

产教融合下慢病膳食 APP 的研发与教学转化

任欣澜^{1,a}, 马静^{1,b,*}

¹ 西安工业大学, 陕西, 西安, 中国

^a 1876037853@qq.com, ^b 410329074@qq.com

摘要: 不少移动健康类应用普遍存在功能堆砌问题, 会给终端设备带来较重的资源负担。高校开设的 Android 移动开发实训课程, 在工程规范方面和行业实际开发要求存在一定差距。针对上述两个现实问题, 文章围绕慢病膳食管控方向开展研究, 选择轻量化 Android 健康管理 APP 作为研究载体, 设计一套可落地的实现方案, 研究重心偏向工程实现与产教融合实践。本系统基于 MVC 分层架构, 用 SQLite 完成本地数据持久化。开发时把非必要业务模块全部去掉, 并据此定下三项轻量化性能指标, 分别管住内存占用、设备兼容适配能力和程序响应效率。本文搭建的三维传导理论框架, 由产业工程量化标准、分层实训实操任务、学生专业能力考核体系三部分构成。方案把商业项目严格的开发验收规范转成阶梯式教学流程, 把完整工程项目拆成入门、进阶、高阶三个层次的实训单元。实践测试表明, 把移动端应用全流程开发引入高校移动开发课程, 能较好改善学生工程实践经验不足的问题, 实训内容覆盖数据库建模、性能优化、多设备适配等关键工程内容。经过两轮教学迭代和校企联合评审验证, 参训学生在 MVC 分层开发、工程目标标准化交付等核心实操任务上的完成质量有明显提升。企业指导教师对学生代码规范性与项目完整交付水平给予高度认可, 证实该产教融合实训方案适配高校课堂教学场景, 具备良好的普及与推广价值。

关键词: 产教融合; Android 开发; 慢病膳食管理; 三维传导框架

基金信息: 西安工业大学省级大学生创新创业项目 (项目编号: S202510702140)

Research and Teaching Transformation of a Lightweight Chronic Disease Dietary Management Android APP Based on Industry-Education Integration

Xinlan Ren^{1,a}, Jing Ma^{1,b,*}

¹ Xi'an Technological University, Xi'an, Shaanxi, China

^a 1876037853@qq.com, ^b 410329074@qq.com

Abstract: Many mobile health applications suffer from feature overload, which places a heavy resource burden on terminal devices. Android mobile development practical courses offered by universities also tend to fall short of industry requirements in terms of engineering standards. To address these two issues, this paper focuses on chronic disease dietary management and takes a lightweight Android health management app as the research carrier. It designs an implementable

solution, with emphasis on engineering implementation and industry-education integration. The system adopts an MVC layered architecture and uses SQLite for local data persistence. All non-essential business modules are removed during development, and three lightweight performance indicators are established to constrain memory usage, device compatibility, and program response efficiency. The three-dimensional transmission theoretical framework constructed in this paper consists of three parts: industry engineering quantitative standards, layered practical training tasks, and a student professional ability assessment system. The solution converts the strict development acceptance standards of commercial projects into a stepwise teaching process and divides a complete engineering project into three training levels: beginner, intermediate, and advanced. Practical tests show that introducing the full development process of mobile applications into university mobile development courses can effectively address students' lack of engineering practice experience. The training content covers key engineering areas such as database modeling, performance optimization, and multi-device adaptation. After two rounds of teaching iteration and joint school-enterprise review, students showed clear improvement in the quality of completing core practical tasks such as MVC layered development and standardized project delivery. Enterprise instructors highly recognized students' code standardization and complete project delivery, confirming that this industry-education integration training program is suitable for university classroom teaching and has good potential for wider adoption and promotion.

Keywords: Industry-Education Integration; Android Development; Chronic Disease Dietary Management; Three-Dimensional Conduction Framework

1 引言

数字健康领域与移动终端技术持续产生交融。居家场景下的慢性病膳食干预手段,在慢病防控工作中逐渐占据主要位置,市场端也在不断释放需求,希望获得轻量化、定位清晰的膳食管理数字工具。市面上综合类健康 APP 通常集成了问诊、社交、运动监测等多项附加模块。这类程序占用内存资源较多,整体操作流程也偏复杂,很难契合中老年慢病群体的使用特点。部分定向膳食软件,并未针对糖尿病、代谢类病症设计区分化的膳食指导方案。市场层面仍然缺少足够的轻量化慢病膳食工具供给[5]。

从学术领域来看,海外研究大多围绕慢病监测算法开展临床效果验证。相关研究并未给出适配高校工程教学的落地实施方案。国内关于数字健康 APP 开发的多数文献,把重心放在业务功能开发上。轻量化对应的各项量化指标未得到足够重视,完整商用产品开发流程也很少被加工成分层设置、具备统一标准的教学材料[6]。

国内高校《Android 移动开发》课程同时还长期存在严重的产教脱节问题:课堂实训大多采用没有真实业务约束的简易 Demo,数据库建模、异步调度、多机型适配、性能优化这些企业刚需实战技能,学生很难掌握到;商用项目耦合度高、业务复杂度也高,直接拆分用于课堂教学是做不到的[7]。现有相关研究存在显著割裂:移动开发教改文献只聚焦教学方法优化,产业开发量化标准与课堂评价体系之间的映射关系没有建立起来;健康应用研发文献只关注产品落地,配套分层实训转化方案比较缺少,贯通产业需求、项目开发、分层实训、量化考核的一体化理论与实践体系还没能形成[4]。

现有市场供给、专业教学和学术研究体系都存在明显短板。本文结合这三类现实问题,从理论建构、产品研发、教学落地三个角度设计了一整套创新方案。研究构建了专属的三维理论框架,依托产业工程约束来对接校园教学标准。这套架构能够衔接企业开发准则与课堂考核体系,消除两类场景之间的衔接断层。

