Software Project and Process Management
Course Project Design

“ICE”—Online Entity Game Store

Team Member

1754060 Zhe Zhang

1753414 Di Bu

1753188 Kaixin Chen

1. General View

1.1 Background

With the development of platform like steam and GOG, digital games seem to be the very first choice for computer gamers nowadays. However, gamers of other gaming console still tend to buy entity games for purpose of collecting or others, and this market is still booming with more and more gamers buying game consoles. However, there doesn't exist a comprehensive platform for players to buy entity games of different consoles from different video game companies. So here comes ICE!

ICE is an online store selling entity games from different game companies for gamers owning different gaming consoles. Authorized by the game publishers, we sell only the legal copies and provide an easy access to purchasing entity games. Besides, we encourage players to discuss the game products freely, thus to build a community gathering players from all over the world.

1.2 Vision

A new business of an e-commerce company will be initiated: a virtual game store. The system to support it must manage the acquisition and selling processes of the company.

Before the appearance of ICE, the usual way for gamers to buy entity games is buying them on online retailers like Taobao, Amazon, and so on. There are several problems about this. First, most of those entity games sold on those websites are from personal business rather than official, which will cast doubt on the quality of the products. They could be illegal copies or second handed. Additionally, the prices of the products can float because of the lack of official rules, thus chances are that gamers buy a product at a unnecessary high price. Another relative safer way to buy entity games is buying them on the official websites of the games or the publishers. However, it annoys people that they have to create different accounts for varied websites to buy games they want.

As is demonstrated above, there is a vacancy of a comprehensive entity game market where gamers with different game consoles can buy absolute legal copies of the entity games they want and publishers can attract all sorts of gamers to try their games. Actually, ICE can also serve as a community besides a store.

1.3 Requirement

Access of the game store for customers and management of the company must be accomplished through a Web site. The user can access it via PC or mobile device.

The system must allow customers to search in our system for his or her favorite game and add games to shopping cart or wish list. Gamers with different game consoles can buy all sorts of physical games easily. And what is super significant is that they need to confirm what they buy are legal copies. To ensure that, the system must maintenance a gamers forum that all customers could refer to the comments from others before

buying the specific game and delivery personal opinions on a product after they purchase the game. When the game is ordered, it is delivered immediately if available in stoke, or else, the specific game is ordered to the publisher, and a compatible deadline is informed to the customer.

The system must give publishers a platform to add new games, managing game-related information, and managing orders conveniently. The functions about the information management of games are only belong to the publisher, ordinary users do not have these permissions. Besides, because publishers always look forward to a lower cut of the sale platform, with audience owning all sorts of game consoles and a lower middleman's cut, ICE will provide a suitable share of the profit to fit publishers' demands.

The system must allow a manager to generate reports on bestselling games, and on most profitable customers, as well as suggest games for buying based on past customer's interests. Furthermore the system must have the capability of predicting the sales in order to provide better decision (inventory, reordering products, etc.) with the solid foundation. When an entity game is set to be delivered, the system should be able make a decision of selecting the most economical way provide that the deadline can be met.

1.4 Assumptions

1. No pirate games are stocked or sold.
2. No second-hand games are stocked or sold.
3. No second-hand games are stocked or sold.
4. Marketing manager contract through e-mail rather than the management system.
5. Third-party logistics company contract to deliver all games.
6. Customer service have all mastered relevant knowledge and skills.
7. Customer service are all familiar with how to buy games with the system.
8. Clients pay for their orders with Alipay or Wechat wallet.

2. Initiation and Scope Definition

2.1 Requirement Management

2.1.1 Determination and Negotiation

With the development of platforms such as Steam and GOG, today, digital games seem to have become the first choice for computer gamers. However, players of other gaming machines still tend to purchase physical games for collection or other purposes, and as more and more players purchase gaming machines, this market is still booming.

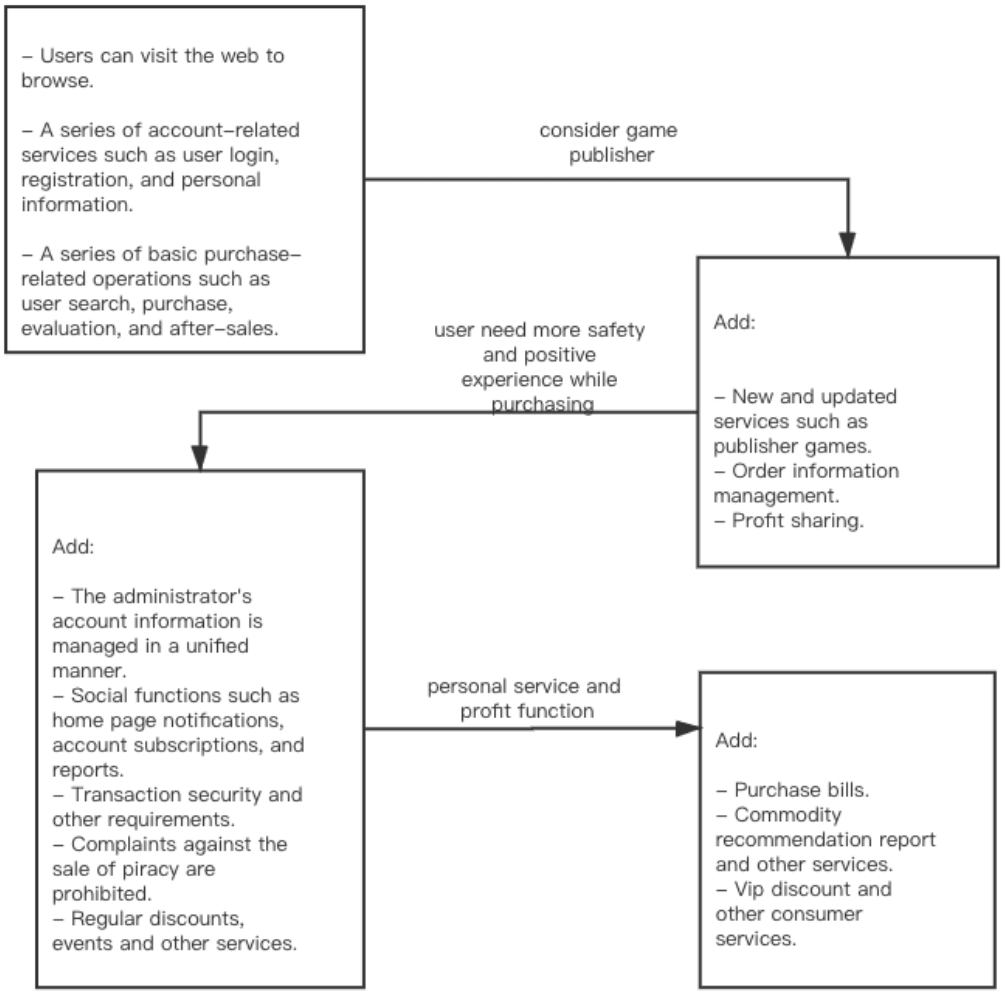
In such a large environment, we decided to create a sales platform specifically for the market, let major game manufacturers sell their products on the platform, users can purchase, evaluate, communicate and other activities on this platform Consultation and game purchase experience.

Therefore, customers must go through the website to access the game store and manage the company. Users can access it via PC or mobile device. The system must allow the customer to search his or her favorite

game in our system and add the game to the shopping cart or wish list. The system must provide a platform for publishers to add new games, manage game-related information, and easily manage orders.

2.1.2 Review and Revision

Here are the revolution of our requirements.



2.2 Feasibility Analysis

2.2.1 Functions

- Functions for users
  - The user logs in to ICE.
  - The user searches for games in ICE.
  - The user views game categories in ICE.
  - The user views game consoles in ICE.
  - The user views game publishers in ICE.
  - The user views personal homepage in ICE.
  - The user views order information in ICE.
  - The user views shopping cart information in ICE.
  - The user views game details information in ICE.

- The user purchases game in ICE.
- The user adds the game to the shopping cart in ICE.
- The user adds the game to the wish list in ICE.
- The user modifies personal information in ICE.
- The user adds delivery address in ICE.
- The user rates the order in ICE.

- Functions for publishers
  - The publisher logs in to ICE.
  - The publisher views personal information in ICE.
  - The publisher modifies personal information in ICE.
  - The publisher views game list in ICE.
  - The publisher modifies game information in ICE.
  - The publisher removes game in ICE.
  - The publisher adds game in ICE.
  - The publisher views order in ICE.
  - The publisher modifies the order information in ICE.
  - The publisher delivery the game product in ICE.

- Functions for administrators
  - The administrator logs in to ICE.
  - The administrator exits from ICE.
  - The administrator makes announcement in ICE.
  - The administrator manages users' account in ICE.
  - The administrator checks users' comments in ICE.
  - The administrator handles with report informations in ICE.
  - The administrator makes recommend information in ICE.

- Functions for analysts
  - Which entity games are better sold in the second quarter than the first quarter?
  - Which categories of entity games are the most profitable ones?
  - What is the average time between the order placed and shipped?
  - Is there any significant difference between entity games published by different publishers in terms of profitability?
  - Basic "Gamers Persona" for customers visit our website.

2.2.2 Constraints

- Customers would pay by credit card, Alipay, or WeChat. All transactions should be secured.
- Access to the system will be available through a web site via PC, mobile devices etc.
- User (Customers) can discuss any games, but ICE only provides entity games from different game companies for gamers owning different gaming consoles.

2.2.3 Features

1. Performance requirements:
   - In 95% of the cases, the response time in the general period does not exceed 1.5 seconds, and the peak period does not exceed 4 seconds.

- Searching according to the specific conditions of number and name during non-peak hours, you can get the search results within 3 seconds.
- The final estimated number of users is 10,000, the number of daily logged-in users is about 3,000, and the network bandwidth is 100M bandwidth.
- The system can satisfy 5,000 user requests at the same time and provide browsing functions for 10,000 concurrent users.

2. Security requirements:
   - Strict permission access control, after identity authentication, users can only access data within their permission range and can only perform operations within their permission range.
   - Different users have different identities and permissions. It is necessary to provide trusted authorization management services under the premise that the user's identity is true and trustworthy, to protect data from illegal / unauthorized access and tampering, and to ensure data confidentiality and integrity.
   - Can withstand general malicious attacks from the Internet. Such as virus (including Trojan horse) attacks, password guessing attacks, hacking, etc.
3. Reliability requirements:
   - There are prompts for input and data are checked to prevent abnormal data.
   - The system is robust and should be able to deal with all kinds of abnormal conditions that occur during the operation of the system, such as: human operation errors, illegal data input, and hardware device failure. The system should be able to handle it properly and avoid it properly.
4. Data confidentiality requirements:
   - Network transmission data should be encrypted. It is necessary to ensure that the data is not peeped, stolen, or tampered with during the collection, transmission, and processing. Business data needs to be encrypted during storage to ensure that it cannot be cracked.
5. Ease of use requirements:
   - 60% of users can master the use and purchase methods through the experience of other platforms and the description of the platform within 5 seconds of first seeing the platform.
6. Maintainability requirements
   - After receiving the modification request, the ordinary modification should be completed within 1 to 2 days; for the evaluation of the major demand or design modification should be completed within 1 week.
   - 90% of the bugs were modified within 1 working day, and others within 2 working days.

2.3 Scope

ICE's strategic goals include continuous growth and profitability, as well as increasing awareness and building a platform atmosphere. This project is based on the entity game market. It hopes to attract users' favor with comprehensive and exquisite games, and optimize the work of managers with concise and refined operation methods. It will improve customer performance with excellent early warning and feedback, and gradually create a user online purchase entity games is the preferred platform and provides long-term and stable services for our buyers and partners. In order to achieve this goal, it is necessary to ensure sufficient game sources, and a certain amount of overhead is allowed in the early stage to increase the popularity of the platform and the complete game purchase experience.

Specific and direct goals are:

1. Meet business needs. ICE must be able to complete the purchase service provided to users, and provide administrators with basic business functions such as income and expenditure inventory reports.
2. Improve work efficiency. ICE needs to respond to some preset scenarios and optimize the management of administrators to improve work efficiency.
3. Improve profitability. ICE needs to complete the summary of sales reports, inventory information, etc., so that administrators can adjust business strategies in time to obtain greater profits.
4. Enhance corporate brand. ICE is committed to improving the buying experience of buyers, enhancing the visibility of the platform, strengthening the construction of atmosphere, and improving profitability from the side.

3. Planning Management

3.1 Planning Activities

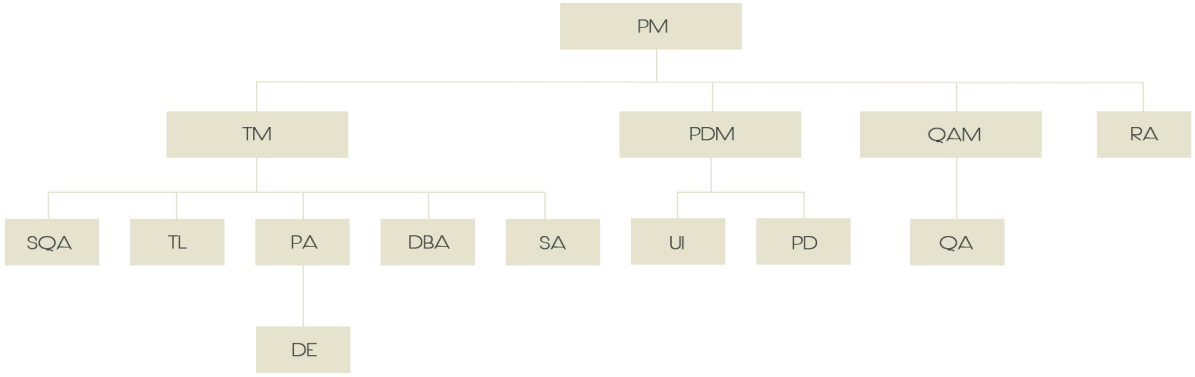
Step	Activity
0 : Select project	ICE: an entity game platform
1 : Identify project scope and objectives 1.1 Identify objectives and measures of effectiveness 1.2 Establish authority 1.3 Identify stakeholders 1.4 Modify objectives in the light of skate holder analysis 1.5 Establish methods of communication	- 1.1: Objectview - 1.2: The project authority is controlled by the project steering committee, and is particularly responsible for setting, monitoring and revising the goals. At present, this work is performed by three people in our group. - 1.3: Stakeholders: Project staff; people outside the project in the same organization, like communication personnel; persons outside the organization game purchasers, game manufacturers. - 1.4: Review and Revision - 1.5: Communication Style
2 : Identify project infrastucture 2.1 Establish relationship between project and strategic planning 2.2 Identify installation standards and procedures 2.3 Identify project team organization	- 2.1: Need to decide in what order to execute these projects, need to establish a framework to accommodate the new system, such as hardware and software standards. - 2.2: There should be standards for change control and configuration management; there may be provisions for quality checks at every point in the project life cycle; there should also be a measurement procedure to control the data that must be collected at each stage; the project manager should be aware of any relevant projects Planning and control standards. - 2.3: The person in charge of a large project may need to control the organizational structure of the project team. While our team has very simple Team Structure.
3 : Analyse project characteristics 3.1 Objective- or product-driven 3.2 Analyze other project characteristics 3.3 Identify high-level project risk 3.4 Take into account user requirements concerning implementation 3.5 Select general life-cycle approach 3.6 Review overall resource estimates	- 3.1: Mostly product-driven. - 3.2: Features - 3.3: Assess the risk level of all projects, make risk prioritization and focus on high-risk projects - 3.4: Customers sometimes have their own regulatory requirements. Some of them in Requirement. - 3.5 Development method and life cycle method：Scrum Process Model - 3.6 Identifying Resource Requirements for all projects, and consider the project's personnel allocation and other issues

