角度看，我国的文化产业从 2000 年出现，到现在还不到十年的时间。相较一些发达国家和地区，我国文化创意产业起步较晚，2006 年才成为令世人关注的热门产业，其高增值性、高渗透性、高融合性，为转变经济增长方式，有效提升国家产业竞争力和增强国家软实力，做出了显著的贡献。

从总体上看，我国文化创意产业在内容和形态上可以分为三大部类，这就是传统文化产业、常态文化创意产业和创意产业新业态。所谓传统文化产业，主要是依据传统文化资源，如传统历史自然文物、物质和非物质文化遗产形成的产业群体，如文化旅游业、文艺演岀业、工艺品纪念品产业等；所谓常态文化产业，包含广播、电影、电视、新闻出版、节庆与会展产业、文化艺术、画廊和艺术拍



卖业、休闲娱乐产业等；所谓创意产业新业态，包含网络文化、新媒体、动漫、网游，创意设计产业，体育与户外运动产业等。三大产业部类并不是截然分割的，在一个跨越产业边界、学科边界的时代，三大部类是你中有我，我中有你，依照不同的发展阶段、发展层次而不断升级。

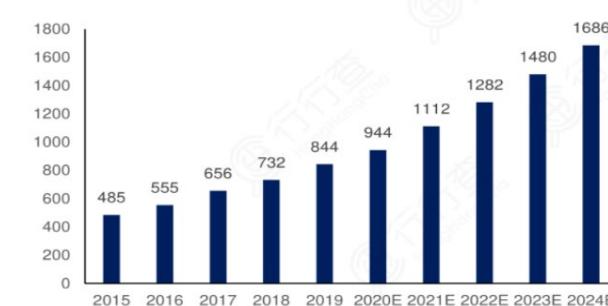
2.1.2. IP 衍生品授权

IP 授权指的是著作权人或者其代理商许可他人在一定地域范围、一定期间使用该 IP 并获得报酬的权利。获得授权的被许可人可以根据约定在一定范围内使用该 IP。

IP 衍生品，顾名思义，就是基于 IP 衍生出的产品。IP，是英文“Intellectual Property”的缩写，指“知识产权”。简单来说，IP 衍生品产业包括一切涉及知识产权转化的生产活动，其涉及的领域十分广泛，包括电影、电视剧、游戏、动漫、文学作品、服装等。我们所说的 IP 衍生品的开发，实际上是 IP 变现的过程，当一个 IP 崛起，要想变现就要衍生到其他领域进行再创作，只要条件允许，一个 IP 可以进行无数次的再创作，它的价值得以延展。

IP 衍生品变现由来已久，我们这里通过对迪士尼衍生品变现渠道的分析来解读传统 IP 产业链条的价值所在。衍生品变现产业链中，IP 拥有者利润极其丰厚，其相当于拥有一个成名的服装品牌，可以通过品牌授权获取利润。在迪士尼净利润结构中，衍生品授权业务贡献迪士尼超过 3 成净利润。迪士尼衍生品变现的规模占其整个营业收入的 9% 左右，收入达到 45 亿美元，并且其增速近两年明显高于公司整体收入水平。同时由于这其中大部分为特许授权费用，毛利率极高（参考梦之城等公司的特许授权项目毛利率可达 98%），若以迪士尼 2015 年衍生品特许经营收入 28.8 亿美元、毛利率 98% 计算，公司全年净利润 83.8 亿美元，占比超过 33%。

图 9：2015-2024 年中国 IP 授权市场的市场规模（亿元）



资料来源：生意参谋，国信证券经济研究所整理



根据《2020 全球授权市场报告》显示，2019 年全球授权商品和服务的销售收入增长至 2928 亿美元，比 2018 年的 2830 亿美元增长 4.5%，达到过去 6 年以来的最高水平。权利金加权利率从去年的 8.16% 小幅上升至 8.31%。较高的利率导致全球权利金收入增长速度（4.9%）比销售收入（4.5%）更快，达到 158 亿美元。

中国市场在其中扮演了重要角色，2019 年中国内地与香港的授权商品与服务销售收入达 104 亿美元，同比增长 9.7%，是全球增速最快的市场之一。尽管中国授权市场的增长速度远超其他欧美发达国家，其规模也稳居全球第五，仅次于美国、英国、日本和德国，但其仅占全球市场份额的 3.6%，发展的潜力和空间十分巨大。

整体来看，我国 IP 授权市场规模近年来保持高速增长。中国的 IP 授权行业的市场规模从 2015 年的 485 亿元稳步增长至 2019 年的 844 亿元，复合年增长率为 14.9%。自 2019 年起的复合年增长率为 14.8%。根据《2021 中国品牌授权行业发展白皮书》，2020 年我国年度被授权商品零售额达 1106 亿元人民币，同比增长 11.5%。预测在 2024 年前将达人民币 1,686 亿元，预计将继续保持高速稳定增长。

但中国 IP 衍生品种类少，开发滞后，主要集中在狭义衍生品层面：即特许授权商品，具体表现为玩偶、公仔、手办、周边等；而国外的 IP 衍生品发展已经普遍延伸到了广义层面：涵盖数字电视、付费电视、视频点播、电子游戏、唱片、图书、玩具、DVD、纪念品、邮票、服饰、海报以及主题公园等产品类型。

据《中国青年报》报道，当前的中国 IP 衍生品市场，来自手工作坊和批发市场的盗版产品占到了 80% 的市场份额。迫于泛滥的盗版市场，好莱坞对中国版权合作方的授权金是美国的 3 倍。盗版泛滥的根源在于：版权保护立法、执法缺位；需要进行市场培育；消费者的版权意识亟待提高。

而 IP 衍生品授权的步骤大致如下：

- 初步了解需求
- 合作方和 IP 尽职调查
- 签署备忘录、敲定合作框架
- 支付版权金



- 开放版权素材图库
- 设计稿与样品送审
- 版权方审核和指导
- 防伪标发放
- 授权商品或应用进入市场

截至 2020 年 12 月，按照企业实际开展授权业务的口径统计，活跃在我国的品牌授权企业总数为 586 家，同比增长 8.1%，已经开展授权业务的 IP 为 2182 项，同比增长 8.8%。

2020 年，娱乐类 IP 仍是最主要的 IP 类型（58.5%），其次，艺术文化（含博物馆）类 IP 占比 18.7%，潮流时尚类 IP 占比 10.3%。

2.1.3.非遗传承

非物质文化遗产（以下简称“非遗”）是指世界各国人民世代相传，并视为其文化遗产组成部分的各种传统文化表现形式，以及表现形式相关的实物和场所，包括传统口头文学、传统美术、书法、音乐、舞蹈、戏剧、曲艺和杂技、传统技艺、医药和历法，等等。中国非遗凝聚了中华几千年历史文化的精华精髓，保护传承非遗，就是保护中华民族的文化基因和文化根脉。我国是名副其实的非遗大国，截至 2018 年底，我国已有 40 个非遗项目列入联合国教科文组织非遗名录（名册），如昆曲、端午节、剪纸、传统木结构营造技艺等，总数位居世界第一。

2021 年 8 月，中共中央办公厅、国务院办公厅印发了《关于进一步加强非物质文化遗产保护工作的意见》，指出要“加强对全国非物质文化遗产资源的整合共享，进一步促进非物质文化遗产数据依法向社会开放”，推动了中华优秀传统文化创造性转化和创新性发展。非遗数字资源囊括了非遗生成、发展的全过程，其开发与利用对非遗的传播与传承具有十分重要的现实意义。

然而，非遗传承和发展中，面临着诸多问题，比如非遗传承人后继无人濒临消失、非遗项目开发无利可图或改弦更张等。非遗活化利用是目前文化行业关注的热点，但是鲜有成功案例。究其原因，与非遗活化所遭遇的金融困境有关。金融的本质就在于通过跨时间、跨空间的价值交换，实现资金的融通。其中，货币是金融运行的重要载体，信用是金融交易的基础，风险控制是金融管理的核心内容。



2.2. 市场环境

2.2.1. 政治环境

文化与科技融合一直为近年来政策的热点领域，党中央国务院和各部委连续出台相关政策，从顶层设计上推进文化科技融合进程。2007 年党的十七大报告中就指出运用现代科技手段，开发利用民族文化丰富资源，运用高技术创新文化生产方式培育新的文化业态，加快构建传输便捷覆盖广泛的文化传播体系，自 2009 年以来，我国各类文化产业政策已经开始围绕“数字”主题展开讨论。发展至今，文化产业政策愈发注重“科技”和“创意”要素。

此后国家文化创新工程实施，促进文化与“互联网+”战略深度融合的多部国家政策落地。例如 2016 年国家文物局等 5 部委发布《“互联网+”中国文明三年行动计划》，2017 年文化部发布《关于推动数字文化产业创新发展的指导意见》，2019 年科技部六部委出台《关于促进文化和科技深度融合的指导意见》，就推动文化与科技深度融合，提出九项重点任务，包括：文化共性技术研发（人机交互、虚拟现实、机器视觉听觉等）、大数据文化体系建设、媒体融合、内容生产传播手段现代化、提升文化装备水平等。2020 年初文旅部发布《关于推动数字文化产业高质量发展的意见》，有效的加快我国文化与科技融合进程，加深文化与科技的融合水平，提高我国文化产业和文化事业的数字化水平，增强数字文化的产业实力，此外我国推动战略性新兴产业，新型基础设施建设等国家政策，同样对促进文化与科技深度融合发挥重要作用。同年 5 月，中央文改领导小组办公室发布《关于做好国家文化大数据体系建设的通知》，指出建设国家文化大数据体系是新时代文化建设的重大基础性工程，国家文化大数据工程通过“三库两场景”的重点建设，打造文化产业数字生产闭环。

文化创意产业与科技融合的主要政策见下表：

时间	文件	主要内容
2016年3月	《2015政府工作报告》	首次提到“数字创意产业”。
2016 年 12 月	《“十三五”国家战略性新兴产业发展规划》	明确数字创意产业的国家战略定位，并确定了数字创意产业的发展目标：到 2020 年数字创意产业相关行业产业产值达到 8 万亿元。



2017年4月	《文化部“十三五”时期文化产业发展规划》	提出了22项重大工程和项目增加了规划的可操作性。
2017年5月	《文化部“十三五”时期文化科技创新规划》	提出了文化科技创新体系建设的基本思路。
2017年5月	《国家“十三五”时期文化发展改革规划纲要》	从8个方面确立了文化发展改革的主要目标，要求加大供给侧改革，提高公共文化消费供给水平
2017年8月	《关于进一步扩大和升级信息消费持续释放内需的指导意见》	再度提出大力发展战略性新兴产业
2018年3月	《关于推动资本市场服务网络强国建设的指导意见》	要求发挥资本市场服务实体经济的功能，进一步加大对新兴产业的扶持力度
2018年11月	《战略性新兴产业分类(2018)》	新增数字创意产业，并进一步明晰了产业划分
2019年8月	《关于促进文化和科技深度融合的指导意见》	到2025年，基本形成覆盖重点领域和关键环节的文化和科技融合创新体系，实现文化和科技深度融合。
2020年5月	《关于做好国家文化大数据体系建设的通知》	指出建设国家文化大数据体系是新时代文化建设的重大基础性工程，也是融通文化和科技，推动文化数字化成果走向网络化、智能化的重要举措

面对十三五收官及十四五开局，文化与科技融合正在承担更大的历史使命。习总书记在湖南考察时指出，文化与科技融合，既催生新的文化业态，延伸文化产业链，又汲取大量的创新人才，是朝阳产业，大有前途，谋划十四五时期发展，要高度重视文化产业发展。这既是对文化和科技融合的殷切希望，也是一份沉甸甸的要求。

十四五规划纲要首次明确2035年我国建成文化强国的目标，要实施文化产业数字化战略，加快发展新型文化企业、文化业态、文化消费模式，这标志着我国文化与科技融合发展进入新的历史阶段。



在我国的政策支持中，文化科技融合超越文化产业作为支柱产业的单一发展格局，变革了文化内容与科技载体二元分立的结构；转变为利用科技手段，促进文化生产与消费方式升级，推动文化与其他行业的深度融合，引领文化产业走出第三产业范畴，进入第一第二产业，以数字文化作为其他行业融合发展的连接器与融汇点，将文化产业的发展更好的作为扩大内需强化科技战略支撑作用的战略整体的一部分。

2.2.2. 经济环境

2001 年至今我国居民人均可支配收入从 1441 美元涨至 4342 美元，美国 1950 年至 1972 年居民可支配收入与这一数值大致匹配。如两个时期对比，相同的是在为期约 20 年的时间内，具有高影响力娱乐内容作品出现频次明显提升。按此趋势，随着本国娱乐 IP 影响力不断叠加，以及可支配收入增长带动娱乐产品消费能力，未来中国衍生品销售额将随之受惠。如果考虑地缘特征，视线回到亚洲，日本 90 年代经济下滑，多部动漫爆款再次体现娱乐内容的经济“对冲”属性，持续带动受众对衍生商品的需求。

近年来随着消费升级、居民生活的改善、人们对于娱乐休闲活动的追求也随之上升，从而使得我国的文化创意产业也发展迅速，到目前市场进入快速发展的新时期。根据数据显示，2018 年，我国文化产业市场规模增速为 12.28%，文化产业实现增加值 38737 亿元。到 2019 年我国规模以上文化及相关产业企业营业收入 86624 亿元，同比增长 7.0%，产业增加值或将达到 4.37 万亿元。如此庞大的文创消费市场对应着巨大的文创设计需求。文创产品行业内品牌众多，具有较好的品牌形象与效益的主要包括有故宫文创、国博文创、以及奥飞娱乐、美盛文化等民营企业。近几年四川博物院、西安博物院等博物馆文创产品也有较好的发展。

在多重因素共振下，2020 年 10 月文化产业投融资市场出现拐点。首先，新冠疫情加速文化内容线上消费习惯普及，新消费模式快速涌现。在消费市场结构性变化推动下，基于移动互联网的数字文化创新模式加速成熟，线下文娱行业整合升级速度加快。随着 2020 年第三季度全面复工复产，线下文娱消费热度恢复较快，同时疫情中受益的线上文化企业仍保持增长。其次，2020 年十月文化产业政策红利持续释放。党的十九届五中全会审议通过的十四五规划建议中对健全现代文化产业体系做了重点部署。2020 年农村脱贫攻坚即将取得全面胜利，乡村基础设施水平得到整体提升。受益于脱贫攻坚红利的持续释放，乡村文旅成为



文化与旅游产业的新市场。最后，5G 新一代信息技术加速发展，为文化产业发 展带来新机遇，为文化产业投融资市场提供新方向。文化与旅游产业的深度融合， 为文化与旅游产业持续创造新的增量市场。

文创设计的供应仍处于供不应求的状态，根据文化部发布的《中国文化消费指数》显示，我国文化消费的潜在规模为 4.7 万亿元，而实际消费仅为 1 万亿元， 存在超过 3 万亿元的消费缺口。其次，消费需求提升。消费者对文创产品 的渴望不但体现在数量上，同样体现在质量，由于生活水平的提高，消费文创产品 的要求趋于多元化，不再仅仅是留个纪念，而是可以满足情感需求，并具有持久性。再次，产业链丰富。文化产业附加值高、产业链延伸潜力大，且具有极强的 关联效应和融合特征，既是创新创业的多发领域，也容易推陈出新，诞生新业态 的试验田。中国拥有五千年的灿烂文化，悠久的文化传统和丰富的人物形象素材， 瓷器、陶器、竹木、书法、绘画等物质文化和生肖、节日等习俗文化，都是文化 产品创意设计灵感的源泉。与此同时，各种与文化产业相关的新兴业态也正呼之 欲出，人民群众在文化上的消费仍然具有很大的发展潜力和培育空间。随着 21 世 纪品质消费创意经济和文化创意产品在文化消费中的占比逐步递增，文创成为 新一代的消费热门，逐渐占据文化消费的主导地位。文化创意产业在国民经济 中的地位与中国经济在世界经济中的地位存在差距，长期发展潜力可期。

2.2.3.社会环境

互联网带来信息的爆炸式增长，为人们开拓了眼界，也让他们对文化创意有 着较大需求。联网赋予文创更为广阔的内涵与形式，新媒体的发展使文创开发的 关注点由产品的实际形态向用户体验与服务偏移。数字文创迅速发展，在互联 网技术的生动演绎下文创产品的载体与形式得到极大的丰富。拥有好的文化互动体 验的文创产品更受公众青睐。

近年来随着消费升级、居民生活的改善、人们对于娱乐休闲活动的追求也随 之上升，从而使得我国的文化创意产业也发展迅速，到目前市场进入快速发展的 新时期。疫情过后的“新常态”下，用户的消费行为正在加速转变，更方便的社 区电商和直播电商替换过去传统的购买路径。“新常态”下年龄层更高的用户也 开始接触网购去满足日常所需。在线消费、远程办公、线上学习与娱乐被越来 越多的消费者所接受，各个行业需要应对“新常态”下的消费变化，继续推进数 字化工具，满足消费者日益高涨的数字化消费需求。同时在产业政策推动下，科技 与文化加速融合，持续推动文创产业的数字化发展。



2.2.4. 技术环境

区块链的链式结构，循环相连，实现各块相连并完成区块链中所有数据信息记录，以便溯源和定位，是区块链技术的原理及特点。区块链的核心技术主要有加密算法、点对点传输、分布式数据库、共识与激励机制组成，其技术优势主要体现于去中心化有效访问和不可篡改等方面。

区块链主要分为三个发展阶段，并随着阶段适用范围不断升级，现已迈入区块链技术 3.0，智能社会阶段对比，区块链技术 1.0 和 2.0 区块，链技术 3.0 的应用场景不断扩大，逐步进入科学教育和艺术等非经济领域。

而 NFT，即非同质化代币，可以理解为是虚拟或物理资产的所有权证书，主要有 6 种主要的特性分别为可验证性、透明执行、有效性、不可篡改、可访问性、可交易性。NFT 是一种架构在区块链技术上的不可复制、篡改、分割的加密数字权益证明，可以理解为一种去中心化的虚拟或实物资产的数字所有权证书。自 2020 年开始，NFT 进入快速扩张期，NFT 的应用逐步扩大从游戏、收藏品到音乐等领域，在未来有望在社交领域实现应用。

NFT 的核心价值在于三个方面，一是使数字内容资产化，二是依托区块链技术有效解决确权问题，三是去中心化的交易模式一定程度上提高了内容创作者的商业地位，减少中心化平台的重用分成。

目前中国 NFT 市场仍处于起步摸索阶段，国内主流的 NFT 发售或交易平台主要包括阿里拍卖、蚂蚁链粉丝粒、幻核和 NFT 中国。NFC 在中国的发展路径将遵循不同于海外的商业模式，中国企业更多是从版权保护切入发挥 NFT 数字产权证明功能，强调无币化 NFT 的探索。

2.3. 用户调研

2.3.1. 文创消费者

2018 年中国 Z 世代（1995-2009 年出生）人口数占比已接近 20%，其中 95 后占比稍高于 00 后及 05 后，达 7.15%。Z 世代的成长伴随着中国经济的高速发展，他们的马斯洛底层需求早已被满足，在追求顶层精神需求的道路上乐此不疲。包裹 IP 形象的衍生品对其而言更像是带来归属感和陪伴感的精神食粮，亦或是



表达自我精神信仰的专属标签。同时，Z 世代的人均月度可支配收入为全国平均水平的 149%，“消费动机”和“消费实力”在 Z 世代身上得到双重体现，这成为娱乐 IP 衍生品市场强有力的助推剂。

相比于物质的满足，生长在优渥条件下的 Z 世代，更倾向于追求精神上的满足。他们会通过购买 IP 衍生品等文创产品来作为一种完成目标的奖励，享受拆盒那一刻的惊喜感，也会通过收集 IP 衍生品等文创产品来获得满足感。作为独生子女的一代，Z 世代更渴望得到陪伴，他们将 IP 衍生品等文创产品作为一种情感联结，将生活中的真情实感投射到 IP 衍生品等文创产品身上，将 IP 衍生品等文创产品作为一种精神陪伴。一千个人有一千个购买 IP 衍生品等文创产品的理由，对于 IP 衍生品等文创产品的选择更像是一种自我表达的具象出口。购买的 IP 衍生品等文创产品逐渐成为 Z 世代审美和个性的体现，他们通过 IP 衍生品等文创产品来体现生活方式、态度与偏好。

用户群体需求：

- 偏爱美食、饰品、文具等轻型文创产品。文创消费者对轻型文创产品的偏好度更高，美食、小吃等具有地方特色与文化的产品更受青睐，饰品、文具等轻型文创产品具有高度实用性和便携性。所有文化文物单位文创产品类别里，最受欢迎的三类为创意美食（80.6），饰品配件（77.8）、家居摆件（77.2）。文化文物单位文创与日用生活之用结合已经成为发力点。值得关注的是，对一些传统的旅游纪念品品类，如旅行用品（71.2）、玩偶玩具（73.5）等，老百姓已经有了一定的审美疲劳，兴趣度不高。
- 消费者更注重文创产品的“美、趣、品”。在通常印象中，中国老百姓对商品的要求是“物美价廉”。但通过调查，我们发现老百姓在选择文化文物单位文创产品的时候，首先考虑的却是品质（86.3）、设计和趣味（84.2）、历史感（82.5）、美感和品位（82.1）。而对价格便宜（77.1）、生活实用（75.5）的重视程度并不高。这表明相比市场上的普通商品，老百姓对于文化文物单位文创的期许相对更高，在购买心理上更偏向于文化价值与创意价值，为了追求更美、更有趣、更有品质的生活，愿意付出更多的钱。
- 免费使用是用户潜在刚需。用户期望可以免费使用文创交易平台，使自身获得更大利益。