研究围绕产品研发建立了新的开发范式,专门适配慢病膳食场景下的轻量化 Android 应用开发工作。研究引入多项可量化性能指标,精准把控产品轻量化运行状态。教学端依据分层培养思路搭建三级实训体系,并配套校企联合制定的量化考核规则。整套体系可为移动开发课程提供标准化的产教融合教学资源。

本次研究以轻量化慢病膳食 Android 应用开发作为实践载体,确定两大核心研究任务。一方面针对慢病膳食相关核心服务开发轻量化工具,保障工具能够适配不同终端设备的运行环境;另一方面搭建标准化落地流程,将商用移动端开发项目改造为层次分明、循序渐进的教学实训案例。

研究拆解产业硬性工程规范,将其转化为适配课堂教学的梯度化实训内容。依托实操学习模式落实产教融合育人思路,形成可复用的实践体系。该体系能够支撑数字健康应用研发工作,同时服务计算机专业工程人才的常态化培养工作[1]。

2 项目定位与三维传导框架

2.1 项目定位

不少面向慢病膳食场景的移动应用存在功能堆砌的问题,也没有建立起对应的轻量化评判标准。高校提供的 Android 实训案例与商用工程标准之间存在偏差,缺少能够直接使用的量化考核依据。本研究选取慢病膳食管控轻量化 Android 应用开展实践,项目同时对接产业落地需求与教学转化工作。

项目面向减脂、糖尿病群体的居家膳食干预诉求,移除了社交、问诊这类非核心业务模块。研究确定了内存占用、机型兼容、加载时延三项硬性轻量化指标,开发出资源开销更低的垂直健康管理工具。

本研究以原生 MVC 架构为基础,结合 SQLite 数据库搭建整套开发技术体系,开发出可分层拆解、适用于课堂实训的标准化教学资源。基于自研的三维传导理论模型,本研究将企业项目上线落地的正式验收规范,转化为契合高校教学场景的阶梯式实操训练体系,整体分为入门、进阶、高阶三个学习层级。同时,研究建立了落地性较强的学生工程能力评价机制,并提炼出一套通用性突出、

可跨领域迁移参考的移动开发产教融合创新教学范式[4]。

2.2 工程约束三维传导框架

本研究把轻量化慢病膳食 Android 应用的原生 MVC 分层架构,还有 SQLite 本地持久化技术栈作为依托,去把适配递进式工程实训的教学载体给构建出来;分层解耦、离线数据存储、异步接口调度这些核心技术模块,都能契合到《Android 移动开发》课程的核心教学知识点上。需求调研、架构设计、数据库建模、迭代开发、多终端兼容性验证这整条完整产品研发链路,会覆盖到全产业工程闭环,也能被拆解成模块化分层实训单元;传统实训教学偏重语法编码、弱化系统工程思维培育的这个固有缺陷就能被弥补掉,学习者也能把标准化商用软件开发范式给建立起来[4]。

针对产业开发规范与高校教学评价体系割裂的现实矛盾,本文创新性提出了产业工程量化约束一分层实训实操任务一专业能力教学标准三维传导理论框架,这也是全文核心理论创新点。该框架用了“输入层—转换层—输出层”闭环传导架构,左域是产业工程量化约束(标准输入源),中域是分层实训实操任务(中介转换载体),右域是专业能力教学标准(评价输出端),三维结构会形成双向映射、动态反馈的传导逻辑,三维传导理论框架图如图 1 所示。

2.2.1 产业工程量化约束

产业工程量化约束是三维传导框架的底层输入基准,它的指标体系是从移动医疗应用上线审核规范、跨终端适配开发准则、垂直慢病业务运营管控条例等产业刚性验收标准里来的,跟高校那种无资源阈值、无业务边界约束的教学演示项目有区别;全部约束指标都有着可量化、可复现、可核验的工程属性,能划分成四类标准化约束体系。

本次研究设置了明确的性能约束标准,对应用运行过程中的内存峰值进行上限管控,将阈值固定为 70MB。同时,程序兼容适配 Android 5.0 及以上版本系统,要求本地膳食数据集的单次加载响应时长控制在 0.5 秒以内。

编码架构提出 MVC 分层解耦的硬性要求，项目执行统一编码规约。模块按照低耦合思路完成拆分，配套接口文档实现标准化管理。

数据业务选用 SQLite+Room 完成离线持久化存储。远程 API 调用遵循异步调度相关机制，明确多表关联查询、数据库事务的处理准则。

项目交付需要产出完整标准化工程文档，Git 版本迭代执行规范化管理流程。业务链路完成闭环校验，还需要开展多轮灰度迭代测试。

其扭转了传统实训偏重功能开发、忽视工程约束的教学问题。它给分层实训体系提供了来自产业端的开发基准，充当三维传导框架的底层规则来源[4]。

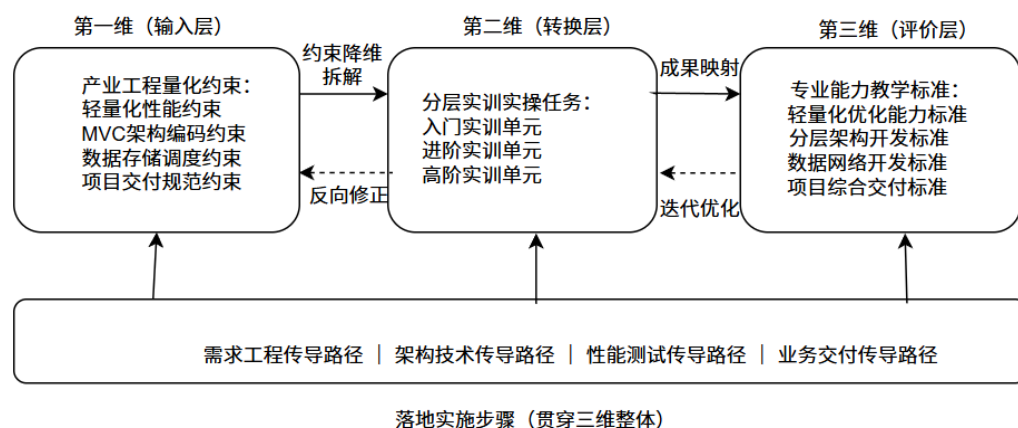


图1.三维传导理论框架图

2.2.2 分层实训实操任务

分层实训实操任务会承担三维传导框架的中介转译功能，核心作用就是对第一维耦合度较高的复合型产业量化约束去实施分层降维拆解，把抽象化、综合性的产业验收规范，转化成难度梯度有序、学习者可分步落地的课堂实操内容。

