ID2207 Homework4

Group 21: Yi Ren, Chongnan Wang

1. Use case stories

（1）　Login：

Login “Any employee in SEP can access the system through the login screen where he/she enters his/her username and password. The system will validate identification and then he/she will be accessible to different functionalities.”
Time estimate: UI 0.5h ,Logic 1h.

(2) Initiate client request

Initiate client request “Customer Service receive request from client, then enter the request in a form.”
Time estimate: UI 0.5h, Logic 0.5h

(3) Send to Senior Service Manager

Send to Senior Service Manager “Customer Service send the Client Request Form to Senior Service Manager, so that she can make a decision on the request.”
Time estimate: Logic 1h

(4) View client request

View client request “Senior service manager checks the request, and she has the power to reject it directly if she feels that it is not feasible at all. Otherwise, she reviews it and redirects it to the first reviewer, the financial manager.”
Time estimate: UI 0.5h Logic: 1.5h

(5) Write budget comment

Write budget comment “Financial manager writes his feedback based on the estimated budget by the client, and redirects it to the administration department manager”
Time estimate: Logic 1h, UI 0.5

(6) Finalize client request

Finalize client request “Administration decides whether to approve or reject it based on the financial

manager’s feedback and his expectation.”
Time estimate: Logic 1.5h, UI 0.5h

(7) Check AM’s decision

Check AM’s decision “Senior customer service check on administration’s decision. If the request is approved, she organizes a meeting to discuss with clients”
Time estimate: logic 0.5h, UI 0.1h

(8) Distribute tasks

Distribute tasks “the production manager fills an application with the client needs from his department, and sends tasks to each sub-team”
Time estimate: logic 0.5h, UI 0.1h

(9) Check task details

Decide event plan “Upon receiving tasks, each sub-team decides a plan for the activities they have to make in the event.:
Time estimate: logic 0.5h, UI 0.1h

(10) Add comments

Add comments “They can mention if they need extra budget for the requested items such as decorations or food. At that time, the application status is set to open. Each sub-team edits the task by filling an expected plan and adding comments for extra budget if required.”
Time estimate: logic 1h, UI 0.5h

(11) Review comments

Review comments “The production/service manager reviews the comments sent by each sub-team, and decide whether request negotiation with financial manager or HR manager”
Time estimate: logic 0.5h, UI 0.1h

(12) Create HR request form

Create HR request form “The production/service manager can fill the HR request form to request more human resources. After the HR request form is prepared, the Production/Service Manager have the ability to forward it to the HR Manager for additional consideration and approval.”
Time estimate: logic 0.5h, UI 0.1h

(13) Review HR request form

Review HR request form “The HR Manager will have access to requests from various departments, enabling them to identify which positions require staffing and initiate the recruitment of appropriate candidates.”
Time estimate: logic 0.3h, UI 0.2h

(14) Change HR request status

Change HR request status “The HR manager can modify the status of HR request form from production manager/service manager. He/she can deny the request and make the request status “denied”, or he/she can start recruit new employees and make the status “ under recruitment”. After interview, new employees are recruited, he/she can change the status to “recruited”. HR manager should update the HR request status timely so that other managers can check the process.”
Time estimate: logic 2h, UI 1h

(15) Check HR request status

Check HR request status “The production/service manger can check the status of HR request after he/she initiate new HR request form to HR manager. He/she can see whether the request is approved or the recruitment is done.”
Time estimate: logic 0.5h, UI 0.2h

(16) Create budget request form

Create budget request form “The production/service manager can fill the budget request form to request extra budget. After the budget request form is prepared, the Production/Service Manager have the ability to forward it to the Financial manager for additional consideration and approval.”
Time estimate: logic 1h, UI 0.5h

(17) Review budget request form

Review budget request form “The financial manager can negotiate the budget request with client. The request will not be processed until client and financial manager agree on the budget issues.”
Time estimate: logic 1h, UI 0.3h

(18) Change budget request status

Change budget request status “The financial manager can modify the status of budget request form from production manager/service manager. He/she can deny the request and make the request status “denied”, or he/she can start negotiation new with client and make the status “ under negotiation”. After negotiation, if client agree on extra budget, he/she can change the status to “approved”. Financial manager should update the budget request status timely so that other managers can check the process.”
Time estimate: logic 3h, UI 1h

(19) Check budget request status

Check budget request status “The production/service manger can check the status of budget request after he/she initiate new budget request form to financial manager. He/she can see whether the request is approved or denied.”
Time estimate: logic 0.5h, UI 0.1h

2. Release and Iteration planning

Use story name	Value	Risk	Iteration
Login	High	Low	1
Initiate client request	High	High	1
Send to Senior Service Manager	Medium	Medium	1

