

Project: Game Collection Management and Sourcing System

Planning and Scheduling

1. Project Goals and Objectives

1.1. Goal

With the rapid development of the multimedia field, the types of games are becoming more novel and diverse. There are so many shops, magazines, websites, and blogs for the player. It is extremely difficult to find and store these resources by individuals. To help players to overcome this tricky problem. This team aims to develop an efficient system that helps the player manage their favorite game collection, connect with the player community, and introduce new games.

1.2. Objectives

To implement this system, this project is divided into three phases. The first phase is analyzing the requirements and design making. At this stage, the team writes the requirements of the system through a group meeting, and the team designs the interface of the system by researching the current game platform. The second Phase is making a concrete development plan. At this stage, the team starts to schedule tasks and analysis risks. The third phase is implementing the main system. At this stage, the team starts to implement this system by using planned technicals.

2. Project Development Scope

2.1. Constraints

1. *Duration*: Total Time is 3 Weeks. The team aims to have 3 weeks to implement the prototype of this system.
2. *Model*: Browser-to-Server model(B/S). The team is going to implement this system with the Client-to-Server model. This model has several advantages; it can largely speed up the rate of development because each component in this model communicates by using network requests. it enables this system to be cross-platform since it runs on any device with the browser. it is easy to maintain functionality in that updates and maintenance can be done centrally on the server, reducing the need for client-side updates.

3. *People*: This project needs four developers to work together, including the two interfaces developer, and two server developers. By using the browser model, two parts of the team can work separately after discussing the design of the API. It reduces the cost of frequently meeting. And it shortens the critical path.
4. *Technicals*: The team aims to use JavaScript and React.js to develop the interface (Browser side), and the server is implemented by using Python and Django Framework structure. These tools are most commonly used in the current industries. There are plenty of reference resources for the developer team to study. It ensures that the team member can train themselves.

2.2. Inclusion

1. Implementing User Management sub-system
2. Implementing Library Management sub-system
3. Friends sub-system
4. game community system

2.3. Exclusion

1. online game platform
2. powerful game-searching system
3. game e-commercial platform
4. chat platform
5. social components

3. Project Tasks

Since this project uses the Browser-to-Server Model, All tasks can be divided into three sub-categories. The first part of the project is the design and communication of the APIs, including data format, API path, and type. The second part is implementing the interface of the prototype system. The third part is implementing the server of the prototype.

3.1. API designs and documents

1. *Building Data Format*: This task figures out what kinds of data need, how to store these data, and how to use these data.

2. *Document API Path and Type*: This task documents the correct API path which is used to communicate between the front-end (Browser client side) and back-end (Server-side). And this task also figures out the type of APIs.

3.2. Implementing Interface function

2. *Implement Registration*: determine the registration form which includes data, such as name, email address, area, username, and password; and, it implements validation rules for each field in the registration form to ensure that the user has entered valid information.
3. *Implement Login*: design a login form that enables the user to input their account and password to enter the game library. If the user inputs the right account and password, then the user can enter the system, otherwise, the system needs to prompt the error message.
4. *Implement Reset Password*: design a reset password form that enables the user to reset their password after forgetting it. The user needs to input a verification code. If the validation pass, the user can input a new password to replace forgot password.
5. *Implement Library Manage page*: Implement a page that needs to display all games the user has. and it displays specific game information after the user selects it.
6. *Implement Community page*: Implement the list view that displays all games' communities. Then, Implement the community page which displays game reviews.
7. *Implement Friend Page*: Implement a list view that displays all friends of a player. Then, design a form that is used to display detailed information about one friend after selecting one of the friends. Design a search bar that enables the player to search for friends by the filter.
8. *Implement Profile page*: Implement a page view that displays the player's detailed information. Enable the modification function which supports the player to update information.

3.3. Implementing Server

1. *Implementing User Controller*: Implementing the User Controller which is responsible for managing user data and activities related to user accounts. It typically involves creating and updating user accounts, handling authentication and authorization, managing user profiles, and controlling access to user-specific resources.
2. *Implementing Social Friend Network Controller*: The Social Friend Network Controller is responsible for managing the social connections between users. It typically involves implementing features such as adding friends, sending messages, and creating groups

related to games. The implementation may involve implementing algorithms for recommending friends based on game preference.