基于认知负荷理论，本研究把实训任务划分成入门、进阶、高阶三级梯度单元，各层级实训任务与对应产业工程约束之间会形成一一映射关系。

入门实训单元匹配界面开发、单表数据存储类基础约束，完成首页膳食浏览模块搭建、静态食材可视化渲染、SQLite 单表 CRUD 操作；

进阶实训单元匹配分层架构、网络交互类综合约束，完成完整 MVC 三层编码实现、远程食材 API 异步数据拉取、JSON 结构化解析与异常容错机制构建；

高阶实训单元对应的是轻量化性能、全链路业务闭环这类高阶约束，多数据表关联检索、膳食风险分级算法开发、内存泄漏定位修复、跨终端兼容适配全流程调优。

各实训单元都会绑定上明确的量化工程指标，商用项目业务复杂度高、课堂落地难

度大的现实障碍就能被有效消解掉，产业开发标准与高校人才培养之间的传导媒介也就被搭建起来了[6]。

2.2.3 专业能力教学标准

专业能力教学标准作为三维传导框架的终端输出模块，会把分层实训任务交付成果当成核心评判依据，构建出分项化、可量化的学生工程能力评价体系，去替代传统定性化、模糊化的课堂考核模式。

本维度依托分层实训任务拆解形成四大类标准化能力评分细则，实现产业验收指标向课程考核标尺的直接转化。

研究针对轻量化性能优化水平设置了量化评分标准，评分依据内存阈值标准、终端适配表现、数据响应速度三项核心参数判定。

分层架构开发的规范程度，主要参考 MVC 分层合规程度、编码规范落地情况、模块耦合控制质量进行综合评判。

数据存储与网络开发的实操水平，重点考察数据库建模规范、异步接口异常处理、结构化数据解析的落地成效。

商用项目整体交付质量，依托工程文档完整程度、Git 版本管控标准、膳食业务闭环落地状态完成综合测评。

这套多元评价模式实现了工程实践能力的量化测评落地。同时补齐了 Android 开发课程缺少标准化评价体系的行业短板。

2.2.4 教学应用价值

本次搭建的三维传导理论框架，能够从多维度为教学改革体系提供支撑。本研究将行业工程规范贯穿于整段实训教学全过程，重新梳理了实训项目的整体开发逻辑。传统课堂教学大多只注重功能能否成功实现，忽视标准化工程规范的培养，而该理论框架可有效弥补这一教学短板，引导学生依托行业量化工程标准完成项目开发与设计。

研究依托 MVC 分层架构拆分阶梯化实训任务，贴合不同阶段学生的知识储备基础。该设计有效降低了商用项目高复杂度带来的学习压力，减轻了学生的认知负担。

课程考核模式得到全面优化，定量测评方式逐步替代传统的定性主观评价。所有考核标准均对标产业一线开发要求，弥补了移动开发课程缺少标准化评价机制的缺陷。

这一理论框架的场景兼容性和迁移应用潜力都不错，适用方向并不止于慢病膳食管理。像电商平台、工业终端、智慧医疗等各类 Android 开发实训场景，都能拿来适配和拓展。总的来说，该框架能为移动开发领域的产教融合教学改革提供比较扎实、通用性也较强的理论支撑。

2.3 落地实施步骤

三维传导框架可以从宏观层面确立产教融合的落地范式。产业工程标准想要真正融入课堂教学，需要细化框架内部的核心执行环节。分层实训实操任务承担着产业与教学的衔接工作，是整套转化体系的核心中枢，实操落地的难度也最高。本研究把核心转化层进一步拆细，梳理出需求工程、架构技术、性能测试、业务交付这四项连贯的实施流程。这套流程并不是三维传导框架之外另加的内容，而是对分层实训实操任务的细化延伸和深度阐释。通过这种精细化设计，产业工程标准能够完整、规范地融入现有课堂教学体系。

2.3.1 需求工程实施

移动医疗行业的项目需求评审规范，是本次需求工程开展的核心约束依据。研究明

确四项量化约束标准，涵盖慢病用户分层画像构建、业务边界管控、标准化文档输出、多轮评审迭代四大内容。研究剔除各类非核心业务模块，高度贴合轻量化产品的整体开发定位。

教学环节依照认知梯度完成实训任务的分层拆分。入门阶段侧重梳理慢病用户群体特征，厘清项目基础功能内容。进阶阶段完成需求规格说明书撰写，同步完成交互原型的设计工作。高阶阶段复刻企业评审流程，落实需求迭代优化与项目文档归档工作。

本次评价体系完全对标产业实际验收标准，从画像完备度、业务合规性、文档完整度、迭代规范性维度搭建评分机制。这套机制可对学生需求规划能力开展全流程量化考评[7]。

2.3.2 架构技术实施

架构技术落地环节参照商用 Android 开发行业规范，设置了四项硬性开发准则。准则涵盖 MVC 分层隔离机制、统一编码标准、低耦合模块拆分、接口文档留存机制，能够有效规避多层代码逻辑混杂的常见工程问题[3]。

