



Miami University
College of Engineering & Computing

Course: CSE 201 - Software Engineering

Project: AppBase (Meta-App)

Team: K-A-I-G-A-N

Instructor: Dr. Alomari

Miami University - College of Engineering & Computing

CSE 201: Software Engineering Project

Project Strategy Layout: AppBase (Meta-App)

Team Information

Team Name: K-A-I-G-A-N

Project Title: AppBase

Instructor: Dr. Alomari

Customer Representative (TA): Amrit Aryal

Weekly Meeting: Mondays, 3 PM, King Library (Reserved Study Room)

1. Project Overview

Goal:

AppBase is a Java-based desktop application designed as an App Catalog System, similar to a miniature app store. It allows users to browse, sort, and discuss applications while enabling developers to submit their apps for listing. Administrators and moderators manage user submissions and forum discussions.

Concept:

AppBase functions as a Meta-App, an application cataloging system for other applications. It bridges the gap between existing platform-specific stores (like Google Play or Apple Store) by offering a unified interface for discovering, comparing, and discussing apps.

Core Objectives:

- Build a responsive, intuitive Java-based interface for app exploration.
- Implement secure login and role-based permissions (admin, moderator, user).
- Support app submissions, approvals, and discussions.
- Maintain persistent storage through a SQL database.

2. Team Roles and Responsibilities

Quinton Davis-Smith – Project Manager / Developer 2: Coordinates tasks, manages Trello & documentation, assists with frontend UI.

Kiera Kuschnerus – Data Layer / SQL / Tester: Designs SQL schema, performs testing and

Michael Anderson – Developer 1: Builds Java Swing UI, handles forms and layout.

Joshua Chisanga – Technical Manager: Oversees backend logic, manages GitHub repo, and handles code reviews.

3. System Architecture

Architecture Pattern: Three-Tier (MVC - Model, View, Controller)

Frontend (View): Java Swing / JavaFX – GUI for user interaction.

Backend (Controller): Java via Eclipse – Core application logic and validation.

Database (Model): MySQL / SQLite – Stores all app and user data.

4. Functional Requirements

- Search & Browse
- Sorting
- App Submission
- Admin Review
- Discussion Forums
- User Authentication
- Data Persistence

5. Tools and Environment

Java (Eclipse IDE) – Development

MySQL / SQLite – Database

GitHub – Version control

Trello – Task tracking

Google Drive – Shared files

6. Workflow and Collaboration

- GitHub branching model
- Agile workflow using Trello
- Weekly stand-up meetings

Branch Structure:

main, feature/frontend-Ui, feature/backend-logic, feature/database-integration

7. Development Timeline

Week 1: Setup repository and roles

Week 2: Requirements gathering

Week 3: System design

Week 4: Implementation I

Week 5: Implementation II

Week 6: Testing & Review

Week 7: Integration & Presentation

8. Design References

- UML: Class, sequence, use-case diagrams
- Design Patterns: MVC, Singleton, Observer

9. Testing

- Unit testing (login, form validation, SQL queries)
- Integration testing (backend-frontend-database)
- Acceptance testing with TA feedback

10. Risk Management

- Merge conflicts → feature branches
- Database issues → test SQL early
- Time conflicts → consistent schedule
- Scope creep → focus on MVP first

11. Future Enhancements

- Rating and recommendation system
- Cloud-based database hosting
- Web-based version

12. Summary

AppBase will be developed in Java using Eclipse IDE and SQL for data persistence, following MVC architecture and agile collaboration through GitHub and Trello.

Submitted by:

K-A-I-G-A-N Team

Joshua Chisanga (Technical Manager)

Quinton Davis-Smith (Project Manager)

Kiera Kuschnerus (SQL / Tester)

Michael Anderson (Frontend Developer)

Appendices

GitHub Repository: <https://github.com/K-A-I-G-A-N/K-A-I-G-A-N-CSE201-PROJECT>

Trello Board: <https://trello.com/b/hFFhFrnF/cse201>