Company

<Kontroller>
Software Development Plan
Version <1.0>

Revision History

Date	Version	Description	Author
02/06/2025	1	Drafting the first version of the report	Nguyễn Gia Nghi
03/06/2025	2	Reviewing and adjusting	The whole team

Table of Contents

Software Development Plan

1. Introduction

Kontroller is a cross-platform web application that allows users to log, rate, and review video games while sharing their gaming experiences with friends. The goal is to provide a centralized platform similar to Letterboxd or RateYourMusic, but for games, leveraging real-time data, social features, and user-generated content. The initial product is a web-based MVP that supports both macOS and Windows platforms.

2. Project Overview

2.1 Project Purpose, Scope, and Objectives

The purpose of this project is to build a functional MVP of Kontroller that supports key user features such as game logging, rating, personal reviews, list curation, and social interaction.

Objectives:

- Deliver a usable MVP within the construction phase
- Integrate iGDB API successfully for game data retrieval
- Design an intuitive UI suitable for desktop and mobile browsers
- Implement user authentication and basic social features

2.2 Assumptions and Constraints

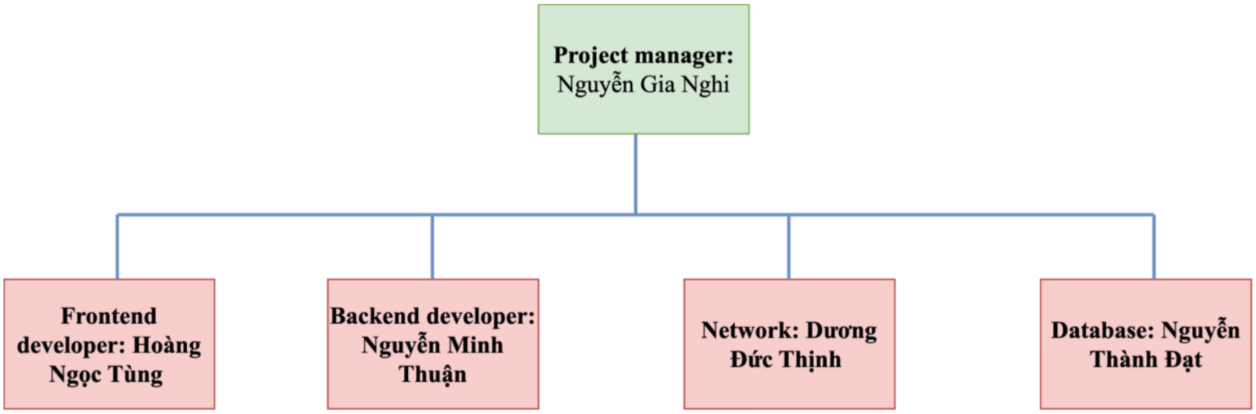
- Project has a fixed schedule of 10 weeks
- This is a zero-budget project
- Project has 5 people, there will be no more people added during the project

2.3 Project Deliverables

- A functional website deployed online
- Full codebase hosted on GitHub
- Project documentation and user manual
- A presentation
- Final report with testing results and user feedback

3. Project Organization

3.1 Organizational Structure



Project manager: Nguyễn Gia Nghi
Frontend developer: Hoàng Ngọc Tùng
Backend developer: Nguyễn Minh Thuận
Network: Dương Đức Thịnh
Database: Nguyễn Thành Đạt

3.2 Roles and Responsibilities

- **Frontend Developers:** Build responsive user interfaces and integrate IGDB game data for display and interaction.
- **Backend Developers:** Develop features for user authentication, data storage, and retrieval using a robust backend framework.
- **Network Engineer:** Handle the networking system, including communication between users and between users and the app's servers.
- **Database Engineer:** Design, implement, and manage the app’s database system. Ensure data integrity, optimize queries, and handle relationships between game data, user profiles, logs, reviews, and other features.
- **Project Manager:** Oversee sprint planning and task distribution, ensuring alignment across team members and managing communication between development roles. Write weekly reports on the progress achieved and make sure the aims are met every week.

Role

Project manager: Nguyễn Gia Nghi

Frontend developer: Hoàng Ngọc Tùng

Backend developer: Nguyễn Minh Thuận

Network engineer: Dương Đức Thịnh

Database engineer: Nguyễn Thành Đạt

4. Management Process

4.1 Project Estimates

- Estimated Schedule**: 10 weeks total, broken down into 3 main phases and 5 development sprints.**
- Estimated Cost**: No direct monetary cost. However, workload is distributed evenly among members based on estimated 6–8 hours/week per person.**
- Basis for Estimates**: Estimates are based on the complexity of features, team experience, and available time per week.**
- Re-estimation Points**:**
  - After Elaboration Phase (Week 4), if API complexity or integration takes longer than expected.
  - After Sprint 3 (Week 8), to assess progress and refine final testing/polishing workload.

