Project Plan

Superior Planning System

IT3C

Ayomide Olagoke - Team Leader

Timothy Adewale - Scrum Master

David Hlavacek – Code Lead

Stefan Tasca- Secretary

Ian Donker

Ioan Mich

A blue and black logo

Description automatically generated

## Table of contents

[Table of contents 2](#_Toc257702073)

[Version Control 2](#_Toc171274409)

[Foreword: 2](#_Toc1871645293)

[Chapter 1: Introduction & Background 2](#_Toc403984427)

[Chapter 2: Project Results 3](#_Toc568101479)

[Chapter 3: Project Activities 5](#_Toc2074456882)

[Sprint Schedule Overview 6](#_Toc814567749)

[Sprint 1: Initial Setup and Basic Framework Development 6](#_Toc1115102482)

[Sprint 2: Core Feature Development 7](#_Toc1857571139)

[Sprint 3: Enhanced Functionality and Validation 8](#_Toc646099386)

[Sprint 4: Final Testing, Deployment, and Handover 8](#_Toc150772266)

[Chapter 4: Intermediate Results 9](#_Toc1652043481)

[Chapter 5: Project Boundaries 11](#_Toc244579670)

[Chapter 6: Quality Control 14](#_Toc2092067508)

[Chapter 7: Project Organization 14](#_Toc522554252)

[Chapter 8: Planning 18](#_Toc1165749799)

[Chapter 9: Cost and Benefits 19](#_Toc1501813884)

[Costs 20](#_Toc1166557654)

[Benefits 20](#_Toc1460816211)

[Chapter 10: Risk Analysis 21](#_Toc923093004)

[Chapter 10: Appendix 24](#_Toc835314796)

[10.1 Value Chain Analysis 24](#_Toc1341354832)

[10.2 GAP Analysis 25](#_Toc45535018)

## Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Content | Editor(s) |
| 0.1 |  |  |  |
| 0.2 | 10/05/25 | Added Chapter 1 | David Hlavacek |

## Chapter 1: Introduction

NHL Stenden University of Applied Sciences is a Dutch institution offering practice-oriented higher education. As part of the Secure Programming minor (Academic Year 2024-2025), student teams are tasked with building real-world applications that integrate security best practices throughout the software development life cycle.

Today’s competitive gamers—and the community managers who support them—must track performance across multiple titles and genres using spreadsheets, disparate websites, or ad-hoc tools. This fragmented approach is time-consuming, prone to data inconsistency, and raises privacy concerns when usernames or third-party credentials are exposed.

Over the coming eight-week period, the team will work under the guidance of module coordinator Rob Loves and lecturer Remco Hassing. Rob and Remco act as both internal stakeholders and product owners; external stakeholders are the players, e-sports teams, and community organizers who will rely on our app to view individual game metrics, genre-level summaries (FPS, MOBA, RPG, etc.), and a consolidated “global” performance dashboard (Profile).

#### Project Details:

* **Project Name:** GamerCV
* **Project Supervisors:** Rob Loves and Remco Hassing

The ultimate goal of this project is to replace manual stat-tracking workflows with a secure, maintainable Android application—implemented in Kotlin with Jetpack Compose—that fetches data from public game-API providers, encrypts sensitive information at rest and in transit, and offers an intuitive interface for gamers to monitor and improve their performance. The detailed requirements and design decisions follow in the chapters below.

## Chapter 2: Project Results

The problem

The team has noticed a growing trend among players who actively track their in-game performance across multiple competitive titles. These players often juggle stats from different games, switching between third-party stat websites, forums, and spreadsheets to monitor progress, compare metrics, and decide what to focus on next. The experience is fragmented, inconsistent, and lacks meaningful cross-game insights.

What makes the problem even worse is that there’s no single tool that pulls all this data together and makes sense of it, like by game genre, play style, or personal goals. Players often don’t notice when their performance starts to drop or when they've been playing for way too long. Plus, there's no personalized, mobile-friendly experience that fits how they play or what games they focus on.

### Solution

To meet this growing need, the team will design and develop a cross-game stat tracking mobile application, built specifically for Android using Kotlin, by June 20th. The goal is to give players a centralized and responsive interface to monitor performance, manage their game profiles, and receive alerts based on custom stat thresholds.

**The app allows users to:**

* Browse and search a curated list of supported games, filterable by genre.
* Add and manage usernames per game, with real-time validation.
* View detailed stats per game, genre-level performance, and global aggregates.
* Receive push notifications when performance dips below or rises above personalized thresholds.
* Customize their profile with display names, avatars, and genre preferences.