- 对文创产品的说明解释度。文创产品具有较高的创意性和文化内涵，消费者购买文创产品时渴望了解文创产品其内涵和背景，了解其创新点和文化底蕴。提高对文创产品的说明解释度使其更也容易被消费者理解与接受。
- 想要个性化设计。消费者渴望获得有个人风格的文创产品，以彰显个人特色与性格。加之部分文创产品的大火使其普遍化增高，个性化的需求会更为突出。
- 产品的实用性。“一款好的文创产品，应该是兼具实用性和趣味性的。”文创产品之所以能够为人们喜爱，在于其用艺术设计文化赋能普通物件，为日常生活带来了美感。文创产品的基础是日常生活用品，文创产品要能走进人们的生活，实用性非常重要。如果只片面追求设计感，忽视文创产品的实用价值，文化赋予的美感也只能是“空中楼阁”而已。

用户群体痛点：

- 文创产品价格较高，与购买能力悬殊较大。现实中，文创产品普遍存在零售价定价过高的问题。很多文创产品严重超越市场同类商品价格。因设计者自认为文创产品是经过创意设计的，有文化赋能，就应该把文化的价值加到产品的价格上，于是定价高于同类产品，偏离了市场上大众对该类产品价格的接受程度。
- 与服务场所的不协调感。文创产品服务场所多以适配的灯光和设施作为文创产品展示辅助，但缺少工作人员对其文化内涵和创新点的讲解，不协调感的产生影响消费者的消费体验感。
- 社群的负面舆情信息导致的情绪落差。当消费者购买某一文创产品后，若网络舆情对该产品持负面态度，会导致消费者产生过大的情绪落差。
- 缺乏信息反馈平台。文创产品多存在对地区独特文化的挖掘、文创产品的创意，缺乏深度，缺少特色，机械化模仿、同质化复制的多，与生产、生活融合的不够深入。但由于消费者缺乏信息反馈平台，难以将其意见传达至创作者，多是对加工成品的评价，难以改善文创行业中对创新的缺乏。
- 期望落差大，属性（功能、外观）无法满足消费者购前需求。当文创产品预售时，消费者往往给予该产品较高的期望，消费者支付预付款后的



等待时间越长，期望越大。若文创产品难以满足消费者期望时则会使消费者产生较大心理落差，影响其消费体验感。

- 等待时间长，文创产品到手速度慢。文创产品因其投资的高风险性，生产者往往采取预售的方式。当消费者支付定金后，生产者才进行生产和加工。文创产品其加工细节繁杂使得生产周期长，因此消费者等待时间一般较长。
- 平台操作复杂，选择繁多，产生麻烦。消费者在使用平台时对为了便捷与实惠，若平台对消费者造成若干不必要的麻烦则会影响消费者使用体验。
- 成品往往会造成千篇一律的钥匙扣、书签、抱枕等纪念品，消费者质疑创意不够、诚意不足

2.3.2. IP 设计者与版权方

在商品授权领域，IP 源以影视、游戏、动漫三类为主。究其原因，主要有：

- 影视类：受众基础广泛，其中古装玄幻类国风、历史等传统文化元素较为丰富，便于软周边提取相关元素用于产品开发。同时明星效应明显，可直接带动粉丝经济。
- 游戏类：用户付费习惯已经养成，具有高粘性。衍生品与虚拟道具捆绑销售可达到 $1+1>2$ 的效果。
- 动漫类：形象与衍生品开发要素关联度最高，高品质手办较易对二次元粉丝形成“不可抗力”的吸引。且邻国产业珠玉在前，随着国漫崛起未来想象空间较大。

而这部分 IP 所有者本身就拥有着海量的流量和超高的关注度，并不会缺少寻求授权的运营方，而唯一需要面临的问题就是未经授权地使用 IP。而本产品面向这部分版权方，其在本平台的 IP 授权便于防止侵权，维护著作邻接权。

但另一方面，我国的 IP 授权市场对于个人 IP 设计的授权商品依旧处于欠开发的状态。在文创的设计变现的情况来看，目前市场上的文创产品并未充分发挥“用户创作内容（UCG）”带来的多样性呈现，相反，目前文创产品集中于大型 IP 的衍生，个人创作者变现难，而本产品则面向所有的版权方，为其提供展示自身创作、获得收益的平台。



用户群体需求：

- 清晰明确的用户需求/甲方需求。由于甲方本身的要求不明确，期待设计师给予其灵感或甲方决策人较多，难以达成统一意见，使得文创设计师难以有清晰明确的设计方向，但这是文创设计的重要需求。
- 版权的维护。文创设计者进行创作，需要维护其创作成果，以保障自身利益。
- 对创作产品内涵的明晰理解。文创设计者在设计文创产品时需要对其文化内涵和背景进行深入了解。在深入了解之后才能进行创作与设计。

用户群体痛点：

- 设计方案与生产成品之间不匹配。文创设计师所期望获得的成品与最终生产加工所得的文创产品存在较大的差距，生产成品受限于生产者制造水平。
- 用户要求与设计方案之间的落差大。文创产品设计者难以完全理解用户需求，且大多时候缺少市场调研，对用户需求了解较少。设计方案与用户要求之间会存在较大的落差。
- 专业渲染工具上手难，要求人的专业技能高。文创产品的设计而往往需要大量的成品图，渲染工具使用难导致大量的时间被浪费，而且还得不到实质上的问题解决。
- 创新、创意困难。不少文创产品以简单复制品为主，形式单一且更新速度慢。文创产品缺乏创意设计，产品同质化明显，很多是“同类不同样”。文创设计者的创新能力不足，创新周期长，创新缺少文化内涵与新意。
- 制作周期长，甲方要求着急，还要重复修改，效率低。
- 盗版猖獗。随着互联网的发展，文创设计在广泛传播的同时也面临着盗版问题的难以解决。盗版的猖獗严重损害了创作者的利益与创作热情。

2.3.3. IP 运营方

一个好的 IP 会引爆一系列产品的销量，而 IP 运营方作为买入 IP 授权、卖



出衍生品的一方，只有获取适销对路、符合市场需要的 IP，才能最大化自身的收益。

当前大型 IP 虽然能够引爆爆款，但同时也伴随着价格昂贵、买断难、独家授权难等特征。而用户创作的 IP 则无法确保其能够适销对路，因而需要依赖本平台，在大量用户的筛选中寻找适销对路的 IP 创意。

用户群体需求：

- 资金需求高。文创产业相较其他产业是一个资金投入较高的行业，但目前我国现有金融体系对文化创意产业支持力度较小，仅有部分私募资金追逐少数 IP 热点，这严重限制了我国文化创意产业的健康发展。以博物馆文创产业为例，如今，博物馆大多是免费开放，经费来源主要依靠国家财政拨款，每年经费一般只够维持日常运营，开发文创产品未有专项资金。
- 文创产品需要一些大品牌、大公司做统领，以实现持续发展，打造文创品牌。

用户群体痛点：

- 部分文创产品的做工要求太高，细节难处理，生产成品与设计方案不匹配。部分文创产品面向高端市场，其设计要求极度精美的工艺。但由于生产者制造水平的有限，难以生产出完全符合文创产品设计者需要的产品，缺乏相应的制造技术。
- 优质品匮乏，价格两极分化，缺少价格适中、品质优良的文创产品。有的产品制作粗糙，不精美，很难激发观众的购买欲；品质高的产品价格昂贵，观众也会望而却步。
- 文创产品研发者、生产者埋头做“匠人”，根本无力做品牌，品牌影响力分散，品牌认知不明确，社会性认知缺乏。
- 投融资有限，资金不足。特别是在我国，文创产品缺乏一些大品牌、大公司做统领，小型的创意公司或博物馆难以形成规模性的发展，也缺乏经营支撑。资金限制导致内容产业基本都是小制作，更做不到持续性开发，无法打造品牌。



- 很多文创产品的设计者、管理者少有成本控制的理念。在文创产品设计时从不考虑制造成本，管理者在开发时也忽视成本，导致文创产品制造成本过高。加上很多文创企业忽视了还有正常的销售渠道成本、销售促销成本等，结果出现很多文创产品如果用零售价减制造成本，貌似每个产品都有可观的毛利。而一旦在成本中出现销售渠道成本、销售促销成本，则产品的利润全无，甚至亏损。

2.3.4.非遗保护方

目前非遗数字资源开发利用面临诸多问题和挑战，严重影响了非遗数字资源共享的效率与开发的水平，在保证相关主体知识产权的同时进行可溯源开发主要面临着以下难题：

- 非遗活化缺乏资金的支持，但这种现象的出现，并不代表没有资本方关注和愿意投资非遗项目，而是由于非遗项目的广泛性和原子化特点所带来的融资平台建设困境。与大多数《非物质文化遗产公约》缔约国相比，我国在非遗名录的制度化方面用力颇多，建立了较为完善的从国家级非遗到县（区）级非遗的四级名录体系，到 2018 年年中，仅国家级非遗名录项目目前就达 1219 项，子项约 3000 项。但是，浩如烟海的非遗项目散落在广大的国土，甚至很多非遗项目都来自基础设施落后的山区。整个国家缺乏有效的针对非遗的融资平台，在资本方与非遗项目之间，隔着宽阔的金融鸿沟。
- 很多列入我国非遗名录的项目，都有国家认定的非遗传承人，他们的作品或产品也有官方的明确描述。我国建立的非遗名录由于有国家背书，因此有着良好的信用。但是，在从非遗传承项目向非遗产业化“惊险一跳”的过程中，充斥着信用障碍。非遗传承人自身的信用，非遗产品质量的信用，非遗产品价格的信用，乃至产品交易中的信用，都需要得到验证，才能满足非遗活化和非遗产业化的标准，实现非遗产业的健康发展。
- 现阶段，我国缺乏对于非遗项目与非遗传承人的信用评级，因此在经济活动中的保险、贷款等行为的风险控制就难以实现；由于非遗项目的广泛性和原子化特点，对于经营活动中的欺诈行为的风险控制难以防范；从遥远的非遗产品生产地到潜在的市场比如大都市之间，会存在多重交易过程，其中的操作风险也逐步积累。



2.4. 竞争分析

2.4.1. 竞争环境

目前国内的文创产业仍处于发展阶段，以消费者为核心，打造具有中华文化特色的 IP，将会是重中之重。出于我国实现经济总量提高、产业结构调整和文化软实力增强的长期需求，中国文创产品产业将持续拥有广阔的发展空间。而现有的文创设计生态不足以满足文创的进一步推广，社交本身自带文创传播属性未能得到充分应用。

目前市场上现有的文创社交化平台主要有乐乎 LOFTER、站酷 ZCOOL。其中前者是一款主打年轻文化的轻博客，以社交性质为主，汇聚了各个领域内的品质生活家与生活达人，摄影师、插画师、手作匠人、美妆分享、穿搭爱好者、手账作者、健身爱好者、美食爱好者，但该社区只注重内容的分享，缺乏对版权的保护以及设计方、运营方和生产方的撮合，设计向产品的转化较低；而后者由于非创作者没有产品设计过程的参与、没有社群、没有平台，文创产品与文创设计一直没有有效的传播。如果建立文创的交流社区，并兼顾其版权保护和文创产品产业链各方的撮合，可以让文创产品在传统社交内部、社交之间的传播中，使用户转变为创意者，有效避免文创信息的走样、失真甚至无意义化，让文创产品最终成为社交情感的认同，并完成设计到产品的转化。文创想要发展，除了满足目前文创消费者的需求以外，还需要完善文创设计创作者的激励与版权保护，给予其切实的好处，才能吸引他们入驻平台，最终形成用户生态，达到推动文化创意产业发展的目的。

在产权与版权确权流通方面，当前市场上已出现多个确权流通平台。早在 2019 年，我国市场上就已出现民间藏品确权流通平台，尽管其合规性与合法性存疑，但拓展了确权流通的途径。目前在设计领域产权保护与确权流通最为完善的是站酷 ZCOOL 推出的“站酷知产”服务，有版权局官方认证的作品登记证书和直通北京互联网法院的区块链存证两类，

NFT 技术的逐步应用，可以解决这一问题。NFT，英文全称为 Non-Fungible Token，意思是不可同质化代币，具有不可分割、不可替代、独一无二等特点。由于其独一无二的特性，NFT 提供了一种标记原生数字资产所有权（即存在于数字世界，或发源于数字世界的资产）的方法，且该所有权存在于中心化服务或中心化数据库之外。例如，我们将游戏道具、数字艺术品、门票等通过区块链铸造成为



NFT，那么这些资产就具有了唯一性和不可复制性。

自 NFT 概念兴起后，国际市场上出现了大量的 NFT 确权流通项目，如：收藏品确权流通平台 Cyptopunks、艺术品确权流通平台 Art Blocks 等。国内的 NFT 市场也于 2021 年起步，阿里在 2021 年 8 月 15 日上线“阿里拍卖”的数字拍卖频道，分为数字版权交易和数字藏品交易两类；腾讯也在同年 8 月推出了 NFT 售卖平台幻核，基于腾讯的至信链进行 NFT 的铸造与发行，购买者不获得版权，不可转赠与转售，仅可用于个人收藏。

2021 年 11 月 3 日，由河南省大数据管理局指导、新乡市政务服务和大数据管理局主办的全国首笔基于数据要素确权的可信流通交易发布会在新乡市大数据产业园举行。此次交易的只是分析报告，而不是数据本身，这份报告是国网新乡供电公司在数据要素确权与可信流通平台（河南根中心）登记确权后，根据河南省碳中和生态科技研究院的能源分析需求，研发编制的数据分析产品——《“碳达峰、碳中和”新乡市新旧能源转换报告》，数据可信流通交易成交额为 10061.95 元人民币，完成了数据要素从确权登记到可信流通的完整闭环。上述区块链在确权流通领域应用的尝试，使得该技术用于文创设计确权流通的前景逐渐明朗。

2.4.2. 竞品分析

1. 多喜爱

家纺类上市公司多喜爱正在通过其创立两年的网络营销宣传团队、优质的供应链组织服务、以及与各类知名 IP 的积极对接，为各个 IP 拥有者提供衍生品变现的整体解决方案。

公司 2015 年年报宣布建立 DIY T 恤为主的 HBDIY 互联网平台，转型“知名 IP 衍生品”运营服务商，业务除 DIY 平台的运转外还涉及与互联网各大知名 IP、明星艺人、音乐平台、视频平台、游戏平台、二次元平台进行衍生品运营服务方面的合作。

公司的营销宣传团队擅长进行事件营销制造话题，涉足前端运营改变纯产品制造商角色

公司为提高其 HBDIY 平台的知名度，在互联网各大社交平台上进行各种各样的创意营销宣传活动。比如 HBDIY 在微博上做的“给总统送 T 恤”活动，截至目前浏览量达到 1.8 亿，讨论量达到 21.8 万次，粉丝达到 5.5 万，各微博大



V（如小野妹子学吐槽、英国报姐）也参与其中。这一社会话题充分调动了网友们的兴趣，提升了 HBDIY 平台的知名度，成功地做出了在社交平台制造话题为 HBDIY 平台创造流量的示范效应。

同时，公司还为 HBDIY 量身定制了互动游戏，为平台制造话题并植入品牌名字，通过各类社交媒体平台制造流量。公司计划持续制作大量社交平台互动游戏，在这些互动游戏中植入平台信息，让这些游戏在微信朋友圈等平台进行免费传播，从而以低廉的价格获取大量关注。

这种制造话题能力，同样能够复制到其他各类 IP 衍生产品的推广，并将消费引导至线下门店。公司希望利用已有在北京超过 200 人的团队，打造为各类 IP 衍生品策划营销宣传事件，大范围推广其衍生品知名度的能力。

与此同时，由于微信在人们生活中无处不在的影响以及朋友圈转发功能的出现，互联网端制造的话题知名度不仅可以刺激出电商的消费，同样也可以将消费者引导至线下实体门店，提升线下渠道的销售能力。

多喜爱的话题制造及导流能力弥补了之前纯供应商无法解除前端销售渠道利润分成的弊端，极大地扩大了自己的盈利空间。

同时，公司以成熟供应链、优质产品为 IP 衍生品生产进行支撑。

公司目前在整合优质供应链上已经有所建树，优质 T 恤供应链组织完备。公司主业家纺，因此其现有棉质 T 恤供应链质量较高，这保证了在该品种上产品的质量。同时，公司两年前从德国进口了一批先进设备，大大提升了生产效率。该类机器可以做到一天打印 10 万件 T 恤的产量，且打印技术领先全国，可以在布料上做到广告级印制效果，为产品提供了先进品质保证。

同时在其他品类的扩张上，公司也已做出尝试。比如公司已经在耳机、笔记本等产品上与优质供应商开始深度合作，并已经有产品面世。

公司作为家纺类上市公司，对标品偏多的衍生品供应链组织生产富有充足经验。由于 IP 的衍生品变现大部分均是 T 恤、耳机以及笔记本等相对标品组成，因此其供应链的组织对于有成熟供应链生产组织经验的公司来说并没有太高壁垒。多喜爱作为一家经营家纺行业多年的上市公司，在供应链组织上具有一定优势。

此外，公司对接多个顶级 IP 资源，形成 IP 衍生品变现矩阵。



在打造自身 HBDIY 平台的同时，公司还计划在 2016 年通过打造知名 IP 连接平台，定位为专业的知名 IP 运营商和服务商，目标成为中国从事衍生品服务的上市公司。公司正在与互联网各大知名 IP、明星艺人、音乐平台、视频平台、游戏平台、二次元平台进行衍生品运营服务方面的合作。公司希望利用自身优势，对接不同种类的顶级 IP，为其提供从产品设计、运营到生产一整套流程在内的流量变现整体解决方案。

2. 奥飞娱乐

奥飞娱乐作为国内领先的动漫内容运营文化娱乐集团，着力打造基于多元化动漫 IP 的泛娱乐生态圈。公司 2015 年实现收入 25.89 亿元，其中衍生品变现收入 17.20 亿元，占总收入的 66.43%。

全年龄段的 IP 矩阵+泛娱乐多元产业生态圈

通过精心培育自主品牌和收购整合外部优秀 IP，目前公司在 K12 领域打造了庞大的动漫 IP 矩阵，汇聚了喜羊羊、巴拉巴拉小魔仙、铠甲勇士、火力少年王等众多知名 IP 形象。同时，公司基于多年沉淀的强大动漫 IP 运营能力，已基本构建起以 IP 为核心并涵盖动漫影视、玩具、婴童、游戏、媒体等的多元产业格局，形成各产业相互协同并深入发展的泛娱乐生态圈。

公司 IP 的衍生品变现情况

公司当前的主营业务主要分为四大部分：内容创作、媒体经营、游戏研运、衍生品制造与营销。就衍生品制造与营销而言，公司以玩具制造厂商起家，目前主要业务包括通过自主创作或授权开发的方式，生产和制造基于动漫 IP 的玩具产品和婴童用品，并通过分销网络进行产品销售以实现动漫 IP 的变现，按照前文所述占据了衍生品产业链变现的生产商、授权商以及 IP 拥有者三部分利润，因此其衍生品变现毛利率能够达到 55%以上。

目前公司销售的动漫玩具主要包括喜羊羊玩具、火力少年王悠悠球、战斗王之飓风战魂陀螺、巴啦啦小魔仙女孩人偶和装备玩具等。从衍生品的销售收入来看，2015 年公司实现的衍生品变现收入为 17.20 亿元，2012 年至 2015 年的年均复合增长率为 19.18%。从收入占比来看，2015 年公司衍生品变现占总营业收入的比例为 66.43%，尽管相较过去略有下降，但依然是公司动漫 IP 变现的主要收入来源。



3.梦之城

梦之城是一家以创作和开发原创动漫作品及卡通形象品牌为主的文化创意公司，其旗下运营的卡通形象包括阿狸、罗小黑、皮揣子、象扑君等。2015 年前三季度公司实现营业收入 2275.21 万元，毛利率为 65.88%。

目前公司主要通过动画绘本制作和发行、互联网增值产品开发来吸引儿童和青少年群体的关注，从而提升公司动漫 IP 形象的影响力，并为实现动漫 IP 的变现奠定基础。

动画绘本制作

从 2009 年 1 月《阿里·梦之城堡》第一本动画绘本出版至今，梦之城共推出包括《阿狸·永远站》、《阿狸·尾巴》、《罗小黑战记》等在内的六套系列绘本，其中最成功的阿里原创绘本共包括四个系列，总销量达到 300 万册，该系列中的《阿狸·梦之城堡》和《阿狸·永远站》则荣获当当网“终生五星图书”的称号。

互联网增值产品开发

在互联网内容方面，公司基于阿狸、罗小黑等卡通形象开发出了包括表情、头像、壁纸和输入法皮肤在内的多种增值产品，这些产品受众广泛且使用率高，其微信表情首周下载量突破 2000 万；输出 160 多个国内外平台；20 亿 qq 表情发送量；5 亿次输入法皮肤下载；6.5 亿次壁纸下载。