4 : Identify project products and activities 4.1 Identify and describe project products 4.2 Document generic product flows 4.3 Recognize product instances 4.4 Produce ideal activity network 4.5 Modify ideal to take into account need for stages and checkpoints	- 4.1: Identifying all the items to be created by the project helps to ensure that all activities that need to be performed have been considered. Including deliverables, intermediate products, etc., including both technical products and products related to project management and quality. These products have their own hierarchical structure, which can be represented by Product Breakdown Structure. - 4.2: Determine the order in which products are created or used through the Product Flow Diagram - 4.3: When the same common PFD fragment is related to multiple instances of a particular type of product, tries to identify each instance. - 4.4: The ideal activity web with sufficient resources. - 4.5: Introduce checkpoint activity to modify activity network.
5 : Estimate effort for each activity 5.1 Carry out bottom-up estimates 5.2 Revise plan to create controllable activities	- 5.1: Estimate the amount of staff work required for each activity, possible time consumption, and required non-human resources with Network Plan - 5.2: Activities that take a long time to split, activities that take a short time to merge. Set the time span of the activity to be the same as the reporting period used to monitor and control the project.
6 : Identify out bottom-up estimates 6.1 Identify and quantify activity-based risks 6.2 Plan risk reduction and contingency measures where appropriate 6.3 Adjust plans and estimates to take account of risks	- 6.1: Review each activity and estimate their risk of success. - 6.2: Some identified risks can be avoided or at least reduced. If there is a risk, the emergency plan specifies the actions to be taken. - 6.3: May change the plan, or add some new activities to reduce risk.
7 : Allocate resources 7.1 Identify and allocate resources 7.2 Revise plans and estimates to take account of resource constraints	- 7.1: Record the type of employees required for each activity, identify the employees available for the project, and temporarily assign to these projects. - 7.2: Establish priorities for tasks to ensure the completion of key tasks; ensure the full work and high utilization rate of available personnel, presented using Gantt charts.
8 : Review/ publicize plan	- 8.1: When each task is completed, determine whether the task can be ended by determining good quality criteria. - 8.2: Document the plan carefully so that the various departments of the project understand the plan and agree to commit to the plan.
9/10 : Execute plan/ lower levels of planning & May require the reiteration of lower level planning	Once the project starts, it is necessary to make a more detailed plan for each phase that is about to begin, and let go of the detailed planning for the subsequent phases.

3.2 Project Organization

3.2.1 Team Structure

There is the organization of our development team. Each manager is responsible for his or her department, report their work and progress to the project manager. The project manager will charge the whole our and make sure the project is under control. The product manager will join in the development process to guarantee that the project meets the requirements and take charge of the later popularizing of the bookstore website.



3.2.2 Roles, Responsibilities and Authority

Every one in our team has a specific responsibility, and the following table shows each of the member's respective responsibility:

Roles	Name	Responsibility
Project Manager	Zhe Zhang	Take full responsibility for the entire project, monitor development progress, make decision on risk control and resource provision, and ensure software quality
Requirements Analyst	Di Bu	Responsible for communicating requirements with customers, assisting project manager to control and follow up requirement change
Product Architect	Kaixin Chen, Sion	Responsible for the design of the software part of the system structure and model, develop the software development plan, determine the software technology selection
Product Design Manager	Rudi	Responsible for monitoring project functional requirements and product design, as well as product functional design and interaction design
Product Designer	Auston	Responsible for the collection and analysis of needs, product design and interaction design
User Interface Designer	Marica	Responsible for prototype design and user experience design
Technical Manager	Sakura	Responsible for system function module coding implementation and correction test feedback product defects
Team Leader	Iwan, Eren	Responsible for management of the development team and monitoring the progress of the project
Development Engineer	Rina, Barkley, Gaia, Tyler	Responsible for system function module coding implementation and correction test feedback product defects

Quality Assurance Manager	Ozzy	Responsible for test plan, and the whole quality assurance activities of the project
Quality Assurance	Cindy, Lily, Zoe	Responsible for test cases design, test execution and evaluation of the test execution process, as well as evaluate test results and document defects found
System Administrator	Lie	Responsible for deployment of software products, completion of project related system engineering work, and customer technical support
Software Quality Assurance	Lucia	Responsible for supervising the process planning and implementation of the project, checking the products produced by the project, and checking the conformity of the project development process
Database Administrator	Liv	Responsible for designing and constructing database system and optimizing database perfo

3.2.3 Communication Style

Communicate in the team through the following collaborative communication methods:

1. Formal, non-personal methods such as software engineering documents and project products, memos, schedules and project control tools, change requests, etc .;
2. Formal, person-to-person communication, focusing on quality assurance, such as status review meetings, design, and code inspection;
3. Informal, person-to-person communication, such as group meetings to exchange information and solve problems;
4. E-mail, mainly to communicate with people outside the project such as instructors and certain technical personnel

3.3 Software Project Planning

3.3.1 WBS

According to the software engineering methodology, we divide the overall system development into six main processes: requirements analysis, outline design, detailed design, coding, testing, and deployment, and then decompose the six major processes respectively.

For demand analysis, it is mainly divided into four parts: demand collection, demand communication, demand analysis and demand confirmation.

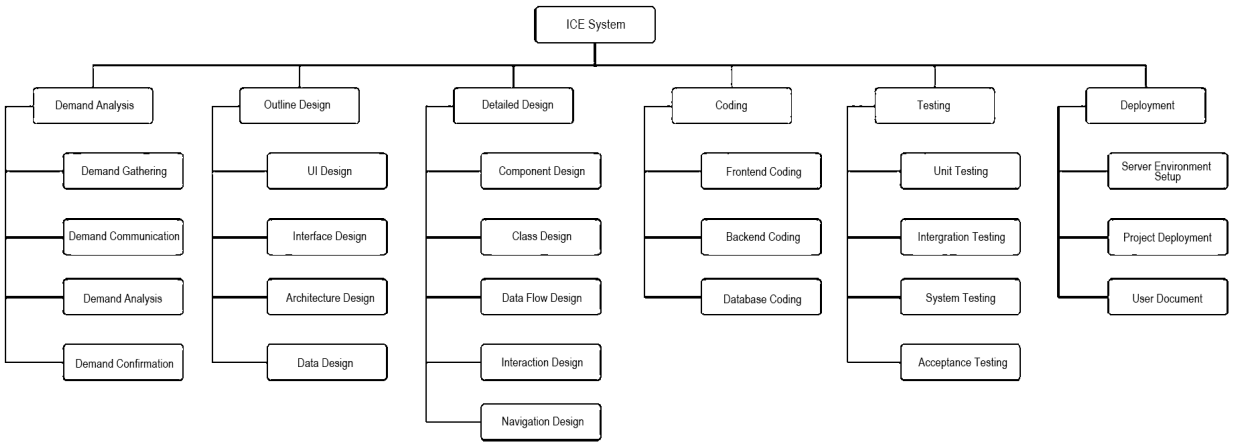
In the outline design stage, we mainly complete the interface design, interface design, architecture design, database design and other contents.

The detailed design stage is subdivided into module design, class design, data flow design, interaction design, and navigation design.

The coding is split into front-end coding, back-end coding and database implementation.

Testing is divided into unit testing, integration testing, system testing and acceptance testing based on software testing theory.

Finally, we configure the server environment for the deployment phase, project deployment and user documentation.

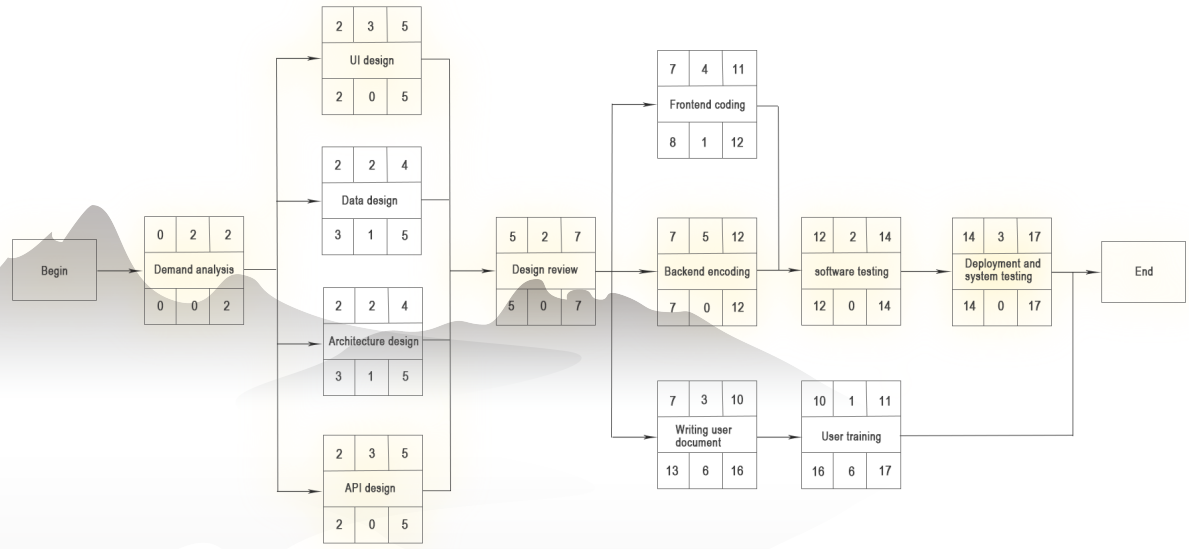


3.3.2 Network Plan & Critical Path

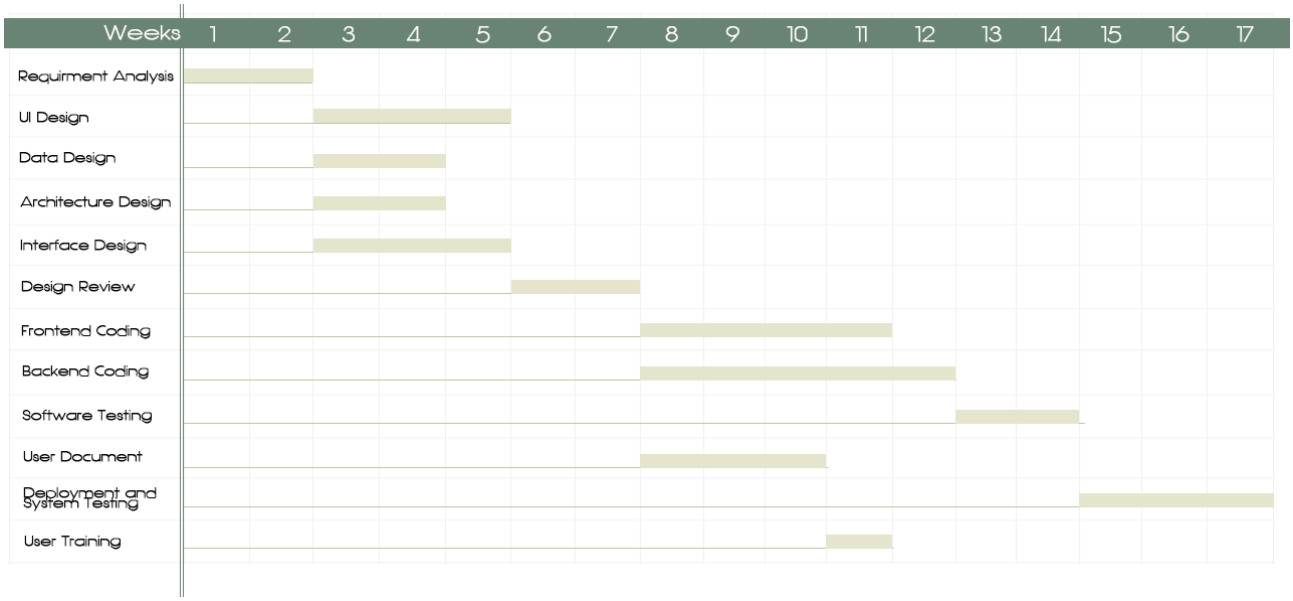
The project uses an activity-based approach to identify activities, divides the project into the main life cycle stages, considers the activities of each stage and its activity cycle separately, and analyzes the pre-activities of individual activities to obtain the following activity list :

No	Activity	Estimated Duration (Weeks)	Depends on
A	Requirment Analysis	2	
B	UI Design	3	A
C	Data Design	2	A
D	Architecture Design	2	A
E	Interface Design	3	A
F	Design Review	2	B, C, D, E
G	Frontend Coding	4	F
H	Backend Coding	5	F
I	Software Testing	2	G, H
J	User Document	3	F
K	Deployment and System Testing	3	I
L	User Training	1	J

Network plan is like below, with the yellow background highlights the **Critical Path** :



Relative Gantt chart:



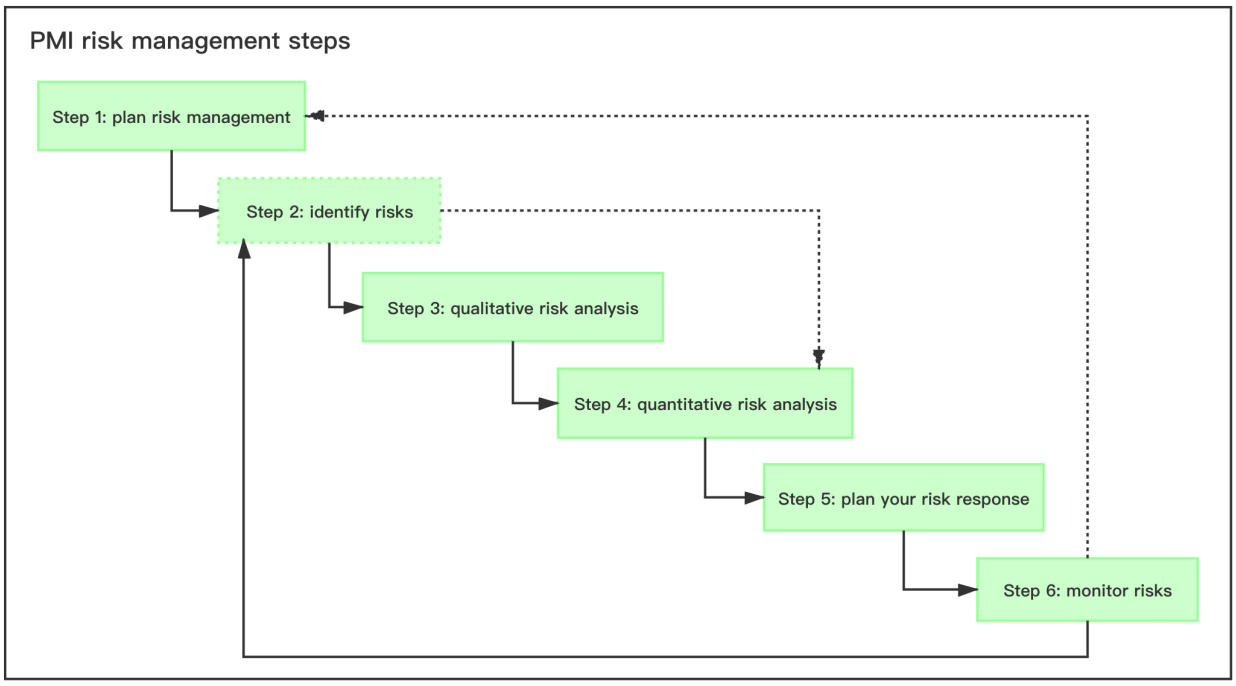
4. Risk Management

4.1 Risk Management Plan

According to Project Management Institute(**PMI**) risk management steps, our group make risk management plan firstly. And then we identify risks from 4 parts, qualitative risk analysis, quantitative risk analysis. Every effort will be made to proactively identify risks ahead of time in order to implement a mitigation strategy from the project's onset.

We manage risks according to priorities, the most likely and highest impact risks are added to the project schedule to ensure that the assigned risk manager take the necessary steps to implement the mitigation response at the appropriate time during the schedule.

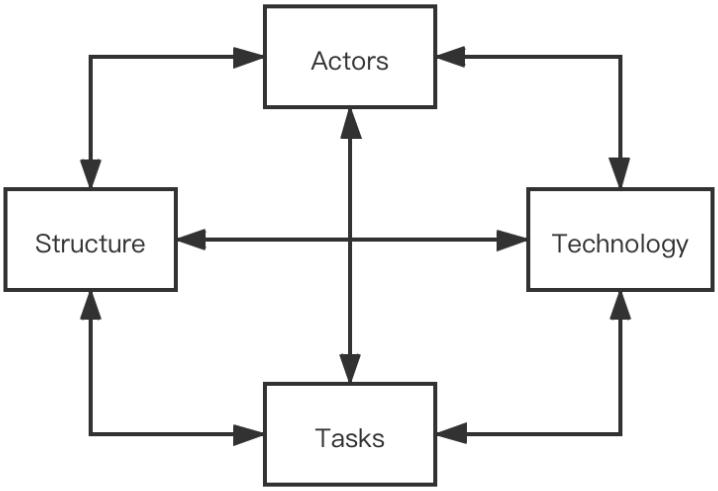
Upon the completion of the project, during the closing process, the project manager will analysis each risk as well as the risk management process. Based on this analysis, the project manager will identify any improvements that could be made to the risk management process for future projects



4.2 Categories of Risk

Project risks are those that could prevent the achievement of the objectives given to the project manager & team.

In the Software Project and Process Management Course Project "**Online Entity Game Store ——ICE**", we reference to "Lyytinen-Mathiassen-Ropponen" risk framework and we divided the risk factors into four parts.



- **Actors**: main participants are the members of the group, among which the typical risks are mention below
    - changing of the staff
    - poor communication within the team leads to loss of valuable information of the project

- **Technology**: technology is the specific knowledge and tools using in the project, typical risk are mention below
    - tools used in the project development and implementation do not meet the project requirements
    - technology used in the project has defects, such as it is not suitable for the project or the team members are not familiar with the technology

- **Structure**: structure contains the planning structure, project structure, management structure and so on, typical risk are mention below
    - project management structure is not clear, which leads to the delay of each team member's timely positioning of their own work

- **Tasks**: tasks involving specific activities of the project, typical risk are mention below
    - risk of requirement change
    - complexity of integration among various project components will delay the progress of the project,
    - quality and performance risk of the ICE system

4.3 Risk Identification

In the risk identification of this project, we adopted the risk identification method combining **Checklist method** and **Brainstorming method**, referred to checklists models such as **lyytinen model** and **Barry Boehm model**, and established our own checklists as follows:

No	Risk	Risk reduction techniques
R1	Team members did not communicate well	Agree on the way of communication; Regular communication of work progress; The minutes of the meeting
R2	Technology adopted is flawed	Use familiar technology; Specially-assigned person to study the new technology needed
R3	Project management structure is not clear	Formulate the management structure; Special personnel to maintain the management structure
R4	Code integration issues	Convention code specification; Incremental development and integration
R5	Personnel changes	More than one person participated in the core work of the project in order to familiar with the project process
R6	Implementation tools do not meet the requirements	Early implementation of the sources of tools; Look for alternative tools
R7	Change to requirements specification during coding	Stringent change control procedures High change threshold Incremental development(deferring changes) Agree on the requirement change control process in writing and record the change request
R8	System quality and performance risks	More exchange of work results; Inspection and review; Using performance test

4.4 Risk Analysis, Assessment and Prioritization

In this project, we define the risk probability using score from 1 to 10, and we divide them into 4 levels from extra-high to extra-low. **Risk Probability Level Table**, **Risk Impact Level Table**, and **Risk Level Decision Table** are shown below:

Risk probability level			
score	probability range	level	description
10	90% ~ 1	High	Almost certainly
9	80% ~ 90%		
8	70% ~ 80%	Significant	Very likely
7	60% ~ 70%		
6	50% ~ 60%	Moderate	Likely
5	40% ~ 50%		
4	30% ~ 40%		
3	20% ~ 30%	Low	Almost impossible
2	10% ~ 20%		
1	0 ~ 10%		

Risk impact level		
score	level	description
10	High	Lead to the failure
9		
8	Significant	Create a big negative impact
7		
6	Moderate	Some influences on the project
5		
4		
3	Low	The negative impact is almost negligible
2		
1		

Risk level decision		
level(cardinal)	level(ordinal)	result
64~100	High	Can't accept
36~64	Significant	Need to make a decision
9~36	Moderate	Need to be reviewed
0~9	Low	Can be neglected

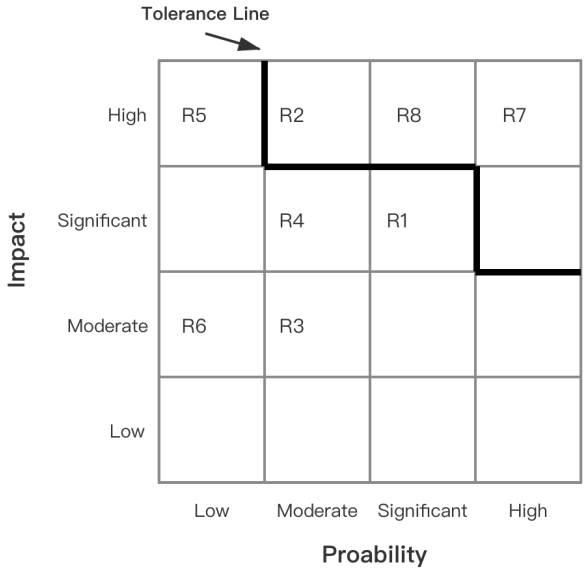
After defining the levels, we analyzed the potential risks in the project from the four aspects mentioned above.

We simulate the risk exposure during our brainstorming, and we use **Barry Boehm's method** for risk exposure assessment. And we use **risk exposure formula** for calculating.

Ref	Hazard	Likelihood	Impact	Risk
R1	Team members did not communicate well	5	5	25
R2	Technology adopted is flawed	3	10	30
R3	Project management structure is not clear	3	4	12
R4	Code integration issues	4	5	20
R5	Personnel changes	2	8	16
R6	Implementation tools do not meet the requirements	2	4	8

R7	Change to requirements specification during coding	8	8	64
R8	System quality and performance risks	6	7	42

And the **probability impact matrix** is as follow:



4.5 Risk Prevention

For risk R1, the group shall agree on the communication method at the beginning of the project and regularly communicate the schedule of work progress. If necessary, the meeting minutes shall be taken to minimize the possibility of risk R1.

For risk R2, when encountering a risk bottleneck, turn the head and select experienced or familiar technologies. At the same time, a team member can be allowed to study and learn new technologies to pave the way for the implementation of the project.

For risk R3, formulate the management structure in advance, and let a team member manage and maintain the management structure;

For risk R4, to start the project code writing, agree on the development environment and code specification, annotation specification, etc., in the development process, incremental development and intergration to reduce the final workload;

For risk R5, everyone is involved in the core work of the project to ensure that the change of personnel will not make the project unable to proceed normally;

For risk R6, identify and implement the source of the tools at the beginning of the project, find the necessary alternative tools, and implement the tools in place before the tools need to be used;

For risk R7, at the beginning of the project construction, agree on the requirement change control process with the user, record and archive the user's requirement change application;

For risk R8, in the regular meeting of the team, the work results are exchanged, the results are checked and reviewed, and the performance test is carried out. After the performance test meets the indicators, the follow-up work is carried out.

4.6 Risk Actions

According to the formula of **Risk reduction leverage(RRL)**

$$\text{RRL} = (RE_{before} - RE_{after})/\text{cost of risk reduction}$$

And we picked out and examined what appear to be the most threatening risks to the project, creating and maintaining our findings in a **Risk Register**

RISK RECORD					
Risk id	R1	**Risk title**	Team members did not communicate well		
Owner	Zhe Zhang	**Data raised**	2019/4/1	**Status**	first update
Risk Description: Due to the low enthusiasm or participation among members, the communication is not in place.					
Impact Description: Project progress is slow, which drags down the progress of the project.					
Recommended risk mitigation: Agree on the way of communication at the beginning of the project and regularly communicate the schedule of work progress. If necessary, take minutes of the meeting to reduce the possibility of it.					
Probability/impact values: 5 / 5					
	Probability	**Impact**			
		Cost	**Duratioin**	**Quality**	
Pre-mitigation	25	2	7	5	
Post-mitigation	10	1	3	3	
Incident/action history					
Date	**Incident/action**		**Actor**	**Outcome/comment**	
2019/4/5	Agree that every Saturday night is the regular meeting time, need to summarize the work of the week		All team members	If necessary, a written record should be made	

RISK RECORD					
Risk id	R2	**Risk title**	Technology adopted is flawed		
Owner	Kaixin Chen	**Data raised**	2019/4/3	**Status**	second update
Risk Description: There are technical flaws in the programming techniques or frameworks used during the project.					
Impact Description: The project is in trouble and may not be able to proceed, requiring an alternative approach.					
Recommended risk mitigation: When it comes to the risk bottleneck, turn the head and select the experienced or familiar technology. At the same time, a team member can study and learn the new technology, which lays the foundation for the project implementation.					
Probability/impact values: 3 / 10					
	Probability	**Impact**			
		Cost	**Duratioin**	**Quality**	
Pre-mitigation	30	6	6	7	
Post-mitigation	8	2	3	3	
Incident/action history					
Date	**Incident/action**		**Actor**	**Outcome/comment**	
2019/4/8	Using SpringBoot framework for back-end; Using origin framework for front-end		Di Bu	/	

RISK RECORD					
Risk id	R3	**Risk title**	Project management structure is not clear		
Owner	Di Bu	**Data raised**	2019/4/9	**Status**	first update

Risk Description:
There is no clear understanding of the overall management structure of the project and no clear explanation.

Impact Description:
Work chaos, slow progress.

Recommended risk mitigation:
Formulate the management structure in advance and let a team member manage and maintain the management structure.

Probability/impact values: 3 / 4

	Probability	**Impact**		
		Cost	**Duratioin**	**Quality**
Pre-mitigation	12	3	3	3
Post-mitigation	5	1	2	1

Incident/action history

Date	**Incident/action**	**Actor**	**Outcome/comment**
2019/4/14	Make sure project management structure and maintance and develop by Kaixin Chen	All team member	/

RISK RECORD					
Risk id	R4	**Risk title**	Code integration issues		
Owner	Zhe Zhang	**Data raised**	2019/4/9	**Status**	third update

Risk Description:
Because of the differences between the code module division and the coding habits of the members, the final code integration lags behind.

Impact Description:
The project process is controlled and a lot of time is spent dealing with coding differences.

Recommended risk mitigation:
To start the project code before the writing of a good agreement on the development environment and code specification, annotation specification, etc., the development process, incremental development and integration, reduce the final workload.

Probability/impact values: 4 / 5

	Probability	**Impact**		
		Cost	**Duratioin**	**Quality**
Pre-mitigation	8	3	3	2
Post-mitigation	2	1	1	1

Incident/action history

Date	**Incident/action**	**Actor**	**Outcome/comment**
2019/4/10 2019/4/15	Integrate currently code for personal responsibility	All team member	Decrease work for later code integration

RISK RECORD					
Risk id	R5	**Risk title**	Personnel changes		
Owner	Zhe Zhang	**Data raised**	2019/4/9	**Status**	third update

Risk Description:
A condition in which a group member cannot continue to work for some reason, resulting in the absence of a member.

Impact Description:
It can affect the project schedule and delay the progress, and it can cause the project to be unable to continue or even die.

Recommended risk mitigation:
Get everyone involved in the core of the project and make sure that changes in people don't derail the project.

Probability/impact values: 2 / 8

	Probability	Impact		
		Cost	Duratioin	Quality
Pre-mitigation	16	4	4	4
Post-mitigation	/	/	/	/

Incident/action history

Date	Incident/action	Actor	Outcome/comment
/	/	/	/

RISK RECORD					
Risk id	R6	**Risk title**	Implementation tools do not meet the requirements		
Owner	Kaixin Chen	**Data raised**	2019/4/2	**Status**	first update

Risk Description:
Whether the management tools, development tools and test tools that must be used in the project development and implementation process can be in place in time and whether the tool versions in place meet the project requirements.