本次实训教学依据学生认知发展规律，按分层递进思路整体设计。入门阶段带学生搭建页面视图布局和基础实体模型，先帮助他们形成分层编码的开发意识，打好程序架构开发的基础认知。进阶阶段重点落在 MVC 三层架构调度运行逻辑的实际落地，完成各业务模块的规范化分包设计，同时编制标准化基础接口文档。高阶阶段以代码迭代优化为主，对项目中高耦合的代码结构进行重构整改，梳理并完善全模块接口体系，完成项目整体代码评审。本次搭建的实训评价体系，围绕分层边界合规性、代码规范落实程度、模块解耦质量、接口文档完整度四项核心指标来评分，能比较准确地反映学生在标准化架构开发上的实操综合能力。

2.3.3 性能测试实施

性能测试工作对标轻量化应用上线硬性标准，搭建了可重复核验的性能指标体系。体系包含内存峰值阈值、系统兼容范围、本地数据响应时延、静态资源压缩四项约束，依托 Android Profiler 工具与多版本真机设备完成客观检测。

实训流程根据学生认知梯度来安排,把性能调优实训任务分成由浅入深的几个层次。入门阶段主要让学生上手性能监测工具,自己把应用的基础运行状态和核心数据采集下来。到了进阶阶段,就围绕各类独立业务模块,做资源精简压缩和低版本系统兼容调试,把单表数据查询的执行效率提上去。高阶阶段则集中排查并修复应用整体的内存泄漏问题,做完多终端设备兼容性测试,再把应用全局资源占用情况统一管起来,最后整理出一份完整规范的性能调优分析报告。

本次评价体系完全依托真机实测所得数据,从轻量化指标达标情况、跨终端适配稳定程度、程序响应效率、测试文档完整程度开展客观量化考评。

2.3.4 业务交付实施

业务交付实施会对标企业软件上线交付标准,Git 规范化版本管理、全流程工程文档归档、慢病业务闭环落地、多轮自测迭代这四项目交付约束会被设定下来,工业软件完整交付流程也就被复刻出来了[4]。实训分阶段推进交付工作,入门阶段把代码仓库搭建起来并输出数据库设计文档;进阶阶段基于分支机制迭代开发业务模块,版本记录得到规范,中期答辩也完成;高阶阶段整合全模块代码、修复业务缺陷,全套工程文档被汇编出来,应用打包后参与校企联合终期验收。综合评价从版本管控规范度、成套文档完备性、业务闭环完整性、项目答辩成效这四个维度,综合评定学生的商用项目全流程交付素养。

2.3.5 路径协同逻辑

四项落地实施环节互为支撑、协同联动,构建出一套完整且支持动态迭代的闭环运行体系。需求工程环节主要梳理并明确业务规则与性能指标标准,为后续架构研发、性能调优及项目交付工作划定统一执行基准;架构设计环节搭建完整的程序代码体系,为性能优化迭代与项目成果落地交付提供核心载体支撑;性能测试环节能够精准定位架构设计存在的缺陷与隐患,依托测试反馈结果反向驱动代码迭代更新与架构结构重构优化。

各阶段实训成果最终统一汇入业务交付环节,集中做综合审核和等级评定。研究以实训考核反馈的真实数据为依据,持续调整

实训体系里的产业约束规范和分层梯度任务内容。这样一来,三维传导理论框架可以动态兼顾课堂教学与产业开发的双重需求,本次搭建的产教融合教学模式也具备长效迭代更新能力,并能在多种教学场景中迁移复用。

3 系统总体架构设计

架构设计会把产业轻量化工程约束与学生认知递进规律统筹到一起,遵循“业务需求驱动模块划分、模块边界定义数据模型”这个核心原则去完成技术方案选型,在满足慢病膳食业务完整闭环的基础上,教学拆解性、多终端适配与低资源占用这三大核心目标也会被兼顾到,整体技术选型优先选用 Android 原生开源组件,实训环境部署门槛就能被降低,高校课堂批量教学场景也能适配[3]。

3.1 需求分析与模块划分

系统需求会被划分成功能性、非功能性这两类指标。功能性需求会面向普通减脂人群、糖尿病患者、日常健康管理用户这三类画像,账号管理、食材检索、热量核算、饮食记录、膳食健康分级、慢病膳食资讯推送这些核心业务都能被覆盖到;非功能性需求则把可量化的性能约束设定出来:Android 5.0 及以上系统能兼容、运行内存峰值不会超过 70MB、本地数据加载响应时长会被控制在 0.5s 以内,量化性能指标能直接转化成性能优化专项教学素材,学生“开发先定性能指标”的工程思维也就被建立起来了[5]。

系统按用户三类核心操作流程完成模块拆分,包含浏览查询、食材检索、健康管理,最终形成三组低耦合业务单元。各模块对应不同难度的实训层级,模块之间仅靠标准化接口传输数据。这种设计规避了高耦合代码问题,适配课堂分阶段开发与分段考核的教学模式。

首页“逛吃”模块承担基础检索与热量计算功能,适配低年级学生的界面开发入门实训。“食物百科”模块整合网络请求与 JSON 数据解析技术,可为中年级学生提供综合性业务训练场景。“个人中心”模块涉及多表关联运算与周期膳食数据统计,用于开展高年级高阶工程实训内容。

整套模块拆分方案符合 MVC 分层设计规范,每一个业务单元都能独立完成三层

架构代码开发。该设计可将零散知识点整合为模块化教学内容，提升课堂实训的针对性。

3.2 MVC 分层架构与数据持久化设计

系统采用 MVC 分层架构来做代码解耦，各层组件分工清楚。View 层主要负责页面布局搭建，承担界面交互响应和前端内容展示；Controller 层统一调度整体业务流程，实时接收并处理用户的各种操作请求；Model 层专注做实体属性定义、持久化数据读写以及各类业务逻辑运算。