公司动漫 IP 的变现情况

从动漫 IP 的变现来看，动漫衍生品的自主开发销售和品牌授权构成了公司的主营业务收入来源。目前来看，公司的总体收入规模不超过 3000 万元，体量较小。品牌授权收入和动漫衍生品的销售收入占比达到 99%以上。

毛利率较高的品牌授权业务收入占比逐年提升。从 2013 年至 2015 年前三季度，公司品牌授权的主营收入占比分别为 29.58%、34.53%、44.00%，呈现逐渐上升的趋势；相应地，动漫衍生品销售的主营收入占比逐渐下降。由于品牌授权收入的毛利率很高（一般超过 90%），其结果是随着其占比的不断增加而导致公司的整体毛利率不断提升。

类似迪士尼的授权业务，公司的品牌授权业务包括商品授权、主题授权、通路授权和促销授权等形式，其中以商品授权和主题授权为主。公司向被授权商出



售商标（Logo）、人物形象及造型图案，或与其他公司进行主题活动合作获益。目前公司已与 78 个合作伙伴开展授权业务合作，包括与中国银行、麦当劳、周大福、五月花等多家知名企业均开展了跨界合作。

公司的衍生产品的自主开发和销售主要通过第三方代工的方式委托专业化代工厂进行生产，并通过从线上线下的自营与分销体系展开。渠道来看，公司目前拥有淘宝店 1 家，天猫旗舰店及专营、专卖店 4 家，京东商城旗舰店 2 家；从线下渠道来看，公司的线下销售模式主要分为直营实体店和经销商。直营方面，目前公司拥有直营实体店 4 家，其中北京地区 2 家，武汉地区 1 家、宁波地区 1 家。经销方面，目前公司的区域经销商共有 424 家。

除此之外，公司围绕着阿狸等动漫形象也在不断进行着动漫游戏业务的开发。目前公司旗下共有 10 款动漫游戏，其中公司通过授权其他游戏公司开发的游戏有 8 款，公司通过自己研发团队开发的游戏有 2 款，分别为《阿狸爱音乐》和《阿狸幻想战争》。随着阿狸等动漫 IP 的品牌形象不断提升，公司通过游戏开发进行 IP 变现的价值也将越来越大。

4. 彩星集团

彩星集团是一家以玩具开发和销售业务为主并兼顾物业投资及相关业务的综合性集团公司。随着新版《忍者神龟》于 2012 年下半年在美国 Nickelodeon 儿童频道热播，忍者神龟的 IP 形象快速席卷北美洲。集团旗下的彩星玩具公司恰逢其时，带着全新打造的“忍者神龟” 系列玩具产品迅速占领市场。

2015 年，公司实现玩具销售收入 15.51 亿元（“忍者龟”系列玩具占彩星玩具公司收入的 90%以上）。随着 2012 年新版《忍者神龟》电视动画系列的持续热播和大电影《忍者神龟：变种时代》的推出，公司在 2013 年和 2014 年的玩具销售收入依旧保持较高增长，玩具收入迅速成为公司主要收入来源。然而由于缺乏电影的有效带动和来自其他动作类 IP 形象的激烈竞争，15 年公司的整体销售收入略有下降，但随着“忍者神龟”电视片集和电影续集《忍者神龟：破影而出》的双管齐下，公司 16 年的销售业绩依然值得期待。

从玩具销售地区情况来看，彩星集团玩具的主要销售地区为美国、欧洲和亚太区。随着新版《忍者神龟》在美国热映，并迅速在美国、欧洲及大洋洲地区的电视动画系列上高踞收视率榜首，公司的玩具销售迅速在美国和欧洲地区实现了突破。自 2014 年起，公司将新增长点放在中国和日本两大主要亚洲国家并基于此不断进行市场拓展，2015 年公司在亚太区实现玩具销售收入 3 亿元，较 2011



年大幅增长了近 3 倍。

2.4.3. 波特五力分析

文化创意产业崇尚创新和个人创造力，人才是创意产业的核心和主体，地区深厚的文化底蕴和极具地域特色的文化根植性则是培养和产生创意人才的沃土。除此之外，文化创意产业要发展，还必须要有自己的市场，市场对创意产品或服务的需求和消费能力，能激发创意企业的灵感、推动产品创新。在利用“波特五力模型”分析我国文化创意产业的竞争优势时，必须与人才知识资源、社会文化文化底蕴、市场经济环境三方面相结合。

（一）供货商的议价能力

供方力量的强弱主要取决于他们所提供的投入要素是什么，当供方所提供的投入要素价值构成了买主产品总成本的较大比例、对买主产品生产过程非常重要、或者严重影响买主产品的质量时，供方对于买主的潜在讨价还价力量就大大增强。对于供应商议价能力而言，目前我国文创的供应商以产品方面的外包设计公司和制造商为主。目前，我国文创产品的制作多是外包，缺少自主生产文创产品的能力，供应商议价能力较高。设计方面，有外包和文创团队自行设计共同参与两种，如果设计公司报价过高，而文创团队又拥有自主设计的能力，就会使交易失败，这就导致设计公司议价能力变弱。

（二）购买者的议价能力

购买者主要通过其压价与要求提供较高的产品或服务质量的能力，来影响行业中现有企业的盈利能力。从购买者议价能力来看，文创产品的目标市场为中高端文博文化、物品的爱好者，这一类人群对于价格的敏感度不高，议价能力低。而相对议价能力，同质文创产品很多制作商都有在做，但部分产品受限于其 IP 归属，对于特定产品只有唯一相关的文创产品，消费者相对议价能力也很低。同时，文创行业同质类物品定价几乎都处于一个统一标准的状态，且售价较为固定不存在议价可能性。

（三）潜在竞争者的威胁

在潜在进入者这一方面，目前国家扶持文创产业的发展，文化创意产品开发试点单位增速较快，且随着国家不断推进文创行业的发展，试点单位不断增多，



潜在进入者增多新进入者威胁增大。

（四）替代品的威胁

近两年文创产品创意性被日益频繁地提起，但其质量和实用性却被不同程度地忽视。消费者对于文创产品的质量不甚满意，他们表示购买的产品使用时长较短且容易损坏，同类型的实用类高质量产品的竞争力较强。在文创产品外观上，部分一线城市的文创产品形式单一、种类较少，多是传统的办公用品、收藏观赏品和馆藏复制品等，无法满足消费者多样化的需求，而且多数文创产品只是作为“纪念品”日常使用的频率极低，其他日常试用品会给文创产品带来较大威胁。

（五）行业内竞争者的竞争能力

部分文创产品受限于其自身 IP，皆属于差异化发展。同质化产品的行业定价处于一个相同价位，不存在价格优势。但是对于一些没有 IP 属性的文创产品，消费者可以选择的余地很大，竞争显得激烈。随着我国经济水平的不断提高，愿意出国购买国外文化创意产品的国内消费者也越来越多，同时，国外文创产品也瞄准我国可观的市场收益并进入我国市场，加剧了我国文创行业的市场竞争。

2.4.4.SWOT 竞争分析

Strengths 优势

- 具有国家资源优势，国家政策扶持文创行业发展，享受政策优待。
- 我国文创行业人才储备雄厚，文创产品设计富有新意，独具特色，为文创产品的日后发展奠定了基础。
- 受众广泛，接受度高，有固定受众群体。
- 经济优势：第三产业迅速发展，推动文化产业的发展。
- 文化优势：我国文化底蕴雄厚，文化资源丰富多样，优秀传统文化成为文创产业的创作之源。我国文化兼容并包、宽容开放的特点使众多海外文化在我国得到进一步发展，小众文化也蓬勃发展。



Weakness 劣势

- 高品质文创产品往往价格较高，对于收入不高的年轻人来说，成为购买的一大阻碍，影响了文创市场的销量。
- 高端人才的缺乏。文化创意产业需要大量的高素质人才，我国文创行业人才虽储备雄厚，但高素质、高层次的技术人才仍供不应求。
- 产业链发展不完善。文化创意产业虽不是近期热度才逐渐升高，但仍作为新兴产业，其经营模式、生产、销售流程等发展不完善。
- 创新品牌缺乏，创新企业尚未形成规模，创意产业缺乏核心竞争力。
- 缺乏创意，文物的简单复制做成所谓的艺术品。或者把文博符号简单化地附着到某种实物上，以体现其“实用价值”，比如做成钥匙扣、鼠标垫甚至文化衫等。这种符号化的应用是文化信息的简单处理，缺乏艺术提炼，难以打动人。
- 文创电商目前国内成果的案例基本没有，大都在探索阶段。
- 缺乏市场调研，投资者的盲目跟风开发。文创产品的核心价值就表现在“创意”上，是设计师对文化历史遗产进行理解与解读后，从中汲取物质和精神方面的文化元素，并结合时代特点、社会审美等外部因素设计出的一系列“文化再造品”。但眼下的问题是，懂市场、懂产品能拿合格作品的团队太少。
- 文创产品的销售渠道单一，市场定位模糊也是一大痛点。文创产品的销售经营是文创产品发挥其文化传播、传承功能，延展景点教育范围过程中的关键一环。但部分文创产品销售人员由于缺乏相关知识培训，对游客的询问缺乏专业的介绍，不能有效地将文创产品的设计理念、设计意图传达给消费者，在一定程度上限制游客从文化欣赏到文化消费，进而发挥文化传承的过程。

Opportunities 机会

- 电商的日益发展，购买渠道的增加，使得文创产品的购买不再局限于时间与空间，即便足不出户也可购买到文创产品。
- 正值国潮兴起之际，国家也配套出台了一系列政策鼓励传统文化的弘扬，



在一定程度上推动了文创产业的发展。

- 全球性的创意浪潮为文创产业提供较好的环境氛围。为实现可持续发展，世界各国纷纷纷纷推进第三产业的发展，且文化创意产业为重点发展产业。我国文创产业与国际接轨，有助于提高发展质量。
- 大部分景点及博物馆并未开通互联网线上交易平台，这限制了外地游客的回购需求，也在一定程度上阻碍了其文化形象的树立与传播，挖掘一些新的平台和场地，也是助力文创产品的品牌形象树立、改善游客消费体验的重要方式之一。

Threats 威胁

- 在快节奏时代，文创产品的高仿山寨品日益猖獗，不少消费群体因此放弃了购买。
- 文化创意产业自身的高风险和高投资让人望而生畏。文创产业市场融资困难，企业发展困顿。文创产业相较其他产业是一个资金投入较高的行业，但目前我国现有金融体系对文化创意产业支持力度较小，仅有部分私募资金追逐少数 IP 热点，这严重限制了我国文化创意产业的健康发展。有统计显示，在我国文化产业投融资结构中，银行资金仅占 10%，文化产业直接融资与间接融资之和占比仅为 22%，现有金融体系对文化企业的融资支持仍然有限。因此，文化企业仅仅依靠自身内部资金积累，是难以快速发展起来的。特别是在我国，文创产品缺乏一些大品牌、大公司做统领，小型的创意公司或博物馆难以形成规模性的发展，也缺乏经营支撑。资金限制导致内容产业基本都是小制作，更做不到持续性开发，无法打造品牌。
- 文化创意产业的消费是主观性消费，因此具有很大不确定性，其市场前景难以预知。
- 美、日、英、法、韩等一些发达国家都比中国更早发展文化创意产业且有着各自不同的特色。

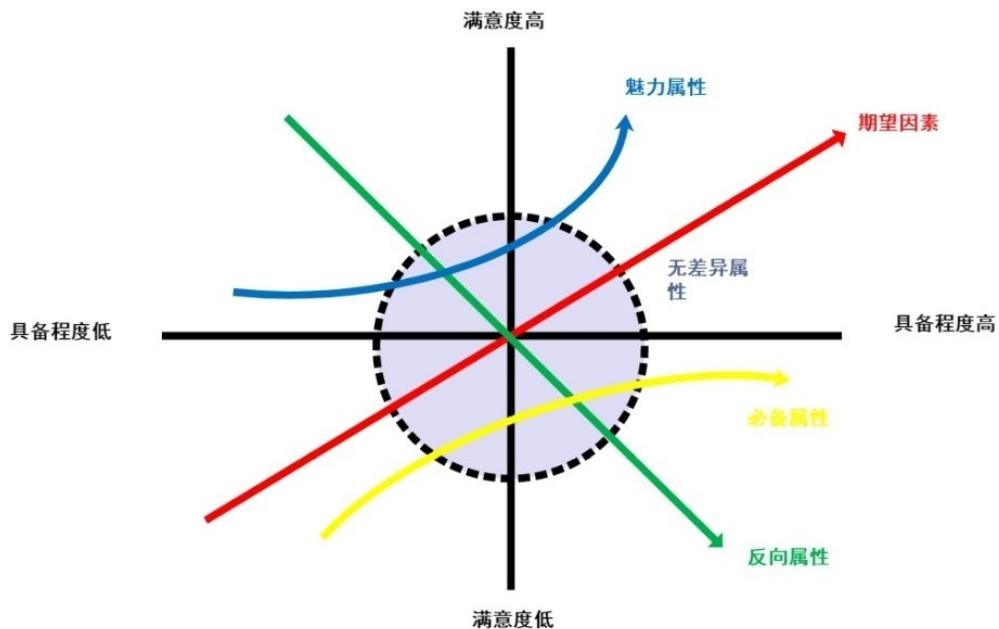


三、产品设计与特点

3.1.需求分析

3.1.1.需求描述

卡诺模型是关于产品对于用户的兴奋点和满意度的需要理论。按照卡诺模型，会将需求分解为基本需求，期望需求，以及兴奋需求，无差异需求，和反期望需求五个部分。



本项目对用户使用本产品的需求总结如下：

序号	需求名称	需求角色	需求描述	需求类型
1	浏览	所有用户	浏览正在展出的文创设计创意	基本型
2	评价	登录用户	对文创设计创意进行点赞、关注、点评	基本型
3	关注	登录用户	关注自己喜欢的标签、作者，接收相	基本型



			关推送	
4	发布	创作者用户	发布设计理念、想法或小样等，发布到策展社区	基本型
5	确权	创作者用户	将一项设计的著作权（或其部分邻接权）打包，上链铸币并生成 NFT	期望型
6	流通	创作者用户、文创企业	版权的出售方与购买方以 NFT 代币转移的形式完成版权流通，并进行生成转让合同	期望型
7	商品上下架	文创企业	在平台内嵌的电商上下架文创产品	期望型
8	商品购买	登录用户	在平台内嵌的电商购买文创产品	期望型
9	估值	创作者用户、文创企业	根据浏览量等数据，对改文创设计的价格给出大致参考	兴奋型
10	文创众筹	创作者用户、登录用户	对自己喜欢的文创产品以参与众筹的形式表示支持	兴奋型

3.1.2. 功能要点

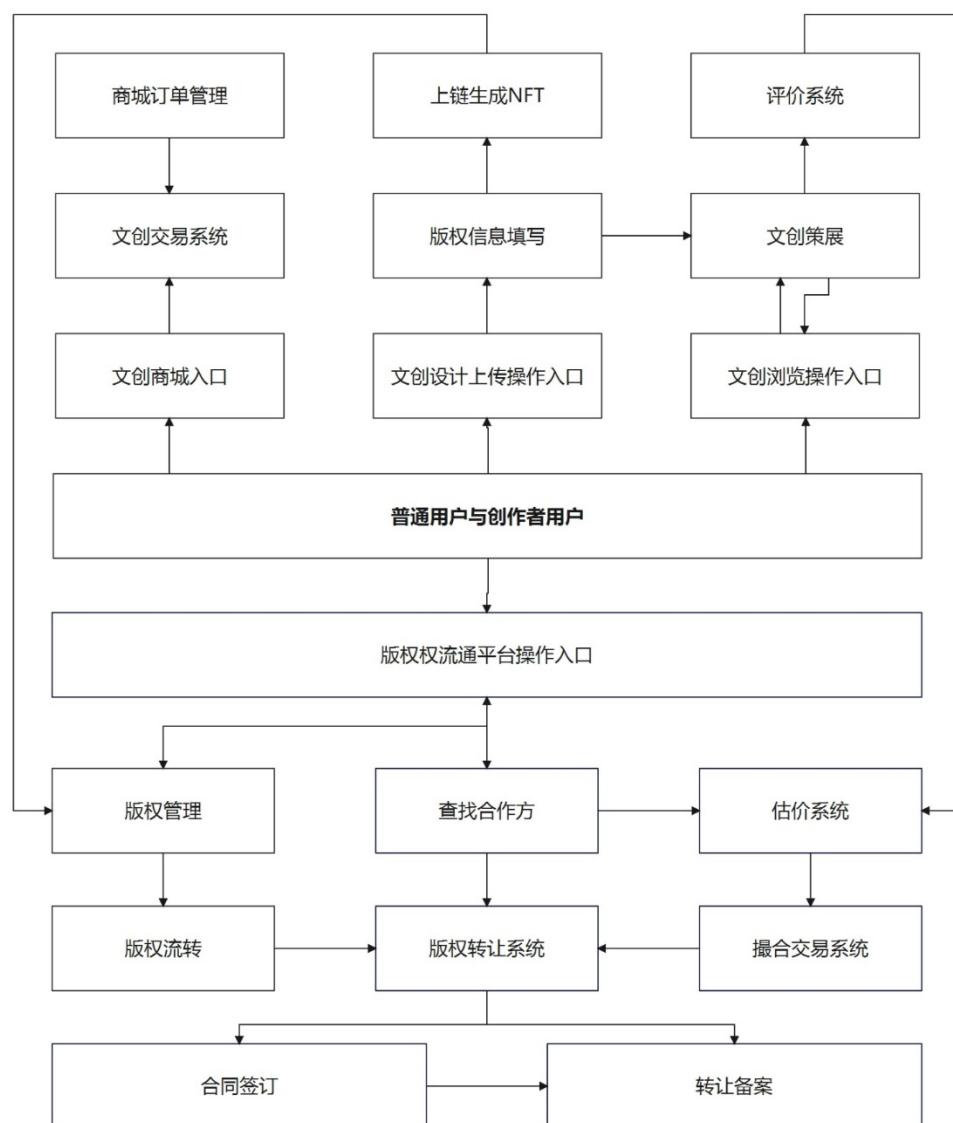
基于上述的需求描述，我们总结出以下功能要点：

设计展示	是本系统的核心子系统，负责将用户所发表的文创设计展示在该子系统的用户界面。所有用户均可在该子系统中对展示的文创创意与文创设计进行打分和评论。
版权确权与版权估值	提供设计者的版权确权服务，同时根据设计展示模块产生的数据进行估价。
合作撮合与版权流通	提供创作版权流转、使用权授权等服务。同时运营、设计与生产可在此达成合作。



文创电商平台	提供文创产品的售卖。
创意市集	所有用户可在本系统内进行任意创意的分享，设计者可在此获得创作灵感与创作创意。
众筹管理	用户可在本系统内发起设计与生产的众筹。众筹模式包括捐赠众筹与预售众筹。

3.2. 产品业务



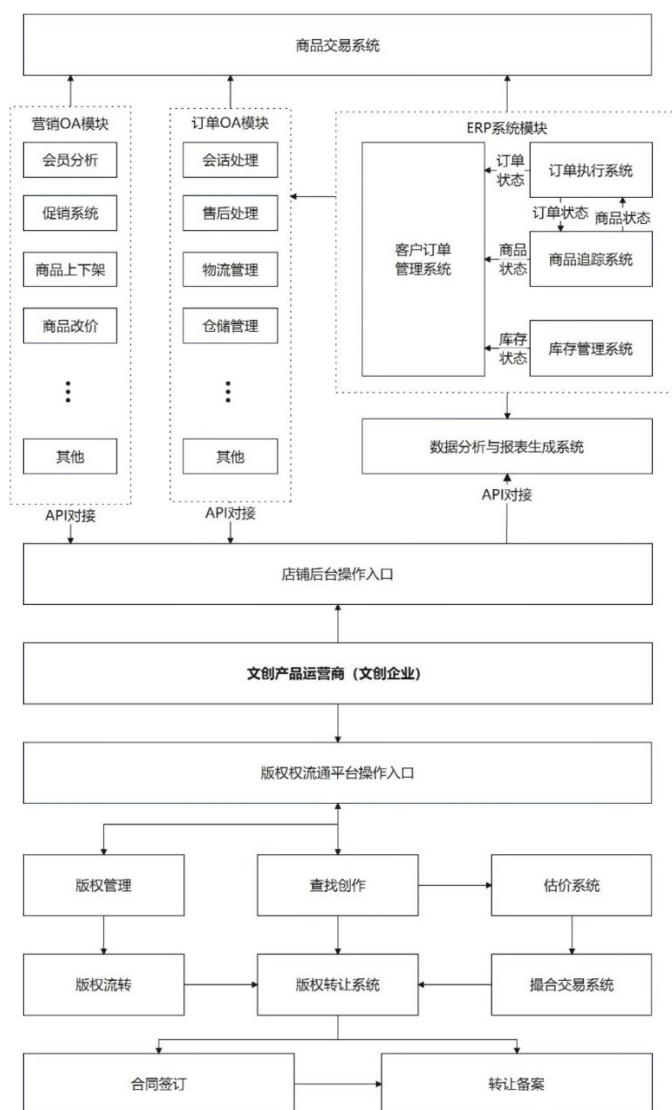
平台首先会区分个人用户和企业用户。对于个人用户及普通用户和创作者用户，在进入平台后会进入首页。首页中有文创浏览的操作入口、文创设计上传的



