PROJECT CHARTER

Project Title: Radice ALM: Application Lifecycle Management (App Store Integrated ALM)	
Start Date: Jun 19, 2025	**End Date:** Jul 9, 2025
Project Manager: Sothea Seng	
Project Sponsor: Paragon International University	
Customer: Neil Ian Uy	
Users: Developers, Investors, Testers	

Stakeholders and Expectations:
Radice ALM App Store Developer: Expected to train and guide the Radice ALM Admin on the functionality and core features of the ALM and the new App Store module.

Radice ALM Team: Expected to maintain the app, accepting incoming project listing requests, sorting them by categories, and managing bug reports and feedback.

Purpose (Problem or opportunity addressed by the project):
Our system operates on a continuous improvement model, where the newly integrated App Store module serves as a crucial hub for fostering ongoing development and quality assurance. Developers can showcase their web apps, mobile apps, or API systems on this centralized platform. Here, End-Users (including other Developers) can explore, test, and, most importantly, provide vital feedback and bug reports. This direct input empowers developers to continuously refine and improve their applications, fostering an iterative development cycle.

Simultaneously, the Radice ALM Team efficiently manages all incoming developer projects. This ensures their legitimacy and quality, preventing the platform from being overwhelmed with irrelevant or low-effort submissions. This rigorous vetting process maintains a professional experience for all viewers, ensuring that the feedback and bug reports gathered are focused on genuinely valuable projects within our improvement cycle.

Goals and Objectives:
The goal and objective are:

- To design, develop, and deploy a public-facing 'App Store' module that integrates seamlessly with the existing Radice ALM.
- Provide a centralized system for testers to report bugs and submit feedback, enabling developers to receive and manage input related to their projects.
- Offer an intuitive platform where Developers can showcase their projects, and potential Investors or Stakeholders can easily discover their projects.

Schedule Information (Major milestones and deliverables):

Week 1: June 19 – June 25
Understand the Existing System

- Study Radicenter ALM's current architecture, codebase, and deployment model
- Identify extension points (APIs, routes, DB structure) for app store integration

Requirements and Planning

- Create the SDA specifically for the app store feature (not the whole system):
    - Component diagram of new modules
    - Data flow between existing system and app store (projects, users, feedback)
    - API interface specifications
    - DB schema extension (e.g., reviews table)
- Sketch wireframes and flow diagrams (e.g., project browsing, feedback UI)
- Define project scope, timeline, risks, and SPMP

Initial Setup

- Create feature branch or module in the existing codebase

Week 2: June 26 – July 2
Documentations

- Draft the SRS and SPMP for the new app store feature
- Restructure existing Radice ALM database and SDA, if needed , to integrate App Store data.

Implementation Phase 1 (Backend):

- Build backend for project publishing endpoints (with status: "live").
- Implement review & rating models and APIs.
- Implement Bug Report submission and viewing models and APIs.

Implementation Phase 1 (Frontend/Admin):

- Build frontend/admin interface to browse projects by tag, category, or recency (linked from navigation bar).
- Develop Admin project verification interface.
- Develop a unified bug reports interface with filters for developers.

Week 3: July 3 – July 9
Implementation Phase 2 (Feature Completion):

- Finish remaining backend features (review moderation, project versioning if applicable).
- Complete UI features:
    - Project details page with predefined text boxes for project information and image/video uploads.
    - Review submission UI.
    - Bug Report submission UI (including upload video/screenshot).
    - Filter/sort/search components for project Browse.
- Implement functionality for developers to write update notes.
- Implement download APK functionality with clear indication of random testing.

Testing & Polishing

- Run functional tests and gather peer feedback
- Fix bugs and finalize frontend/backend integration
- Write unit tests, integration tests (if part of the scope)

Documentation & Final Deliverables

- Finalize SRS, SPMP, SDA
- Prepare final presentation/demo materials

July 12th is the last day of Semester 6.

