KTH

MODERN METHODS IN SOFTWARE ENGINEERING

(ID2207)

PROJECT

GROUP 6

Name: **Yuchen Gao**
Name: **Weikai Zhou**

Date: **23 October, 2021**

1. User stories

1.1. Login story:

Login: “As a member of the SEP, he/she is able to login with password and operate the function exclusive to him/her.”
Time Estimate: Logic:0.25h; UI: 0.25h;

1.2. Create new client request

Create new client request: “As the customer service, he/she would like to record the detailed plan for customers, the request has to be clear and easy to read for the upper class.”
Time Estimate: Logic:1.5h; UI: 1.5h;

1.3. Send to senior service customer

Send to senior service customer: “As the customer service, he/she would like to send the report to Senior Customer Service so that the event plan process can be officially initiated.”
Time Estimate: Logic:0.25h; UI: 0.25h;

1.4. View client requests

View client requests: “As the Senior Customer Service, he/she would like to view the report sent by Customer Service.”
Time Estimate: Logic:0.5h; UI: 1h;

1.5. Change status of the request

Change status of the request: “As the Senior customer service, he/she would like to make a general decision and change the status of the event based on his/her decision.”
Time Estimate: Logic:0.25h; UI: 0.25h;

1.6. Add budget comment

Add budget comment: “As the Financial Manager, he/she would like to make a report to make it clear for his/her upper class to think more easily of the feasibility of the event based on the professional economic perspective.”
Time Estimate: Logic:0.5h; UI: 0.25h;

1.7. Finalize the request

Finalize the request: “As the Administration Manager, he/she would like to make a final decision of the initialization of the whole activity based on his experience and financial manager’s report.”
Time Estimate: Logic:0.5h; UI: 0.25h;

1.8. Check the request

Check the request: “As the Senior Customer Service, he/she should be able to check the feedback from the Administration Manager. If the request is approved, the contact information of the client will be displayed to the Senior Customer Service to arrange a meeting. Otherwise, the system should display the termination of the case.
Time Estimate: Logic: 1h, UI: 0.5h

1.9. Create task distribution form

Create task distribution form: “As the Production/Service Manager, he/she would like to distribute tasks to the sub-team, this process is realized by initializing a formal task distribution form.”
Time Estimate: Logic:1h; UI: 1h;

1.10. Send task distribution form

Send task distribution form: “As the Production/Service Manager, he/she would like to send the task distribution form to the event-relative sub-team members, this operation should visualize the individual of different departments .”
Time Estimate: Logic:0.5h; UI: 0.5h;

1.11. View task detail

View Task details: “As the Sub-team members, they will have the right to see the official initializing form of the event, they can see the requirements and budget of the event and make adjustments based on their experience .”
Time Estimate: Logic:0.5h; UI: 1h;

1.12. Add plan

Add plan: “As the sub-team members, he/she should be able to add plans based on the request details to satisfy the clients’ needs.”

Time Estimate: Logic:0.5h; UI: 0.5h;

1.13. Add comment

Add comment: "As the Sub-team members, they will have the right to add comments if more budget is needed when they plan the part they play in the event, and they have the interface of adding comments to the form so that the manager can see it ."
Time Estimate: Logic:0.5h; UI: 0.5h;

1.14. Check plan and comment

Check plan and comment: "As the Production/Service Manager, he/she will have the ability to view the plan and comment added by the sub-team members."
Time Estimate: Logic:0.5h; UI: 0.5h;

1.15. Create HR request form

Create HR request form: "As the Production/Service Manager, he/she will have the right to allocate members to their team, so that the event will be assigned properly, and they can access the request form to ask the HR team for extra members ."
Time Estimate: Logic:0.5h; UI: 1h;

1.16. Send HR request form

Send HR request form: "As the Production/Service Manager, after the HR request form is created, he/she should be able to send the form to the HR manager for further decision."
Time Estimate: Logic:0.25h; UI: 0.25h;

1.17. View HR request

View HR request: "As the HR manager, he/she will have access to the request from different departments, so that they can tell which position is in need and recruit relative members."
Time Estimate: Logic:0.5h; UI: 1h;