操作入口以及文创商城的操作入口。在进入文创浏览的操作入口后，将进入文创策展系统，文创策展系统中会详细的展示创作者所上传的文创设计，用户可对该文创设计进行评价和打分。在进入文创设计上传操作入口后，将进行版权信息的填写，如该设计已取得著作权，将直接进入上链铸币阶段，若设计尚未取得著作权，将进入著作权申请界面。而进入文创商城入口后，将可选择购买喜欢的文创产品。

与此同时创作者用户还可进入版权确权流通平台，版权流通平台首先将展示该用户所拥有的所有版权，用户可选择流转。选择流转后，将查找合作方并对版权进行股价，最后通过撮合交易系统完成版权转让，完成版权转让后将签订电子合同并进行转让备案。

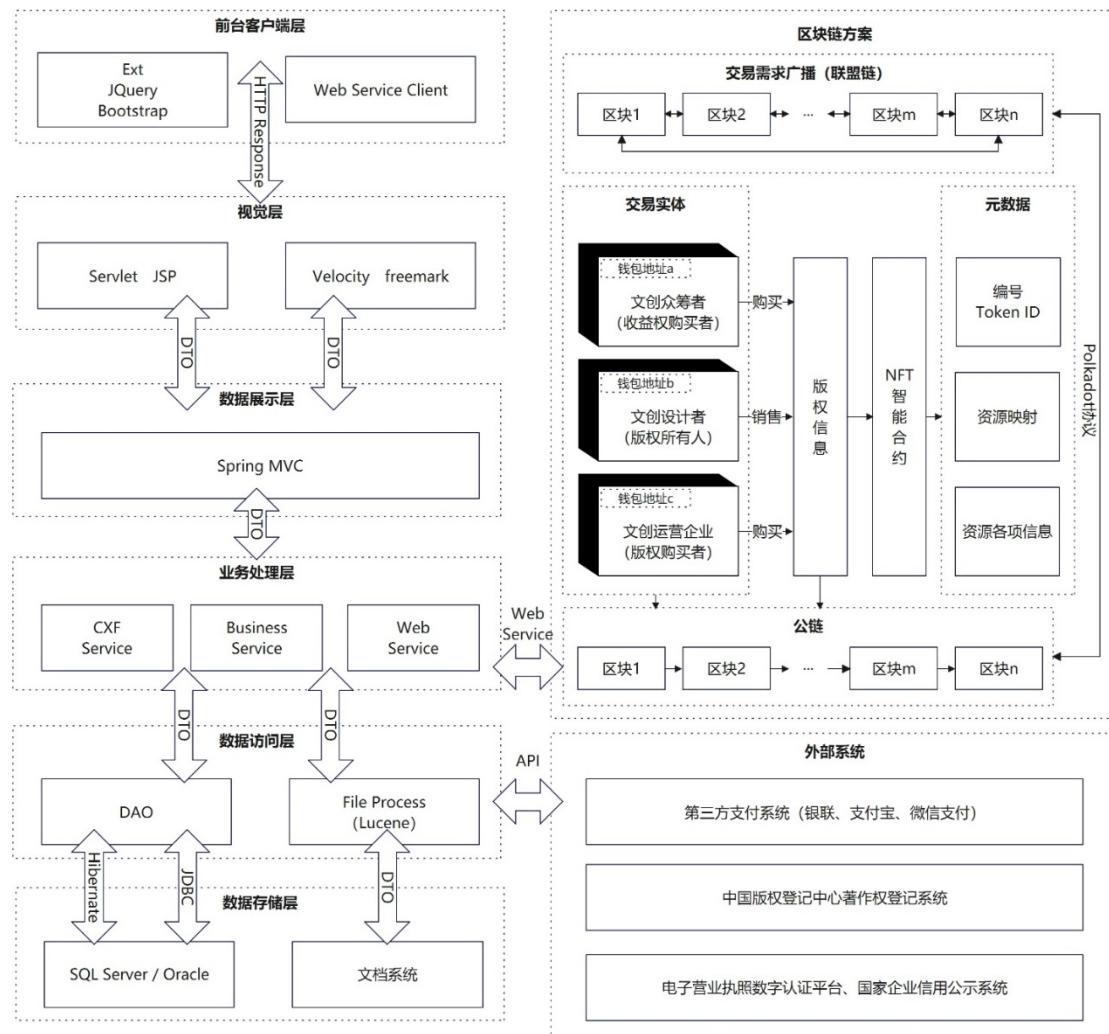
而对于机构用户，除了可购买被转让的版权以外，还可进入店铺后台进行文创商品售卖。





3.3.技术架构

基于区块链的文创设计策展社区和版权确权流通平台的技术框架包含八大板块，分别为前台客户端层、视觉层、数据展示层、业务处理层、数据访问层、数据存储层、区块链方案和外部系统，板块间相互联系共同支撑该平台设计展示、版权确权与版权估值、合作撮合与版权流通、文创电商平台、创意市集、众筹管理等功能的实现。



3.3.1. 前台客户端层

前台客户端层是直接与客户联结的板块，对外向客户提供平台各类服务，如设计展示服务。板块内包含 Web 客户服务端和扩展文件系统、JQuery 与 Bootstrap 等框架。



3.3.2. 视觉层

视觉层通过 HTTP 响应与前台客户端层连接，其是将可视化的界面与数据分离的板块。包含服务连接器、JAVA 服务器界面和 Velocity、freemarker 等模板引擎。

3.3.3. 数据展示层

数据展示层将来自最底层的数据存储层的各类数据通过数据传输对象 DTO 直接传输给视觉层，数据在视觉层与 Web 客户服务端的可视化界面分离。数据展示层的上述功能依托 Spring MVC 实现，Spring MVC 可以分离控制器、模型对象、过滤器以及处理程序对象的角色，而此分离让它们更容易进行定制。

3.3.4. 业务处理层

业务处理层与上层的数据展示层和下层的数据访问层均通过数据传输对象 DTO 联结，其包含 3 个服务模块，分别为 CXF 服务、商务服务和网络服务，这一板块实现直接对平台各项业务进行处理的功能。

3.3.5. 数据访问层

数据访问层的核心功能是对来自最底层的数据存储层中存储的数据进行访问，其内部的数据库访问对象通过对象关系映射框架与 Java 数据库连接与数据存储层中的 SQL Server/Oracle 连接，而文件处理（全文搜索引擎）通过数据传输对象 DTO 与数据存储层中的文档系统相连接。数据在数据访问层被访问后将被传输至业务处理层，在平台各项业务处理过程中被进一步处理，并在数据展示层展示。

3.3.6. 数据存储层

数据存储层是平台中存储各类数据的数据仓库，是整个平台搭建的基础，由 SQL Server/Oracle 和文档系统两个数据存储体系共同构成。

3.3.7. 区块链方案

区块链方案是平台所有板块中最核心的板块，通过网络服务与业务处理层直接联系，实现数据上链存证等功能。文创众筹者（收益权购买者）、文创设计者（版权所有人）和文创运营企业（版权购买者）均有各自对应的钱包地址，分别



通过购买、销售和购买生成版权信息。

3.3.8.外部系统

外部系统拓宽了平台的业务外延，通过应用程序编程接口 API 与平台内部系统中的数据访问层联结，数据访问层在对数据存储层的数据进行了访问后可以通过 API 与处于外部的外部系统联系，外部系统包含各类相对独立于平台之外的系统，如第三方支付系统（如银联、支付宝、微信支付等）、中国版权登记中心著作权登记系统和电子营业执照数字认证平台、国家企业信用公示系统。

3.4.技术核心

NFT 代表“不可替代的代币”。在基本层面上，NFT 是一种数字资产，将所有权与独特的物理或数字项目联系起来，例如艺术品、房地产、音乐或视频。

NFT 可以被视为现代收藏品。它们在网上买卖，代表任何给定物品的所有权的数字证明。NFT 被安全地记录在区块链上——与加密货币背后的技术相同——这确保了资产是独一无二的。该技术还可以使更改或伪造 NFT 变得困难。

可互换的物品可以轻松地相互交换，因为它们的价值与其独特性无关。例如，您可以将一张 1 美元的钞票换成另一张 1 美元的钞票，即使您的新钞票具有不同的序列号，您仍将拥有 1 美元。

不可替代的物品不可互换。对于 NFT，每个代币都具有独特的属性，其价值与其他类似代币的价值不同。

创建 NFT 的过程如下：

- 获取物质的唯一标志性中间产物。如使用哈希函数，计算其哈希值，作为该物质的标志。
- 通证化。在区块链上进行智能合约的开发，通过交易的方式，将物质的唯一标志性中间产物等相关数据存储到链上。过程见附录。
- 展示。所谓展示，就是对 NFT 的数据进行读取再展示。根据当初设置 NFT 数据到链上时获得的相关数据使用智能合约读取信息，将获得的信息通过某介质应用还原出原始的 NFT 数据。



3.5.未来改进方向

技术的后续发展方向可着眼于算法开发极其性能优化、计算法则的设计与生态的构建上，其中，在算法开发及其性能优化方面，我们希望以市场运作与平台运营为核心导向，开发与之相关联的算法，如估价算法、防欺诈算法等，不断优化平台功能，使得版权确权与版权估值等服务的信服度持续提升，以此吸引更多用户入驻平台。

3.5.1.算法开发及其性能优化

估价算法

在平台的版权确权与版权估值服务模块，根据设计展示模块产生的数据进行估价这一环节可以依托估价算法来实现。在平台开发完成并具备了一定规模的用户数据量后，利用大数据技术，基于此算法可以针对用户依托某一设计的浏览量、点赞量和评分等信息，生成建议交易价格并提供给用户作交易参考，同时对于文字类的评论，可以开发能实现语义识别与提取功能的预处理算法，通过赋权重的方式将文字类的评论量化，并进一步参与后续估价算法的计算。

防欺诈算法

针对用户依托某一设计的各类信息如浏览量、点赞量和评分以及文字类评论等直接应用估价算法进行版权估值的模式存在一定的欺诈隐患，如某些用户可能通过恶意刷分、恶意刷浏览量等违规操作进行欺诈，而引入欺诈算法可以通过智能识别刷分、刷浏览量等违规操作及时制止并处理平台交易中存在的欺诈行为，进而维护产业生态。

3.5.2.计算法则设计

安全多方计算法则

为了进一步提升平台交易的安全性，项目组未来可以利用密码学知识等，设计安全多方计算法则，使得在数据加密的情况下可以实现多方计算，进而方便监管或引入其他合作商，并使得用户隐私得到充分保护。



3.5.3. 确权功能

著作权身份标识的唯一性+著作权登记认证机制

项目在确权功能方面的创新在于引入了自主知识产权的确权水印技术，利用该技术可提取数字内容素材的特征信息，进而基于获取的特征信息为其生成全网唯一的身份标识，使得著作权身份标识具有了唯一性，由确权引起的各类纠纷如侵权等问题发生的几率由此将可以大幅降低。此外，自主知识产权的确权水印技术可抵抗旋转、扭曲、裁剪、涂改等操作对数字内容素材带来的各类影响，如识别时因为裁剪等操作而使得识别率降低等，由此可以确保被修改后的素材身份标识与数字内容原始素材仍然保持高度的一致性，保证了著作权登记认证机制的正常运作。

3.5.4. 溯源功能

数字内容素材追踪器

本项目在溯源功能方面同样有创新点，我们可以在数字内容素材中嵌入自主知识产权的溯源水印，并基于此打造具有不可修改属性的数字内容素材追踪器。利用嵌入式的溯源水印可以快速确认数字内容素材的核心信息，如素材原始归属、素材交易历史与素材现有归属等。因此，当侵权事件发生时，我们就可以依托上述数字内容素材追踪器，快速发现侵权行为并定位侵权主体，然后输出链上的侵权证据，为发起侵权链上司法公证提供坚实的技术基础。

3.5.5. 隐私保护下的安全交易功能

我们的项目在安全交易功能方面的着眼点在于隐私保护，功能的实现强调隐私保护的技术语境。针对数字内容素材交易过程中可能暴露的隐私泄露与不诚实问题，我们基于加解密与智能合约技术，可以实现分布式场景下的多方公平交易功能，由此将确保用户间交易内容的不可篡改性、交易行为的不可抵赖性与整个交易过程的隐私安全性，构建隐私被充分保护的安全交易生态。



四、项目创新与价值

4.1.商业模式创新

4.1.1. 产业链集约化

文化创意产业总体发展模式不完善，虽然数量上急剧膨胀，但是发展的质量普遍不高，集约化程度低。文创产业链主要可以分为 IP 培育及产品设计、生产制造、销售、社区及文化打造几大环节。从由于文创的人物 IP 是影响其成功的核心因素之一，因此早期文创通常是以设计师及其工作室主导整个流程，产量通常较低。产业链之间彼此割裂，导致产业链各环节不能与其他环节沟通交流、共享资源，最终各个环节都无法单独创造效益，导致总体产出较低。以动漫产业为例，北京某动漫企业负责人说，北京尚未完全摆脱“漫画→读者”“动画→投资”“厂商→购买少部分动漫形象→消费者”单项赢利模式，“衍生产品制造商→漫画→衍生产品制造商互动开发→动画制作→衍生产品→消费者”的产业链尚未形成无缝对接。

本项目创造了一种商业模式联通各个环节，就能使得文创的设计、生产效率大幅提高，销售渠道铺设加速，并促进了文创社区、潮玩文化的形成。链城之壁希望在 IP 衍生品领域建立一种开放模式，在这样一种产业生态环境中，不但消费者拥有很大的信息获取和选择自由度，而且 IP 衍生品价值链上的各方力量也能够自由参与其中，获得利益，从而形成一种开放的产业格局。以往的 IP 衍生品价值链呈垂直封闭模式，集约化程度低，“人”、“财”、“物”相互割裂。消费者缺乏足够开放的信息渠道，被动接受 IP 运营商的供应产品，难以真正实现自我的消费意愿。版权方和运营方难以获悉市场动态与消费者的动态消费偏好，衍生品产品与市场需求匹配度低。产业链之间彼此割裂，各环节不能与其他环节沟通交流、共享资源，最终各个环节都无法单独创造效益，导致总体产出较低。本项目创造出一种开放互联的商业模式，可以有效整合产业链上的相关环节，满足各方，包括消费者、版权商、运营方等的利益诉求，为未来 IP 衍生品行业发展提供了一种新的可能。同时，开放性的商业模式也有可能改变 IP 衍生品传统的内容销售盈利模式，开拓出新的盈利渠道。



4.1.2.产品社会化与创作激励

由于非创作者没有产品设计过程的参与、没有社群、没有平台，文创产品与文创设计一直没有有效的传播。文化为大众所认同方显其魅力，文化创意产业发展的市场基础是大众化消费的形成。文化氛围的增强首先需要依靠大众文化消费的自然积累。文创企业通过提供多元化的文化创意产品，不仅能够创造社会经济效益，而且有助于培育文化消费市场、营造文化消费氛围。文化行业最大的特点就是对人才的依赖性很强，而人有很多不确定因素，很多时候和也在人，散也在人。不同于传统工业有一条完整的生产线，要有很强的团队，需要日积月累的考验，文创企业的成功有很多偶然因素。由于文创企业的创业门槛相对较低，可能有个好项目，或者有个好点子、好编剧，就可以创业，甚至创造票房奇迹，一战成名。如果建立文创的交流社区，并兼顾其版权保护和文创产品产业链各方的撮合，可以让文创产品在传统社交内部、社交之间的传播中，使用户转变为创意者，有效避免文创信息的走样、失真甚至无意义化，让文创产品最终成为社交情感的认同，并完成设计到产品的转化。

此外，文创产品设计一旦在互联网发布，出现在公众视野，侵权、剽窃、抄袭等问题将接踵而至。目前，传统维权和区块链维权并行中，两者共同发展，但仍有不足。亟需利用一种技术，开发出既能对商业模式进行创新，又能保证侵权证据的收集，减少侵权事件的发生。

4.1.3.解放消费者

由于非创作者没有产品设计过程的参与、没有社群、没有平台，文创产品与文创设计一直没有有效的传播。文化为大众所认同方显其魅力，文化创意产业发展的市场基础是大众化消费的形成。文化氛围的增强首先需要依靠大众文化消费的自然积累。本项目通过搭建开放的社群生态场域，使原来被动接受的消费者最大程度上参与 IP 衍生品的创意、设计和生产全流程，使用户转变为创意者，有效避免文创信息的走样、失真甚至无意义化，让文创产品最终成为社交情感的认同，并完成设计到产品的转化。将消费者从传统的文创产业消费模式中解放出来，极大提升了获得感与认同感，将积极促进文创社区、潮玩文化的形成和发展。



4.2.技术特色

4.2.1.去中心化

项目在去中心化方面的创新在于采用了非同质化技术，使得 NFT 在链上实现公开记录与交易可溯，加速信息可信流转，利用该技术可提取数字内容素材的特征信息，进而基于获取的特征信息为其生成全网唯一的身份标识，使得著作权身份标识具有了唯一性，由确权引起的各类纠纷如侵权等问题发生的几率由此将可以大幅降低；同时采用了 IPFS 去中心化存储，实现去中心化特性，同时存储更为便捷安全；另外，我们还采用了 DID 分布式身份验证技术，更大程度上实现去中心化。

4.2.2.权益凭证可信流转

本项目在权益凭证可信流转方面同样有创新点，我们使用 NFT 技术实现权益凭证的可信流转，IP 出版方将邻接权借助 NFT 流转至各方，并借助区块链技术赋能，实现流转过程全记录，流转周期共识可信，打造具有不可修改属性的数字内容，可以将 IP 与唯一性的 NFT 绑定在一起，杜绝了虚假权益凭证的出现和顶替，同时可以通过区块链技术进行快速验证，确保 IP 出版方的权利不受侵害。因此，当侵权事件发生时，我们就可以依托上述数字内容，快速发现侵权行为并定位侵权主体，然后输出链上的侵权证据，为发起侵权链上司法公证提供坚实的技术基础。

4.2.3.便捷可移植系统

我们的项目在便捷可移植系统方面的着眼点在于构建匿名交易组件，加密中间件，链上监管中间件等可插拔组件，满足用户以及监管机构的个性化需求针对数字内容素材交易过程中复杂繁琐，各个系统相互阻隔的现象，我们借鉴了 Rarible 智能合约，实现免手续费离线签名，帮助用户在交易 NFT 的同时，减少手续费消耗；同时，我们还兼顾兼容性：可部署在以太坊、币安、火币、MATIC 等流行的公链并逐步拓展至 Hyperledger Fabric/ Fisco Bcos 等联盟链，实现分布式场景下的公平交易。



4.3.社会效益

4.1.1.促进经济创新

在知识经济的背景下，文化创意不仅是走出危机的先导产业和经济实现加快发展的新战略，更是成为改变世界越来越重要的力量。文化创意产业属于创新型业态，是指依靠创意人的智慧、技能和天赋，借助于高科技对文化资源进行创造与提升，通过知识产权的开发和运用，产生出高附加值产品，具有创造财富和就业潜力的产业，对加快经济创新具有积极的促进意义。

2008 年的全球金融危机中，文化创意产业成为经济寒冬中的一股暖流。数据显示，当各地经济出现负增长的同时，创意产业则呈现了良好的发展态势，许多地方创意产业增速普遍高于当地经济增幅。在后危机时代，各国经济为应对危机促进经济复苏，把发展文化创意产业作为推进创新和调整产业结构的重点领域，引领经济复苏的新战略。比如，美国经济中心纽约在金融危机前以金融（Finance）、保险（Insurance）和房地产（RealEstate），即 FIRE 为支柱产业，2005 年上半年 FIRE 产业占纽约所有产业中工资支出的 84%。由于 FIRE 发展过火而失控，引发了金融危机。金融危机爆发后，纽约吸取教训，转向以智力（Intellectual）、文化（Cultural）和教育（Educational）为代表的 ICE 产业。目前，美国整体经济也在从 FIRE 转向 ICE，后者也包括信息（Information）、创意（Creative）、能源和环境（Energy&Environment）等领域。

综观世界经济发展的实践，不难发现，在知识经济的背景下，文化创意不仅是走出危机的先导产业，是经济实现加快发展的新战略，而且已成为改变世界的重要力量。任何一场经济危机发生之后，总需要由创新带来突破，由发现新市场战胜困难。上世纪 30 年代经济大萧条时，工业领域率先突破、力挽狂澜；1998 年东南亚金融危机爆发时，IT 和内容产业异军突起，逆转了经济下坡路；2008 年的金融危机，文化创意产业逆势崛起，不但成为化危为机，促进区域经济增长的创新战略，也为大中华地区开启了一条走向创新的发展之路。

经过 30 多年的改革开放和发展，中国成为了世界制造业大国，但在“中国制造”的盛名之下，要清醒地认识到低端制造模式的不可持续性。目前我们正面临劳动力成本上升、人民币升值、环境资源等瓶颈的约束，中国传统制造业正面临一场转型和升级的严峻考验。大中华地区要在国际产业竞争中立于不败之地，必须改变“中国制造”之困境，唱响“中华创造”之品牌，加快实现从制造到创