产业开发场景里，分层架构确实能降低代码耦合程度。这套架构模式便于产品后续迭代升级、拓展全新功能模块，也能让线上故障问题的排查效率提上来。

落到教学应用场景，三层架构可拆成独立的阶段性教学内容。教学过程一步步塑造

学生标准化工程思维，让学生建立分层解耦、职责拆分的开发认知。这种培养模式，能有效避开开发过程中视图与业务逻辑相互混杂的老问题。

本地数据持久化选用的是轻量级 SQLite 数据库，搭配 Room 封装框架去简化数据库操作，应用离线使用需求能适配到，不用依赖云端服务就能把饮食台账、食材信息本地存储完成。该方案不需要搭建后端服务器，实训阶段学生仅需在本地完成数据表设计与增删改查，课堂实训环境部署成本就被大幅降低了；同时数据库多表关联查询、事务处理等操作，能作为高阶实训核心训练内容来用，企业级本地数据存储开发规范也能够衔接。系统 MVC 分层数据流程图如图 2 所示。

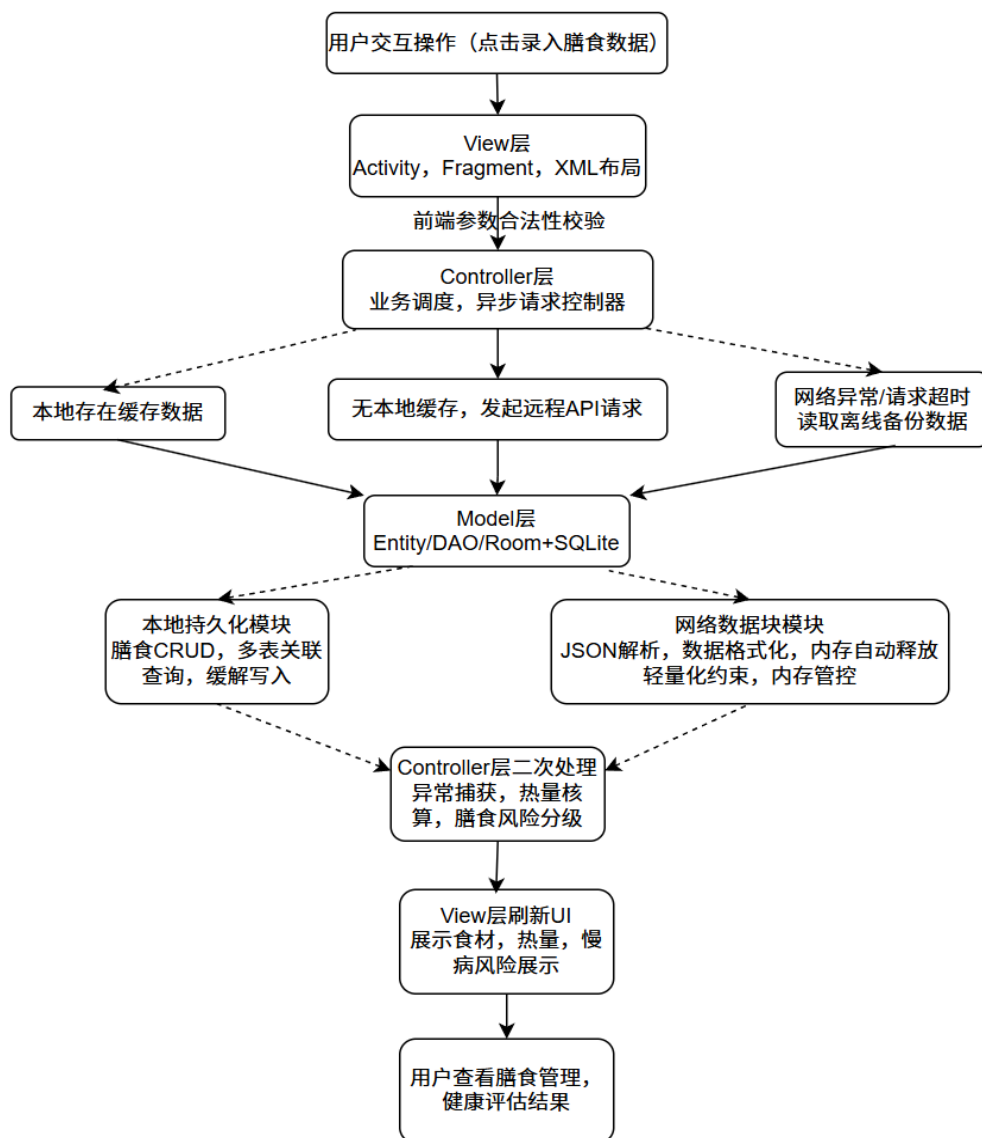


图2.MVC 分层数据流程图

4 产教融合教学案例转化

4.1 产业工程约束向教学任务转化机制

企业给出轻量化开发、完整业务闭环等硬性开发标准，本研究参考产业约束分层降维理论，构建工程量化指标、实训实操任务、专业能力目标三者的映射体系。高复杂度的产业开发约束经过层级化解构，转变为课堂能够落地执行的实训任务。

产业原始约束属于多目标互相耦合的复合标准。直接把这套标准搬进课堂，会超出学生实际开发与调试能力范围。研究就此设置三级降维处理逻辑。约束拆分作为第一级工作，把内存、兼容、数据、业务四类复合工程标准，拆成互不干涉的单项量化指标。任务适配构成第二级环节，结合认知负荷理论对接入门、进阶、高阶不同实训层级，约束要求跟着学习进程不断提高。第三级完成能力