**Behind the scenes, the backend:**

* Is built using scalable technologies such as Spring Boot, FastAPI, or Express.
* Integrates securely with game-specific APIs to fetch user stats.
* Uses PostgreSQL with encryption and Redis for fast stat caching.
* Serves a clean RESTful API structure for user management, game data, and notifications.

**Boundaries & Ethical Scope**

* The application is strictly read-only with respect to game data; it cannot alter any in-game progression or real-time match data.
* It collects only essential user data (usernames and preferences), ensuring high privacy standards with AES-256 encryption and TLS 1.2+ transport security.
* It does not handle financial transactions or health-related data.

Value Proposition  
By offering this centralized and customizable experience, the app not only empowers players with data-driven insights into their performance but also supports healthier play habits through proactive alerts. Whether a casual player trying to improve or a competitive gamer optimizing strategy, the app serves as a useful tool.

## Chapter 3: Project Activities

This chapter provides a comprehensive overview of the project activities, organized into weekly sprints that guide the team's efforts throughout the duration of the project.

#### **Sprint Schedule Overview**

1. **Sprint 1:** November 18, 2024 - December 1, 2024
2. **Sprint 2:** December 2, 2024 - December 15, 2024
3. **Sprint 3:** December 16, 2024 - January 10, 2025
4. **Sprint 4:** January 10, 2025 - January 24, 2025

## Sprint 1: Initial Setup and Basic Framework Development

**Activities:**

1. **Requirement Gathering:**
   1. First Client meeting to get information from the client.
   2. Communicate with the client to confirm user stories, system requirements, acceptance criteria and the project's definition of done.
2. **Backlog Creation:**
   1. Develop a prioritized product backlog.
   2. Break down tasks into manageable sprint activities.
3. **Set Up Infrastructure:**
   1. Create and configure the GitLab repository.
   2. Set up project management tools and communication channels (Teams).
4. **System Architecture:**
   1. Research on Frameworks and Solutions
   2. Define the initial architecture for database design (courses, cohorts, periods).
5. **Mock up Development:**
   1. Begin Figma designs for student planning and teacher dashboards.
6. **Document Analysis**
7. Stakeholder Analysis: Identify key stakeholders (Client, Student, NHL Stenden).
8. Value Chain Analysis: Determine how the project add value to the client company.
9. Gap Analysis: Conduct a gap analysis.

**Sprint Demo (November 29, 2024):**

* Present initial Figma design to the client.
* Discuss the product backlog and sprint roadmap.
* Collect feedback for refinement.

## Sprint 2: Core Feature Development

**Activities:**

1. **Drag-and-Drop Course Planning:**
   1. Implement the drag-and-drop functionality for students to plan courses.
   2. Design and implement the drag-and-drop interface for students.
   3. Ensure courses are visually represented with essential details (e.g., course name, credits).
   4. Enable students to move courses between periods and dynamically update their plans.
2. **Backend Integration:**
   1. Create database functions for managing cohorts, courses, and periods.
3. **Warning System Implementation:**
   1. Develop basic warnings for overloading periods or academic years.
   2. Display visual indicators (e.g., warning icons or messages) when a period or year exceeds workload thresholds.
   3. Provide additional information explaining why a warning was triggered.
4. **Student Dashboard Development:**
5. Build the initial student dashboard, integrating the drag-and-drop feature and warning notifications.
6. Ensure it is responsive and adheres to NHL Stenden's branding guidelines.

**Sprint Demo (December 13, 2024):**

* Demonstrate drag-and-drop functionality and database integration.
* Present the basic warning system for student planning.
* Gather feedback for the next sprint.

## Sprint 3: Enhanced Functionality and Validation

**Activities:**

1. **Teacher Dashboard Development:**
   1. Implement aggregated warning views for teachers.
2. **Notifications and Prompts:**
   1. Finalize notification prompts for students (e.g., contact study career coach).
3. **Test Plan and Testing:**
   1. Conduct unit testing for drag-and-drop, warning systems, and notifications.
   2. Create a Test plan
4. **Branding and UI Refinement:**
   1. Align the user interface with NHL Stenden’s style guide.
   2. Include sorting and filtering options for easier data navigation.
   3. Ensure notifications are visually distinct and accessible from the dashboard.
5. **Initial Version of Design Document**

**Sprint Demo (January 10, 2025):**

* Present a functional system with teacher dashboard and notification features.
* Showcase the updated user interface with NHL Stenden branding.
* Collect feedback for the final sprint.