Impact Description:
Additional time will be spent reselecting and implementing tools to slow down the project.

Recommended risk mitigation:
Identify and implement the source of the tools at the start of the project, while finding the necessary alternative tools, and implement the tools in place before they need to be used.

Probability/impact values: 2 / 4

	Probability	Impact		
		Cost	Duratioin	Quality
Pre-mitigation	16	5	5	5
Post-mitigation	10	3	1	3

Incident/action history

Date	Incident/action	Actor	Outcome/comment
2019/4/16	IDE for front-end we choose JetBrains WebStorm; IDE for back-end we choose JetBrains IntelliJ IDEA; Database we choose MySQL Workbench	Di Bu	Decrease work later

RISK RECORD					
Risk id	R7	**Risk title**	Change to requirements specification during coding		
Owner	Di Bu	**Data raised**	2019/3/21	**Status**	second update

Risk Description:
Users may frequently propose new requirements or modify existing requirements.

Impact Description:
At the very least, new work needs to be added, and at the very least, the whole project may have to be overhauled.

Recommended risk mitigation:
Identify and implement the source of the tools at the start of the project, while finding the necessary alternative tools, and implement the tools in place before they need to be used.

Probability/impact values: 8 / 8

	Probability	**Impact**		
		Cost	**Duratioin**	**Quality**
Pre-mitigation	64	8	7	7
Post-mitigation	8	2	2	2

Incident/action history

Date	**Incident/action**	**Actor**	**Outcome/comment**
2019/4/1 2019/4/10	Requirement changing	Zhe Zhang	/

RISK RECORD					
Risk id	R8	**Risk title**	System quality and performance risks		
Owner	Zhe Zhang	**Data raised**	2019/4/9	**Status**	none update

Risk Description:
Users will have a high demand for software quality and the software system will have a high demand for performance.

Impact Description:
/

Recommended risk mitigation:
In the regular meeting of the team, the work results are exchanged, the results are checked and reviewed, and the performance test is carried out. After the performance test meets the indicators, the follow-up work is carried out.

Probability/impact values: 6 / 7

	Probability	**Impact**		
		Cost	**Duratioin**	**Quality**
Pre-mitigation	42	7	8	8
Post-mitigation	/	/	/	/

Incident/action history

Date	**Incident/action**	**Actor**	**Outcome/comment**
/	/	/	/

4.7 Risk Evaluation

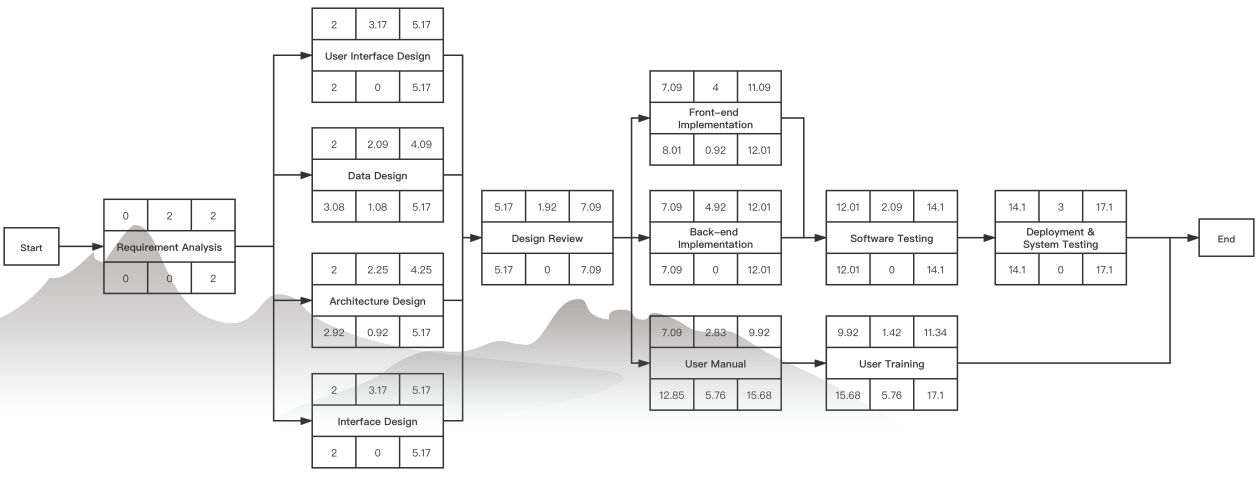
We use **PERT(Program Evaluation and Review Technique)** for risk evaluation. PERT was developed to deal with the uncertainly surrounding estimates of task durations.

First we divided the global task into sub-task, and estimating for three times, **Most likely time(a)**, **Optimistic time(m)** and **Pessimistic time(b)**, then using formula to calculating **Expected Duration** and **Activity Standard Deviations**

4.7.1 PERT Activities Schedule

Serial Numver	Activity	Optimistic Time	Most Likely Time	Pessimistic time	Expected Duration	Standard Deviations
A	Requirement Analysis	1	2	3	2	0.33
B	User Interface Design	2	3	5	3.17	0.5
C	Data Design	1.5	2	3	2.09	0.25
D	Architecture Design	1.5	2	4	2.25	0.42
E	Interface Design	2	3	5	3.17	0.5
F	Design Review	1	2	2.5	1.92	0.25
G	Front-end Implementation	3	4	5	4	0.33
H	Back-end Implementation	3.5	5	6	4.92	0.42
I	Software Testing	1.5	2	3	2.09	0.25
J	User Manual	1	3	4	2.83	0.5
K	Deployment & System Testing	2	3	4	3	0.33
L	User Training	0.5	1	2	1.42	0.25

4.7.2 Project Cycle Activities Network Diagram



5. Resource Allocation

5.1 Identifying Resource Requirements

We use **activity network analysis techniques** to plan when activities should take place in **3.3.2**. They was calculated as a time span during which an activity should take place - bounded by the earliest start and latest finish dates. We also use **PERT technique** in **4.7.1** forecasting a range of expected dates by which activities would be completed. And after further consideration, we match the activity plan to available resources and assess the efficacy of changing the plan to fit the resources in some aspects.

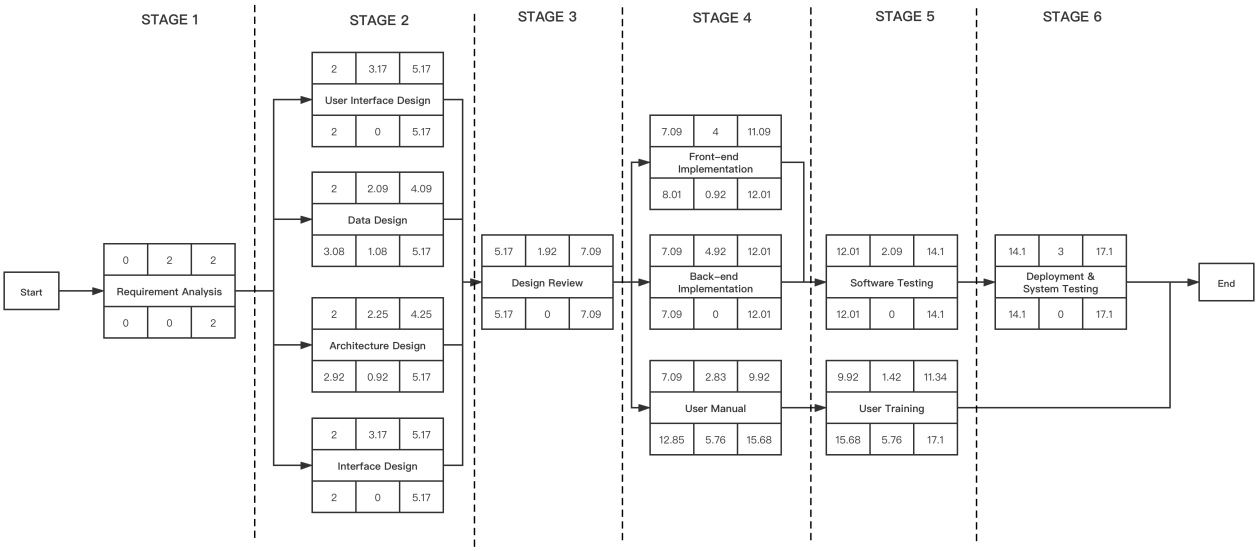
The allocation of resources to activities will lead us to review and modify the ideal activity plan. And we revise state and project completion dates after resource allocation.

Firstly, we product a resource allocation plan to list the resources that will be required along with the expected level of demand. Some of them should considering each activity in turn and identifying the resources required, and there will also be resources required that are not activity specific, but are part of the project's infrastructure or required to support other resources. The **table of identified resource requirements** are shown below:

Category	**Specification**	**Qualitative/Quantitative**
Labour	The main items in this category will be members of the development project team such as the project manager, systems analysts and software developers. Equally important will be the quality assurance team and other support staff and any employees of the client organization who might be required to undertake to participate in specific activities.	- PM - Requirement Analyst - Product Architect - PD Manager & PD - UI Designer - Technical Manager - Team Leaser - Development Engineer - QA - Administrators
Equipment	Obvious items will include workstations and other computing and office equipment. Also basic equipment such as desks and chairs are also should be considered.	- Laptop * n - Work table * n - Work chairs * n - Display * n - A number of hard disks
Materials	Items that are consumed, rather than equipment that is used. They are of little consequence in most software projects but can be important for some software that is to be widely distributed.	- floppy disk (our project don't need)
Space	For projects that are undertaken with existing staff, space is normally readily available.	- most time work online, need a basic meeting room
Services	Procurement of specialist services - development of a wide area distributed system.	- some specific knowledge about entity game manufactures

Time	The resource that is being offset against the other primary resources - project timescales ca sometimes be reduced by increasing other resources and will almost certainly be extended if they are unexpectedly reduced.	- according to schedule is about 17weeks, a floating number
Money	Secondary resource - it is used to buy other resources and will be consumed as other resources are used. It is similar to other resources in that it is available at a cost - in this case interest charges.	- money for employee people - money for purchase equipment and material - money to rent space

The stage of our ICE project is shown below:



Resource Requirement List (mainly take Labour into consideration)

Stage	Activities	Resources	Time	Amount	Appendix
1	ALL Requirement Analysis	PM Workstation Senior analyst	104F/T	34	Check software availability
2	ALL User Interface Design Data Design Architecture Design Interface Design	WorkStation Analyst/Designer Analyst/Designer Analyst/Designer	34F/T — 20F/T 15F/T 25F/T 15F/T	3	One per person is ideal
3	ALL Design Review	Workstation Senior analyst	2F/T	2	May use Analyst/Designer
4	ALL Font-end Implementation Back-end Implementation User Manual	WorkStation Analyst/Designer Analyst/Designer Analyst/Designer	2F/T — 7F/T 6F/T 4F/T 4F/T	3	The same as Stage 2

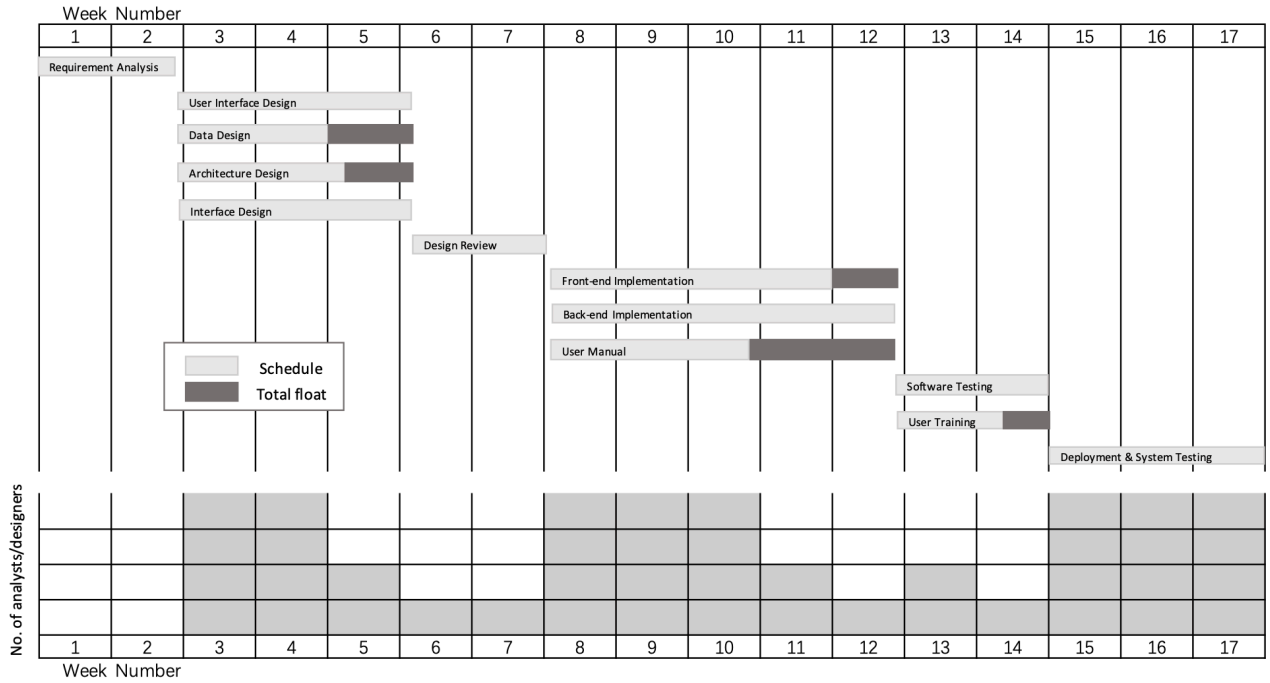
5	ALL Software Testing User Training	WorkStation Office place Programmer Programmer Programmer	— 30F/T 28F/T 15F/T 25F/T	4	One per programmer
6	ALL Deployment & System Testing	Full collection access Analyst/Designer	— 6F/T		The full system test took about 20 hours

5.2 Scheduling Resources

- **Activity schedule**: The activity schedule indicates the planned start and completion dates for each activity.
- **Resource schedule**: The resource schedule shows the dates on which each resource will be required and the level of that requirement.
- **Cost schedule**: The cost schedule shows the planned cumulative expenditure incurred by the use of resources over time.