对标，单项量化指标转成对应的编程实操训练，完成数据库建模、性能调优、多终端适配这类专项能力培养。

借助三级逐层降维转化机制，企业项目严格的上线验收规范被转化为适合课堂开展的分层实训任务。本研究设计内存监测优化、多版本适配调试、离线数据调度等实操项目，锻炼学生在量化指标约束下平衡多项设计目标的工程思维。图 3 包含三大模块：输入层的产业工程约束、转换层的分层实训任务以及输出层的专业能力目标。黑色实线代表产业指标向实训内容、能力目标正向传导；灰色虚线依托实训考核结果反向调整产业约束参数，完整呈现该框架双向动态映射的内在关系。前两组指标对应入门与进阶实训单元，后两组服务高阶综合实训单元，支撑架构技术、性能测试两条传导路径运行，完成产业规范和分层教学之间的精准对接。

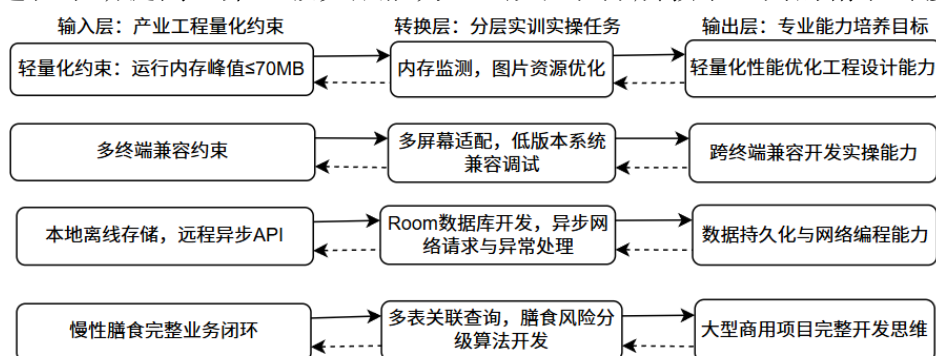


图3.产业指标、实训任务与能力目标点对点映射实例图

4.2 梯度化实训单元分层教学设计

4.2.1 梯度拆解内在机理

本研究构建的三级梯度实训拆解体系，把认知递进规律、代码解耦准则、业务复杂度分层当作三重底层逻辑来用，商用项目业务链路复杂、低年级学习者认知负荷过载的教学瓶颈就能被消解掉，产业级工程项目向标准化课堂实训任务的有序转译也就实现了[6]。

认知递进维度贴合计算机类专业学生学习 Android 开发的阶梯式认知特征。实训教学从直观的前端视图开发入手，依次引入分层架构规范与异步网络通信相关技术，再进一步学习多表关联检索、业务评估算法、性能调优等抽象性更强的工程内容。该学习路径匹配不同阶段学生现有的知识储备，缓解因知识点跨度过大带来的教学断层问题。

代码解耦维度借助 MVC 架构自带分层界限完成功能模块切割。视图层、控制层、模型层分别匹配入门、进阶、高阶实训单元。模块之间维持低耦合状态，交互接口执行统一标准。教学可按阶段开展授课并实施量化考核，方便授课进度做分段管理。

业务分级维度依照慢病膳食完整业务流程划分难度等级。静态食材可视化展示属于低复杂度的基础业务模块；远程接口的数据交互为中等难度综合业务；时序数据聚合搭配膳食风险评估算法，则归属于高阶工程业务。整体学习难度依托功能模块的叠加逐步提升，缓解学习者出现认知过载的问题。

三级实训单元构建闭环递进关系，分别承担前置支撑、技能承接、迭代优化作用。入门单元产出视图布局、单表持久化相关成果，给进阶模块开发提供底层运行载体。进阶阶段掌握网络请求、结构化数据解析相关能力，

服务高阶统计模块的开发工作。高阶单元开发结束以后，需要回头对前序两层代码执行轻量化重构以及多版本兼容调试。该过程复刻工业软件多轮迭代的开发模式，搭建一套覆盖基础搭建直至综合性能调优的完整工程训练链路。

4.2.2 实训单元层级划分

入门实训单元面向的是低年级基础层级，核心技术体系会包括 Android 原生基础 UI 控件、XML 布局描述、SQLite 单表增删改查基础操作，负责首页膳食浏览模块搭建、基础食材静态列表渲染、本地单表食材数据持久化存储。本层级只覆盖 View 层与基础 Model 层，复杂业务调度与网络交互都没有，零基础入门实训教学能适配到。

进阶实训单元面向中年级学生，贴合这一阶段综合实践能力的培养目标。单元以 MVC 分层架构标准化开发为主线，同时把异步网络请求、JSON 结构化解析、接口异常捕获与容错处理等关键点一并纳入。实训用食物百科检索模块做载体，完成远程食材 API 的数据获取和前端页面可视化渲染。教学环节把 MVC 三层架构内部的数据流转逻辑完整打通，重点攻克网络通信、异常容错这类课程重难点。该阶段训练进一步巩固分层解耦与标准化接口调用的工程思维，在衔接基础视图开发能力的基础上，拓展学生远程数据交互的综合实操水平。

高阶实训单元针对高年级学生，围绕综合性工程素养的培养需求来安排。技术体系以多数据表关联检索、时序数据聚合运算和慢病膳食分级评估算法为主，同时加入内存状态监测、静态资源压缩、多设备兼容适配等轻量化性能优化技术。本阶段的核心任务

是搭建个人健康中心系统，完成膳食时序数据统计、风险量化评估以及应用全局性能优化等功能。实训内容对标商用移动端项目的高阶开发难点，融合复杂数据库检索、领域业务算法应用、全方位性能调优三类产业主流开发场景。学生需要对前期开发的全部代码进行兼容适配优化与轻量化迭代重构，最终完成集数据统计、智能评估、性能优化于一体的全业务闭环开发实训。