## Sprint 4: Final Testing, Deployment, and Handover

**Activities:**

1. **Final System Testing:**
   1. Perform system-wide testing with real-world scenarios and user data.
2. **Refinements:**
   1. Address feedback from Sprint 3 demo.
3. **Documentation:**
   1. Prepare user manuals for Super Admins, teachers, and students.
4. **Deployment:**
   1. Deploy the proof of concept for client review in a live environment.
5. **Final Presentation:**
   1. Prepare and rehearse the final presentation for project handover.
   2. Present the product
6. **Final Version of Project Plan**
7. **Final Version of Design Document**
8. **Advisory Document**

**Sprint Demo (January 24, 2025):**

* Present the fully functional system to the client.
* Provide a walkthrough of all features and functionalities.
* Deliver documentation and finalize project handover.

## Chapter 4: Intermediate Results

* **Sprint 1**
  + Focus: MVP scope definition, system and architecture planning.
  + Security Activities:
    - Initial threat modeling
    - GitLab repository and CI setup
    - Planning cryptographic use
  + Deliverables & Report Chapters:
    - Project plan
    - Product backlog
    - Draft threat model
    - Report Chapters: Introduction, Project Planning
* **Sprint 2**
  + Focus: Implementation of core user persona management.
  + Security Activities:
    - Cryptographic implementation
    - Static code analysis
  + Deliverables & Report Chapters:
    - Working persona module
    - Encrypted local storage
    - CI/CD pipeline foundation
    - Report Chapters: Concise Summary, Research, Secure Programming, Cryptography
* **Sprint 3**
  + Focus: Development of social features and secure login/auth.
  + Security Activities:
    - Secure communication implementation
    - Authentication logic
    - Expanded test coverage
  + Deliverables & Report Chapters:
    - Friend comparison functionality
    - Authentication system
    - Test reports
    - Report Chapters: Threat Modeling, Code Review, Secure SDLC
* **Sprint 4**
  + Focus: Final UI polish, integrations, and documentation.
  + Security Activities:
    - Secure UI review
    - Final threat model update
  + Deliverables & Report Chapters:
    - Final production-ready app version
    - Complete documentation
    - Code review summary
    - Report Chapters: Conclusions, References, Appendices

## Chapter 5: Project Boundaries

**Project Scope**

We are building a **web application** for our client, NHL Stenden, designed to serve as a **planning system for students**. The system aims to facilitate students' academic journey by enabling them to visually organize their courses on a timeline using a **drag-and-drop interface**. This application will provide a **proof of concept** to demonstrate its feasibility, usability, and functionality.

**Project Inclusions**

The following elements are within the scope of this project:

* **User Roles and Permissions:**
  + The system will support distinct roles, including Super Admin, Teacher, and Student, with specific permissions for each.
* **Planning Functionality:**
* Students can drag and drop courses onto a visual timeline, making it easier to plan their academic period.
* **Warning System:**
* The application will generate warnings for excessive workloads, providing students with feedback about overloading periods or the year.
* **Cohort Management:**
* The system will manage courses based on different cohorts (starting years) and periods, ensuring relevance to each student.
* **Mock Database:**
* A functioning database populated with mock data will be created to support the proof of concept.
* **Design Adherence:**
* The application will be aligned with the NHL Stenden brand book to maintain consistency with institutional design guidelines.

**Project Exclusions**

The following items are outside the scope of this project:

* **Deployment and Hosting:**
* The application will not be deployed or hosted. Only the proof of concept will be developed and demonstrated.
* **Database Hosting:**
* The database will not be hosted externally; it will exist solely for the proof of concept within the development environment.
* **Communication Features:**
* The application will not include a chat or messaging function between teachers and students.
* **Third-Party Integrations:**
* Integration with external systems or tools (e.g., email notifications, SMS alerts) is excluded.
* **Advanced Customization:**
* Features requiring extensive personalization, such as custom course selection algorithms, Artificial Intelligence will not be implemented.

**Project Constraints**

* **Time Constraint:**
* The project follows a defined timeline, with deliverables completed within a 5-sprint framework.
* **Budget Constraint:**
* Budget considerations are limited as the project is for academic purposes and focuses on proof of concept.
* **Resource Constraint:**

Development is restricted to the tools and technologies agreed upon (e.g., Django for the backend, Vue for the frontend).

**Conclusion:**

The project boundaries focus on delivering a functional proof of concept for a student planning system, emphasizing core features like drag-and-drop course management, cohort-specific organization, and workload warnings. Exclusions such as deployment, hosting, and advanced integrations ensure the project remains feasible within the given time and resource constraints. This approach allows us to meet the client’s needs effectively while staying aligned with the project's scope and objectives.

## Chapter 6: Quality Control

To deliver a high-quality, secure, and maintainable Android application, the team will implement rigorous quality checks throughout design, planning, development, and testing phases.