造的新跨越。

从文化创意产业的功能来看，文化创意是促进产业转型的重要途径。文化创意产业倡导开发人类创造力、解放文化生产力、提升产业竞争力、增强国家软实力，强调创意和创新，强调把文化、技术、产品和市场有机结合起来，不仅能够为人们提供文化含量较高的产品和服务，满足人们的精神需求，形成新的消费市场，更重要的是它还可以和其他产业融合发展，促进产业创新和结构优化，有效地推动中国的经济转型和经济创新。

从文化创意产业的发展实践来看，大中华地区通过文化创意的发展，产业转型取得了积极成效。比如香港特区政府为优化经济结构，全力支持创意产业，目前创意产业已成为香港快速增长的产业之一。以设计产业为例，2005年到2010年间，其对香港本地生产总值的贡献增加近3倍，同期就业人数增加1/4。

随着人们生活水平的不断提高，消费者往往不再满足于商品本身的使用价值，而更关注商品中的观念价值，即其中被注入的文化要素。文化创意产业正是通过观念、感情和品位的传达，赋予传统意义的商品某种独特的“象征意义”，提升其文化附加值，从而满足人们的精神需求和个性化消费，并加快促进消费增长。

发展文化创意产业也是创造一种新型生活方式，在促进消费的同时，还有利于提升生活品质。比如台湾文化创意产业发展初期就强调，要通过发展创意生活产业，提升人们的生活品质，提倡“把生意做成文化”，创造出具有文化品位和情调的创意生活方式，不仅满足了人们的个性化消费需求，还在精致农业、休闲生活等领域积极与大陆合作，大大地拓展了产业消费的市场空间。

4.1.2. 推动文化创新

文化创意精品的传播影响远大于说教式的宣传，优秀的创意产品不仅可以传播和普及文化知识，而且会潜移默化地影响人们的思想观念、价值判断和道德情操。只有提升创意转化力，才能将资源优势转为经济优势，并借势扩大中华文化影响力。

随着现代化进程的加快，许多传统文化濒临灭绝，而文化创意产业利用高科技和多媒体等创新手段将传统文化中的精髓延续下来，既有效地传承，又在内容或形式上有所创新。中华文化是大中华地区共同的情感记忆、精神遗产，发展文化创意产业，有利于推动大中华地区文化创新力、文化影响力和文化吸引力的整体提升。



中华文化创新力的提升，是基于对传统优秀文化的创新性传承、对外来先进文化的包容性吸收，以及对历史文化资源的创意性转化。传统文化只有在创新中传承，才能得到发扬光大。大中华地区发展文化创意产业，就是促进中华文化的创新创意性发展，是对中华传统文化赋予现代阐释，使其在服务当代人的文化精神需求中焕发出新的生命力。

发展文化创意产业不仅需要增添新的内容，而且需要对异质文化的吸收和融合，这种融合性不仅体现在产业运作上，还体现在对文化内容和形式的重新编码和整合上，有利于推动中华文化的价值创新，进一步优化中华文化基因。为此，我们应以包容和开放的胸怀来吸纳和借鉴其他民族的先进文化，扩大国际文化创意的交流，从而拓展创意空间、提升创新能力。

中华文化是一个巨大的财富宝库，只有提升创意转化力，才能将资源优势转为经济优势，并借势扩大中华文化影响力，这其中的关键环节就是对历史文化资源的创意性转化。比如《花木兰》、《功夫熊猫》中的人物、动物原本是中国特有的文化和生物资源，但却被创造力强大的好莱坞所开发和利用，成为美国人获取财富的资源。以美国的梦工厂《功夫熊猫》为例，导演花 30 年时间研究中国文化，使整部电影充满了中国元素，《功夫熊猫 2》全球票房高达 6.5 亿美元，其衍生出来的产业市场还在不断扩大，后续效应也在持续发酵。

这给我们的启发是，文化创意精品的传播影响远大于说教式的宣传。优秀的创意产品不仅可以传播和普及文化知识，而且会潜移默化地影响人们的思想观念、价值判断和道德情操。美国就是通过好莱坞电影向全世界传播美国精神和价值观，从而成为文化强国的。比如 3D 电影《阿凡达》取得了很好的票房，其 DVD 热卖 4 天销量 670 万张，创收 1.3 亿美元。观众之所以乐意观看，是因为“保护生态”的主题是人类共同面临的问题。但是吸引眼球的宏大叙述中渗透进了美国的主流价值观，观众在无形中接受了。同样，韩国也是通过文化影视，形成“韩流”，推广韩国产品和价值理念。

那么，如何增强中华文化的吸引力，展示其独特魅力呢？不二法门还是要借助现代高新科技成果，大力发展战略性新兴产业，推进文化交流和传播手段的升级换代，改造传统文化的生产经营和传播模式，促进传统艺术样式的升级换代。比如传统舞台美术与多媒体技术结合形成的视觉效果使人目不暇接，由此创新形成的实景文化旅游演艺节目吸引了众多游人，各地游人纷至沓来。以有“上海演艺产业名片”之称的《时空之旅》为例，从 2005 年 9 月首演至今，天天上演，演出场次近 2000 场，票房过亿元，其成功之处是在传统杂技基础上，运用多媒体



声光音响技术，融合现代音乐舞蹈，并打出“秀一个上海给世界看”的主题，巧妙融合各项艺术元素，营造出绚丽的多媒体效果，实现了“艺术与技术共融，传统与现代对接”，让人震撼于现代高科技和文化结合的魅力。

文化创意产业通过美学符号的诠释，既塑造了区域文化的个性，也增强了城市的文化吸引力。文化创意产业与旧城区改造形成有机互动，有利于历史文化遗产的保护和城市文化品位的提升。一方面，通过保留具有历史文化价值的建筑，可以避免城市文脉的中断，使得历史与未来、传统与现代、东方与西洋、经典与流行在这里的交叉融会，为城市增添了历史与现代交融的文化景观，给人以城市的繁华感、文化底蕴的厚重感和时代的生机感；另一方面，孕育了新的产业业态，避免了产业的空心化，对城市经济的更新和持续发展，以及就业率的提高等产生了巨大的推动作用。上海目前近 100 个创意产业集聚区中，有 2/3 都是由 20 世纪中期上海工业大发展时建造的厂房、仓库改造而成，比如 M50、8 号桥、田子坊、红坊等。又如，被列入“世界文化遗产名录”的澳门历史城区，保存了百年中西文化交流的历史精髓，展示了独特的文化魅力，而澳门特区的望德堂区仁慈堂婆仔屋最近经重新改造也成为创意产业区，这些文化创意空间将为澳门吸引更多的游客。

4.1.3. 加深交流合作

利用中华文化元素和价值理念发展文化创意产业，既能够使大中华区以鲜明的文化特征区别于世界其他地区，又能增强区内中华儿女的文化认同感。

发展文化创意产业不仅能共享共赢，还将加深区域间的交流合作，大大增进文化认同，增强民族凝聚力，共同为大中华区域同胞谋福祉，为两岸谋和平，为中华民族谋复兴。

大中华区域具有共同的文化母体，发展文化创意产业，在加强文化交流的同时，又能促进产业发展。一方面可以进一步增进文化认同感，加深友情；另一方面还能够产生社会效益，文化与经济双赢的交流，有利于增进合作共识。

迄今为止已经举办了七届的“两岸经贸文化论坛”，将经济和文化作为重要研讨议题，历届论坛到现在已经获得了 100 多项的共同结论和共识，为文化、交通、旅游、金融等方面提供了更多的双赢的机会，在第七届论坛上形成的“共同建议”中，把加快文化创意产业发展列为重点合作内容，提出继续推动两岸文化产业交流与合作。扩大两岸出版物贸易、版权贸易及影视业合作，加强技术合作



与项目合作，共同开拓海外华文市场，增强中华文化的国际传播力和影响力。拓展交流合作管道，加强业务往来和沟通协调，并探讨市场准入与通路议题，增强两岸文化产业实力。在合作共识的基础上，近年来台湾和大陆的许多城市建立了文化创意产业项目的合作关系，不仅满足了台湾的投资需求，还将台湾的美食文化、声光数字、动漫世界、时尚品牌、游艺购物向大陆推广，从而推动了海峡两岸的文化交流，进一步促进和平发展。又如香港资讯丰富，高新技术的研究能力处于世界一流水平，许多香港高科技企业将厂址迁移至营运成本较低的内陆，并以内陆著名高校的人力资源为强大后盾，促进了各自的发展。

文化是大中华区域血脉相连的纽带，如果说经济合作像手携手，文化交流就是心连心，文化创意产业融合了经济和文化，其发展必将进一步促进大中华区域的文化认同，增强中华民族的凝聚力。

利用中华文化元素和价值理念发展文化创意产业，既能够使大中华区以鲜明的文化特征区别于世界其他地区，又能增强区内中华儿女的文化认同感。比如，香港出品的华语卡通动画片《麦兜响当当》，以麦兜母子俩的经历为线索，幽默风趣地反映了港人北上的辛酸，片中所折射出来的公正、至诚、仁爱、孝悌、信义等儒家价值理念，因为大中华地区的观众对这种儒家价值理念具有认同感，使得该片广受欢迎，无形中还增强了中华民族的凝聚力。

从文化层面看，大中华地区发展文化创意产业，首先是彼此间文化的认同与合作。比如，“和”文化是中华文化的精髓，通过“和谐”、“和睦”、“和美”、“和顺”、“和悦”等的创意发展，既可以加深中华民族的血肉联系，也有助于强化中华文化精神的弘扬和传承。从产业层面看，发展文化创意产业，对于开辟中华民族共同创造和积累财富的新路径，提升中华民族整体经济实力，增强大中华地区的国际竞争力也不无裨益。比如，北京深厚的文化积淀、良好的文化氛围，吸引了一大批台湾文化创意人士驻扎，从事与文化创意相关的产业，“两岸四地文化研究与交流中心”落户杭州西溪创意产业园，大中华地区共同打造的“妈祖文化旅游圈”项目启动，“海峡两岸文化产业园”、“两岸艺术品交易中心”也在厦门建设，等等，这些均有利于大中华区域内部各方发挥各自优势，形成具有互补效应的文化创意产业体系。

4.1.4. 提升国际形象

发展文化创意产业，在国际市场上营销“中华牌”文化创意产品，不仅能够提升大中华地区的制造业和产品附加值，还能够输出中华文化品牌，形成持续的



区域竞争优势。

我们自豪于祖先的创造，使大中华地区拥有积淀了五千年的历史文化资源。通过发展文化创意产业，一方面，我们可以挖掘和整合本土文化资源，推陈出新，将丰富多彩的中华文化资源宝库活化为财富金矿和精神乐园，造福中华民族；另一方面，我们可以通过树立良好的国际文化形象，将丰厚的中华文化遗产转化为世界人民共享的精神财富，在推进世界文明的进程中发挥应有的作用，贡献应有的力量。

文化创意产业的发展有助于我们以全球的视角、时代的眼光、创意的手段演绎大中华区域的国际形象，围绕中华文化的核心价值，用世界语言叙述中国故事，用创意建立起鲜明的中华文化形象识别体系，包括理念识别（mindidentity）、行为识别（behavior identity）和视觉识别（visual identity），实现中华文化形象的创新发展，提升中华文化的国际认知度和认同度。

当前，全球范围的产业结构、生产方式与国际分工正面临着大调整、大变革，围绕市场、资源、人才、技术、标准等方面的竞争更加激烈，文化日渐成为提升区域软实力，获取竞争优势的关键要素。法国学者马特尔说过一句话：世界文化大战已经爆发，各大国之间为了国家利益已然不复动用传统意义上的军事手段，而是通过电影、电视、流行音乐、时尚服饰、餐饮、媒体等各种文化创意产业的手段展开博弈和竞争。比如美国通过销售好莱坞大片、米老鼠卡通、百老汇歌剧、NBA 篮球，以及星巴克、麦当劳、可口可乐等传播美国文化和价值观。因此，大中华地区发展文化创意产业，在国际市场上营销“中华牌”文化创意产品，不仅能够提升本区制造业和产品附加值，在全球产业分工体系中占据优势地位，其意义还在于能够输出中华文化品牌，形成持续的区域竞争优势。

必须强调的是，在现代经济发展中，品牌已经成为一种战略性资源，企业和产品在市场上的竞争实质上是其所体现的品牌文化之间的竞争，品牌正主导世界财富的分配。我们是文化资源大国，但还不是文化品牌大国，对此，我们应有足够的重视和义不容辞的责任，以创新的态度，加快发展文化创意产业，创造具有中华文化内涵的世界级品牌产品，在引领消费的同时，传播中华文化。

需要指出的是，创意人才也是一个国家和地区国际文化形象体系中的重要标识。比如美国苹果公司的前 CEO 乔布斯、Facebook 的创始人扎克伯格等，都已成为美国国家创新文化的形象符号，其国际影响力和号召力巨大，在他们的背后都形成了消费忠诚度高的“粉丝群”。因此，鼓励创新创意企业发展，培育和推



广大中华地区的“乔布斯式”人才，应是提升中华文化国际影响力的有效举措之一。除了重视推广上游的创新创意专业人才，还要注重发展下游的策划、咨询、中介、营销等创意服务人才，只有形成完整的文化创意产业人才链，才能有效加快实现文化创意产品走向国际市场，才能为文化创意人才集聚起“国际粉丝”。

4.4.商业价值

衍生品带来 IP 溢价，是内容变现重要手段。数据显示，2015 年全球衍生品市场（只计算特许授权商品）规模为 2517 亿美元，其中娱乐/角色形象市场 1132 亿，是全球 383 亿电影票房的 2.9 倍。

衍生品可以延长 IP 的生命周期。IP 衍生品可以极大的丰富 IP 的宣传渠道，譬如相关的玩具、服装、主题乐园等，使得 IP 形象渗透到各个细分领域市场，增加 IP 与用户的接触机会，从而帮助 IP 长期热度不会消减。

传统 IP 衍生品生产商可以通过转型衍生品整体变现服务提供商获取更高利润。

虽然 IP 普遍自带流量，但是 IP 衍生品的推广在互联网时代仍需要专业团队运营流量、策划事件话题以获取足够关注。虽然 IP 衍生品在 IP 自身的人气下天生拥有粉丝基础，使其营销推广成本相对其他品牌依旧较低。但是为了在目前移动社交信息爆炸的时代背景下获得足够的粉丝关注，IP 本身仍需要通过向社交平台索要资源、设计各类创意营销活动以及进行各类图片文字推广等多种方式尽可能多地吸引流量，继而进行 IP 衍生品的营销。因此这就需要专业的互联网流量运营、事件策划营销团队，为 IP 衍生品进行宣传。

传统 IP 衍生品制造商可以通过转型 IP 衍生品整体变现服务提供商，进而承接衍生品整体的产品创意设计、互联网营销事件活动策划宣传、以及各类广告推广，来分享产业链中原先由经销商所享有的高额利润。由于 IP 衍生品服务商往往在终端零售上没有大量的传统渠道资源积淀，因此大部分 IP 衍生品服务商很难在渠道上对已有经销商造成威胁。但是若 IP 衍生品服务商通过在产品设计、宣传事件策划等方面建设，拥有了移动互联网上的营销推广、事件策划能力，结合自身原有的生产设计能力以及已有的成熟互联网电商平台，其可以为 IP 拥有者提供从产品设计生产、到营销宣传再到最后变现的一整套衍生品变现服务工



作。之后，在提供全产业链变现服务的过程中，其也能与 IP 拥有者分享由 IP 带来的高利润空间，获得原先由传统渠道经销商所获得的那部分营销利润。



五、项目运营

5.1.商业模式

5.1.1.盈利模式设计

与合作伙伴内容销售分成收入

链城之壁提供了消费者购买 IP 衍生品的平台，通过与运营方合作出售文创产品，进而收取手续费获得利润分成。

链城之壁平台提供了消费者和 IP 衍生品版权方的交流渠道，通过与版权方分成消费者对版权方的打赏等获得利润。初步设计分成情况可见下表。

	合作方分成	链城之壁平台分成
IP 衍生品运营方	90%	10%
IP 衍生品版权方（IP 设计者）	80%	20%

IP 授权流程撮合佣金收入

链城之壁平台连接起了版权方与运营方的 IP 授权交易，覆盖了从撮合交易双方到签订授权合同的授权全流程。平台通过收取撮合授权交易流程的佣金获得利润。

广告收入

作为文创产品爱好者、设计者、生产者群体聚集的交互平台，为 IP 衍生品领域的精准的广告投放提供了可能。广告收入也预计成为平台的主要收入来源。

5.1.2.关键业务和产品形态

商业模式叙事

链城之壁搭建的创意社区便捷消费者之间进行信息交互与分享（价值 1），平台识别消费者喜好热点并将数据提供至版权方和运营方（交易 1），为版权方



的创作设计提供实时信息和建议（价值 2），帮助运营方识别市场热点，精确销售方向（价值 3）。授权平台撮合版权方与运营方合作（交易 2），版权方获得运营方支付的授权金（价值 4），运营方获得版权方的 IP 授权，进行衍生品生产（价值 5）。消费者通过平台购买衍生品（交易 3）。

关键业务描述

链城之壁连接 IP 衍生品版权方（文创产品创作者）、IP 衍生品运营方（文创企业）和潜在产品消费者的交互平台，为版权方提供设计、估值、交易全流程的实时信息和建议，为运营方提供准确市场需求数据，为消费者提供信息浏览、分享和产品购买渠道，并线上撮合版权方和运营方的 IP 授权全流程。真正实现 IP 衍生品领域“人、财、物”的集聚。

5.1.3. 商业模式画布

商业环节	构成要素	含义
产品/服务	价值主张	对 IP 设计者（版权方）：IP 设计变现 对 IP 运营方：便捷获得热点 IP 授权 对 IP 衍生品消费者：参与 IP 衍生品的设计生产
用户界面	客户细分	IP 设计者（版权方），IP 衍生品消费者和 IP 运营方（文创公司）
	渠道通路	搜索引擎；链城之壁网站；广告平台推介
	客户关系	衍生品消费者市场和版权方-运营方授权交易双边市场的维系
资产管理	关键业务	设计者-运营者-消费者交互社区； 版权方-运营方授权交易撮合平台
	核心资源	全价值链资源整合能力；数据建设能力；网站建设与运维能力；客户关系管理能力；客户商业安全保障能力
	重要合作	IP 衍生品设计者（版权方）；IP 渠道与运营；广告



		投放商
财务方面	成本结构	技术研发与设备投入；系统运行维护费用；人力资源成本；财务管理费用
	收入来源	授权流程交易佣金；文创产品售卖手续费；合作方内容售卖分成；广告收入

5.2.发展战略

项目初期建设完成后先在本地高校进行小规模推广，此时主要的入驻设计者均为各校的高校特色文创设计方。以西安电子科技大学为例，校内的校团委宣传部融媒体中心、校党委宣传部大学生网络文化工作室，以及与学校关联较大的企业，如西安西军电文化传媒有限公司（西电印象）等。这些主体都是项目完成后的首批合作对象。

而对于用户，同样来自于高校内部。校内文创爱好者可对校内文创设计提出建议，与高校文创设计方交流，达到供需对接。与此同时，校内学生如有个体设计者，也可在本平台发布设计创意。

初步推广证明商业模式可行后，将会形成文创社区生态的雏形，此时平台即可在本地范围内走出高校向社会公众开放。作为历史悠久的文化古城，西安具备文化创意产业发展的先天优势。陕西历史博物馆、西安博物院（小雁塔）、大明宫遗址公园、半坡博物馆等众多旅游景点早已开始探索文化周边与文创产品，西安本地的文创消费市场也相当广阔。

本地推广预期与本地较为熟知的涉及文化创意产业的企事业单位进行合作，如陕西文化产业投资控股集团（陕文投）、陕西历史博物馆（陕西省文物交流中心）等，邀请其以机构用户身份入驻，在平台发布其预售信息与设计创意。对于本地推广的社会用户，既有普通社区用户参与文创产品设计的过程，参与打分、评价和推荐，也有个体设计者在平台发布创意，社区生态将在本轮次推广过程中初步形成。

形成地区生态后，即可逐步向全国范围推广，扩张模式沿用地区推广的模式即可。在全国推广的同时，需要善用互联网媒介，通过线上广告的形式逐步将产品进入目标人群的视野。



5.3.人才战略

5.3.1.人才引进战略

为满足公司对人才的需求，为公司的发展提供高质量的人力资源保证，为实现企业的战略目标和长期利益，公司需要不断吸纳杰出人才，制定出合理的人才引进战略，为公司的长远发展保驾护航。

公司将结合需求和人才特质采用外部招聘和定向引进两种方式进行人才引进。

在外部招聘方面，公司将通过发布自己在国内处于技术领先的基本情况，以及发布公司未来发展方向、方针政策、企业文化、产品特征等信息，吸引社会高科技人才及致力于 web 应用防护方向人才的加盟。公司将根据所需求人才确定合适的招聘范围。

在定向引进方面，公司将从国内外企业、科研机构引进各种信息安全方面的专家和技师，同时将把校园招聘作为定向引进的重要环节，从学校吸收大量网络安全和计算机科技人才，为大学生就业创造机会。