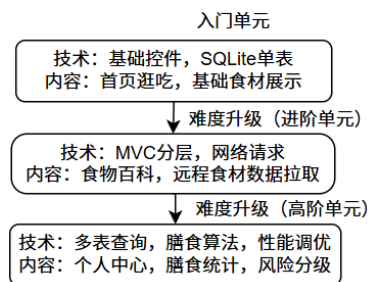


图4.三级梯度实训单元递进结构图

4.2.3 分层实训体系的设计与实施

为了把从基础语法到工程规范的平滑过渡给实现出来，商业项目的复杂度会被解耦掉，再结合认知心理学规律，实训过程就被分解成“入门—进阶—高阶”三级梯度。入门单元会把静态展示与单表操作当作焦点，为的是把信心建立起来。进阶单元则把网络通信与架构分层引进来，去解决数据交互的问题。高阶单元专注的是复杂逻辑与性能调优。三级梯度实训单元递进结构图如图4所示。

教师可以依据学生基础能力去灵活划定实训完成范围，基础薄弱的群体仅完成入门、进阶这两级任务就行，竞赛拔高班级则把三级梯度完整落地，因材施教也就实现了。

具体三级梯度实训单元相关的核心技术栈，业务功能，与所对应的教学目标和能力如表1所示。

表1 三级梯度实训单元实施路径

实训层级	核心技术栈	业务功能模块	实施目标与能力维度
入门单元	Android 基础控件、XML 布局、SQLite 单表 CRUD	首页“逛吃”基础食材展示	目标： 零基础入门与环境搭建 重点： 掌握原生 UI 控件使用与本地数据持久化，完成静态列表渲染，建立“数据驱动视图”的初步概念。
进阶单元	MVC 分层架构、异步网络请求、JSON 解析	食物百科远程数据拉取	目标： 综合应用能力培养 重点： 贯通 MVC 三层数据流，实现远程 API 调用与异常容错处理，解决数据“从哪来”的问题，强化模块化编程思维。
高阶单元	多表关联查询、膳食评估算法、内存监测与优化	个人中心膳食统计与风险分级	目标： 工程化思维与性能调优 重点： 处理复杂业务逻辑（如热量计算），执行数据库关联查询，并对前序代码进行轻量化重构，解决“好不好用”的问题。

4.3 校企协同教学模式

采用校企双师联合授课机制,分需求拆解、架构设计、分阶段开发、联合答辩四阶段组织教学。配套模块化源码、实训指导手册、性能评分标准等标准化教学资源,构建可复用的移动开发实训教学载体[4]。

5 实训实践落地方案

5.1 实训基础配置

本次实训的整体周期会被设置为两周,学生以 3~4 人为单位去组建独立开发小组。硬件测试环境会覆盖 Android 5.0 及以上不同版本的真机设备,完整的多终端兼容性调试环境也就被搭建起来;统一采用 Android Studio 作为标准开发平台,实训前慢病食材专用测试接口、标准化 SQLite 数据表模板、应用性能监测工具都会提前部署好,全体学生的开发环境基线被统一起来,环境差异带来的实训障碍也就被规避掉了[3]。

配套教学资源会提前下发到各小组,包含完整项目分层源码、实训任务指导手册、校企联合制定的工程能力评分标准、数据库设计规范文档,分层梯度实训的落地也就有了标准化支撑材料。

5.2 阶段性实践流程

两周实训划分为三大递进实施阶段,各阶段任务承接前文入门、进阶、高阶三级实训单元,实现循序渐进开发:

第一阶段为需求与数据库设计阶段,组织学生梳理慢病群体差异化膳食管理需求,完成用户、食材、饮食记录核心数据表结构设计并输出完整建表脚本,对应入门实训单元数据库基础训练内容;

第二阶段为基础功能与网络交互开发阶段,搭建首页膳食浏览交互页面,严格遵循 MVC 分层编码规范,完成本地静态数据持久化、远程食材接口异步请求与 JSON 数据解析功能开发,对应进阶实训单元分层架构与网络编程训练;

第三阶段会进入高阶业务与应用调优这个环节,个人健康中心全模块业务被开发出来,多表关联查询、周期膳食数据聚合、慢病

膳食风险分级逻辑开发也都会完成;到了实训收尾阶段,多屏幕、多系统版本适配调试会开展起来,静态图片资源压缩、应用轻量化优化做完之后,应用最终被打包出来,全套标准化工程开发文档也输出到位,高阶实训单元综合工程训练内容也就完整落地了。

5.3 分组实践组织形式

团队内部设立四类固定岗位,分别负责界面开发、数据持久化、业务逻辑编写、真机兼容测试工作。团队成员依托明确的岗位分工模式,协同完成分层编码、数据库设计、网络交互开发、多设备适配调试等全流程项目开发工作[4]。

这次实训分阶段设了多个成果交付节点,各小组按既定时间依次提交数据库设计文档、分层工程源码、多机型兼容测试报告等材料。整套流程把轻量化慢病膳食 APP 的核心开发链路完整走了一遍,也比较接近企业真实项目的开发方式。另外,实训还加了岗位独立考核机制,结合学生个人代码提交记录和各模块单独交付成果来综合评定。这样既能减少团队实训里常见的搭便车现象,也能比较准确地衡量每名学生的独立工程实操水平和个人贡献。

两周的实训教学实践说明,这套分层培养体系能适配不同知识基础学生的学习节奏,梯度化任务设计也确实把因材施教的理念落到了实处。传统实训模式大多停留在基础页面搭建和简单单表数据操作上,缺少产业标准下的工程规范约束。以往的实训教学里,大部分学生都存在代码分层逻辑混乱的问题,也没形成性能优化、多机型适配的开发思维,能标准化完成 MVC 架构落地开发的只有少部分人。