**Sprint Planning & Progress Tracking**

* Bi-weekly sprint planning and retrospectives to review completed work against acceptance criteria, identify blockers, and adjust priorities.
* Weekly (minimum) stand-ups for quick alignment; all tasks and bugs tracked in GitHub Issues under the Scrum Master’s oversight.

**Design Reviews**

* Initial screen sketches are made in Figma, then shared with the team and client for feedback.
* Before code starts, we confirm layouts are clean, easy to navigate, and match our style guide.

**Code Quality & Security**

* Enforce a shared Kotlin style guide via ktlint in the CI pipeline—any style error fails the build.
* Every merge request needs at least one approving code review before it’s merged.
* Run an automated dependency scan each build to spot known vulnerabilities.

**Testing**

* Unit tests cover all core logic and data-handling code.
* UI tests verify key flows (login, game list, stats view).
* CI builds are blocked if any test fails, ensuring new changes don’t break existing features.

**Performance & Reliability**

* Track app startup time and screen-load speeds.
* Use Android Studio profilers to catch memory leaks or excessive CPU/network use.

**Documentation & Release Notes**

* Maintain all setup guides, API specs, and architecture notes as Markdown in the repo.
* After each sprint, publish release notes summarizing new features, bug fixes, and security updates.
* Include screenshots or sample API calls for any major UI or endpoint changes.

By embedding these straightforward checks into our two-week sprints and making quality a shared responsibility, we’ll keep the app stable, secure, and on track.

## Chapter 7: Project Organization

**The job distribution of the team goes as follows:**

* Ayomide Olagoke - Team Leader
* Timothy Adewale - Scrum Master
* Radu Dumitrache - Code Lead
* Stefan Tasca - Secretary
* Ioan Mich - Document reviewer
* Vladislav Chitac - Developer

 Name: Ayomide Olagoke

Telephone number: +31 6 13102423

Email: [ayomide.olagoke@student.nhlstenden.com](mailto:ayomide.olagoke@student.nhlstenden.com)

 Name: Timothy Adewale

Telephone number: +234 813 348 0312

Email: [adewale.adewale@student.nhlstenden.com](mailto:adewale.adewale@student.nhlstenden.com)

 Name: Radu Dumitrache

Telephone number: +31 6 44544516

Email: [radu.dumitrache@student.nhlstenden.com](mailto:radu.dumitrache@student.nhlstenden.com)

 Name: Radu Dumitrache

Telephone number: +31 6 44544516

Email: [radu.dumitrache@student.nhlstenden.com](mailto:radu.dumitrache@student.nhlstenden.com)

Name: Stefan Ionut Tasca

Telephone number: +40 787210982

Email: [stefan.tasca@student.nhlstenden.com](mailto:stefan.tasca@student.nhlstenden.com)

Name: Ioan Mich

Telephone number: +31 6 25387906

Email: [ioan.mich@student.nhlstenden.com](mailto:ioan.mich@student.nhlstenden.com)

Name: Vlad Chitac

Telephone number: +380 66 878 8863

Email: [vlad.chitac@student.nhlstenden.com](mailto:vlad.chitac@student.nhlstenden.com)

Client: Name: Arvid Fens

Role: Project Leader

Email: [arvid.fens@nhlstenden.com](mailto:arvid.fens@nhlstenden.com)

## Stakeholder Groups

We categorize the stakeholders into the following groups:

• Manage Closely

• Keep Satisfied

• Keep Informed

• Monitor

1. Manage Closely

Primary Stakeholder:  
Project Client (Mr. Fens), Team Members, Teachers/Lecturers (Mr. Laan and Mr. Meijerink).

Strategy: Direct and consistent engagement is essential. Communication will be handled through emails and regular scheduled meetings to ensure that our project's objectives align with the client's, lecturers', and team members' expectations and requirements.

2. Keep Satisfied

Stakeholders: Potential investors and Board Members (NHL Stenden).

Strategy: While securing their strategic support and funding is our focus, maintaining a positive relationship with other potential clients or partners in the industry is vital. We will engage them through occasional project updates and showcasing relevant milestones.

3. Keep Informed

Stakeholders: Teachers, educational bodies, and students at NHL Stenden.

Strategy: Regular updates during "Atelier" hours at NHL Stenden will be provided. This keeps the academic stakeholders in the loop about our project's progression and educational relevance. Updates will also include usability feedback sessions to address student needs.

4. Monitor

Stakeholders: Government bodies, media, competitors, and suppliers.

Strategy: Monitoring these entities is crucial for staying informed about market trends, technological advancements, and regulatory changes. This will primarily involve keeping an eye on their public releases, market strategies, and any relevant industry news.