不断扩大招聘范围能够增加企业人才选择的空间，引进更加优秀的人才；对于应聘者，能够更好地选择适合自己的岗位，发挥才干，为社会创造更大的效益。

5.3.2.人才培养战略

招聘来的员工虽然经过层层挑选，但不一定完全符合公司的要求，需要企业通过培训与开发来提高他们的素质。

公司的人力资源部门要在招聘之初了解所招聘员工的特长及发展意愿，并根据每位员工的特质安排相应的工作岗位。尤其对于技术人员，要了解他们在区块链开发、算法设计等方面的技术功底和软硬件特长，并针对员工不同的技术要求进行培训，着眼于为公司培养专业技术过硬、具有国际化视野、热爱企业文化的高素质员工。

在培训内容方面，公司要与时俱进，不断更新相应的专业知识，着力抓好在计算机、行政和经营管理、外语、信息安全部新技术与新方法方面的培训，致力于提高公司员工在创新方法与创新理念方面的思想与技能，为公司的发展提供更加坚实的人才基础。



5.4. 市场战略

5.4.1. 线上传播策略

首先，移动互联网传播信息的便捷性使得通过输出“价值观”带来的“冲动消费”逐渐成为常态，在互联网下 IP 能利用其自身优质内容吸引更多具有相同“价值观”的粉丝实现精准营销。目前随着移动互联技术的发展，以微信、微博为代表的移动端社交平台使各种信息能够以每个用户为自媒体的碎片化方式极快地传播。这种传播方式使得每个用户的搜索行为相对减少，转而更加依赖他人的“推荐”和“浏览”。因此用户在互联网海量信息中注意到的往往是自身社交圈中与自身兴趣、价值观相仿的信息。正因如此，当用户在社交圈中看到包含自己感兴趣内容的商品信息时，更容易引起“冲动消费”。

由于移动互联网进一步增大了 IP 所吸引的免费流量，IP 的周边衍生品通过移动互联网大大增加了销售机会。在移动互联网时代，通过事件策划、话题制造及营销内容的推送，IP 拥有者能够针对其粉丝进行以往传统线下渠道所不能做到的大范围精准营销，通过各类小视屏、小游戏的转发分享互动大大促进 IP 衍生品的销售。因此，IP 与互联网的结合使得 IP 衍生品的营销能够充分利用 IP 自有粉丝流量，使其衍生品营销更为精准，传播范围更广、效率更高。

随着互联网行业的飞速发展，我们已经进入新媒体传播的黄金时代。青年人群也逐渐成为短视频平台的主要受众。因此，借助短视频平台和微信公众平台对学生和白领进行广告宣传营销或成为现阶段重要营销途径。

短视频即短片视频，是一种互联网内容传播方式，一般是在互联网新媒体上传播的时长在 5 分钟以内的视频；随着移动终端普及和网络的提速，短平快的大流量传播内容逐渐获得各大平台、粉丝和资本的青睐。截至 2020 年 6 月，中国短视频用户规模已达 8.18 亿，日均使用时长 110 分钟。目前，主要短视频有：抖音、快手、西瓜视频等。

微信公众号是开发者或商家在微信公众平台上申请的应用账号，该帐号与 QQ 账号互通，平台上实现和特定群体的文字、图片、语音、视频的全方位沟通、互动。形成了一种主流的线上线下微信互动营销方式。

我们可以借助微信平台及公共社交平台向用户发布的关于产品或服务的信息，以文字、图片、语音和视频进行广告创意，从而将企业和品牌的信息传达给



用户。

此外，还可进行 App 内嵌广告的形式进行推广，触达用户群体。

5.4.2.线下传播策略

高校广告

赞助推广逐渐成为了一种被广泛应用的营销媒介手段。赞助营销在早期主要是为了提高公司或品牌的知名度，为此，赞助商往往投入巨资以制造出其品牌超强覆盖的印象。

公司将同高校信息安全部进行合作，以高校作为信息安全防护的首批试点；同时，在专业优势高校赞助信息安全相关竞赛，推广公司产品，以此来拓展顾客群的同时提高学生信息安全意识与能力，专业优势高校往往具有较高的行业关注度，协办活动竞赛有利于增加公司的曝光度，树立良好企业形象。

通过对高校竞赛的赞助，公司将对获奖项目与作者协商一同进行深度开发并研发产品投入到市场中打造完整的产品体系从而实现知名度建设与产品研发一道完成。此外，赞助高校竞赛有利于在高校形成良好的宣传效果，便于公司打开高校市场。学生群体中的公司形象建设也增进了学生对于公司的了解，将会为公司吸引人才加入，为公司的持续发展贡献力量。

与此同时，食堂桌贴广告也能起到较好的效果。根据《全国高校大学校园餐厅食堂桌贴广告媒体》表示，近五成大学生每天去食堂 3 次，平均每日到食堂 2.5 次。大学生在食堂的停留时间也是极具传播价值的，大约有 51.3% 的大学生每次在食堂用餐在 11-20 分钟之间，平均一餐用时 19.7 分钟。就餐时间对于大学生来说相当于是一段“无聊时间”，他们迫切需要更加丰富多彩的“食堂生活”。

以餐桌平面出现的广告形式，其最大的特点是信息接收的自觉性。当在就餐环境中，目标受众在心理上处于信息真空状态，广告信息以精准、直接、短距离的方式，与学生面对面交流，使大学生在感官上受到强烈的吸引；在广告接触频率方面，桌贴媒体有着超长的单次广告持续时间和超长的广告发布周期，受众平均停留关注时间为 15-30 分钟，每日阅读高达两至三次，日积月累，潜移默化，品牌自然深植脑海。桌面广告由于其具有固定目标群体，因此投放商可以对投入的资金所带来的广告效益有一个比较科学、准确认知，为下一次的投入计划做比较理性、准确的对比。



地铁广告

地铁广告主要面向的对象为白领。根据广发银行与艾瑞咨询 2020 年联合发布的《中国新白领消费行为研究报告》显示，白领日常出行乘坐地铁的比例高达 51.3%，且较为频繁地出入地铁站。西安地铁发布 2019 年度数据显示，西安地铁日均运送乘客 258.59 万人次，全年共计运送乘客 9.4 亿人次。在此条件下，强制性（人群强制阅读且干扰度低）、流动性（通过广告在人群流动过程中的反复呈现有效增加广告受众人群的数量及接触频率）、强视觉冲击力（广告的画面面积比较大，视觉冲击力强）等特点更是有效提高了地铁广告的到达率，地铁广告已成为高效、直接的广告投放方式。



六、投融资方案

6.1. 初始融资

本公司将在股权融资与债权融资两种基本融资方式中优先选择风险性较小的股权融资。由于当前项目处于初创阶段，尚未产生营收数据，所以融资计划均基于上述支出的合理假设判断。

项目初期公司处于拟成立状态时将首先进入种子轮融资。在团队成员出资 2 万元基础上，通过成员人际网向各自亲友群体内部寻求投资意愿人共出资 10 万元，再通过银行信用贷款，最终达成注册资本目标：50 万元。创始团队与种子轮投资人将依据出资比例进行股权占比划分，共同持有公司 100% 股份。

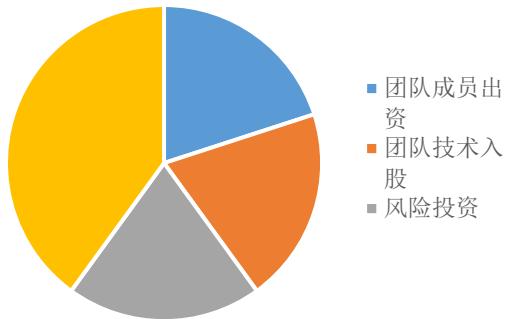
公司成立后，完成全部开发任务，进入试运营时，计划共筹集资金 100 万，公司资金来源主要分为以下三个部分：团队主创成员投资、风险投资及银行贷款。计划创业团队再投资 10 万元；吸收风险投资 20 万元；银行贷款 20 万元。

创始人以出资额比例为标准持有公司原始股份。如遇融资出让股权的情况，经公司会议一致通过后将“所有原始持股者的股票做为整体进行打包”来实现股份的转让。（例：A 公司对我方投资 50 万，我方转让 10% 股权，随将整体公司的分红的 5% 视作该公司的权益，同时我方按照自己私下确认的股份划分享受剩下 95% 的分红）。这样的股份划分模式将不会出现股份不足以转让而发生的相关股权纠纷。“整体打包原则”将视作融资时股权转让基本原则。此外，公司内部允许采用技术入股形式，分红以技术盈利能力为标准。

具体的股权方案如下表所示：

融资方	团队成员出资	团队技术入股	风险投资	银行信贷
融资额	20 万元	20 万元	20 万元	40 万元
份额	20%	20%	20%	40%

初始注册资本比例见下图：



6.2.二次融资

公司在规模扩大的程度下，根据需要进行二次融资，此时早期投资者可通过协议合同或者股权取得回报。也可以通过资产重组的方式进行股权转让。

项目立项、公司注册成立并完成基本研发与初步产品投放后，在预留市场估值溢价空间的基础上，根据软件进一步研发所需资金的情况以及初入市场的用户规模数据表现，判断市场潜力与盈利能力，进一步根据市场扩张、员工招募需求做出计划，通过商业路演、项目比赛宣传等途径寻找投资人，进入天使轮融资。本阶段计划根据资金需求情况与盈利能力，通过整体打包方式出让 10%—30% 的股份。

在产品逐步大量投放市场后，将根据市场运营数据及公司盈亏情况，确定是否要马上进入 Pre-A 轮风险融资。此阶段会完成从面向个人融资到争取机构融资的转变；市场部搜集市场上现有大量机构，了解各个机构的投资方向、资金规模，有目标地寻求更适合本公司的风投机构，达成更高的融资对接效率，在 A 轮完成后还可通过 B、C 轮继续提供资金流支持。

完成占领市场、用户规模变现计划之后，将根据自身实力与资金需求确定何时进一步进入 Pre-IPO 阶段寻求私募股权基金支持以及是否在科创板上市进行首次公开募股集资。

6.3.分红机制

在公司建立初期，为了保证外部风险投资的积极性，我们将在第二个会计年



度结束时开始分红。对创业者投资方面，使用期权方式，公司销售收入不断提高，保持并提高创业者投资的优势。客观来讲，风险投资者并不能取得公司的控制权，但在投资中可以获得企业的利润分配，通过资金退出，股东就可以从股权增值中获得高回报。

6.4. 风险撤出机制

按照我们协议规定，在协议期内，投资商将按照投资资本所对应的公司股份参与利润分配。在协议期满后，按照协议规定，由投资商选择是否回收所投资成本。

风险资本的退出是否畅通，是关系到企业能否吸引投资，以及未来能否持续、健康发展的关键一环。基于与风险投资商实现双赢以及把公司做大做强的信念，我们对三种主要的风险资本退出方式：公司上市、对外出售、MBO，结合中国市场进行了优缺点及可行性分析、评估，再根据我公司的自身特点，最终决定采用对外出售的形式实现风险资本撤出。现有的风险资金撤出机制主要有公开上市、兼并、资产清算、出售股权和管理层回购等。经营达到稳定时的出售股权和管理层回购将是本项目首先要考虑的撤出方式，撤出的定价可采用简单的市盈率法进行计算，即以退出时的年度净利润乘以市盈率计算出企业价值并作为转让基价。



七、财务预估

7.1. 合理财务假设

1. 云服务、区块链等外包费用

项目以华为云、浪潮、阿里云等企业的相关产品报价作为外包费用的计算，成本见下表：

计划支出项	预估成本（元/首年）
云服务器	360.00
AI 解决方案	1,650.00
云计算与云专线	32,000.00
云数据库	110,740.00
区块链服务	30,000.00
公司注册	4,000.00
营销费用	5,000.00
管理费用	2,000.00
固定资产购置	30,000.00
总支出	216,750.00

2. 工资及福利费用假设

根据互联网行业人员平均薪酬以及相关人员的福利政策（包括福利费、保险费等）对比分析，公司人员的薪酬如下：

项目明细	首年人数	年平均薪资	总薪资预估
技术人员	3	¥15,000.00	¥45,000.00
管理人员	4	¥12,000.00	¥48,000.00
其他人员（售后等）	3	¥8,000.00	¥24,000.00



合计	10	/	¥117,000.00
----	----	---	-------------

3.固定资产购置费用

该项目所需的固定资产为技术人员以及管理人员操作所用设备，技术人员以 8000 元/人计算，管理人员以 3000 元/人。

4.折旧与摊销假设

该项目采用综合计提折旧的方式，采用直线法计提折旧，折旧年限是 10 年，净残值率为 5%。

摊销费用包括无形资产摊销和开办费用，其中开办费用在项目开始投入生产的第一年全部进行摊销。

5.研发费用假设

首年不计提，后续研发费用按年营业收入的 10%计提.

6.其它费用假设

项目的其他费用预估包含销售费用、管理费用、财务费用等各项费用.

其它费用按利息支出之前的费用合计的 10%计算。

7.应交税费与税金估算

公司为创新型企业，享受特殊的税收优惠政策，即在公司成立的前三年免征所得税，第四年至第五年所得税税率为 15%（正常税率为 33%）；税后利润按照 15%提取盈余公积金和法定公益金。

8.坏账处理假设

财务预估的坏账处理采用直接转销法，企业在实际发生坏账时，确认坏账损失，直接记入当期损益（资产减值损失）。

采用直接转销法仅为财务预估便利而使用，实际合并财务报表时需采用本企业事先确定的坏账准备计提方法进行坏账处理。

坏账率按中国人民银行发布的全国平均贷款不良率 2.8%估算。



7.2. 财务报表

损益与利润分配表

项目	2022	2023	2024
一、主营业务收入	¥720,000.00	¥1,350,000.00	¥1,680,000.00
(-)主营业务成本	¥290,858.00	¥505,242.00	¥635,228.00
(-)主营业务税金及附加	¥23,896.10	¥56,449.65	¥72,534.99
二、主营业务利润	¥405,245.90	¥788,308.35	¥972,237.01
(-)销售费用	¥44,000.00	¥80,000.00	¥100,000.00
(-)管理费用	¥50,000.00	¥60,000.00	¥70,000.00
(-)财务费用	¥3,750.00	¥3,750.00	¥3,750.00
(-)其他业务支出	¥-	¥-	¥-
三、营业利润	¥307,495.90	¥644,558.35	¥798,487.01
(+)其他收入净额	¥-	¥-	¥-
四、利润总额	¥307,495.90	¥644,558.35	¥798,487.01
(-)所得税	¥-	¥19,336.75	¥23,954.61
五、净利润	¥307,495.90	¥625,221.60	¥774,532.40
六、可分配的利润	¥307,495.90	¥625,221.60	¥774,532.40
(-)法定盈余公积	¥46,124.39	¥93,783.24	¥116,179.86
(-)法定公益金	¥30,749.59	¥62,522.16	¥77,453.24
七、可分配股东利润	¥-	¥-	¥-
八、未分配股东利润	¥230,621.93	¥468,916.20	¥580,899.30



现金流量表

项目	2022	2023	2024
固定资产投资	¥3,000.00	¥6,000.00	¥9,000.00
流动资金	¥257,600.00	¥400,000.00	¥500,000.00
销售收入	¥720,000.00	¥1,350,000.00	¥1,680,000.00
(-)成本	¥314,754.10	¥561,691.65	¥707,762.99
税前利润	¥405,245.90	¥788,308.35	¥972,237.01
(-)税收	¥-	¥19,336.75	¥23,954.61
税后利润	¥405,245.90	¥768,971.60	¥948,282.40
(-)折旧	¥600.00	¥1,200.00	¥1,800.00
(+)无形资产摊销	¥200,000.00	¥300,000.00	¥400,000.00
净现金流量	¥604,645.90	¥1,067,771.60	¥1,346,482.40

资产负债表

项目	2022	2023	2024
资产			
流动资产			
货币资金	¥722,095.90	¥977,758.35	¥1,052,567.01
应收账款	¥-	¥-	¥-
流动资产合计	¥722,095.90	¥977,758.35	¥1,052,567.01
固定资产			
固定资产原价	¥3,000.00	¥9,000.00	¥18,000.00
(-)累计折旧	¥600.00	¥1,800.00	¥3,600.00



固定资产净值	¥2,400.00	¥7,200.00	¥14,400.00
固定资产合计	¥2,400.00	¥7,200.00	¥14,400.00
无形资产			
无形资产	¥200,000.00	¥300,000.00	¥400,000.00
资产总计	¥924,495.90	¥1,284,958.35	¥1,466,967.01
负债及所有者权益			
流动负债	¥200,000.00	¥200,000.00	¥200,000.00
应付工资	¥117,000.00	¥140,400.00	¥168,480.00
应交税费	¥-	¥19,336.75	¥23,954.61
负债合计	¥317,000.00	¥359,736.75	¥392,434.61
所有者权益			
实收资本	¥300,000.00	¥300,000.00	¥300,000.00
盈余公积	¥46,124.39	¥93,783.24	¥116,179.86
法定公益金	¥30,749.59	¥62,522.16	¥77,453.24
未分配利润	¥230,621.93	¥468,916.20	¥580,899.30
所有者权益合计	¥607,495.90	¥925,221.60	¥1,074,532.40
负债和所有者权益合计	¥924,495.90	¥1,284,958.35	¥1,466,967.01

7.3.财务指标分析

7.3.1.盈利能力比率

销售净利率=(净利润/销售收入)×100%;

总资产净利率=(净利润/资产总额)×100%;

权益净利率=(净利润/所有者权益) ×100%;



盈利能力是指企业获取利润的能力，也称为企业的资金或资本增值能力，通常表现为一定时期内企业收益数额的多少及其水平的高低。对于经营者来讲，通过对盈利能力的分析，可以发现经营管理环节出现的问题。从以上盈利能力各指标来看，我公司的销售净利率、总资产净利率和权益净利率都较高，并且有逐年上升的趋势，说明我公司具有较强的盈利能力和较大的盈利上涨空间。

项目	2022	2023	2024
销售净利率	0.22	0.32	0.40
总资产净利率	-0.06	0.17	0.23
权益净利率	-0.06	0.17	0.26

7.3.2.营运能力比率

总资产周转率=(销售收入/资产总额)×100%;

应收账款周转率=(销售收入/应收账款平均额)×100%;

营运能力是指企业的经营运行能力，即企业运用各项资产以赚取利润的能力。企业营运能力的财务分析比率有：存货周转率、应收账款周转率、营业周期、流动资产周转率和总资产周转率等。这些比率揭示了企业资金运营周转的情况，反映了企业对经济资源管理、运用的效率高低。我们选取了总资产周转率和应收账款周转率两个指标来衡量我公司的营运状况，可以看出，我公司的总资产和应收账款处于有效运转状态，尤其是应收账款，周转速度较快。

项目	2022	2023	2024
总资产周转率	0.71	0.53	0.58
应收账款周转率	3.45	1.91	2.25

7.3.3.偿债能力比率

资产负债率=(负债总额/资产总额)×100%;

速动比率=(货币资金/流动负债)×100%;

流动比率=(流动资产/流动负债)×100%;



偿债能力是指企业偿还到期债务（包含本金及利息）的能力。能否及时偿还到期债务，是反映企业财务状况好坏的重要标志，因此，为了维护企业的良好信誉，增强投资者的信心，必须十分重视对债务的偿还能力。由于公司处于创业初期，本着安全性的原则，公司适用较低的财务杠杆，资产负债率比较低，财务风险也比较低。同时，资产负债率较低也反映了公司具有较强的融资能力，在发生资金危机时，可以用较低的成本筹到资金。

项目	2021	2022	2023
资产负债率	0.01	0.01	0.12
速动比率	--	248.05	6.13
流动比率	--	344.02	8.05

7.4.投资回报分析

7.4.1.投资净现值分析

我们进行投资净现值（NPV）分析。净现值指未来资金（现金）流入（收入）现值与未来资金（现金）流出（支出）现值的差额，是项目评估中净现值法的基本指标。未来的资金流入与资金流出均按预计折现率各个时期的现值系数换算为现值后，再确定其净现值。投资净现值 NPV 的计算公式如下：

$$NPV = \sum_{t=1}^n (CI - CO)_t (1 + i)^{-t}$$

2017 年至 2020 年的 A 股上市的软件开发公司年复合增长率为 7.35%，若该笔投入资金不用于投资 A 股同行业上市公司，则对投资者而言，将会产生的机会成本，这部分可以以 7.35% 计算，因而公式中的折现率（基准收益率）以 7.35% 计。经计算，项目投入运营后第三年，投资净现值为 43489.15，远高于 0，项目净效益抵付了以行业基准收益率计算的盈利后仍有大量盈余，计算期内盈利能力高于行业平均水平，因此，项目在财务上可行。