3. *Implementing Game Controller*: Implementing Game Controller which is responsible for showing extensive information about each game in the collection, such as reviews and ratings. The implementation may involve a detection algorithm for blocking useless, irrelevant or offensive, discriminatory, etc. comments.
4. *Implementing Community Controller*: Implementing the Community Controller which is responsible for managing the community-related activities of the application related to games. It includes implementing features such as creating and managing communities for specific games, handling community events related to games, and managing community-related data for games. The implementation may involve letting users post their thoughts in the community and implementing algorithms for recommending communities for games, and designing user interfaces for managing communities.
5. *Implementing Game Suggestion Controller*: The Game Suggestion Controller is to provide personalized game recommendations to users based on their gaming preferences. The implementation tasks typically involve implementing algorithms for analyzing user data related to gaming and recommending games based on user preferences and also implementing recommendation algorithm.

4. Changes to Structure Design

1. *MVC to B/S (Browser to Server)*: In the original structure design, the MVC (Model-View-Controller) is deprecated for several reasons. First, It is hard to fully follow the MVC; The structure of the MVC model is too complex for small or medium projects. Sometimes, it would be a better choice that the View component directly communicates with the Model component to retrieve data. Strictly following the MVC will extend the duration of the project. Secondly, Tasks can't be finished in parallel by following the MVC model. It is hard to shorten critical paths. B/S mode divides the project into two parts, the Browser interface, and the server. The team can develop two parts in parallel. It speeds up the rate of development. And, The maintenance of B/S mode is easier than MVC.

5. Scheduling

1. *Gantt Chart*

Board Game Collection System

Software Development Group

Team Member:

Project Start: Thu, 3/23/2023

Xuanqi Li, Yuchen Pu
Meghna Rajee, Xinneng Chen

Xuanqi Li, Yuchen Pu Meghna Rajee, Xinnmeng Chen						Mar 20, 2023							Mar 27, 2023							Apr 3, 2023							Apr 10, 2023						
						M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S	M	T	W	T	F	S	S
TASK	ASSIGNED TO	PROGRESS	START	END	DAYS																												
API designs and documents																																	
Building Data Format	All Member	100%	3/23/23	3/25/23	3																												
Document API Path and Type	All Member	100%	3/25/23	3/26/23	2																												
Implementing Interface function																																	
Implement Registration	Xinnmeng Chen & Yuchen Pu	100%	3/23/23	3/24/23	2																												
Implement Login	Xinnmeng Chen & Yuchen Pu	100%	3/23/23	3/24/23	2																												
Implement Reset Password	Xinnmeng Chen & Yuchen Pu	100%	3/23/23	3/24/23	2																												
Implement Library Manage page	Xinnmeng Chen & Yuchen Pu	50%	3/25/23	3/28/23	4																												
Implement Community page	Xinnmeng Chen & Yuchen Pu	0%	3/29/23	4/1/23	4																												
Implement Friend Page	Xinnmeng Chen & Yuchen Pu	0%	4/2/23	4/3/23	2																												
Implement Profile page	Xinnmeng Chen & Yuchen Pu	0%	4/4/23	4/5/23	2																												
Implementing Server																																	
Implementing User controller	Xunnqi Li & Megha Rajee	100%	3/23/23	3/25/23	3																												
Implementing Social friend network controller	Xunnqi Li & Megha Rajee	90%	3/26/23	3/28/23	3																												
Implementing Game controller	Xunnqi Li & Megha Rajee	0%	3/29/23	4/2/23	5																												
Implementing Community controlle	Xunnqi Li & Megha Rajee	0%	4/3/23	4/6/23	4																												
Implementing Game suggestion controller	Xunnqi Li & Megha Rajee	0%	4/6/23	4/7/23	2																												
Implement Database																																	
Define and Plan Database Schema	Xuanqi Li & Megha Rajee	100%	3/23/23	3/24/23	2																												
Implementing Database	Xuanqi Li & Megha Rajee	90%	3/25/23	3/28/23	4																												
Populating Database	Xuanqi Li & Megha Rajee	0%	3/29/23	3/30/23	2																												
Testing, Evaluating and Potimizing Database	Xuanqi Li & Megha Rajee	0%	3/30/23	4/1/23	3																												
Combining Three Part																																	
Database and Backend	Xunaqi Li & Meghna Rajee	0%	4/7/23	4/8/23	2																												
Front-end, Back-end and Database	All Member	0%	4/8/23	4/9/23	2																												
Test and Evalution	All Member	0%	4/9/23	4/12/23	4																												

