
<The Next Stage Ltd.>

<Kontroller>
Software Development Plan
Version **<1.0>**

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

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
15/06/2025	2.1	Update project estimation, project plan, project monitoring and control	Nguyễn Gia Nghi

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

Table of Contents

- 1. Introduction 4**
- 2. Project Overview 4**
 - 2.1 Project Purpose, Scope, and Objectives 4
 - 2.2 Assumptions and Constraints 4
 - 2.3 Project Deliverables 4
- 3. Project Organization 5**
 - 3.1 Organizational Structure 5
 - 3.2 Roles and Responsibilities 5
- 4. Management Process 6**
 - 4.1 Project Estimates 6
 - 4.2 Project Plan 6
 - 4.2.1 Phase and Iteration Plan 6
 - 4.2.2 Releases 7
 - 4.2.3 Project Schedule 7
 - 4.3 Project Monitoring and Control 7
 - 4.3.1 Reporting 7
 - 4.3.2 Risk Management 8
 - 4.3.3 Configuration Management 8

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

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

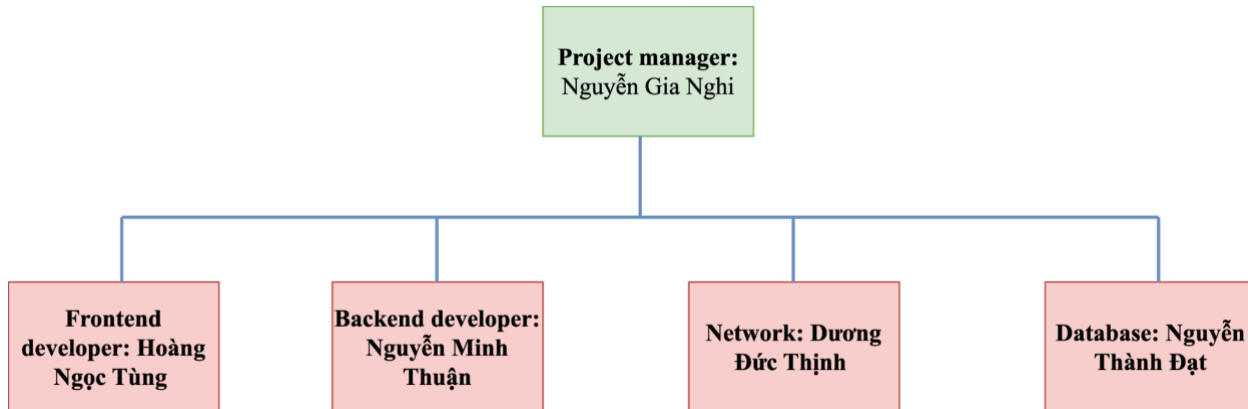
2.3 Project Deliverables

- 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

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

3. Project Organization

3.1 Organizational Structure



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

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

4. Management Process

4.1 Project Estimates

- Estimated Schedule: 10 weeks total, broken down into 3 main phases and 5 development sprints. [The project would likely be completed by July 24, 2025.](#)
- 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: **Time and effort estimates are based on:**
 - Complexity of each feature and its integration with the IGDB API.
 - Time available per week during the academic semester.
 - Technical risks such as external API behavior, deployment issues, or integration blockers.
- 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

Phase	Start Date	End Date	Objectives
Inception Phase	May 23, 2025	June 5, 2025	Establish team roles and responsibilities, select development tools, gather initial requirements, and draft the vision document.
Elaboration Phase	June 6, 2025	June 19, 2025	Finalize detailed UI/UX wireframes, define system architecture, design database schema, and test IGDB API connectivity.
Construction Phase	June 20, 2025	July 24, 2025	Implement all planned features, conduct testing, integrate UI components, and prepare the application for final release and presentation.

Milestone	Phase	Achievement Criteria
Team Setup & Planning	Inception	All members assigned roles; development tools selected; project scope and timeline confirmed.
Design Completion	Elaboration	Wireframes, ERD, and architectural diagrams finalized and approved; API endpoints tested.
Core System Demo (Beta)	Construction (Mid)	Game logging, authentication, review, and list features working; ready for internal testing/demo.
Full Feature Demo	Construction (Late)	All social and statistical features implemented; system stable and testable.
Final Build & Presentation	Construction (End)	Polished UI, fully tested system, deployment-ready version complete with bug fixes and documentation.

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

4.2.2 Releases

Release Name	Timeframe	Type	Content
Build 1	After Sprint 2	Internal	Early concept walkthrough: project plan and mockups
Build 2	After Sprint 4	Alpha	Basic UI layout, game logging input UI, and database test-ready
Build 3	After Sprint 6	Beta	Game logging, user auth, review system, and list creation ready for demo
Build 4	After Sprint 8	Pre-final	Social features and search filters working, design complete
Final Build	After Sprint 9	Release	Fully tested version, ready for end-user interaction

4.2.3 Project Schedule

Sprint	Date Range	Phase	Iteration	Key Deliverables
Sprint 1	May 23 – May 29	Inception	Sprint 1	Team established, roles assigned, timeline & tools selected
Sprint 2	May 30 – June 5	Inception	Sprint 2	Vision document completed and reviewed
Sprint 3	June 6 – June 12	Elaboration	Sprint 3	Wireframes created; ERD and database tool confirmed, schema , tech decisions
Sprint 4	June 13 – June 19	Elaboration	Sprint 4	GitHub repo setup, tech stack finalized, API structure planned
Sprint 5	June 20 – June 26	Construction	Sprint 5	Game logging UI/backend, user profile creation implemented (Alpha)
Sprint 6	June 27 – July 3	Construction	Sprint 6	Review system, public/private lists functional (Beta)
Sprint 7	July 4 – July 10	Construction	Sprint 7	Activity feed visible, basic stats charting added
Sprint 8	July 11 – July 17	Construction	Sprint 8	Advanced search, filters by genre/platform/year implemented (Pre-final build)
Sprint 9	July 18 – July 24	Construction	Sprint 9	System tested, UX issues addressed, UI finalized (Final build)

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 [Slack](#) group channel or instant messaging.

	Version: <1.0>
Software Development Plan	Date: 02/06/2025
Kontroller	

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	Security gaps in login system	Medium	High	Medium	Medium	Use OAuth / hashed passwords, conduct penetration testing
R5	Lack of team coordination or absence	Low	High	Medium	Medium	Weekly sync meetings, shared task board

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.
 - **Slack** for communication.
 - **Figma** for modeling.