Financial Information (Cost estimate and budget information):

Category	**Amount**	**Percentage**
Personnel	$3,300	64.00%
Infrastructure & Tools	$200	4.00%
Testing & QA	$350	7.00%
Contingency (15%)	$750	15.00%
Documentation & Launch	$500	10.00%
TOTAL PROJECT COST	**$5,000**	**100%**

Project Priorities and degrees of freedom:

Project Priorities

Core Functionalities: Prioritize core functionalities that are crucial to the usage of the App Store module and its purpose within ALM.
Usability: Ensure a clean and usable user interface while also ensuring an intuitive user experience without sacrificing the functionalities of ALM.
Flexibility: Involves best practices and design ideas that incorporate the idea of future developments.
Security: Unwanted access is crucial to maintain the quality and usability of the ALM.

Degree of Freedom:

Role Responsibility: Member work distributions can be shifted to ensure efficiency.
Scope: The Project Manager can adjust the scope, adjusting it to be reasonable in the given time frame.
Approach: The methodology (Scrum with weekly sprints) is adaptable based on project needs.

Approach:

Methodology: Incremental/Iterative

Phase 1: Foundation & Planning

Duration: June 19 – June 25
Milestone: System Understanding + Planning Documents + Initial Designs

Objectives:

- Analyze Radicenter ALM architecture, APIs, and extension points.
- Identify integration strategy (DB, routes, services).
- Create:
    - **SDA** (Component diagram, data flow, interfaces, schema changes)
    - **SRS & SPMP** (Scope, goals, risk, timeline)
- Design initial wireframes (Browse, Feedback, Admin UI)
- Prepare environment (branching, module structure)

Deliverables:

- Drafted SRS, SPMP, SDA
- Wireframes and flow diagrams
- Codebase/module initialized

Phase 2: Backend Development & UI Scaffolding

Duration: June 26 – July 2
Milestone: Core Backend Ready + Initial UI Functional

Objectives:

- Implement:
    - Project Publishing backend
    - Bug Reports & Feedback model and APIs
- Setup Admin Dashboard (verify projects, view bug reports)
- Setup basic Browse UI (by tag, category, recency)

Deliverables:

- Working API endpoints (Postman, Insomnia tested)
- Admin, End User, Developer UI (basic functional prototype)
- Updated SDA and DB schema (if changes occurred)

Phase 3: Completion, Testing, & Documentation

Duration: July 3 – July 9
Milestone: Feature-Complete App Store + Ready for Final Review

Objectives:

- Complete UI:
    - Project details page with file upload (media)
    - Review & Bug Report forms

    - Search/Filter components
- Perform:
    - Peer Testing
    - Bug fixing and UI polishing
- Finalize all documentation (SRS, SPMP, SDA)
- Begin demo prep

Deliverables:

- Fully functional App Store module
- Finalized documentation (SRS, SPMP, SDA)
- Internal demo-ready build

Final Handoff & Presentation

Duration: July 10 – July 12
Milestone: Presentation-Ready Product

Objectives:

- Final polish and UI refinements
- Rehearse demo with the team
- Document known issues, dev notes, handover guides
- Present to stakeholders

Deliverables:

- Stable build for live demonstration
- Knowledge transfer documents
- Final presentation deck

Constraints:

Time Constraints: The project faces a **hard deadline of July 9, 2025**, for all core App Store features, testing, and final documentation (SRS, SPMP, SDA). The absolute final handover is **July 12, 2025**. Each of the three sprints is strictly **one week** long.

Scope Constraints: The scope is **fixed** for this App Store integration, covering project publishing, review/rating systems, Browse/discovery, and administrative verification/moderation. This must seamlessly leverage Radicenter ALM's existing architecture and codebase. **Deviations from this fixed scope would take away the value and purpose of ALM** by causing delays or introducing unnecessary complexity.

Resource Constraints: The project relies solely on the **current team members** and **existing development and staging environments**. There's no provision for additional personnel or new infrastructure acquisition.