7.4.2. 内含报酬率分析

内部收益率是指项目净现值为零时的折现率或现金流入量现值与现金流出量现值相等时的折现率，用来衡量项目的获利能力，是根据项目本身的现金流量计算的，与资本成本无关。

当 NPV 等于 0 时，方程变为：

$$NPV = \sum_{t=1}^n (CI - CO)_t (1 + IRR)^{-t} = 0$$

计算得：首年 $IRR=29.84\%$ 。

内含报酬率达到 29.84%，远大于资金成本率 7.35%，项目在财务上可行。



八、风险提示

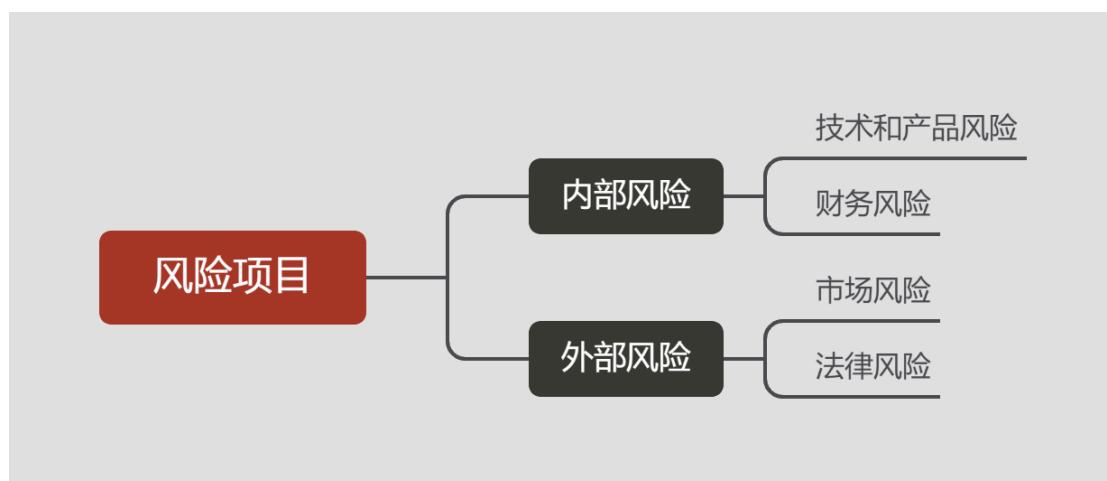
本项目在运行过程中可能面临来自内部和外部复杂多样的风险，本团队将对本项目运行过程中可能产生的风险进行甄别与确定，并制定针对性的风险规避和转化策略，实现对风险的管理和控制，使本项目保持健康生存发展。

8.1. 风险识别

本项目面临的风险主要可分为企业内部风险和企业外部风险，技术与产品的研发、和企业的财务运转等过程中产生的风险构成了本项目的内部风险。外部环境下市场的变化和 IP 授权流程中法律的合规性风险等，也是风险管理所考虑的部分。

团队主要通过以下三个方面提前预测项目风险：1.项目过往风险的经验；2.其它企业已发生过的风险经验；3.一些有可能产生的风险预兆。

8.2. 风险项目





8.3.技术与产品风险

8.3.1 技术风险分析

技术研发风险

公司在运营过程中线上部分在大数据遍历过程中容易造成后台分析数据冗杂系统运转低效的后果，导致出现线上控制风险，不利于线上便捷化端口的推广与应用。

技术保护风险

本项目技术密集程度高，技术研发过程产生的专利较多，在经营和发展中有频繁的专利活动，近年区块链技术发展迅速，对专利的竞争激烈，因此在技术的专利保护上面临一定风险。

产品同质化风险

本项目技术应用创新程度高，但随着现代信息手段、检测分析手段的不断发展以及黑客攻击技术的提高，产品在推向市场后被仿制的速度越来越快，面临同质产品挑战的奉献。

8.3.2 应对策略

加强知识产权保护

对于核心性技术申请技术专利，从法律上保证核心技术的市场竞争力。注重完善技术合同的约定，更好地防范技术泄露可能带来的风险。

强化技术优势

针对线上控制问题对后台运行分析系统进行算法改进，对系统运行算法进行优化改进并增强线上运营推广。持续追踪市场需求，不断更新升级现有技术，形成并维持技术优势。



8.4.财务风险

8.4.1 财务风险分析

融资风险

本项目的前期投入研发费用较大且具备一定的公益性。产品进入市场后，仍需大量的资金投入进行技术与产品的研发，因此作为初创型项目，预计将面对融资渠道和融资额度的风险。

现金流量周转风险

现金流量的周转情况与储备情况关乎项目后续新技术的研发，新产品的开发。无力站稳市场，严重情况下，与客户的财务往来中出现逾期、坏账等意外状况，会对项目的资金链维系造成严重打击。

8.4.2 应对策略

扩大资金来源渠道

一方面吸引风险投资或政府基金，寻求与资金雄厚的企业或者政府机构合作，争取稳定投资，控制资金风险。；另一方面，产品运营的部分利润也将会持续投入到产品研发当中，做好资金的“开源”。

合理利用资金

对于通过各种渠道获得的技术创新资金，在使用时一定要保证合理使用，团队共同努力来做到“节流”，研发中做好规划，运营过程控制成本，推广中精准识别目标客户，减少不必要的开支。



8.5.市场风险

8.5.1 市场风险分析

市场接受程度风险

作为覆盖文创产品市场领域全价值链的创意社区平台，项目产品拥有创新性的技术支持，但产品尚未进行大规模推广应用，面对全新的市场，没有可靠的经验可供借鉴，难以确定消费者的认可程度，同时，也由于消费的惯性力使然，客户群体对新产品的接受能力有一定的风险性。

市场竞争风险

项目产品研发到进入市场等必要流程的用时较长，如何抓住市场窗口期，在短时间内创立优秀知名品牌，形成完整的平台覆盖，并抢占一定的市场份额，将成为一个市场风险问题。

8.5.2 应对策略

加强产品和消费者的衔接

充分掌握市场信息，做好市场预测。通过用户体验机制、市场调研等方式等在产品正式投入市场前收集用户反馈数据，细化潜在消费者需求，预测目标市场对产品的接受程度，并进一步更新产品以更满足用户需要，增强产品的市场应变能力。加紧布局平台的搭建开发，在产品和市场的衔接中，及时占据市场先机。

8.6.法律风险

8.6.1 法律风险分析

IP 授权流程须符合我国《著作权法》等相关法律规定，但 IP 授权行业领域的法律实务分析相对较少，平台在规范授权方与被授权方之间的权利、义务以及利益分配和授权合同审查方面面临法律风险。



8.6.2 应对策略

在项目设计过程中，咨询或聘请专业的法律人员对授权流程的各种风险事先进行预测、规避，针对 IP 授权过程中可能发生的侵权行为及合同审查的法律风险进行相应的流程设计，推动 IP 授权行业规范发展。

8.7.系统运行风险

8.7.1.系统运行风险分析

公司在运营过程中线上部分在大数据遍历过程中容易造成后台分析数据冗杂系统运转低效的后果，导致出现线上控制风险，不利于线上便捷化端口的推广与应用。

8.7.2.应对措施

针对线上控制问题对后台运行分析系统进行算法改进，对系统运行算法进行优化改进并增强线上运营推广。



附 1：参考资料

- [1] 郭万超. 博物馆文创的市场逻辑及提升路向 ——对"故宫文创热"的思考[J]. 人民论坛,2019(9):127-128. DOI:10.3969/j.issn.1004-3381.2019.09.063.
- [2] 李中南. 文创与消费者的新链接——以当今国内市场为例[J]. 汽车博览,2021(1):140.
- [3] 王爱莉. 文博单位文创商品研发的市场开发与定位研究[J]. 中国新通信,2018,20(18):228-230. DOI:10.3969/j.issn.1673-4866.2018.18.201.
- [4] 陈奇伟. 文创企业借助资本市场转型升级战略研究 [J]. 经济师,2016(4):278-279. DOI:10.3969/j.issn.1004-4914.2016.04.144.
- [5] 刘扶全,陈锦航. 互联网巨头时代的中国博物馆文创市场分析 ——从文化消费到品牌消费[J]. 现代企业文化,2021(24):26-28.
- [6] 乌日勒. 地域文化创意产品设计研究[J]. 包装工程,2022,43(8):257-259,265. DOI:10.19554/j.cnki.1001-3563.2022.08.033.
- [7] 鲁小艳. 文化创意产业如何优化创新[J]. 人民论坛,2020(3):138-139. DOI:10.3969/j.issn.1004-3381.2020.03.043.
- [8] 张建东. 区块链版权产业应用探析 [J]. 中国版权 ,2020(3):53-58. DOI:10.3969/j.issn.1671-4717.2020.03.011.
- [9] 孙静. 基于"区块链+版权平台"的原创短视频版权保护路径探析[J]. 中国出版,2022(7):58-62. DOI:10.3969/j.issn.1002-4166.2022.07.014.
- [10] 王怀营,曹三省. 基于国密区块链的融合媒体版权保护系统设计与实现 [J]. 中国传媒科技,2022(4):128-132. DOI:10.19483/j.cnki.11-4653/n.2022.04.039.
- [11] 廖柏寒. 区块链在版权存证上的技术优势与未来应用 ——基于"版权链 + 天平链" 协同取证第一案的启示 [J]. 出版广角 ,2021(21):84-87. DOI:10.16491/j.cnki.cn45-1216/g2.2021.21.023.
- [12] 陈策. 区块链经济视域下 NFT 艺术品的著作权利益保护[J]. 中国商论,2022(6):109-111. DOI:10.19699/j.cnki.issn2096-0298.2022.06.109.
- [13] 江哲丰,彭祝斌. 加密数字艺术产业发展过程中的监管逻辑 ——基于



NFT 艺术的快速传播与行业影响研究[J]. 学术论坛,2021,44(4):122-132.
DOI:10.16524/j.45-1002.2021.04.012.

[14] 王功明. NFT 艺术品的价值分析和问题探讨[J]. 中国美术,2021(4):38-43.
DOI:10.3969/j.issn.1003-045X.2021.04.008.

[15] 唐榜. 基于 Node.js 的 Web 服务端框架研究与实现[D]. 四川:西南科技大学,2021.

[16] 银河证券: 助力文化强国扬帆起航, 游戏乘东风之势出海

[17] 清华五道口国家金融研究院: 2012-2020 年文化产业投融资报告



2



3

政策导向 · 两大政策导向支持项目实施

Policy Orientation: Two major policy directions to support project implementation

文化与科技融合发展

加强版权保护 促进创新发展

近五年我国国家层面共出台支持文化与科技融合发展的相关政策**11条**
仅2021年一年我国国家层面共出台加强版权保护相关政策**6条**

4

市场潜力 · 需求扩张和技术发展带来广阔市场前景

Market Potential: Broad Market Outlook due to Demand Expansion and Technology Development

IP授权需求扩展

- 伴随着文创市场的扩张IP衍生品授权商品的需求增大，授权商品销售额已达到**1374亿元**
- 相应地，授权服务市场规模已达到**53.2亿元**

年份	销售额 (亿元)	市场规模 (亿元)
2019	992	38.2
2020	1106	41.5
2021	1374	53.2

技术不断发展完善

创作来源丰富 IP多样化

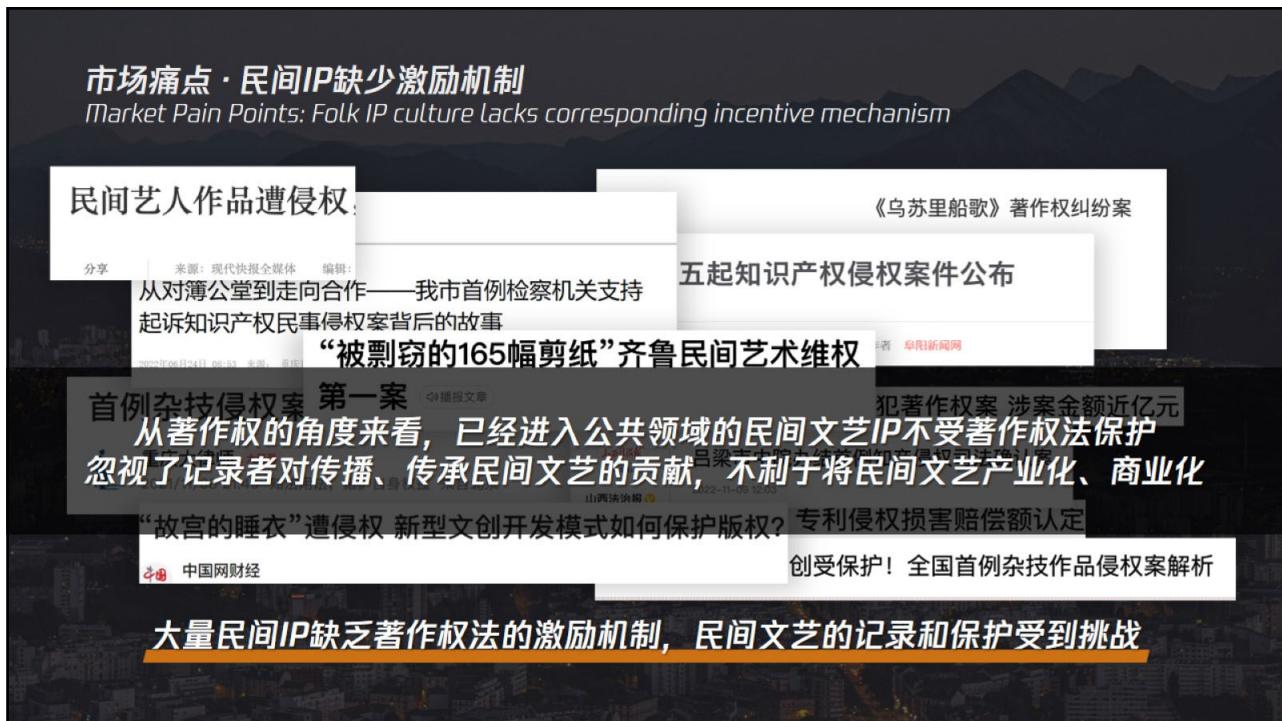
产出效率提升 成本降低

传播范围更广 触达用户

5



6



7



8



9

核心技术 · 通过三大特性打通价值链
Core Technologies: Open up the Value Chain by Achieving Three Features

大数据推荐算法
 我们基于大数据4V+特性的数据评价模型，定义影响因子和传播因子等因素共同作用IP的价值属性与UP调性，并设计供需匹配算法。

智能合约
 依靠智能合约实现实现权益凭证可信流转，确保交易内容不可篡改、交易行为不可抵赖与整个交易过程的隐私安全性。

非同质化通证 (NFT)
 随授权交易完成后，为每一件授权商品生成全网唯一且不可篡改的非同质化通证，有效防伪的同时限制了被授权方超额使用授权IP。

供需匹配

安全交易

防伪授权

10

产品功能1 · IP推荐算法
Product Feature 1: IP Recommendation Algorithm

影响因子	购买欲因子	IP自身调性	讨论者人口特征
<ul style="list-style-type: none"> 话题点赞数 话题评论数 	<ul style="list-style-type: none"> 电商搜索数 相关讨论数 	<ul style="list-style-type: none"> IP亚文化标签 IP形象属性 IP情感属性 IP出处与背景 IP内容根基 	<ul style="list-style-type: none"> 性别特征 年龄特征 地域特征 职业特征 兴趣标签
传播因子	转化因子		
<ul style="list-style-type: none"> 话题转发数 话题分享数 	<ul style="list-style-type: none"> 客单价 月销量 		

讨论热度 **变现能力** **IP调性** **受众画像**

根据社交媒体上用户对IP的讨论建立IP推荐系统以便购买方选择合适IP

11



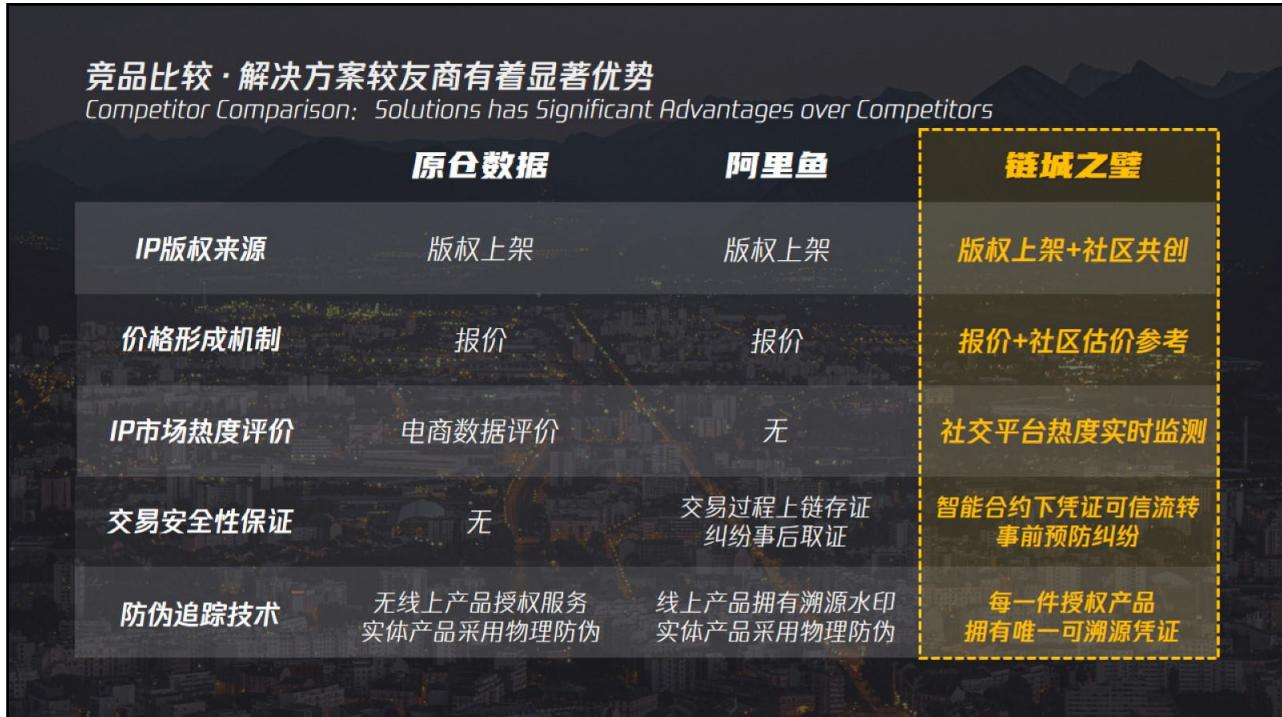
12



13



14



15

商业模式 · 提供可信交易与防伪方案的B端服务
Business Model: B-side service providing trusted transactions and anti-counterfeiting solutions

提供交易平台、提供IP供需对接与IP咨询服务、提供防伪方案、提供交易纠纷取证服务

交易前服务 → **交易中后服务**

- 提供APP付费会员，会员权限包括IP推荐系统的“IP调性”分析权限与“受众画像”分析权限，定价为**68元/月**
- 后期数据完善后将采取按月使用次数分梯度定价，并开放单一权限购买
- 向交易双方收取交易佣金，交易佣金为双方各收取**成交额1%**
- 出现交易纠纷后链上数据取证服务为**280元/次**，申请者支付
- 防伪卡制作费为**1.2元/件**，含数字藏品支持

16

市场推广 · 由UGC版权与校园版权服务起步向PGC版权服务扩展
Marketing: Expanding from UGC copyright and campus copyright services to PGC copyright services

01
校园版权与UGC IP
Campus copyright and UGC copyright

02
PGC IP与网文、影视版权
PGC copyright and online literature or Film copyright

现有校园UGC IP案例:
夏夏与她的朋友们
作者: 夏卟夏卟味
[中央美术学院]

17



22



23

项目进展·积极向社会各界寻求合作机会
Project Progress: Actively Seeking Cooperation Opportunities from all Sectors of Society

2022-07-09 20:25:52
HUAWEI
 行业: 信息技术服务业
 地区: 广东省-深圳市-龙岗区
 企业名称: 基于华为云区块链的应用设计和实现
 ● 有对接意向

2022-07-09 20:25:12
零数科
 行业: 信息技术服务业
 地区: 上海市-上海市-浦东新区
 企业名称: 基于华为云区块链的应用设计和实现
 ● 有对接意向

西安市大学生创新创业成长计划“创业实践”活动
创业双选会
入驻邀请函
 邀请人: 链城之堡
 邀请单位: 西安市微软创新中心

零数科 LINGSHU.NET **HUAWEI 云** **Microsoft** **NLVC** NORTHERN LIGHT VENTURE CAPITAL

在上届互联网+产业赛道中
 与零数科技、华为两家企业达成对接意向

获得陕西微软创新中心的入驻邀请
 已向北极光创投TMT团队递交商业计划

24

产教融合·依托西安电子科技大学区块链工作室开展产教融合工作
Industry-Rducation Integration: Relying on Xidian University Blockchain Studio to Commence Work