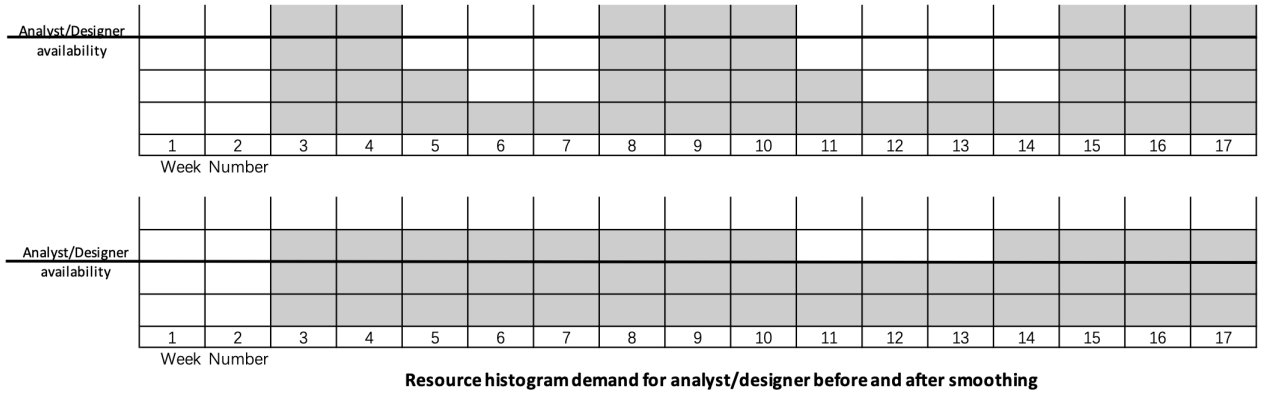
Secondly, map the resource requirements produced in the first step on to the activity plan to assess the distribution of resources over the duration of the project.

Using **Bar Chart** to produce a **resource histogram** for resource(mainly take Labour into consideration)



Each activity has been scheduled to start at its earliest start date - a sensible initial strategy, and we also wish to save any float to allow for contingencies. Earliest start date scheduling frequently creates resource histograms that start with a peak and then tail off.

By adjusting the start date of some activities and splitting other, our resource histogram can, subject to constraints such as precedence requirement, be smoothed to contain resource demand at available levels. So we smooth the resource histogram demand of analyst/designer ideally, update **smoothing resource histogram chart** is shown below:



Resource histogram demand for analyst/designer before and after smoothing

But we don't take resource conflict into conversation, therefore, we should prioritize activities so that resources can be allocated to competing activities in some ration order. The priority must almost always be to allocate resources to critical path activities and the to those activities that are most likely to affect others. In that way, lower-priority activities are made to fit around the more critical, already scheduled activities.

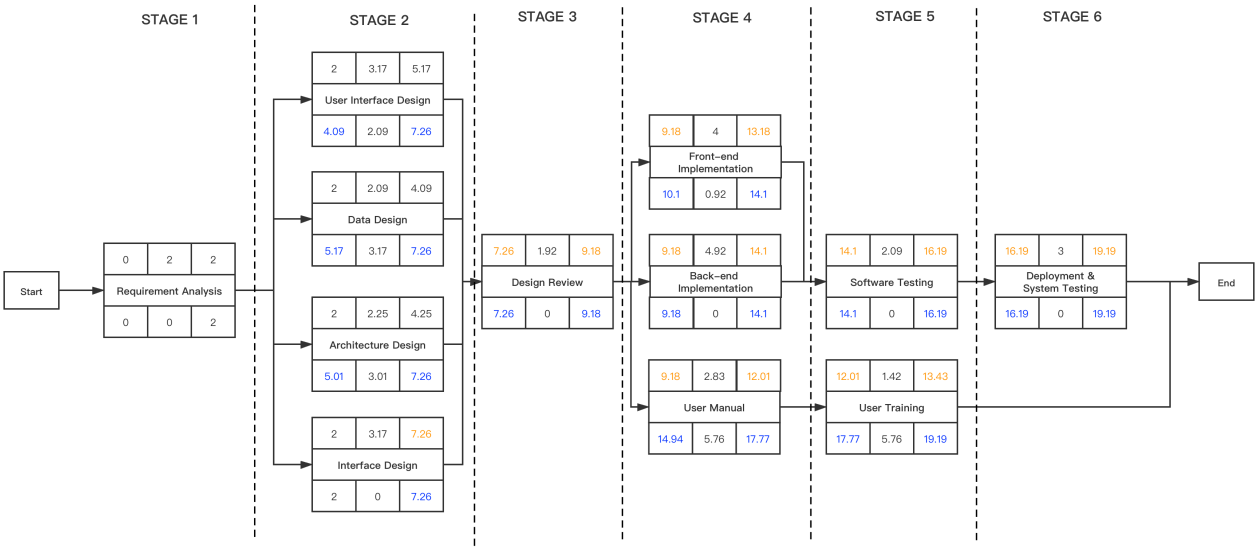
We use both **Total float priority** to smallest float has the highest priority and **Ordered list priority** according to the criteria.

Total float priority: ordered according to the total float, those with the smallest total float having the highest priority. In the simplest application of this method, activities are allocated resources in ascending order of total float. However, as scheduling proceeds, activities will be delayed and total floats will be reduced. It is therefore desirable to recalculate floats each time an activity is delayed.

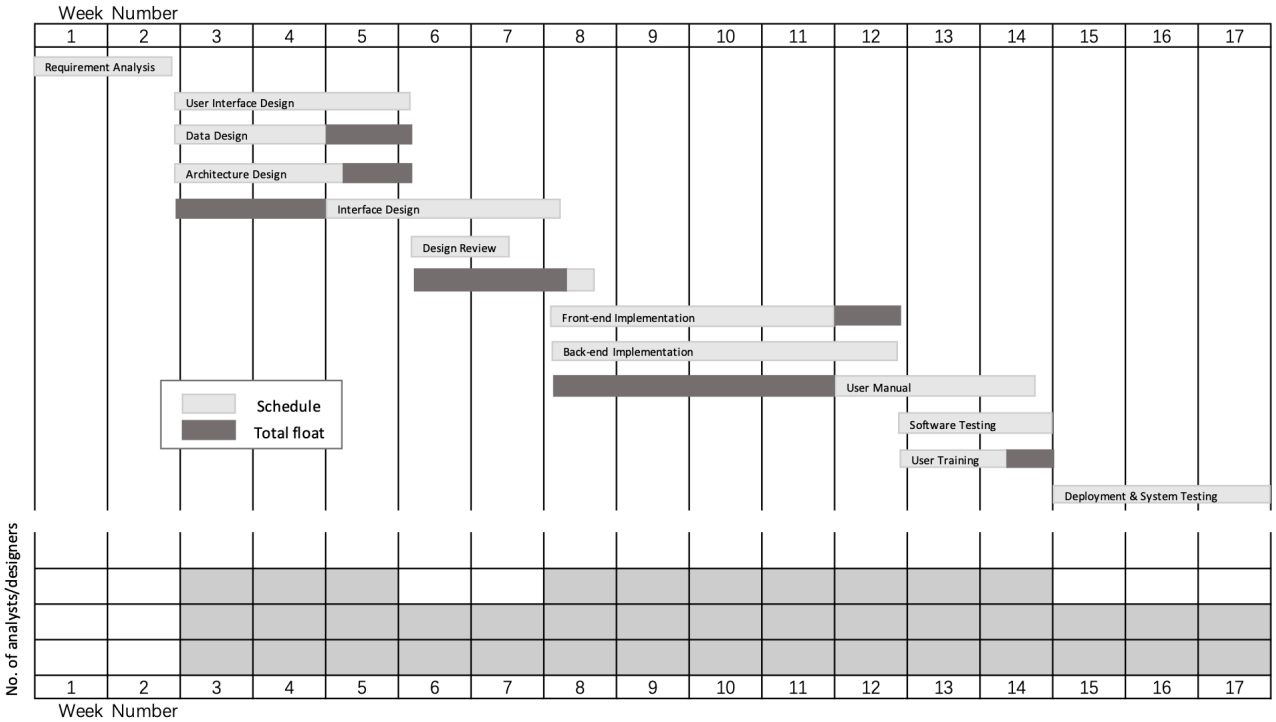
Ordered list priority: activities that can proceed at the same time are ordered according to a set of simple criteria.

- shortest critical activity
- critical activities
- shortest non-critical activity
- non-critical activity with least float
- non-critical activities

Project cycle Activity Network Diagram under Total float priority and Ordered list priority is shown below:

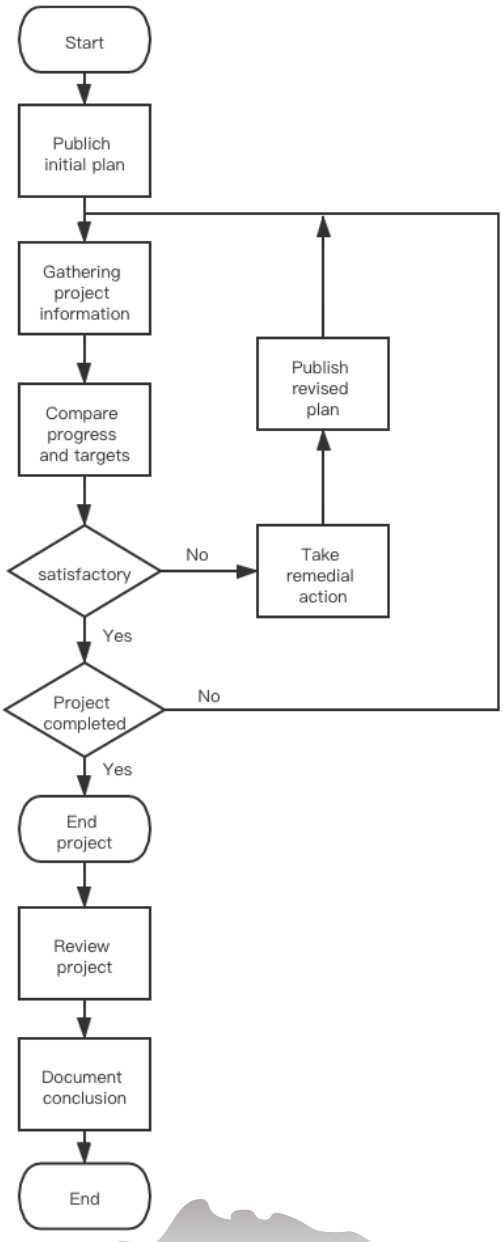


Bar Chart and **resource histogram** for resource under labour resource constraint are shown below:



6. Monitor & Control

6.1 Project control framework



We monitor the progress of the project, compare the difference between the actual progress and the plan, we modify the plan so that the project can return to the desired track. The product manager in charge of the business pays attention to the progress of the project every day, and the project manager conducts a weekly project report. Ask team members who work harder to be effective, or allocate additional resources to tune resources on non-critical paths to critical paths. Of course, other members in the team also need to give their reports in different ways and frequencies.

6.2 Data Collection

Every one should gather the tasks which are partially completed. In the process of collecting data, there are two methods: **partial completion report** and **risk report**. Considering that the partial completion of the report will make employees less focused and may cause delays, the task of making the partial completion report is left to the product manager for production and analysis. Each employee confirms the risk report weekly.

The two reports are as follows. Because there is no actual process reference, the textbook has been appropriately modified to the sample report.

Time Sheet

Staff Zhezhang Week ending 12/4/2020

Rechargeable hours

Project	Activity code	Description	Hours this week	% complete	Scheduled completion	Estimated completion
P06	A223	Code mod A2	13	50	20/4/2020	20/4/2020
P25	D347	Document take-on	6	10	5/5/2020	3/5/2020

Total recharged hours	19

Non-rechargeable hours

Code	Description	Hours this week	Comment and authorization
Z14	work for a emergency work	15	Authorized by RB