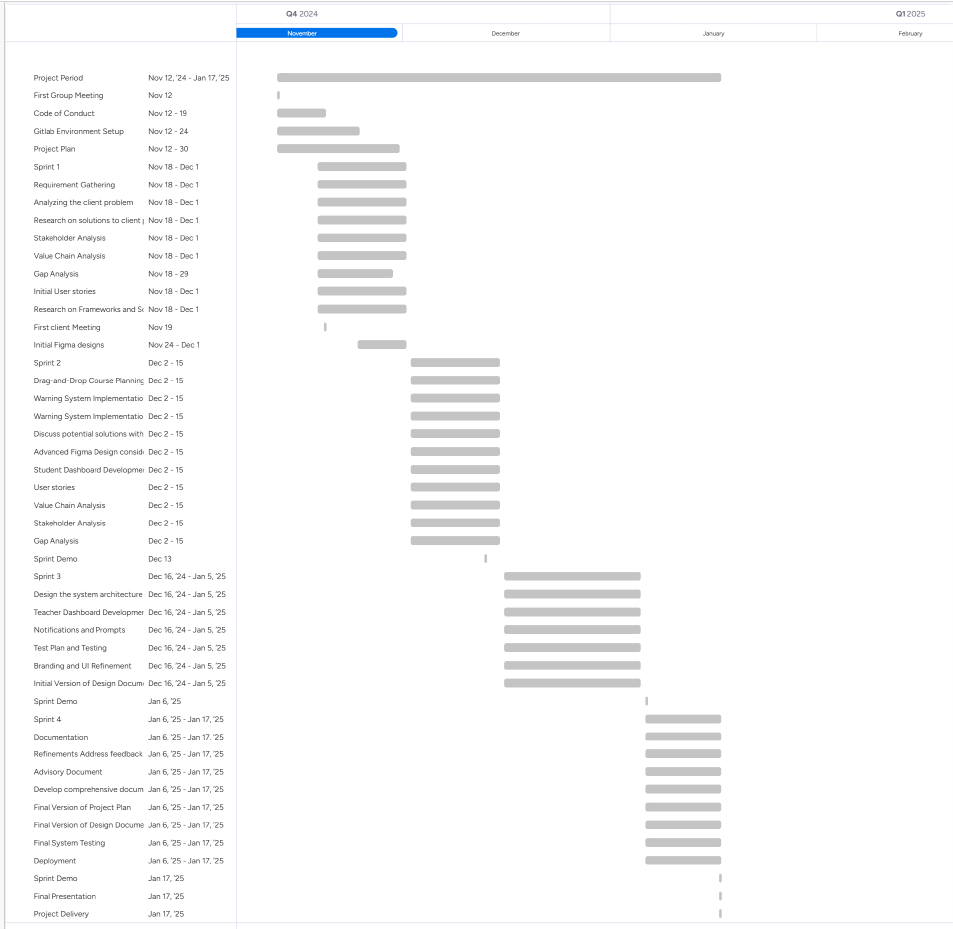
The connections between the stakeholders are detailed in the Stakeholder Analysis document, which is attached to this file under the name ["Stakeholder Analysis."](https://newuniversity.sharepoint.com/:b:/r/sites/Group-Group-Project2.1IT224_25-IT3C/Shared%20Documents/IT3C%20(SPS)/Project%202.2%20resit/Deliverables/Team%20files/Analysis%20Documents/SPS%20Stakeholder%20Analysis.pdf?csf=1&web=1&e=pfXcf3)

Information:

During this project, team members will meet during the atelier, and contact the client by Email or Teams regularly. When team members cannot join meetings physically, he/she will have online meetings on Microsoft Teams. For each official meeting, the project team will take minutes of the meeting. If any emergency occurs, the project team will contact the client as soon as possible.

## Chapter 8: Planning

Additional elaboration concerning the planning will be done via the Gantt chart, which will be attached in addition to this document. The Gantt chart is consistent with the project activities stated in chapter 3.  For a more detailed and clearer view check the file name SPS Gant in the documents folder.



## Chapter 9: Cost and Benefits

#### **Costs**

1. **Development Costs:**
   1. **Team Resources:** Allocation of time and effort from developers, designers, and project managers.
   2. **Infrastructure:** Local servers for development and testing.
2. **Maintenance Costs:**
   1. Regular updates to ensure system security, fix bugs, and add new features.
   2. Training and documentation updates for administrators and users.
3. **Time Investment:**
   1. Time spent by stakeholders (students, teachers, and administrators) in feedback sessions and testing the system.
   2. Time required for team meetings, planning, and coordination.
   3. Time spent on development of the system.