2. Task Time Allocation Principle

The time allocated to the task was determined by the complexity of the task implementation, the delivery date of the assignment and the average time spent by the group on the project per day. Due to the need to take into account both the final project research and other subject assignments, the group members agreed to spend an average of 4 to 5 hours per day on this project.

The level of complexity of the tasks was variable from one task to another. For example, in the front-end development phase of the task the user interfaces for registration, login and password change were similar enough that all three pages could be implemented in two days. However, the front-end aspects of the community function and management system, which were the focus of this assignment, were much more complex than the login interface, so with the agreement of the front-end developers yucheng and xinneng, it was decided to allocate four days each for the community function and management system front-end user interfaces. Similarly for the back-end implementation, the game controller and community controller were the most challenging to implement as the main part of the application, so meghna and xuanqi agreed to spend a total of 9 days on these two parts. The task of merging the front-end, back-end and database was less time consuming as it was a joint effort between all the team members. However, for the application evaluation, given that the evaluation part of the assignment is an extremely important part, we decided to allocate 4 days to complete

Risk Analysis

1. Risk analysis methods employed

As part of our risk assessment process, we selected the risk matrix as our primary risk analysis method. We chose this method because it provides a clear and concise way to assess and prioritize risks based on their likelihood and impact. Compared to other methods, such as SWOT analysis or fault tree analysis, the risk matrix is simpler to use and provides a standardized approach to risk assessment.

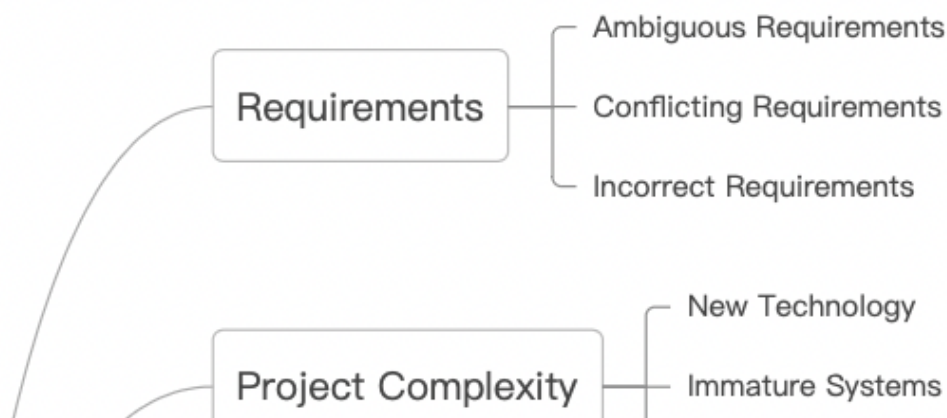
One of the key benefits of the risk matrix is its ease of communication, which allows us to share the results of the risk assessment with the group in a consistent and objective way. By prioritizing risks based on their potential impact on the project, we can focus our efforts on managing high-priority risks and ensuring the success of the project.