Total non-rechargeable hours	15

Activity Assessment Sheet

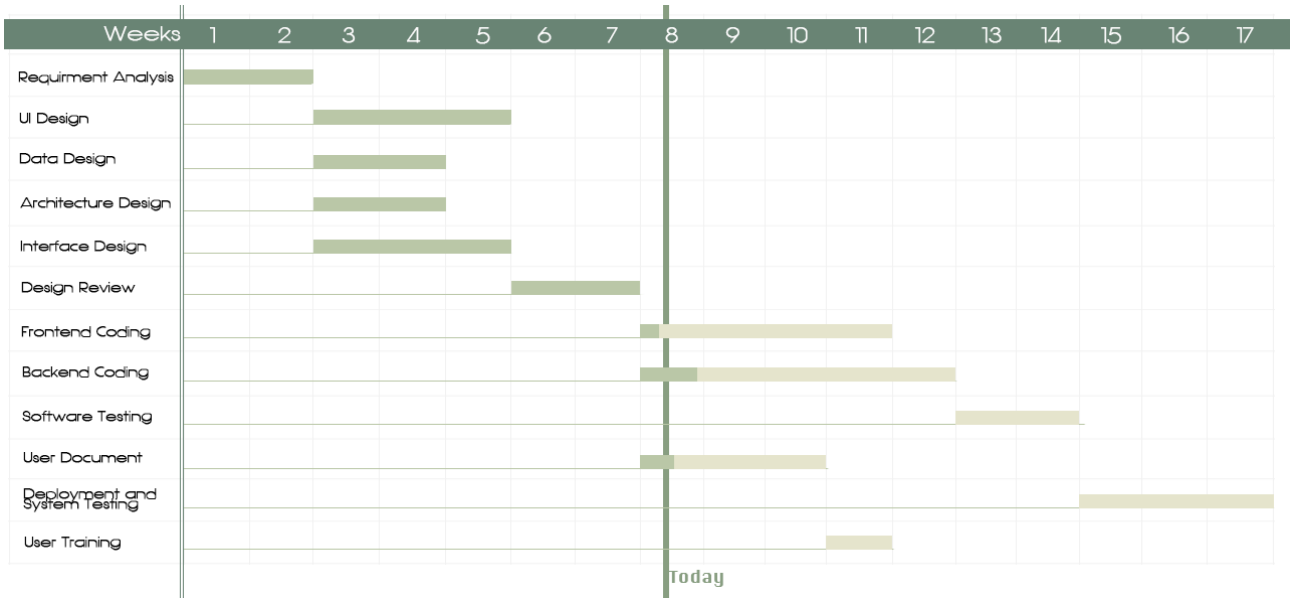
Staff Budidi

Ref: G/3 Activity: Frontend Coding Module 3

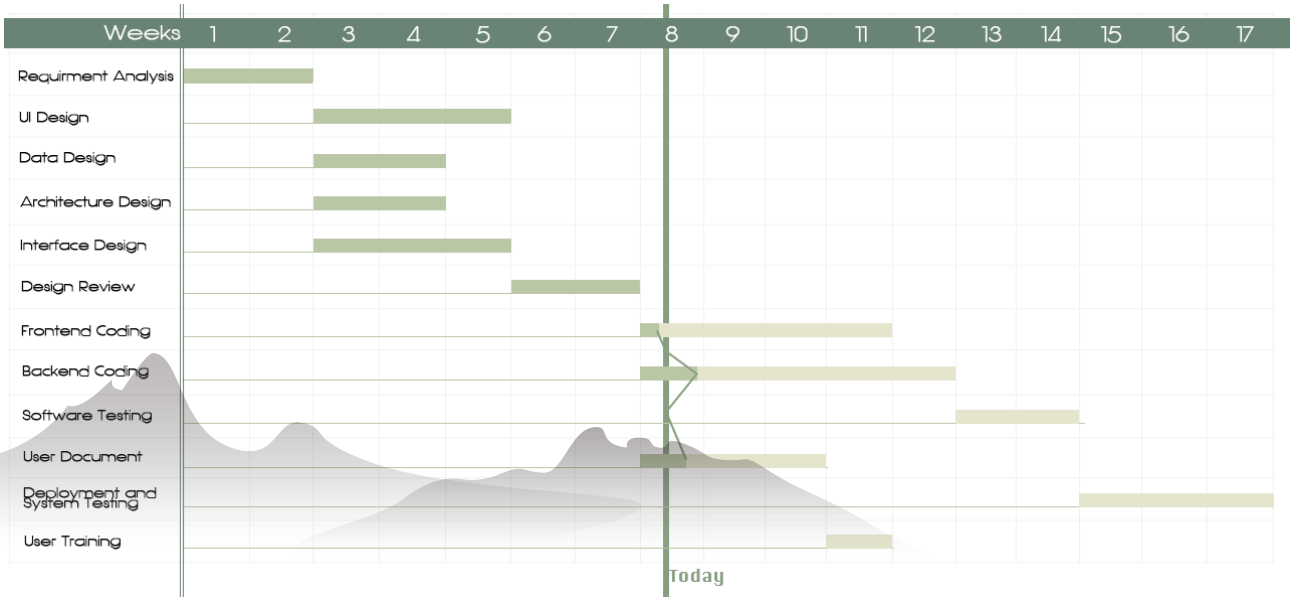
Week number	13	14	15	16	17	18	
Activity summary	G	A	A	R			
Component							Comments
Screen handling procedures	G	G	A	G			
File update procedures	G	A	R	A			
Housekeeping procedures	G	G	G	A			
Compilation	G	G	G	R			
Test data runs	G	G	G	A			
Program documentation	G	G	A	R			

6.3 Progress Visualization

The project manager can modify the Gantt chart when an activity is completed, and the visualization of it should given to all the member.

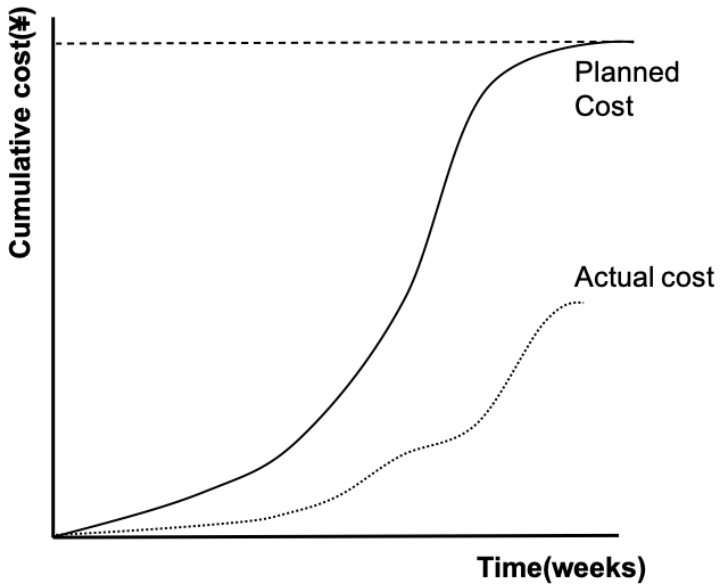


Someone prefer **slip chart**. A slip char is a very similar alternative favored by some project manager, who believe it provides a more striking visual indication of those activities that are not progressing to schedule. The two chart are both a simple example of project.

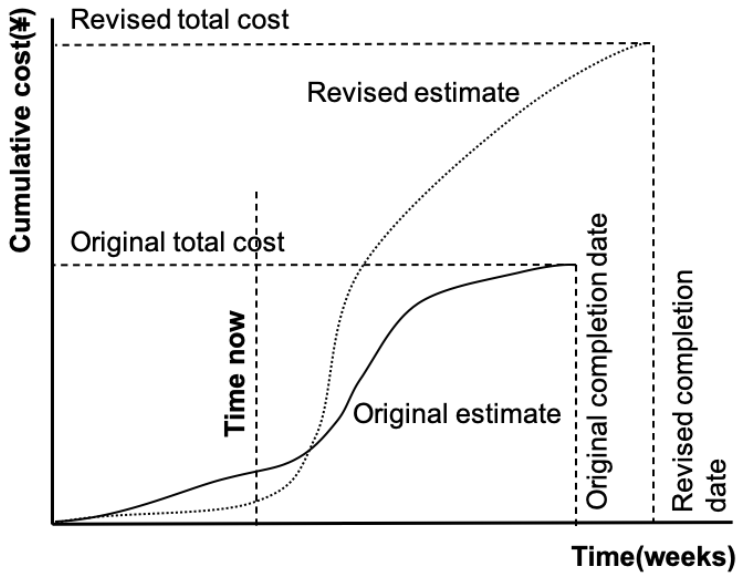


6.4 Cost Monitoring

Expenditure monitoring is an important component of project control, it provides an indication of the effort that has gone into our ICE project. Our ICE project might be on time but only because more monty has been spent on activities than originally budgeted. **Our Cumulative Expenditure Chart** is shown below, which provides a simple method of comparing actual and planned expenditure. Also we need to take account of the current status of our ICE project activities before attempting to interpret the meaning of recorded expenditure.



Then we add projected future costs calculated by adding the estimated costs of uncompleted work to the costs already incurred. Where a computer-based planning tool is used, revision of cost schedules is generally provided automatically once actual expenditure has been record. We update the Cumulative Expenditure Chart and including additional information available once the revised cost schedule.



6.5 Earned Value Analysis

Earned Value Management measures progress against a baseline. It involves calculating three key value for each activity in the WBS.

- **Planned Value(PV)**: the portion of the approved cost estimate planned to be spent on the given activity during a given period
- **Actual Cost(AC)**: the total of the costs incurred in accomplishing work on the activity in a given period. It must correspond to whatever was budgeted for the Planned Value and the Earned Value
- **Earned Value(EV)**: the value of the work actually completed

These three values are combined to determine at that point in time whether or not work is being accomplished as planned. The most commonly used measures are the cost variance:

$$CV = EV - AC$$

and the schedule variance:

$$SV = EV - PV$$

These two values can be converted to efficiency indicators to reflect the cost and schedule performance of the project. The most commonly used cost-efficiency indicator is the **Cost Performance Index(CPI)**. It is calculated as:

$$CPI = \frac{EV}{AC}$$

The sum of all individual EV budgets divided by the sum of all individual AC's is known as the cumulative CPI, and is generally used to forecast the cost to complete a project. The **Schedule Performance Index(SPI)** is often used with the CPI to forecast overall project completion estimates, and it can be calculated by:

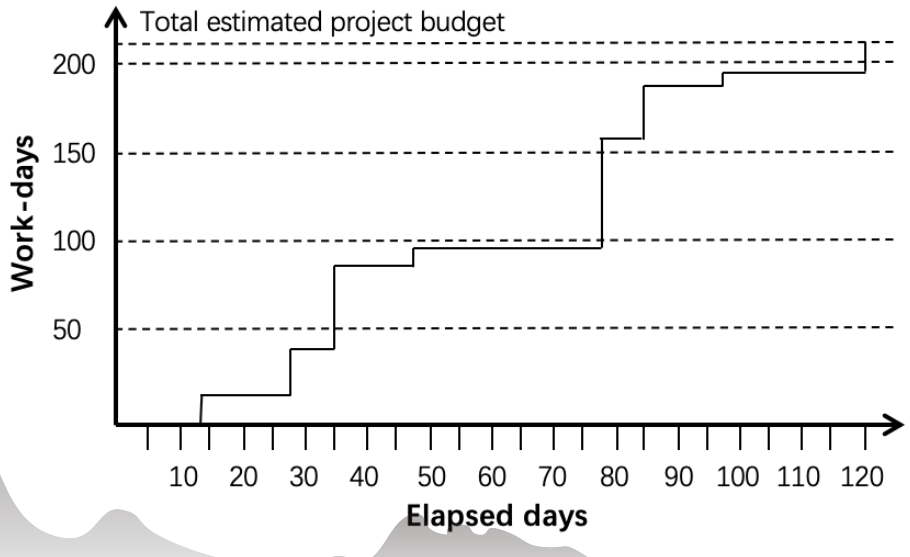
$$SPI = \frac{EV}{PV}$$

A negative schedule variance(SV) calculated at a given point in time means the project is behind schedule, while a negative cost variance(CV) means the project is over budget.

We combined the **Baseline Budget** and **0/100 technique**, assigning 0 to task unfinished and 100% to task completed of the budget value

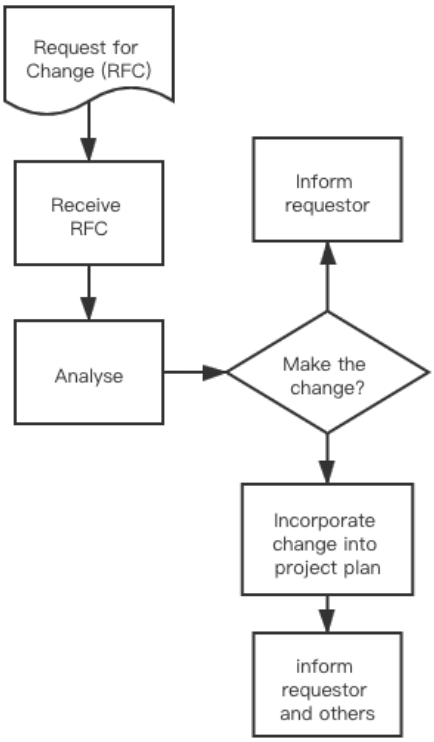
Task	Budgeted workdays	Scheduled completion	Cumulative workdays	% cumulative earned value

Requirement Analysis	14	14	14	6.25
Architecture Design	14	28	42	18.75
Data Design	14	28		
User Interface Design	21	35	84	37.5
Interface Design	21	35		
Design Review	14	49	98	43.75
User Manual	21	70	119	53.125
User Training	7	77	154	68.75
Front-end Implementation	28	77		
Back-end Implementation	35	84	189	84.375
Software Testing	14	98	203	90.625
Deployment & System Testing	21	119	224	100



6.6 Change Control

Because that the changes including requirements changes and staff changes couldn't be avoided in the process, so we must control and manage the changes. The change control managing process is showed in the following figure:



The key points is to plan the change beforehand, then we estimate the risk of controlling the change, and include verification of the success, the testing must be done to the change to ensure that we can control it, at last, we assign our staff with new responsibilities to solve the problems caused by the changes.

7. Project Implementation

7.1 Process Model

Scrum Process Model

In this project, we chose to use the Scrum process model for software development. Like all other forms of agile software processes, Scrum has frequent intermediate deliverables that contain functions that can work. This allows customers to get working software earlier, and at the same time allows projects to change project requirements to adapt to changing needs. Frequent risk and mitigation plans are developed by our development team.

In the process of using Scrum for software development, different members play different roles.

Role	**Person**	**Task**
Product Owner	Prof. Huang	1. Determine the function of the project. 2. For each sprint, adjust features and priorities as needed. 3. Inspection word of the development team.
Scrum Master	Zhe Zhang	1. Link Team and Product owner. 2. Organize Daily Scrum, Sprint Review and Sprint Planning meetings. 3. Ensure good collaboration between members. 4. Resolve obstacles in team development.
Team	Di Bu, Kaixin Chen	1. Mainly responsible for product development. 2. Deliver potential deliverable product increments after each Iteration. Ensure that the goals of the Sprint are achieved.

Meetings are an important part of Scrum. During the meeting, the product owner tells the development team which order items he needs to complete in the product order. The development team decides how many line items they can promise to complete in the next sprint. During the sprint, no one can change the sprint backlog, which means that demand is frozen during a sprint.

In the process of using Scrum for software development, we will hold the following types of meetings:

Meeting	**Task**
Planning Meeting	Every Thursday at 19:00. Arrange next week's task
Daily Meeting	Discuss how to get the job done in the team every day
Review Meeting	On the project display day, Prof.Huang participated in the discussion, and we show the results of the pervious stage to the teacher.
Retrospective Meeting	Every Thursday at 19:00, review last week's accomplishments before the planning meeting and dynamically adjust the tasks for the next week

In the process of software development using Scrum, we need to generate a series of documents as a guide and summary in our development process. In our development process, we will also write the following documents.

Document	Function
Product backlog	The product backlog is a summary document for the entire project. The product order includes a rough description of all required features.
Sprint backlog	The sprint backlog is a greatly refined document that contains information on how the team will fulfill the requirements of the next sprint.
Burn down chart	The burn down chart is a publicly displayed chart showing the number of unfinished tasks in the current sprint or the number of unfinished line items on the sprint order.