Quality Constraints: Focus is on **functional correctness** and **stable integration** with existing ALM. Basic usability is required, and mandatory documentation (SRS, SPMP, SDA) must be finalized by July 9th. Extensive automated testing is secondary due to the tight timeline. **For API projects, developers must host their API documentation externally, as our system will not host these documents.**

Assumptions:

Assumptions Regarding Users of Our App Store

- **User Access & Authentication:** We assume users (Paragon students and viewers), developers (Paragon students and Independence Developers), and ALM admins will reliably access Radicenter ALM (and thus our App Store) by **logging in via their Google accounts** from standard internet-enabled devices.
- **Defined User Behavior & Engagement:** Our App Store's design and functionality rely on users Browse and submitting reviews; developers actively listing and managing their projects; and admins performing verification and moderation. We assume all these user types are motivated to engage and will adhere to established policies.

Assumptions Regarding the Radicenter ALM System

- **Core Platform Stability & Capacity:** We assume the underlying Radicenter ALM platform provides a stable, 24/7 available environment with sufficient performance capacity to seamlessly host and operate our new App Store feature.
- **Seamless Integration Points:** Our App Store development relies on the existence of well-defined data structures within the existing Radicenter ALM for smooth integration.
- **External Content & Moderation Management:** We assume that accurate project listings will be maintained by developers, and that ALM administrators will consistently perform timely project verification and review moderation outside of our core App Store development process.

Success Criteria:

Successfully implement all defined core functionalities for Users, Developers, and Admins in the Radice ALM App Store, ensuring they are fully operational before the July 9th deadline.

Scope:

User Management

- **Registration:**
  - End-User Registration
  - Developer Registration
- **Authentication & Authorization:**
  - User Login (End-User, Developer, Admin)
  - User Logout (End-User, Developer, Admin)
- **Profile Management:**
  - Update User Profile Information (End-User)

App Management

- **App Submission & Upload:**
  - Upload App Files (Developer)
  - Upload App URL Link To The Site (Developer)
  - Enter App Information Such As Description, Version, Title, (Developer)
  - Upload App Media Such As Screenshots, Preview, Banner (Developer)
- **App Viewing & Discovery:**
  - View All Apps (End-User)
  - Search Apps (End-User)
  - Filter Apps by Category (End-User)
  - View Apps Update History (End-User)
- **App Administration & Moderation:**
  - Verify Apps Legitimacy and Quality (Admin)
  - Approve Apps (Admin)
  - Reject Apps with Reasonings (Admin)
  - Notify Developers of App Status (Rejection, Approval, Developer)

App Interaction & Feedback

- **App Rating:**
  - Rate The Apps By Giving It Star Rating (For All Users)
- **Feedback Management:**
  - Submit Public Feedback on Apps (End-User, Developer)
  - View App-Specific Feedback (Developer)
  - View All Feedback Reports (Admin)
  - Prevent Feedback Submission for Shelved Apps (End-User, Developer)
- **Bug Reporting:**
  - Can Submit Bug Reports with Media Upload Such As Video and Screenshot (End-User, Developer)
  - View App-Specific Bug Reports (Developer)
  - View All Bug Reports (Admin)
  - Prevent Bug Report Submission for Shelved Apps (End-User, Developer)

Project Testing

- **Random App Testing:**
  - Test Listed Apps (End-User)
  - Download APK for Testing (End-User)

System Administration

- **Data Management:**
  - Modify All System Data (Admin)

- View All System Data (Admin)

User Interface & Experience

- **Responsive Frontend Display:**
  - Display Responsive Frontend (End-User)

Risks and obstacles to success:

Technical Integration Challenges: Unexpected difficulties in connecting with Radicenter ALM's existing systems (APIs, codebase) could lead to **significant delays**, potential **bugs**, or **security vulnerabilities** if implementation is rushed.

Post-Launch Content Management: The App Store's long-term value depends on **consistent project submissions** from developers and **timely verification** by admins. A lack of engagement here could leave the platform underutilized.

Signatures

_____

Project Manager

_____

Project Sponsor

_____

Customer

_____

Technical Lead