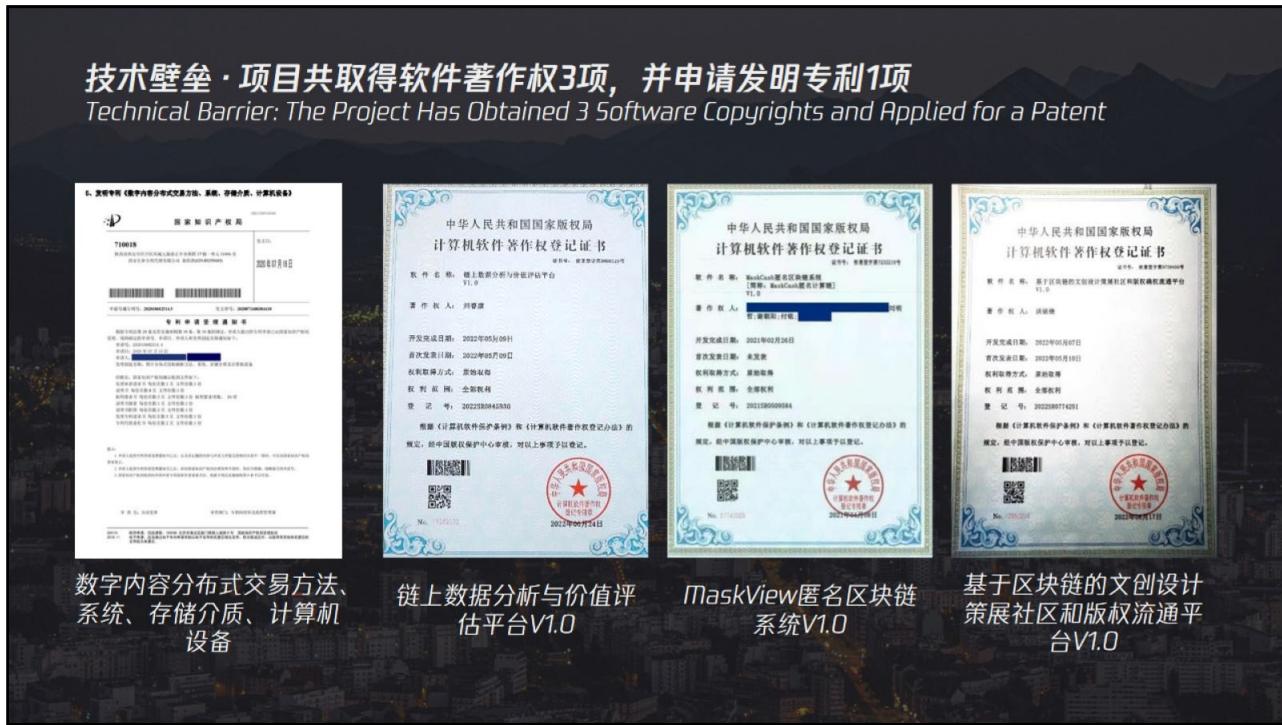
XDLR 西电链融
 陕西省区块链与安全计算重点实验室
 西安电子科技大学区块链应用与评测研究中心
 西安市移动边缘计算及安全重点实验室
 于全院士工作站
 起航 创新 发展 合作 共赢

团队成员
 基于FISCO BCOS的学历证书系统—融信链
 “融信链”是一个使用FISCO BCOS联盟链内为底层架构的电子学历证书系统，利用FISCO BCOS的多方参与共识，数据不可篡改和智能合约自动执行特性，确保证书数据来源安全可靠性。为学生提供方便的企业用人信息便捷机制，以及对评价机构和数据提供者的数据审核机制，开放、公平公正和高效的国内生态，为建立诚信社会打下坚实基础。

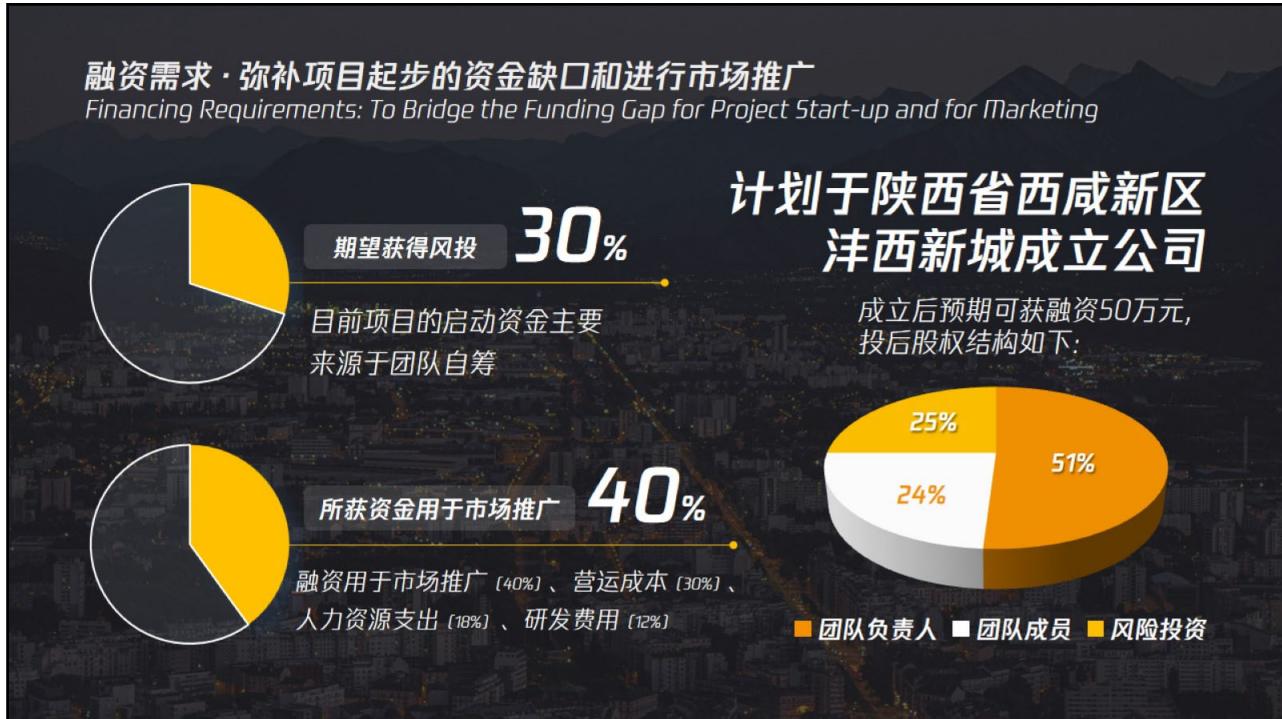
教育引领
 团队依托西安电子科技大学星火众创空间区块链工作室展开科技创新工作，在原始技术创新、区块链+产业创新应用探索、区块链人才培养取得较大进展，开展开发培训、竞赛培训等数场

实践育人
 团队成员同时依托区块链工作室，与西电链融科技、微众银行、蚂蚁科技等企业开展生产实习，参与蚂蚁链、工商银行、微众银行Fisco Bcos等举办的各类孵化、竞赛活动

25



26



27



西安电子科技大学

大学生创新创业训练计划项目 结题验收报告

项目级别：	<input checked="" type="checkbox"/> 国家级 <input type="checkbox"/> 省级
项目编号：	S202210701007X
项目名称：	基于区块链的文创设计策展社区和版权确权流通平台
项目类别：	<input type="checkbox"/> 创新训练 <input checked="" type="checkbox"/> 创业训练 <input type="checkbox"/> 创业实践
作品形式：	<input type="checkbox"/> 硬件 <input checked="" type="checkbox"/> 软件 <input type="checkbox"/> 研究报告 <input checked="" type="checkbox"/> 计划书 <input type="checkbox"/> 其它
项目组成员：	洪铭锋、刘睿康、王渝普、王立文、刘博涵
指导教师：	崔江涛
项目申请单位：	西安电子科技大学
项目起止时间：	2022 年 9 月 至 2023 年 6 月

二〇二三年六月

西安电子科技大学教务处制

填 表 说 明

1. 本表供西安电子科技大学国家大学生创新创业训练计划项目结题验收使用。
2. 结题验收报告须用 A4 纸打印并正反面套印装订，内容要具体、真实，并经审核认可。
3. 作品形式分“硬件、软件、研究报告、计划书、其它”可多项选择。
4. 结题验收附件材料含有 1-5 项，需另外打印装订，要求同上。
5. 填写表格内容统一使用宋体，小四号字，要求单倍行距。
6. 表格各栏目纵向长度可随填写内容的多少而调整，但不要改变表格各栏目固有格式和前后顺序。
7. 结题验收报告及附件材料各一式三份。
8. 此页不必打印。

一、项目组成员（成果排名以所列序号为准，姓名栏本人签字）										
序号	姓名	学号	学院	成员分工	工作量化比例					
1	洪铭锋	20069100191	经济与管理学院	产品、商业策划	50%					
2	刘睿康	20009200584	计算机科学与技术学院	开发	15%					
3	王渝普	20012100012	通信工程学院	开发	15%					
4	王立文	20009200043	计算机科学与技术学院	开发	10%					
5	刘博涵	20009201455	计算机科学与技术学院	开发	10%					
6										
二、本项目经费使用情况（明细单见附件）										
1. 项目计划经费： 1 万元	2. 项目使用经费： 0.13 万元									
三、项目实施情况										
<p>完成平台搭建，提供源码编译方式，云端 API 接口； （已完成，源码、软件需求文档、用户操作手册见附件，平台已申请软件著作权 2 项）</p> <p>进行模拟企业运营，在项目开发结束后完成详尽的可行性分析和小规模的试点测试，并作出报告； （已完成，项目小规模试点已获技术应用证明与入驻协议，项目先后获得：西咸新区沣西新城微软创新中心、西安高新技术产业风险投资有限责任公司、西安曲江文化产业投资（集团）有限公司、远诺科技前研科投（西安）科技有限公司、黄陵工业园区陕西省高新技术产业开发区共 5 家企业/孵化器/产业园区的投资意向，以及上海零数科技有限公司、华为技术有限公司共 2 家企业的合作对接意向，商业计划书见附件）</p> <p>参加创新创业相关大赛并取得成果。 （已完成，项目先后获得：西安市大学生创新创业成长计划优秀创新创业项目、第 8 届中国国际“互联网+”大学生创新创业大赛陕西省银奖、第 21 届微软 Imagine Cup 复赛赛区优秀奖、第 13 届全国大学生电子商务“创新、创意及创业”挑战赛陕西省一等奖、陕西省第 14 届工业工程改善创意竞赛校一等奖（未完赛，已推省赛），相关证明材料见附件）</p>										

项目主要成果

1. 论文发表及专利申请等（论文共____篇；专利/软著申请共2项）
2. 其它成果：竞赛获奖 5 项

四、项目的创新点

该项目采用区块链技术来实现版权交易和确权，这是一种全新的技术手段，具有去中心化、不可篡改、安全可靠等优点。该平台利用智能合约实现自动化处理、快速结算和透明管理，使得版权交易和确权变得更加高效和便捷。与此同时采用区块链技术实现数据的去中心化存储和管理，同时保护用户的隐私数据。所有数据都被加密存储，只有用户本人持有私钥才能访问和控制数据，从而保障了用户的数据隐私和安全。

该项目将文创设计师和投资人、收藏家等文创产业链的各个环节紧密连接起来，实现了版权交易、文创设计策展、投资和收藏等多种功能，从而促进了文创产业的良性发展。该平台还提供了个性化的服务和定制化的用户体验，满足了用户对个性化和定制化服务的需求。不仅实现了文创设计作品的版权交易和确权，还实现了投资和收藏的功能。投资人可以通过该平台获得高品质的文创设计作品，同时可以获得作品增值的收益。收藏家可以通过该平台方便地收藏和管理自己喜爱的文创设计作品，同时可以获得作品的增值和流通价值。还将社交功能与版权交易和文创设计策展功能相结合，实现了社区化的文创设计平台。用户可以通过该平台展示自己的作品、与其他用户交流和合作、获得更多的文创设计资源和信息等，从而促进了文创设计行业的交流和合作。

五、项目过程开展情况

洪铭锋：研究市场需求，了解文创设计师和版权交易市场的需求和趋势。收集用户需求和反馈，提出产品需求和功能规划。根据市场需求和用户反馈，制定产品的特性和用户界面设计。确定产品的价值主张和定位，制定产品营销策略。确认产品的功能和特性，维护产品路线图，并制定产品优先级。作为负责人组织和协调团队开发工作，包括前端开发、后端开发、智能合约开发和测试等。确定项目的范围、进度和质量目标，制定项目计划和工作分配。

监督项目进展，并及时调整计划和资源分配，确保项目按时交付。跟进团队成员的工作进展，解决团队成员之间的协作问题。管理项目的风险和变更，制定应对方案和风险管理计划。（完成情况：撰写产品需求文档、用户操作手册、申报软件著作权）

刘睿康：确定后端技术架构，在多个技术选项中选择最适合项目的技术，例如使用哪种编程语言、使用哪种数据库管理系统等。设计和开发数据库，确保其能够支持高并发和高可靠性的数据存储和访问。设计和开发系统的API，确保其能够支持各种类型的请求和响应。（完成情况：撰写用户操作手册、申报软件著作权）

王渝普：设计和开发智能合约，确保其能够实现版权确权和交易等功能。此外，还需要考虑智能合约的安全性和可靠性。设计和开发分布式账本，确保其能够实现版权交易的可追溯性和安全性。（完成情况：链上部署、智能合约部署）

王立文：收集大量的文创设计市场数据，包括作品的类型、风格、时代背景等方面的信息，以及市场交易数据、价格等信息。在多个算法选项中选择最适合项目的算法，例如基于机器学习、基于统计分析等算法。基于收集到的市场数据和算法设计，算法工程师需要设计和开发估价算法。算法需要考虑作品的特征和市场需求，计算出作品的价值。（完成情况：算法开发）

刘博涵：设计和开发用户界面，确保其能够支持用户上传文创设计作品、进行版权确权和交易等操作。此外，还需要考虑用户界面的易用性和用户体验。设计和开发用户交互功能，确保其能够支持用户上传作品、进行版权确权和交易等操作。此外，还需要考虑用户交互功能的易用性。（完成情况：前端开发）

工作组核 查（签章）：

① 优 良 ② 合 格 ③ 不 合 格

年 月

六、指导教师意见

指导教师（签章）：

年 月

七、学院评价意见

主管院长签字：

公 章

年 月 日

八、专家鉴定意见

专家意见：

鉴定结果：推荐优秀□ 合格□ 不合格□

全体专家组成员（签章）：

年 月 日

九、学校审批意见

负责人（签章）： 公 章

年 月 日



西安电子科技大学

大学生创新创业训练计划项目 结题验收附件材料

项目级别：	<input checked="" type="checkbox"/> 国家级 <input type="checkbox"/> 省级
项目编号：	S202210701007X
项目名称：	基于区块链的文创设计策展社区和版权确权流通平台
项目类别：	<input type="checkbox"/> 创新训练 <input checked="" type="checkbox"/> 创业训练 <input type="checkbox"/> 创业实践
作品形式：	<input type="checkbox"/> 硬件 <input checked="" type="checkbox"/> 软件 <input type="checkbox"/> 研究报告 <input checked="" type="checkbox"/> 计划书 <input type="checkbox"/> 其它
项目组成员：	洪铭锋、刘睿康、王渝普、王立文、刘博涵
指导教师：	崔江涛
项目申请单位：	西安电子科技大学
项目起止时间：	2022 年 9 月 至 2023 年 6 月

二〇三〇年六月

西安电子科技大学教务处制

附件 1:

国家大学生创新创业训练计划项目经费使用情况表

项目编号:

项目名称		项目负责人			联系方式		
所在学院				指导教师			
材料费 (器件注明 名称型号, 可 另附页)	明细	单价 (元)	数量	总金额 (元)	用途		
	腾讯云轻量应用服务器(1000GB 流量包 (MED4-60GB-1000GB), 2 核 CPU、4GB 内存 (MED4-60GB-1000GB), 36 个月)	540	1	540	平台服务器		
其他费用	创赛圈-创赛材料费用	98	1	98	参赛		
	《IP 授权商业化: 从入门到精通》电子 工业出版社	43.12	1	43.12	项目参考		
合计	0.13 万元						

附件 2:

项目成果支撑材料



第八届互联网+ 陕西省银奖（产业命题赛道）



第二十一届微软“创新杯”大赛复赛赛区 20 强

西安市大学生创新创业成长计划“创业实践”活动

创业双选会

入驻邀请函

邀 请：链城之堡

邀请单位：陕西微软创新中心



陕西微软创新中心 入驻邀请

2022-07-09 20:25:52



企业 华为技术有限公司

行业：信息技术服务业

HUAWEI

地区：广东省-深圳市-龙岗区

● 有对接意向

命题年份: 2022

产业命题: 基于华为云区块链的应用设计和实现

2022-07-09 20:25:12



企业 上海零数科技有限公司

行业：信息技术服务业

地区：上海市-上海市-浦东新区

● 有对接意向

企业对接意向



软著



软著



平台入驻协议

产品应用证明

兹证明《链城之壁》系统，通过与我公司的合作，向我公司提供了IP版权交易。

应用该系统，我公司旨在解决公司年终奖品IP的购买问题。经过产品使用后，我公司认为其面向IP著作权的创投交易具有安全可信、防伪性强等特点，在IP交易方式上具有较强的创新性。

自我公司使用该平台购买IP以来，IP授权权益凭证按时交付，公司员工反馈良好，有着很广的应用前景，期待该系统进一步的推广与应用。



产品应用证明

附件 3:

项目组成员心得体会

基于区块链的文创设计社区和版权交易平台项目的实施过程中，我们遇到了很多挑战和机遇，也获得了很多经验和收获。

首先，我们在技术方面采用了区块链、智能合约、分布式账本、数据库、API、前端、后端等多个技术领域的知识和技能，实现了版权交易和确权的安全、快速、便捷。在这个过程中，我们不断学习和探索最新的技术，选择最适合项目的技术架构，并不断优化和改进技术方案，以提高系统的可靠性和稳定性。

其次，我们在创业方面学习了市场调研、产品设计、团队协作、资源管理、风险管理、营销等方面的知识和经验。我们深入了解用户需求和市场趋势，设计出具有竞争力的产品和服务，并不断优化和改进创业方案，以提高项目的成功率和可持续性。

第三，在团队协作方面，我们建立起高效的沟通和协作机制，分工明确，任务清晰，以提高工作效率和项目成功率。我们通过协同工作，共同完成了项目的开发和实施。

第四，在价值链分析与产业调研方面，我们深入了解文创设计行业的发展趋势和特点，设计出支持版权保护和价值实现的系统和算法方案，以实现文创设计师的利益最大化。

第五，在交流学习方面，我们与其他技术人员、文创设计师、投资人等进行交流，学习他们的经验、观点和建议。我们建立起良好的人际关系和合作伙伴关系，不断学习和探索最新的技术和市场动态，以提高项目的成功率和可持续性。

第六，在实践和创新方面，我们不断地实践和创新，以验证和改进我们的想法和方案。我们积极探索新的业务领域和模式，借鉴和应用最新的技术和方法，以推动项目的发展和创新。

总的来说，基于区块链的文创设计社区和版权交易平台项目是一个具有挑战性和机遇的项目。在这个过程中，我们不断学习和探索，不断优化和改进，以实现项目的成功和可持续性。我们相信，这个项目将为文创设计行业的发展和创新提供新的思路和途径。

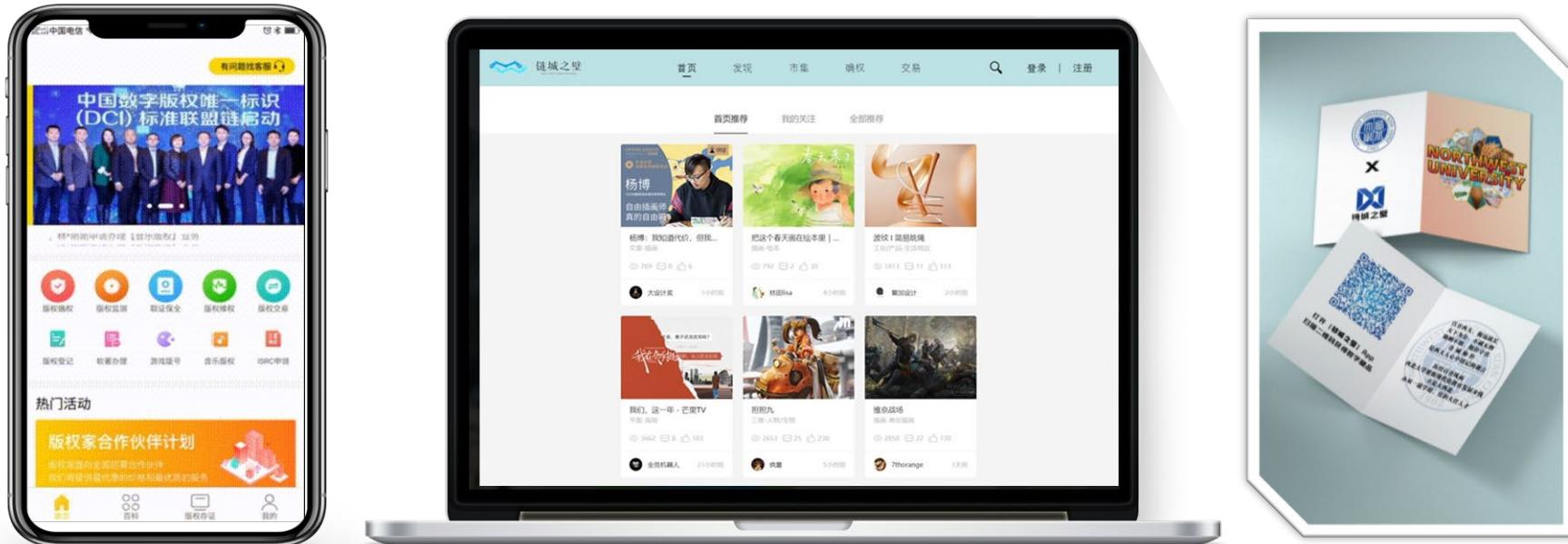
附件 4：

项目组成员联系方式

附件 5:

照片资料（含项目成员合影，指导教师照片，作品特写照片共 3 张）

作品特写:





指导老师照片：



成员合影：