1.18. Change status of HR request

Change status of HR request: "As the HR manager, he/she should be able to update the status of the HR request so that the Production/Service Manager can know the latest status of their requests. If

corresponding people are recruited, the status should be updated as ‘Recruited’. If the recruitment is ongoing, the status should be updated as ‘Under Recruitment’. If the HR denies the request, the status should be updated as ‘Denied’.”
Time Estimate: Logic:0.25h; UI: 0.5h;

1.19. Check HR request

Check HR request: “As the Production/Service Manager, he/she will have access to the form they submitted to the HR manager so that they can follow the updates of the recruitment: if it is recruited successfully or not.”
Time Estimate: Logic:0.5h; UI: 1h;

1.20. Create financial request form

Create financial request form: “As the Production/Service Manager, they will have the right to initiate an extra budget requirement form so that when the comments are added by the sub-team, the Production/Service Manager can ask the financial manager for extra budget.”
Time Estimate: Logic:1h; UI: 1h;

1.21. Send financial request form

Send financial request form: “As the Production/Service Manager, after the financial request form is created, he/she should be able to send the form to the Financial manager for further decision.”
Time Estimate: Logic:0.25h; UI: 0.25h;

1.22. Check financial request

Check financial request: “As the Financial manager, they are able to check the financial requests from different departments and make some changes to them.”
Time Estimate: Logic:0.5h; UI: 1h;

1.23. Change status of financial request

Change status of financial request: “As the Financial manager, he/she should be able to update the status of the financial request so that the Production/Service Manager can know the latest status of their requests. If the financial manager denies the request or the client refuses to add budget, the status should be updated as ‘Denied’. If the budget is added, the status should be updated as ‘Approved’.”

Time Estimate: Logic:0.25h; UI: 0.5h;

1.24. Check status of financial request

Check status of financial request: “As the Production/Service Manager, they will have the right to follow the updates of the extra budget allocation request and see if it is approved or rejected.”
Time Estimate: Logic:0.5h; UI: 1h;

2. Release & Iteration planning

User story name	Value	Risk	Iteration
Login	High	Low	1
Create new request	High	High	1
Send to SCS	Medium	Low	1
View client request	High	Medium	1
Change request status	Medium	Low	1
Add budget comment	High	Medium	1
Finalize the request	High	Medium	1
Check the request	Medium	Medium	1
Create task distribution form	High	High	2
Send task distribution form	Medium	Low	2
View task detail	High	Medium	2
Add plan	High	Low	2
Add comment	Medium	Low	2
Check plan and comment	Medium	Medium	2
Create HR request form	High	High	3
Send HR request form	Medium	Low	3
View HR request	High	Medium	3
Change status of HR request	Medium	Low	3
Check HR request	Medium	Medium	3

Create financial request form	High	High	4
Send financial request form	Medium	Low	4
Check financial request	High	Medium	4
Change status of financial request	Medium	Low	4
Check status of financial request	Medium	Medium	4

	High value	Medium value	Low value
High risk	4	0	0
Medium risk	6	4	0
Low risk	2	8	0

3. Metaphor

Metaphor	System
Restaurant	SEP company
Waiter/Waitress	Customer Service
Head Waiter/Waitress	Senior customer service
Cashier	Financial Manager
Restaurant manager	Administration Manager
Head Chef	Production/Service Manager
Chef	Sub-team
Order	Client request
Dishes	Activity plan
Hire more chef	Staff recruitment
Rise cost of dishes	Financial request

4. Test-driven pair programming

We applied test-driven pair programming by writing examples that can run based on our database and user exclusive operations. First, we decompose the project into 4 parts and come up with the data structure to implement. For example, we decide to use 4 different classes to represent the different requests, several functions to implement user functions, and 4 vectors to store the information. Then, based on the data structure, we write the tests to check the functionality of each workflow. At this time, the tests cannot be compiled and run. After that, we start to implement the detailed classes and functions. When we think we have written the code to the extent that the tests should be able to run, we will run the tests to check our implementation. If we find some unexpected behaviors, we will go deep into the code to fix them and run the tests again until our implementation can pass all the tests we write. During this process, both of the team members participated to write code, run tests and debug.