View client request	High	Low	1
Write budget comment	Medium	Low	1
Finalize client request	High	Low	1
Check AM’s decision	Medium	Medium	1
Distribute tasks	High	High	2
Check task details	Medium	Low	2
Add comments	High	Low	2
Review comments	High	Low	2
Create HR request form	High	High	3
Review HR request form	Medium	Low	3
Change HR request status	Medium	Medium	3
Check HR request status	Low	Low	3
Create budget request form	High	High	3
Review budget request form	Medium	Low	3
Change budget request status	Medium	Medium	3
Check budget request status	Low	Low	3

Number of stories for each combination:

Value\Risk	High	Medium	Low
High	4	0	5
Medium	0	4	4
Low	0	0	2

3. Metaphor

Metaphor	System
Restaurant	SEP
Restaurant manager	Administration manager
Top chef	Production/Service manager
Chef	Subteam
Senior waitress	Senior customer service
Waitor/waitress	Customer service
Order	Client request
Dish	Event plan
Account	Financial manager
Prepare ingredients	Distribute tasks
Change menu price	Budget request
Recruit new chefs	HR request

4. test-driven pair programming

In project development, we use test-driven programing. Because there are 4 workflows, we designed 4 groups of test cases for different workflows. We carefully considered various situations that may occur when users use the system and tried to cover all situations as comprehensively as possible. For example, we designed a test case for users to log in to the system based on the sample data in the database to detect whether the system can correctly verify the user's identity information. When designing the system, we included classes such as PM and AM, and we also specially designed many tests to detect whether their methods are running normally. After writing the test cases and completing the code, we executed the test cases and modified the code and fixed the bugs according to the errors found.

5. Applied refactoring

At first we designed many classes to implement the system, however we later found that it is more reasonable to design a class for ever character in the system. For example, we design administration manager class, financial manager, department manager, senior customer service, customer service and so on. according to the documentation, we carefully designed their attributes and functions. We also designed classes of different forms to make the flow of information more intuitive and proper. Additionally we design a database and any user could only access the functionality that is exposed to them. In each iteration, we modify classes and their functions to make it better for the system.

We just devided the whole refactoring into 3 iterations, in the first iteration we implemented the workflow1 and its relative classes, in the second iteration we implemented the workflow2 and its relative classes. In the last iteration we implemented the last two workflow because they have similar logic and we modify some details or fix bugs we found of the system in each iteration. Generally speaking, we design classes according to the need of 4 workflows.

6. Acceptance test

   1. Login

Test Case Name	Login
Expected Actions	1. Navigate to the login page of

	the system. 2. Enter user name: “username” 3. Enter password: “password”
Expected Results	If the user name and password is validate, user can enter the system and see the corresponding functionalities. Otherwise, he/she will be refused by the system.
Test Result	Successful

2. Create new budget request

Test case name	Create new budget reqeust
Expected Actions	1. The production manager login successfully. 2. Production manager choose “5) Send a budget negotiation to financial manager” 3. Production manager fill the content of requirements on budget. 4. The budget request is created.

Expected Results	1. System shows : “Login Success!” 2. System shows: “Please enter your requirements on budget, press Ctrl + Z to end input”. 3. System shows: “Requirements added!” 4. System creates new budget request, and UI goes back to Production Manager functions..
Test Results	Successful

7. Daily Meeting

Meeting Dates	9 Oct, 2024
Participants	Yi Ren, Chongnan Wang
Meeting Notes	1. Look through documents; 2. Discuss about system design a. Think about using what program language. b. Think about how design classes. c. Think about how to implement the login function 3. Review on 4 workflows, think about

	how to design functions of classes to implement system functionalities. 4. Assign work and get to programming about the login function.
Comment	Expected outcome for today: accomplish the general system design, assign work, and finish the login function codes.

Meeting Dates	13 Oct, 2024
Participants	Yi Ren, Chongnan Wang
Meeting Notes	1. Summary of our activities in the previous day; a. Review the codes and corresponding use case stories. 2. Today expected actions: a. Writing the Metaphor. b. Accomplish implementing creating HR request and budget request. c. Thinking of the implementation of class Department Manager. 3. Problems: Not much.

Comment	Expected outcome for today: today we write the metaphor, summarize previous work and go on writing codes about HR /budget request and Department Manager.

8. Comparison

   In the project development, we use Extreme Programming. We discovered that it is effective to find the point of the system and get to code. Instead of designing abstract classes and modelling, this saves a lot of time and energy. During development, we refactored the codes frequently to make the program functioning better. We also design comprehensive test units to assure the reliability and usability of system. Extreme development is an agile practice that is more inclined to development methodology and team collaboration, while object-oriented development is a programming paradigm and design philosophy. XP and OOP can be used in combination. XP practices such as TDD, refactoring, and pair programming can help developers better achieve OOP design goals, while OOP provides the basic tools for building modular and maintainable systems.