7.2 Development Techniques

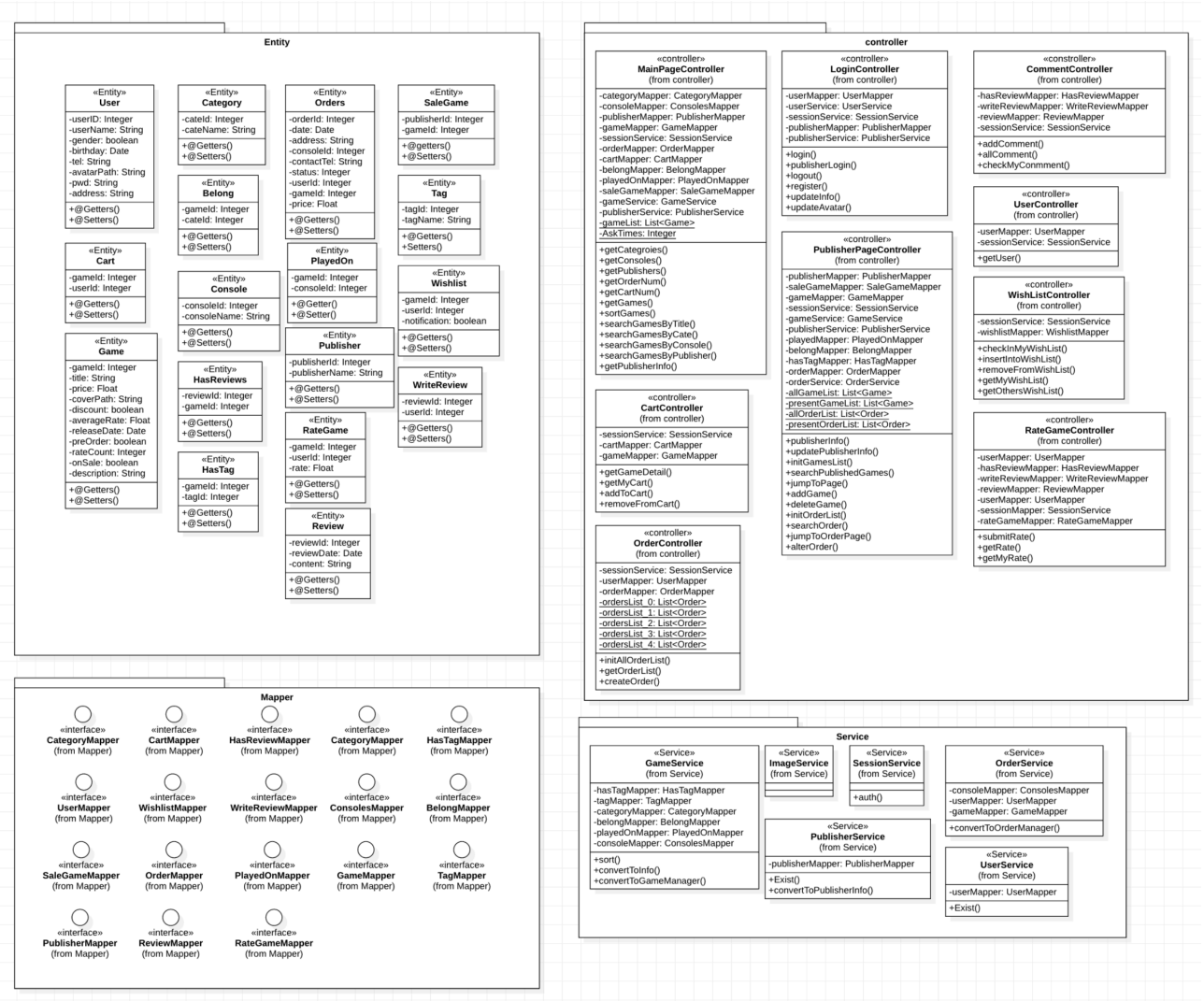
1. Development Tools

Tool	Illustration	Official Web
Eclipse	IDE	https://www.eclipse.org
IDEA	IDE	https://www.jetbrains.com/idea/download
Navicat	Database connecting tool	https://www.formsql.com/xiazai.html
PowerDesigner	Database designing tool	http://powerdesigner.de/
Edraw	Flowchart drawing tool	http://www.edrawsoft.cn
Postman	API platform	https://www.postman.com

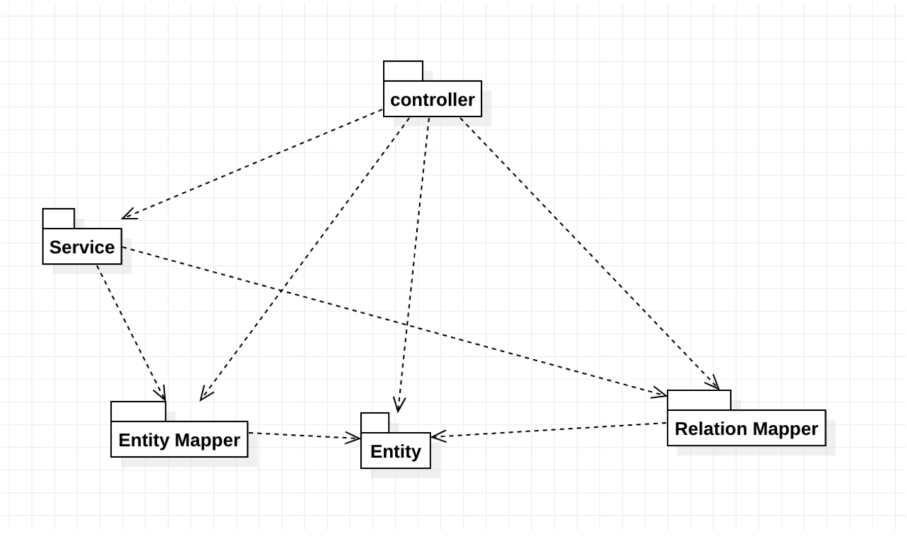
2. Development Environment

Tool	Version	Download
JDK	1.8	https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html
Mysql	5.7	https://www.mysql.com/
Nginx	1.10	http://nginx.org/en/download.html

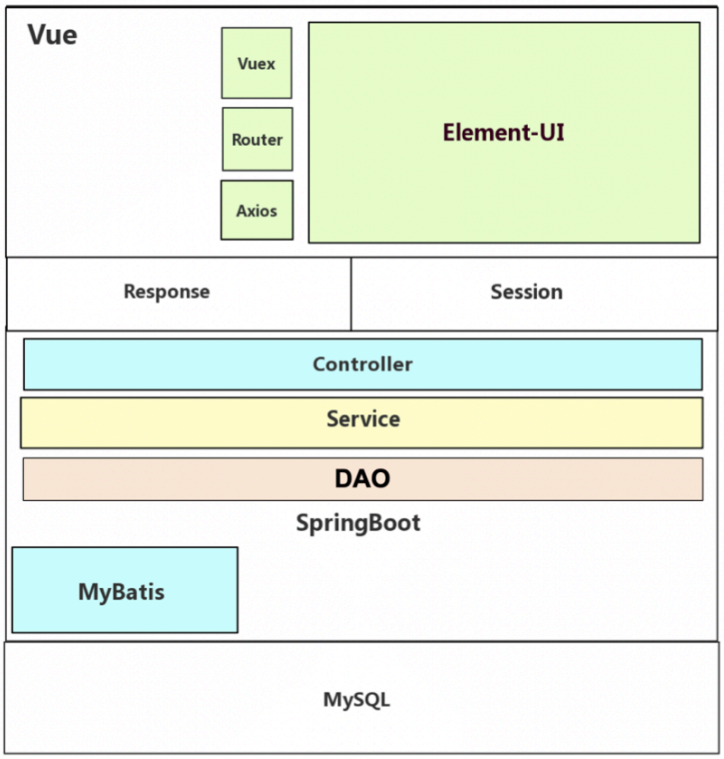
3. Design Elements



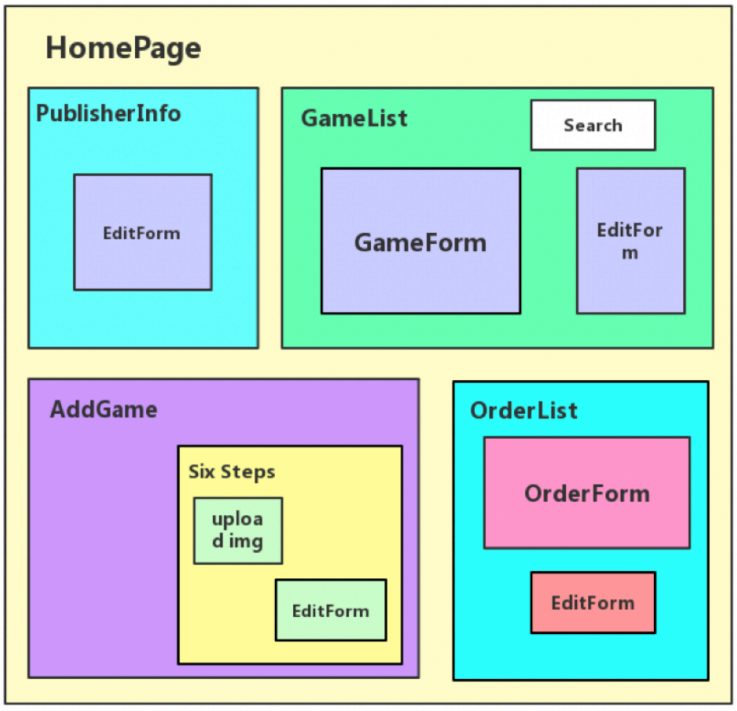
4. Package Diagram



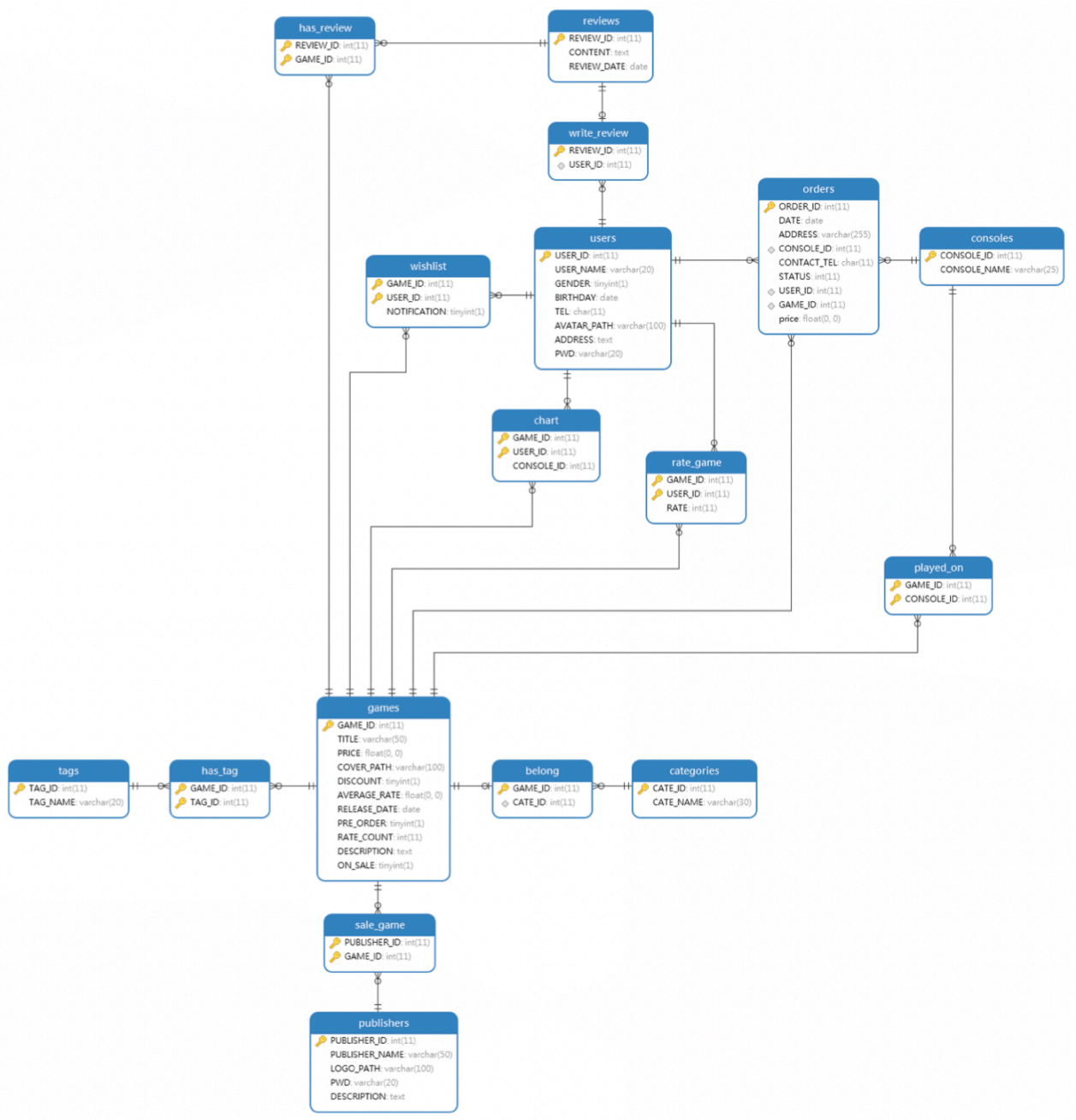
5. Design Architecture: The whole scope of the project and how we organize it.