#### **Benefits**

1. **Administrative Efficiency:**
   1. The system simplifies and centralizes course management, significantly reducing the workload for administrators.
   2. Automated warnings for overloaded academic periods or years save time compared to manual checks.
2. **Enhanced Student Experience:**
   1. **Visual Clarity:** Students gain a clear view of their curriculum for the academic year, enabling better planning and prioritization.
   2. **Flexibility:** Students can choose the order of courses to redo if necessary, empowering them to take control of their academic paths.
   3. **Error Reduction:** The drag-and-drop functionality minimizes manual errors in planning.
3. **Improved Communication:**
   1. Teachers can view students’ study plans and provide targeted feedback.
   2. Clear prioritization ensures alignment between students and faculty.
4. **Long-Term Savings:**
   1. Reduced need for email-based or manual consultations regarding study plans, saving time and effort for both students and lecturers.
5. **Scalability:**
   1. The platform serves as a proof of concept, potentially extending its utility to other departments or institutions, increasing its overall value.

In conclusion, while the Superior Planning System requires investment in resources and time, the benefits it offers in terms of efficiency, clarity, and user experience justify these costs. This system not only reduces the workload for administrators but also empowers students with tools to plan their academic journeys effectively, aligning with NHL Stenden’s mission to enhance the educational experience.

## Chapter 10: Risk Analysis

The following risk analysis is based on the model from Grit (2021) and adapted to fit the GamerCV project. Since the project has no external clients or budget, the focus is on internal project risks, technical dependencies, and external events.

## Internal risks

|  |  |  |
| --- | --- | --- |
| Risk | Description | Mitigation Strategy |
| Inadequate planning | Poor mapping of features and tasks can cause delay. | Create a clear feature list and technical design document early in development. |
| Poor team coordination | Miscommunication or unclear roles can cause duplicated or missing work. | Use GitLab actively; Assign roles clearly; Conduct regular syncs, stand-ups and meetings; |
| Feature creep | Adding extra features (e.g. more games, UI flair) may delay core development and the whole premise of the project. | Lock MVP scope early. Track extras as ‘future features’ |

## Technical risks

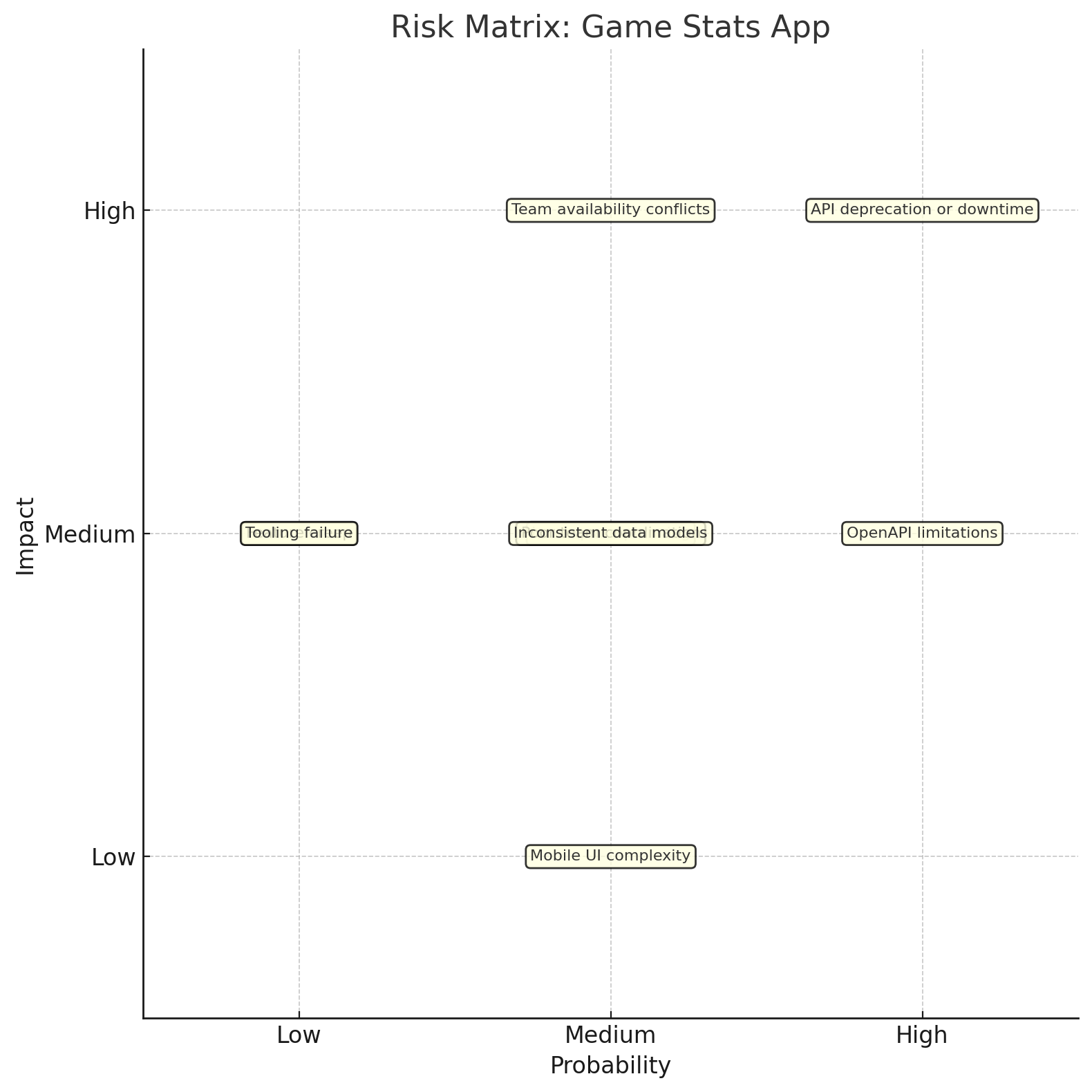
|  |  |  |
| --- | --- | --- |
| Risk | Description | Mitigation Strategy |
| OpenAPI limitations | APIs might have rate limits or missing data. | Test endpoints early. Mock responses. Use retry logic. |
| Inconsistent data models | Games expose data in different structures. | Normalize API data using mappers. |
| Mobile UI complexity | Stat-heavy screens might not fit well on mobile devices. | Use wireframes. Prioritize clarity and minimal design. |