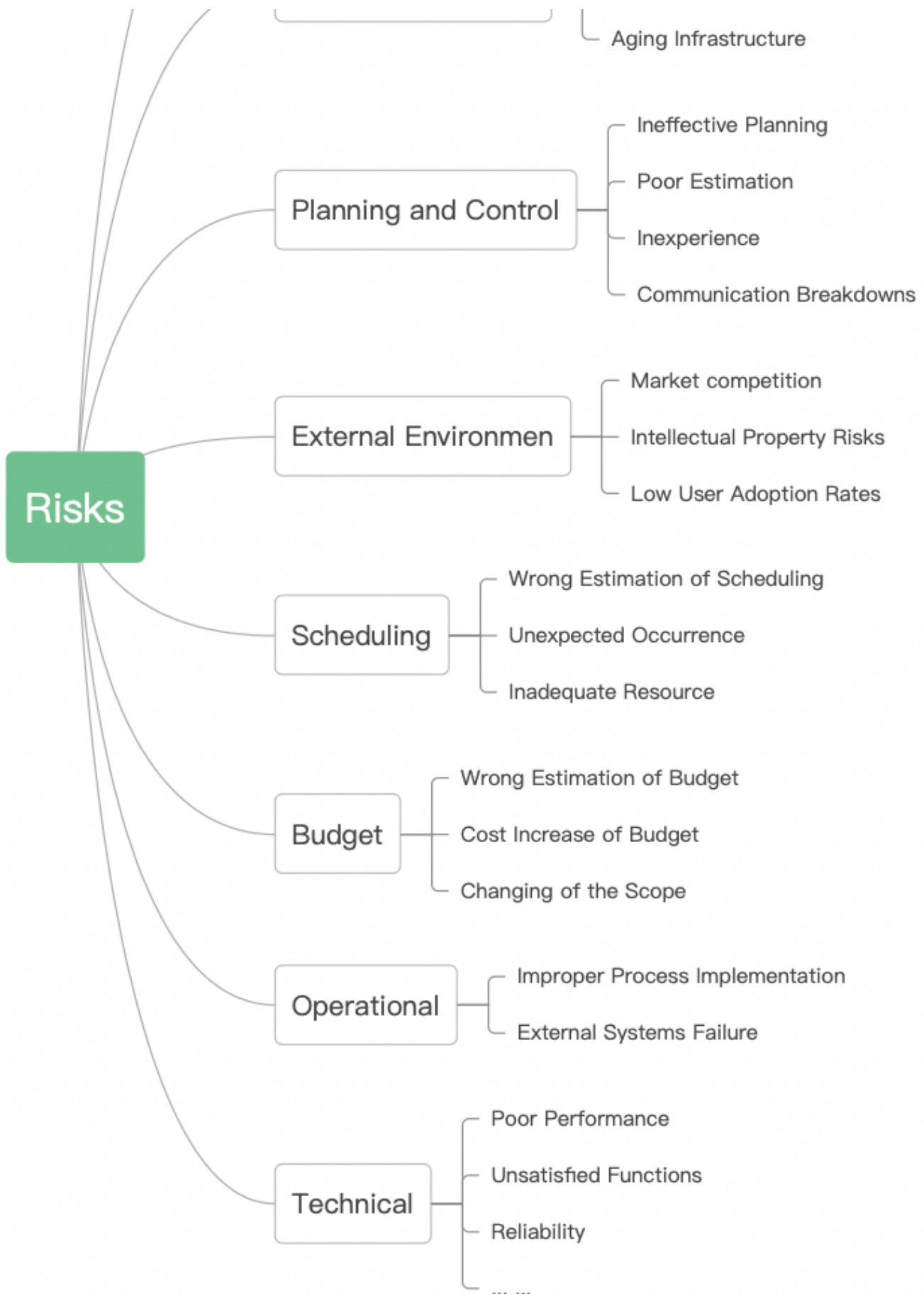
Overall, we believe that the risk matrix is a valuable tool for assessing and managing risks, and we are confident that it will help us to identify potential issues and take proactive measures to address them.

2. Risks identified and applied risk mitigation strategies

To ensure the success of our project, we have conducted a thorough risk assessment and categorized the potential risks into eight distinct categories: Requirements, Project Complexity, Planning and Control, External Environment, Scheduling, Budget, Operational and Technical. Each category represents a different area of potential risk, such as challenges related to project requirements, the complexity of the project, planning and control processes, team dynamics, and the overall organizational environment. By identifying and evaluating risks in each category, we can develop a comprehensive risk management plan that addresses potential threats and helps us proactively mitigate risks throughout the project's lifecycle.

The chart indicating the categories of risks is shown below.