6. Design Elements: All the components and the function methods supporting those components.

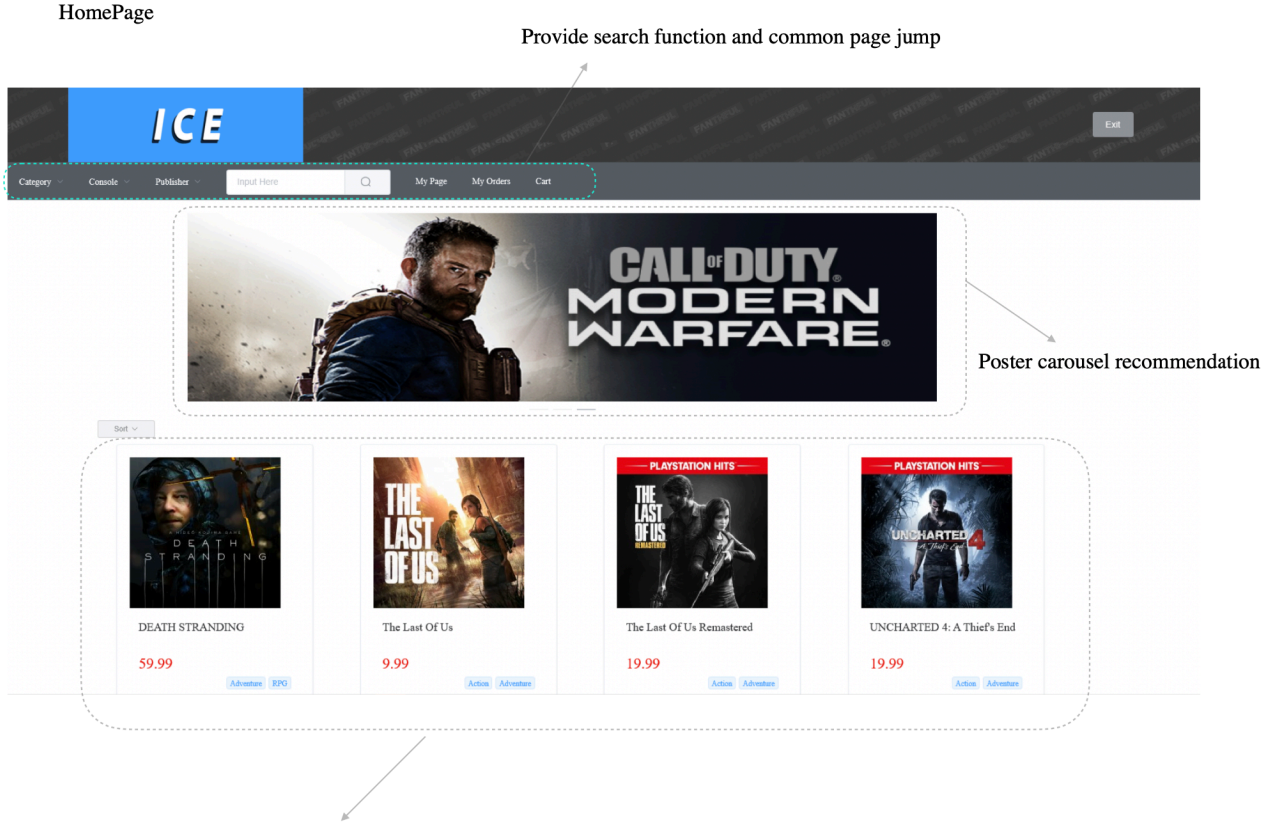


7. Design constraints: value constraints, relationships exclusivity constraints, navigability, generalization sets, multiplicity, derivation, changeability, initial value, qualifier, ordering, static, pre-condition, post-condition, and generalization set constraints. Those constrains are mainly designed and realized in the database design part, as is shown below

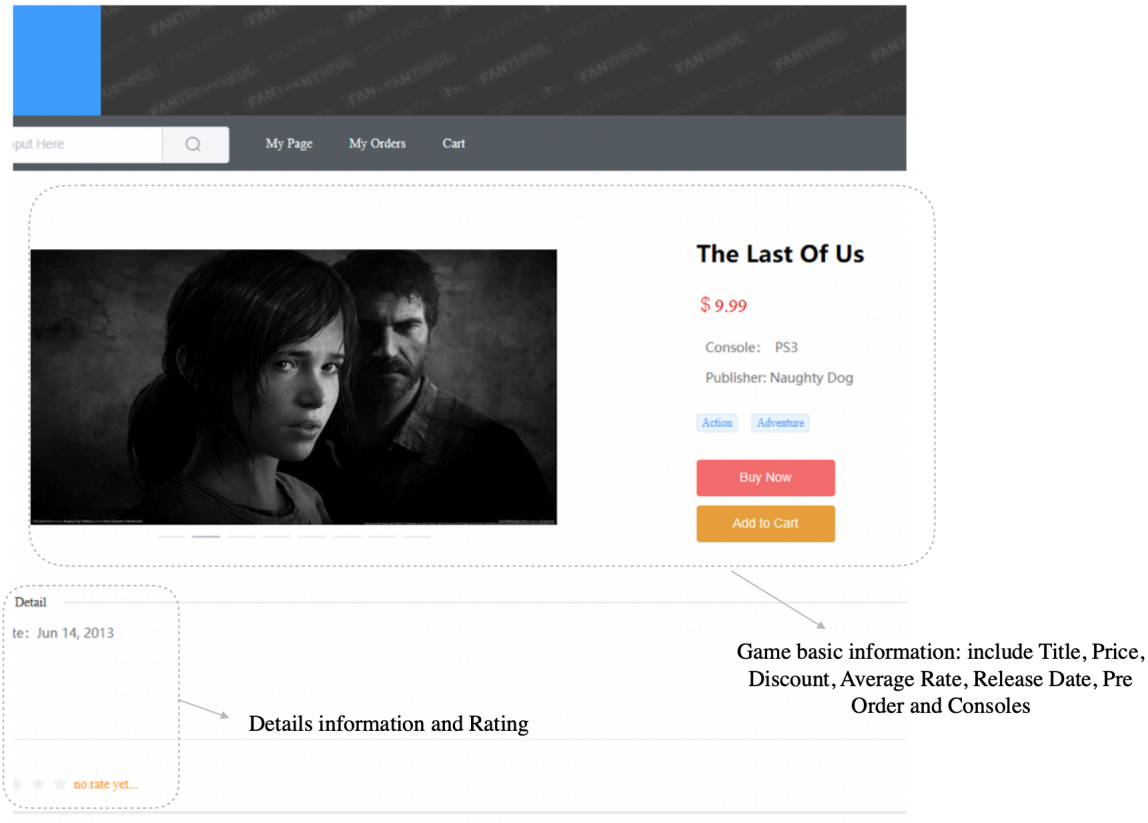


8. Prototype

HomePage

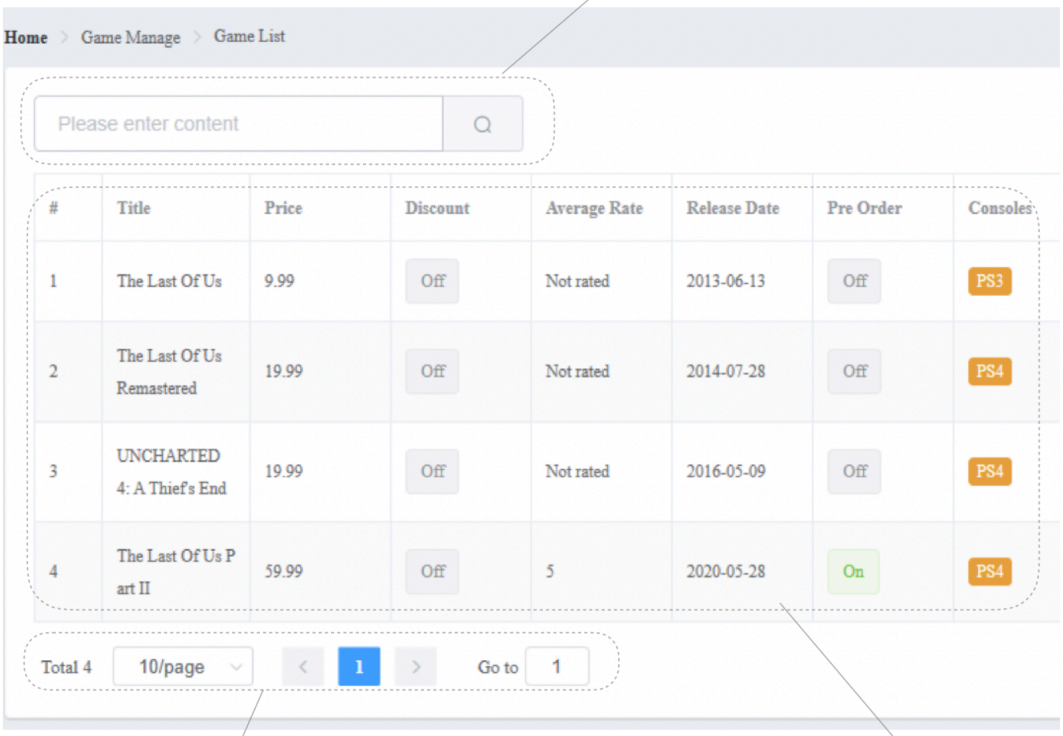


Game Detail Page



Game List Page

Search bar: Publisher can search game by title



Paging mechanism: realize page turning browsing and page jumping

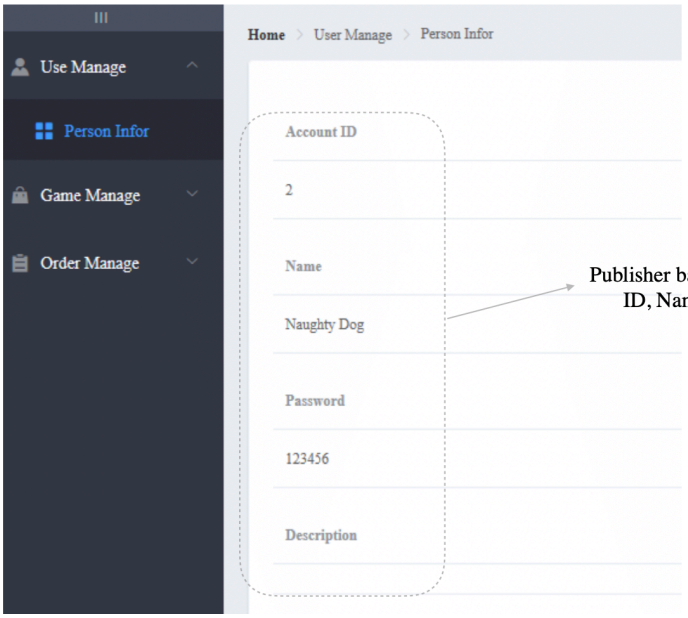
Game basic information: include Title, Price, Discount, Average Rate, Release Date, Pre Order and Consoles

Order List Page

Cart basic information: include Game Cover, Name, Consoles, Category and Price

Options button: User can buy or delete game in the shopping cart

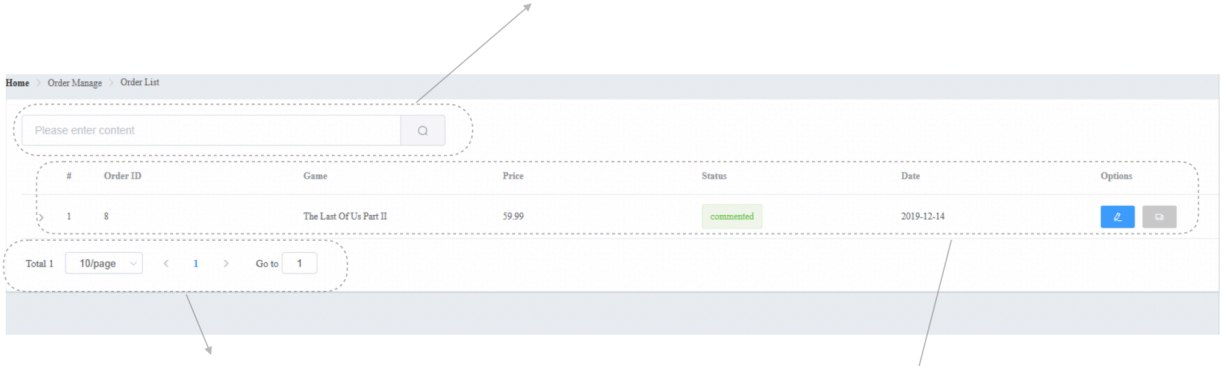
Publisher Person Page



Publisher basic information: include Account ID, Name, Password and Description

Orders Page

Search bar: Publisher can search game by title



Paging mechanism: realize page turning browsing and page jumping

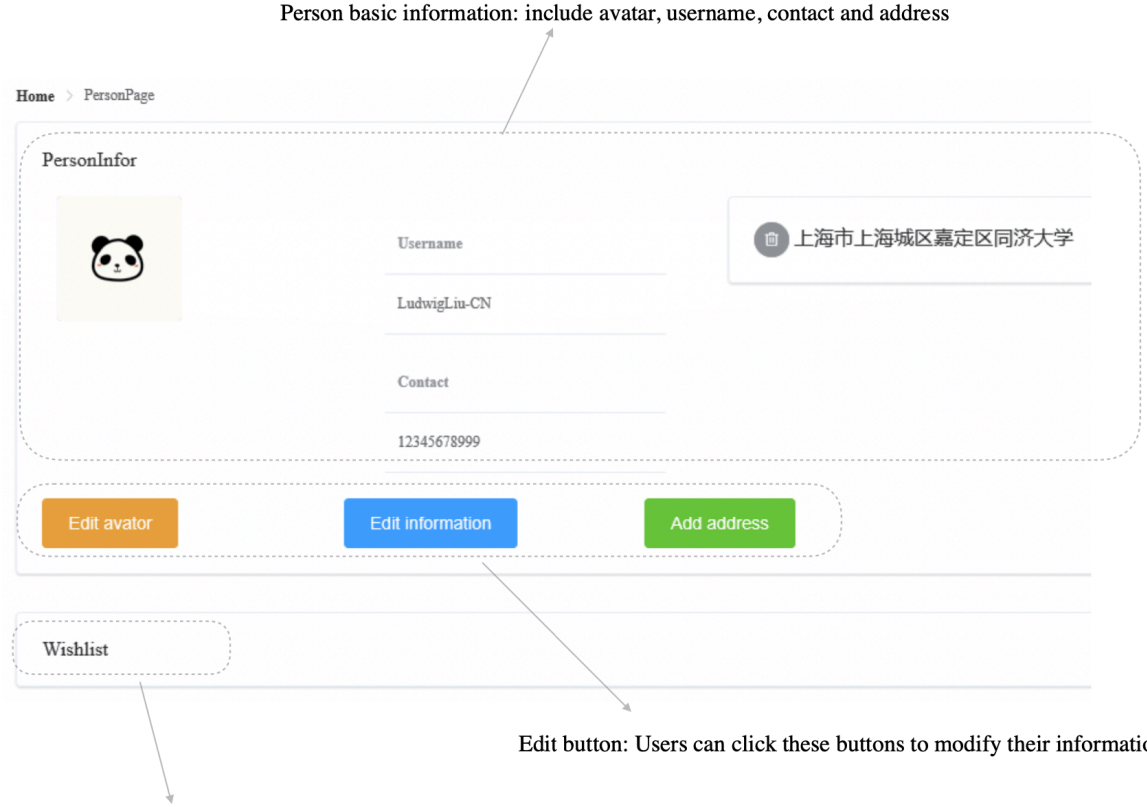
Here publisher can publish their games step by step, and of course they can edit those information later

Add Game Page

Here publisher can publish their games step by step, and of course they can edit those information later

Personal Page

Person basic information: include avatar, username, contact and address



Edit button: Users can click these buttons to modify their information

Wishlist part: Show games added to the wish list by users

9. References

Pierre, BOurque. Richard, E. (Dick) Fairley. SWEBOK: Guide to the Software Engineering Body of Knowledge. IEEE, 2014.

Kathy, Schwalbe. Information Technology Project Management. Boston, MA, 2011.

Bob, Hughes. Mike, Cotterell. *Software Project Management*. 2009.

PMBOK® Guide and Standards. Project Management Institute, Inc. www.pmi.org/pmbok-guide-standards. Accessed 12 April 2020.

ISO 31000 RISK MANAGEMENT. ISO. www.iso.org/iso-31000-risk-management.html. Accessed 13 April 2020.

Dmitriy, Nizhebetskiy. Software Development Life Cycle and Project Management. pmbasics101.com/software-development-life-cycle/. Accessed 13 April 2020.