## External risks

|  |  |  |
| --- | --- | --- |
| Risk | Description | Mitigation Strategy |
| API deprecation/downtime | Public APIs may be disabled or change format without notice. | Document all endpoints. Add fallback logic. Cache key stats locally. |
| Team availability conflicts | Personals conflicts like exams or illness can reduce team capacity. | Set up flexible workload distribution. Conduct weekly stand-ups and have clear communication. |
| Tooling failure | Dependency errors or broken dev environments might block progress. | Use Git and commit regularly. Keep setup guides updated. |

## Risk assessment criteria

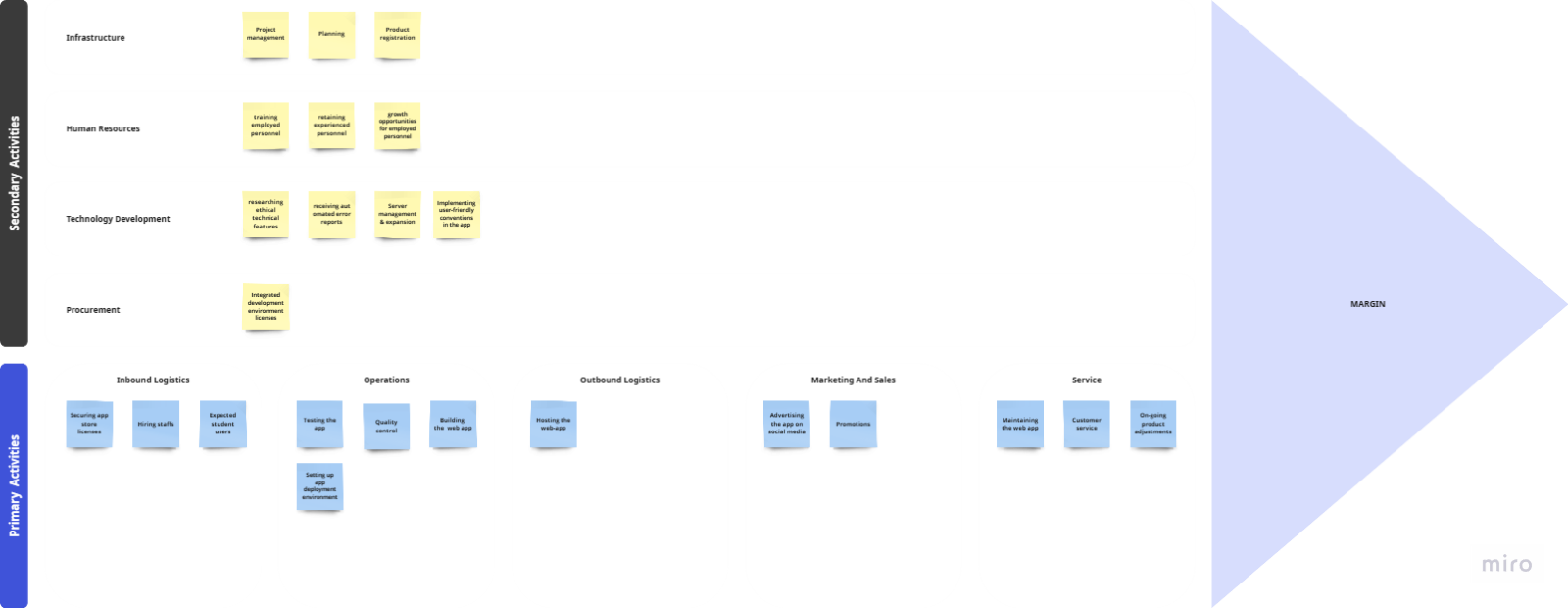
A risk matrix has been created to help classify and prioritize the risks in a visual manner.