Risk Number	01
Risk Name	Ambiguous Requirements
Risk Category	Requirements
Risk Description	Confusion, rework, and delays in the completion of a project can result from unclear, incomplete, or changing requirements.
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	Regularly review and clarify requirements within team, document the changes and seek feedback

Risk Number	02
Risk Name	Conflicting Requirements
Risk Category	Requirements
Risk Description	Competing or contradictory requirements may result in disputes between team members and project delays.
Risk Possibility	Medium
Risk Impact	Medium
Risk Priority	High
Risk Mitigation	Facilitate team meetings to settle disputes, record requirements, and seek consensus

Risk Number	03
Risk Name	Incorrect Requirements
Risk Category	Requirements
Risk Description	Requirements that don't match the project's objectives, user demands, or industry standards might result in lost work, missed opportunities, and project failure.
Risk Possibility	Low
Risk Impact	High
Risk Priority	High
Risk Mitigation	Perform user research, go through industry standards, and double-check requirements

Risk Number	04
Risk Name	New Technology
Risk Category	Project Complexity
Risk Description	Inexperience with new or emerging technologies like django and react can cause delays, errors, and increased costs.
Risk Possibility	Low
Risk Impact	High
Risk Priority	High
Risk Mitigation	Conduct thorough research and testing of new technologies, start training on Django and react in available time

Risk Number	05
Risk Name	Immature Systems
Risk Category	Project Complexity
Risk Description	Systems or architectures that are unproven or experimental may result in technical challenges and longer development time
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Research and test new systems or architectures thoroughly, keep track of all decisions, and consult experts as needed.

Risk Number	06
Risk Name	Aging Infrastructure
Risk Category	Project Complexity
Risk Description	Outdated or obsolete systems, technologies or processes can lead to system failures
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Plan for replacements, carry out routine maintenance, and update, and get professional assistance.

Risk Number	07
Risk Name	Ineffective Planning
Risk Category	Planning and Control
Risk Description	Poorly developed, incomplete or unrealistic project plans can lead to delays and missed objectives
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	Create comprehensive, achievable, and realistic project plans, record all presumptions, and conduct regular team reviews of them.

Risk Number	08
Risk Name	Poor Estimation
Risk Category	Planning and Control
Risk Description	Overly optimistic or incorrect estimates may result in missing deadlines and resource limitations.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Conduct in-depth research and analysis to create precise estimates; record all presumptions; and periodically evaluate and modify estimates as necessary.

Risk Number	09
Risk Name	Inexperience
Risk Category	Planning and Control
Risk Description	Lack of experience or understanding with technologies or tools can lead missed deadlines
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Give team members the chance to learn and grow, and use efficient techniques.

Risk Number	10
Risk Name	Communication Breakdowns
Risk Category	Planning and Control
Risk Description	Having poor or ineffective team communications can result in delays, rework, and misunderstandings.
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	Create and carry out efficient communication strategies, frequent meetings and feedback channels, and encourage transparency and collaboration.

Risk Number	11
Risk Name	Market competition
Risk Category	External Risk
Risk Description	Increased competition from other companies can impact the project's market share
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	Conducting regular market analysis, identifying unique selling points

Risk Number	12
Risk Name	Intellectual Property Risks
Risk Category	External Risk
Risk Description	The system may use logos or names that infringe on registered trademarks, resulting in legal disputes and financial penalties.
Risk Possibility	Low
Risk Impact	High
Risk Priority	High
Risk Mitigation	Conduct a thorough search to ensure that all logos and names used are not registered trademarks.

Risk Number	13
Risk Name	Low User Adoption Rates
Risk Category	External Risk
Risk Description	The system might not be adopted by enough users, which would lead to low utilization rates and diminished effectiveness. The lack of user knowledge, the difficulty utilizing the technology, or the lack of perceived value might all be contributing factors.
Risk Possibility	Low
Risk Impact	High
Risk Priority	High
Risk Mitigation	To mitigate this risk, we will conduct user testing and feedback sessions during development to ensure that the system is user-friendly and meets the needs of the target user group.

Risk Number	14
Risk Name	Wrong Estimation of Scheduling
Risk Category	Scheduling Risks
Risk Description	Wrong estimation of the scheduling, including time scheduling and resources scheduling may result in the delay or failure of the project and the waste of resources .
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Double-check the time and schedule we assign to each task, describe the scheduling more clearly by drawing a Gantt chart.