4.2 Project Plan

4.2.1 Phase and Iteration Plan

Inception Phase (Week 1 – 2)

- Objectives: Team formation, project planning, tool selection, requirement gathering.
- Milestone: Initial project proposal submitted and reviewed.

Elaboration Phase (Week 3 – 4)

- Objectives: Continue on planning, wireframing, architecture planning, database schema design, and IGDB API testing.
- Milestone: Completion of detailed design documents and mockups.

Construction Phase (Week 5 – 10)

- Sprint 1 (Week 5–6): Implement game logging system, user authentication module.
- Sprint 2 (Week 6–7): Add rating and review system; integrate search functionality.
- Sprint 3 (Week 7–8): Implement user profiles and list creation.
- Sprint 4 (Week 8–9): Build activity feed and game recap stats.
- Sprint 5 (Week 9–10): Final testing, UI/UX polishing, and deployment preparation.
- Milestones:
  - After Sprint 2: Demo of core features
  - After Sprint 4: Demo of full feature set
  - Sprint 5: Final build and user acceptance test

Work Breakdown Structure (WBS), Gantt Chart, and Milestone timeline to be attached/exported from MS Project separately.

4.2.2 Releases

Release Name	Timeframe	Type	Description
Build 1	After Sprint 2	Alpha	Core features: Game logging, browsing, basic auth
Build 2	After Sprint 4	Beta	Full features: social elements, profiles, search
Final Build	Sprint 5	Release	Final version with bug fixes

4.2.3 Project Schedule

Sprint	Date Range	Phase / Activity	Milestones / Deliverables
Sprint 1	May 23 – May 29	Team Formation & Project Planning	Team established, roles assigned, timeline & tools selected
Sprint 2	May 30 – June 5	Requirement Analysis & Vision Report	Vision document completed and reviewed
Sprint 3	June 6 – June 12	UI/UX Design & Database Schema	Wireframes created; ERD and database tool confirmed
Sprint 4	June 13 – June 19	Project Setup & API Planning	GitHub repo setup, tech stack finalized, API structure planned
Sprint 5	June 20 – June 26	Implement Game Logging & Profile System (Part 1)	Game logging UI/backend, user profile creation implemented
Sprint 6	June 27 – July 3	Implement Review & List Features (Part 2)	Review system, public/private lists functional
Sprint 7	July 4 – July 10	Activity Feed & Stats	Activity feed visible, basic stats charting added
Sprint 8	July 11 – July 17	Final Features, Search, and Filters	Advanced search, filters by genre/platform/year implemented
Sprint 9	July 18 – July 24	Bug Fixing, Testing, and UI Polish	System tested, UX issues addressed, UI finalized

4.3 Project Monitoring and Control

4.3.1 Reporting

Project progress will be monitored through:

- **Weekly Team Meetings**: Discuss progress, blockers, and upcoming goals.
- **Weekly Status Reports**: Each member completes their assigned tasks, the project manager will go through each of the tasks, assess, adjust and write reports.
- **Informal Chats**: Daily coordination through Discord group channel or instant messaging.

4.3.2 Risk Management

Risk ID	Risk Description	Probability	Impact	Risk Exposure	Priority	Mitigation Strategy
R1	IGDB API limitation or outage	Medium	High	Medium-High	High	Implement API caching, have sample backup data
R2	Schedule overrun due to underestimated tasks	High	Medium	High	High	Weekly review of task load, redistribute work
R3	Git merge conflicts or data loss	Medium	Medium	Medium	Medium	Use Git branches, commit regularly, code reviews
R4	Lack of team coordination or absence	Low	High	Medium	Medium	Weekly sync meetings, shared task board
R5	User authentication vulnerabilities	Low	High	Medium	Medium	Use secure frameworks, test thoroughly

4.3.3 Configuration Management

- **Source Code and Version Control:**
  - **Git** via **GitHub** for managing and tracking source code and commits.
  - Feature branches and pull requests for new features to prevent merge conflicts.
- **Document Sharing and Collaboration:**
  - **Google Drive** for storing documentation, planning materials, and presentation files.
  - **Trello** for weekly scheduling.
  - **Discord** for communication.