*Figure x. Risk assessment matrix*

## Chapter 11: Appendix

## 11.1 Value Chain Analysis

  
The key sub-activities for the company are quality control, app development, and promotions. Supporting these are product adjustments, testing, project management, planning, and social media advertising.

Effective communication between these activities is crucial for success, fostering coordination and quick issue resolution, while poor communication risks project failure.

Promotions and app maintenance provide the most value by boosting visibility, user base, and reliability.

Our project enhances the client's value chain by offering an app for students to visualize their study plan, sharpening their academic year awareness & reducing the time it takes to realize their study plan.

## 11.2 GAP Analysis

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Structure** |  | **Current situation** | **Desired situation** | **GAP** | **Action** |
| **Management** | -There is no current system   that allows the client to view the study plan of students.  Due to this, the client must deal with many requests to review study plans via mail.  -The entire process is time consuming | An application that allows the students to create their study plan for an academic year, gives teachers access to view the study plan of their assigned student cohort. | Currently, the client does not have any application that the students can use to organize their study plan | The team will build a web-app that students can use to organize their study plan by academic year. |
| **IT management** | The client currently uses progresswww.nl to view the study progress of students | An application that provides the client a feature to view the study progress of students via their current study plan & past study plans | Client does not have an application that shows a student's study plan | The team will develop an application for the client so they can work more conveniently. |
| **User group X** | The client does not have an application | The client will have a web-app that allows him to specify courses for a study by academic year | Our client does not have an application yet. | The team will build a web-app for the client that accomplishes the desired situation. |
| **Technology** | **Hardware** | There is no hardware part for this system |  | There is no gap at this moment. | There is no action to be taken during this project |
| **Software** | The client does not have an application yet, currently they use paper/spreadsheets to visualize a student's study plan. | Clients will receive a web-app that has a visual feature that allows the client to view the study plans of students. | The client does not have an application. | The team will develop a web-app that the client can use to view the study plans of students |
| **Network** | The clients' uses proxmox for web hosting | No desired situation is applicable to the boundaries of this project | - | - |
|  |  |  |  |  |
| **Staff** | **System management** | Our client has partial access to the systems of the company. | The client will have a web-app that is connected to related parts of the system (Microsoft OAuth) | The clients do not yet have the application. | The team will build a web-app for the clients, and they will be granted full access to it. |
| **Functional management** | The client has no application that fulfills the expected functionalities. | An application that fulfills all functions the client has agreed to. | The clients do not have a system with any functionality they can use yet. | The team will develop a web application system with working functionalities, so the clients can use all the functions of the system. |
| **User X** | Students create their study plan in paper/ spreadsheet, then send it to the client via email. | -Reduced manual efforts in creating a study plan.  -Less time spent on conversations with students over their study progress by having the ability to view a student's study plan via an app. | The clients lack a system that reduces the effort in the process of reviewing a student's study plan/progress. | -Implement a system that reduces the time & effort required to review a study plan/progress-Create a user manual that describes how to use the systems |
|  |  |  |  |  |
| **Culture** | **Access** | The client do not have access to any systems of the web-app since it is not developed yet. | The client will have access to the web app once it is developed. | The clients do not have a web app yet. | The team will build a web-app for the clients, and they will get full access to the new system. |
| **Training level** | The client has experience in using built applications | Client will know how to use the application completely | The client does not know how to navigate through the app  The client does not know all the app's features | -The team will present to the client how to use the web app  -The team will provide the client with a video that explains all the created features in the app. |
| **Behavior** | The client must deal with many requests to review study plans via mail.  -The entire process is time consuming | The clients will have an app that shortens the process of reviewing study plan/progress related requests. | Clients need an app that reduces the time spent on study plan/progress related requests. | The team will develop an application to help the client save time on reviewing study plan/progress of students. |
|  |  |  |  |  |

According to the gap analysis, the client's non-existing system of managing request to create/review study plan and progress of students is slow and time consuming. To close this gap, a comprehensible web application that allows students to create a visual representation of their study plan and a feature that allows the client to view the study plans of students. The client will save time by streamlining operations, reducing conversations required to create an effective study plan, increasing efficiency.