Risk Number	15
Risk Name	Unexpected Occurrence
Risk Category	Scheduling Risks
Risk Description	Unexpected occurrence, e.g., one of the teammates falls sick and cannot complete his work on time may result in the delay of the completion of the project, leading to the unsucccess of the project.
Risk Possibility	Low
Risk Impact	High
Risk Priority	High
Risk Mitigation	List occurrences likely to happen in the process of the projects and make preparation for them.

Risk Number	16
Risk Name	Inadequate Resource
Risk Category	Scheduling Risks
Risk Description	Without sufficient resources, e.g., some libraries to code, or enough capacity of computers, we are unable to complete some tasks, leading to the failure of our project.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Preprare the resource which we may need in the process of our project as early as possible.

Risk Number	17
Risk Name	Wrong Estimation of Budget
Risk Category	Budget Risks
Risk Description	Failure to make a good estimate of the project budget before the project starts will lead to project failure or over-budget problems.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Use a variety of budget estimation methods to estimate as early as possible, we can ask professional and experienced cost estimators for help or use emerging machine learning methods.

Risk Number	18
Risk Name	Cost Increase of Budget
Risk Category	Budget Risks
Risk Description	Due to improper budget estimation in the early stage, the budget needs to be increased during the progress of the project. The project may fail because there are no more budget sources.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Use a variety of budget estimation methods to estimate as early as possible, and re-evaluate the budget we still need in the process of the project.

Risk Number	19
Risk Name	Changing of the Scope
Risk Category	Budget Risks
Risk Description	Failure to make a good estimate of the project budget before the project starts will lead to project failure or over-budget problems.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	During the project development process, the customer's needs may change, and we may find some previous misunderstandings of user needs, which will cause us to re-estimate our budget. If our budget is insufficient after the scope change, it may lead to project failure.

Risk Number	20
Risk Name	Improper Process Implementation
Risk Category	Operational Risks
Risk Description	In the software development process, each developer may have his own development process and code writing habits, which may cause conflicts in the development process and increase the workload of the project.
Risk Possibility	Medium
Risk Impact	Medium
Risk Priority	Medium
Risk Mitigation	Before development starts, establish a set of standard development processes, and let each developer follow this process to avoid conflicts.

Risk Number	21
Risk Name	External Systems Failure
Risk Category	Operational Risks
Risk Description	The software we develop may run on different external systems, and when these external systems fail, the software cannot run.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	Improve the compatibility and fault tolerance rate of software, so that the software can run normally when failures occur on different external systems. We can get in touch with the developer of the external system to obtain more fault information and troubleshoot early.

Risk Number	22
Risk Name	Poor Performance
Risk Category	Technical Risks
Risk Description	Performace includes the response speed and processing power of this software. For example, the response speed of the system to the query when searching for games, and the loading capacity of the software when multiple users use the software at the same time. If the performance is poor, the software will not be able to meet the customer's requirements, and it will be a failed project.
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	Optimize the algorithm to improve the response speed of the system, and test it with a large number of use cases to ensure that the performance of the system is good.

Risk Number	23
Risk Name	Unsatisfied Functions
Risk Category	Technical Risks
Risk Description	Due to failure to understand user needs or code logic problems, many functions may not be implemented as expected by customers, which will lead to project failure.
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	We can conduct more code reviews during the development process to ensure that there is no problem with the logic of the code. We can communicate with customers more about their needs to ensure that functions are carried out according to their requirements.

Risk Number	24
Risk Name	Reliability
Risk Category	Technical Risks
Risk Description	The system may experience unexpected downtime due to hardware or software failures.
Risk Possibility	Moderate
Risk Impact	High
Risk Priority	High
Risk Mitigation	Regular system maintenance and testing, redundancy in hardware and software, and quick response and recovery procedures.

Risk Number	25
Risk Name	Dependency
Risk Category	Technical Risks
Risk Description	The project relies on third-party services, libraries, or frameworks, therefore there is a risk of dependency issues, such as version conflicts or service disruptions.
Risk Possibility	Moderate
Risk Impact	High
Risk Priority	High
Risk Mitigation	To mitigate the dependency risk, we will carefully evaluate the third-party services, libraries, and frameworks we use and ensure that they are reliable and well-maintained. We will also regularly monitor for updates and potential conflicts, and have a backup plan in case of service disruptions.