5. Applied Refactoring:

At first, we would like to implement many classes to write operations inside of different classes, but we realized the exclusive operation is owned by each individual, so during implementation, the classes have to be modified. So we discarded the solution and began to construct the program by analyzing the operations owned by each one and how they can access the database. They will only be able to access the database that is exposed to them, and when they have the same specific purpose on one procedure, they will be able to access the same database. And we cannot come up with all the needed functions and variables in each class at the beginning. During each iteration, we continually add more variables and functions to each class to make them more suitable for the whole project.

6. Iteration management:

We manage different iterations based on the decomposition of the whole system, we believe each iteration serves a specific purpose and during one iteration these included people have to have the same goal, accessing the same database when necessary. The system is divided into 4 parts and we can operate functions to access different request forms inside the process.

7. Acceptance test

Test case name	Login
Expected actions	1. Go to the homepage of the system 2. Enter “Login” and press Enter on the keyboard to go to the login interface 3. Enter username “CS” and press Enter on the keyboard to start login as customer service. 4. Enter password “000” and press Enter on the keyboard.
Expected results	1. The system should display: “Please choose a function: Login Exit” 2. The system should display: “Please enter username: ” 3. The system should display:

	“Please input password: ” 4. The system should display: “Login successfully! Please choose a function: CreateNewClientRequest SendToSCS Logout”
Test result	Successful

Test case name	Create new client request
Expected actions	1. The customer service login successfully. 2. Enter “CreateNewClientRequest” and press Enter to choose the function. 3. Enter the client name “Alice” and press Enter. 4. Enter the client phone number “1234567” and press Enter. 5. Enter the ID “1” and press Enter. 6. Enter the date “20211015” and press Enter. 7. Enter the request “1...2...3…” and press Enter. 8. Enter the priority “high” and press Enter. 9. Enter the ClientBudget “10” and press Enter.
Expected results	1. The system should display: “Login successfully! Please choose a function: CreateNewClientRequest SendToSCS Logout” 2. The system should display: “Please input ClientName: ” 3. The system should display: “Please input ClientPhoneNumber:” 4. The system should display: “Please input ID: ” 5. The system should display: “Please input Date: ” 6. The system should display: “Please input Request: ” 7. The system should display: “Please input Priority: ” 8. The system should display: “Please input ClientBudget: ” 9. The system should display: “Create successfully! Please choose a function: CreateNewClientRequest SendToSCS Logout”
Test result	Successful

8. Daily meeting:

Meeting Dates	13 Oct, 2021
Participants	Yuchen; Weikai
Meeting Notes	1. Review of the Login realization; a. Achieved by a while loop 2. Review of the Client Request realization; a. Including different information b. Sent between users 3. Proposing the realization of task distribution; a. The managers should be able to fill in a form of detailed information about the task b. The form can be sent to certain people c. Sub-team can add plans and comment d. The manager can check the plan and comment 4. Implementation of task distribution.
Comment	Expected outcome for today: Finish coding the task distribution task and debugging the code. Written by Yuchen.

Meeting Dates	15 Oct, 2021
Participants	Yuchen; Weikai
Meeting Notes	1. Summary of the realization of task distribution a. Implemented by a class and a function 2. Proposing the realization of HR request a. The managers should be able to create a HR request form and fill in all the needed information b. The HR team should be able to view the form and update the status c. The managers should be able to check the latest update of the statu 3. Implementation of HR request
Comment	Expected outcome for today: Finish coding the HR request task and debugging the code. Written by Weikai.

9. Comparision:

In this approach of realization, we find it easy to capture the main idea of the program and focus on what it really matters, such as the function realization and user interface, instead of the abstract classes inside of each class this whole system has. The XP approach is faster to realize and easier to comprehend. We can deploy the simplistic way of the system in a normal way, both for developers and customers. The OOP is a more professional way, also can be really complicated. The hard to understand developing method is not trivial. However, the relationship between objects is better described by OOP rather than XP, we have to select wisely. In this specific example, it is a more intuitive description by XP, we can have it in our mind about the structure of the program and explain it without illustration. If we use OOP, things could be different, as we have to keep track of classes and operations, it may be a wide layer and a loose couple, high cohesive way, but really difficult to understand.