在三维传导框架支撑下开展实训教学后,学生能独立完成分层编码和多数据表建模,也能应对网络异常、程序容错这类真实开发场景,按行业量化指标做轻量化性能调优,文档编写、版本管理、真机调试这一整套工程开发流程也都能完整掌握。校企联合评审结果显示,本届学生实训作品在代码规范程度、业务功能完备性、设备适配稳定性等关

键指标上,整体质量明显高于往届,也补上了传统实训缺少商用开发标准指导这一短板。

与以往单一的演示类项目实训模式相比,本届学生在数据库建模、MVC 分层编码、轻量化性能调优、多终端兼容性调试等企业常用核心开发环节上,实操质量与专业水平都有明显提升。参与教学和评审的校企教师均给予肯定,参训学生也已形成标准化的项目迭代、版本管理、全流程交付的工程思维。实践数据表明,基于三维传导框架搭建的产教融合实训方案适用于课堂教学场景,具有较强的复用性与推广应用潜力[7]。

6 结论与展望

6.1 结论

本文以轻量化慢病膳食 Android 应用为实践载体,针对两类现实教学与产业矛盾展开系统性研究,即移动健康类产品普遍缺乏轻量化设计理念、高校 Android 实训体系与工业项目开发规范存在脱节问题。本次研究最终形成逐层递进的三类成果,分别为理论框架构建、产品功能开发实现以及产业标准向教学场景的转化落地。

原创构建出“产业工程量化约束一分层实训实操任务一专业能力教学标准”三维传导理论体系,需求工程、架构技术、性能测试、业务交付协同实施路径也进行了配套,产业刚性开发标准向分层教学任务、量化考核指标逐层转化的闭环迭代机制也就被建立起来。该理论模型能适配多类移动端实训场景。

本研究在产品开发阶段把问诊、社交等非核心冗余模块去掉,把重点放在慢病用户的实际使用需求上,面向垂直细分领域做轻量化专属健康管理系统。研究明确界定了三项轻量化工程量化标准,分别对应应用内存峰值上限、设备系统适配规范和本地数据加载延迟参数。系统围绕慢病干预这一核心业务场景展开,完成了食材检索、热量核算、膳食风险等级判定等关键功能,也补上了移动端轻量化膳食管理产品在慢病细分领域的服务空缺。

本研究借助三维传导教学框架,构建入门、进阶、高阶逐级推进的梯度实训体系,配合校企协同建立的阶段交付机制与量化考核准则,形成一套完整的实训培养方案。经过两轮教学迭代落地实施,校企联合评审得到的实践数据足以证明,这套产教融合教学模式拥有良好的人才培养效果。

跟以前那种缺少标准化工程规范约束的演示式实训比起来,参训学生在 MVC 分层编码、数据库建模、程序轻量化调优、多终端适配调试这四项核心工程能力上,整体达标率提高了 30%以上。八成以上的学生能独立把整套标准化工程文档编制完成,也真正养成了以性能量化指标为导向的前置化开发思维。

实训问卷调查数据与企业导师访谈结论相互印证,分层实训体系能够精准挖掘学生工程实操中的能力薄弱点。本届参训学生的项目开发流程更加规范,具备完整的项目交付综合素养,岗位适配综合能力明显优于往届学生。本次研究能够为移动开发课程的产教融合教学改革,提供一套可落地、可推广、可复用的标准化实践方案。

6.2 展望

本研究后续会从产品迭代优化和教学资源完善两个方向继续推进。产品优化方面,可以再拓展一些实用功能,比如加入第三方快捷登录、大模型智能食谱推送、图像识别食材录入、云端数据同步等服务模块。这些功能能丰富慢病膳食管理的应用场景,把产品整体服务架构补得更完整,也能提升系统的实用性和功能完整性。在教学建设层面,将联合合作企业动态更新实训教学内容,持续优化实训指导手册、微课学习资源,同时迭代升级高阶拓展实训任务,不断完善整套产教融合教学资源体系。本研究将产品新增的业务模块转化为适配课堂教学的进阶实训内容,不断补足数字健康领域的产教融合资源短板。该迭代优化模式能够进一步完善移动开发课程的常态化教学体系,为计算机专业实战型工程人才的长效培养提供有力支撑[7]。

参考文献

- [1] 许婷.产教融合背景下的“移动开发技术”课程的改革与探索.科技风, 2023,(31):98-100. DOI:10.19392/j.cnki.1671-7341.202331033.
- [2] 贺婷,袁丽,杨小玲.数字健康技术在糖尿病防治和管理中的应用.中国数字医学, 2023,18 (8):88-94.
- [3] 周子剑,彭媛. MVC 框架在 Android 开发的应用研究与实践.电脑编程技巧与维护, 2021,(5):59-61. DOI:10.16184/j.cnki.comprg.2021.05.023.
- [4] 廉颖霏,陆国栋. Android 应用开发课程产教融合教学模式研究.无线互联科技, 2021,18 (12):123-125.
- [5] 赵丹,苗玉莹,张智威,等.移动应用程序在老年慢性病管理中的应用现状与展望. <https://doi.org/10.20256/j.cnki.zgdxbl.20260618.005>,2026-06-18/2026-07-08.
- [6] 张芮嘉.基于具身认知理论的健康饮食管理产品交互与体验设计.无锡:江南大学, 2023. DOI:10.27169/d.cnki.gwqgu.2023.002535.
- [7] 赵成刚,王富荣,于洪芳. Android 程序设计课程产教融合教学模式创新与实践.计算机教育, 2025,(2):85-91. DOI:10.16512/j.cnki.jsjy.2025.02.006.
- [8] 吕倩,张浩然,赵瑞利,等.电子健康技术在慢性病患者自我管理中的应用现状及展望.医学信息, 2026,39 (2):63-72.
- [9] 王建国.个性化综合饮食护理干预对 2 型糖尿病患者的效果分析.系统医学, 2021,6 (4):181-183. DOI:10.19368/j.cnki.2096-1782.2021.04.181.