Risk Number	26
Risk Name	Cultural Risks
Risk Category	Team
Risk Description	The team is composed of individuals from different cultural backgrounds, leading to potential misunderstandings and miscommunications.
Risk Possibility	Moderate
Risk Impact	High
Risk Priority	High
Risk Mitigation	Establishing clear communication protocols, promoting cultural awareness and sensitivity, and fostering an inclusive team environment.

Risk Number	27
Risk Name	Limited Technical Expertise
Risk Category	Team
Risk Description	The project team may lack the necessary technical expertise required to complete the project successfully. This can result in delays, errors, and a lower quality of work.
Risk Possibility	Moderate
Risk Impact	High
Risk Priority	High
Risk Mitigation	Identify areas where the team lacks expertise and provide training or support to team members to fill those gaps.

Risk Number	28
Risk Name	Inadequate Documentation
Risk Category	Technical
Risk Description	Insufficient documentation may lead to errors, delays, and miscommunication during development, testing, and maintenance phases of the project.
Risk Possibility	Moderate
Risk Impact	High
Risk Priority	High
Risk Mitigation	Develop and maintain comprehensive documentation throughout the project lifecycle, including requirements, design, testing, and maintenance documentation. Establish documentation standards and review processes to ensure completeness, accuracy, and consistency of documentation.

Risk Number	29
Risk Name	Data Security Risk
Risk Category	Technical
Risk Description	Our systems may be at risk of data breaches and cyber attacks. Our system will store the important information of the user, such as the user's gender, location and preferences. If this information is leaked, it will pose a great threat to user privacy and involve legal issues.
Risk Possibility	Moderate
Risk Impact	High
Risk Priority	High
Risk Mitigation	We can encrypt the data to make it difficult for the data to be stolen during transmission. We can also add access control to operations, only specific users can perform certain operations and obtain data.

Risk Number	30
Risk Name	Maintenance and Support Risk
Risk Category	Technical
Risk Description	After the software goes online, the system may fail or some functions may not be well implemented, which will cause huge obstacles to customers' use.
Risk Possibility	High
Risk Impact	High
Risk Priority	High
Risk Mitigation	When developers write code, they must pay attention to the maintainability, scalability and readability of the code. In this way, during the maintenance phase, it is easier to find problems in the code and modify them. In the code writing stage, we can also add some explanatory documents to the code for explanation, and review the code frequently to avoid failures as much as possible.

Risk Number	31
Risk Name	Scalability Risk
Risk Category	Technical
Risk Description	The system may need to handle a large number of users and game collections. There may be risks associated with scalability, such as system crashes, slow loading times, and poor system performance.
Risk Possibility	Medium
Risk Impact	Medium
Risk Priority	Medium
Risk Mitigation	We should adopt a scalable architecture for the software, so that when the software needs to support more users to use it at the same time, it is convenient to increase the capacity of the software. In addition, we can also use techniques such as partitioning, indexing, and caching to improve database performance and avoid slow response times.

Risk Number	32
Risk Name	Testing and Quality Assurance Risk
Risk Category	Technical
Risk Description	To ensure the proper functioning of the system, it may require rigorous testing and quality assurance. However, this process may come with certain risks, including testing delays, software bugs, and user acceptance issues.
Risk Possibility	Medium
Risk Impact	Medium
Risk Priority	Medium
Risk Mitigation	We can make a detailed test plan. The test plan should cover all testing phases and specify the testing objectives, test cases, testing schedule and testing resources. We can also adopt automated testing methods, which can reduce human-caused bugs.

Risk Number	33
Risk Name	Infrastructure Risk
Risk Category	Technical
Risk Description	There are risks associated with the infrastructure components that the system relies on, such as servers, networks, and databases. These risks may include server downtime, network outages, and data center disruptions.
Risk Possibility	Medium
Risk Impact	High
Risk Priority	High
Risk Mitigation	We can choose reliable infrastructure providers, such as choosing reliable servers and network providers. Prepare data backups and develop a recovery plan, so that when an infrastructure crash occurs, we can try our best to restore the original state and reduce losses.