ID2207 Final Report

User stories

Login story

Any employees can access the system through the login GUI where he/she enters his/her user name and the password. After verification of the user's authority, he/she will be able to access different functionality.

> Time estimate: GUI: 0.5 day, logic: 0.5 day

Manage client request story

1. Initialize client request

The customer service (CS) creates a new client request form, filling it with client request details and then will be able to submit it.

> Time estimate: GUI: 3h, logic: 1h

2. Redirect client request

The CS, senior customer service (SCS), financial manager (FM) can redirect the client request to SCS, FM and administration manager (AM) respectively.

> Time estimate: GUI: 0h, logic: 8h

3. View the list of client requests

The CS, SCS, FM, AM, PM and SM can access to the list of the client requests that they are working on.

> Time estimate: GUI: 3h, logic: 1h

4. View the client request details

The CS, SCS, FM, AM, PM and SM can access to every single client request in the request list

and look into the request details.

> Time estimate: GUI: 1h, logic: 1h

5. Write feedback on request

After the FM review the client request, he/she can write his/her feedback on the request.

> Time estimate: GUI: 1h, logic: 1h

6. Make decision on request

The SCS and AM have the right to accept or reject the client request.

> Time estimate: GUI: 1h, logic: 1h

7. Organize business meeting

The SCS check the status of the client request, and if its status is set to be accepted by AM, the SCS will creates a business meeting schedule and mentions the staff involved in.

> Time estimate: GUI: 3h, logic: 1h

8. Update the client request

Based on the result of the business meeting, the SCS can update the client request with more detailed information.

> Time estimate: GUI: 0h, logic: 1h

9. View business meeting

The staff can check the details of the business meeting.

> Time estimate: GUI: 1h, logic: 1h

Manage task story

1. Create an application

After reviewing the details of the client request, the PM/SM can create a new application, filling

it with client requirements.

> Time estimate: GUI: 1h, logic: 1h

2. Review the history of applications

The PM/SM can review the list of the applications.

> Time estimate: GUI: 1h, logic: 1h

3. Check the staff availability

The PM/SM can check the availability of staff in the department to confirm whether there is any conflict in schedules.

> Time estimate: GUI: 0h, logic: 1h

4. Create task

The PM/SM creates task, fill it with the task description and send it to the staff in the department sub-team.

> Time estimate: GUI: 1h, logic: 1h

5. Review the list of tasks

The PM/SM can review the list of the tasks they have sent,
and all sub-teams in product and service department can review the list of the tasks they are working on.

> Time estimate: GUI: 1h, logic: 1h

6. Review the task details

The PM/SM can review the details of the tasks they have sent, and all sub-teams in product and service department can review the details of the tasks they are working on.

> Time estimate: GUI: 1h, logic: 1h

8. Update the task

The sub-team should make a plan according to the task, and mention the extra budget if needed. Then the sub-team attaches the plan to the task and updates it.

> Time estimate: GUI: 1h, logic: 1h

9. Update the status of the application

The PM/SM can set the status of the application to open, in progress, close or achieved. Once the application is created, the status will be set to open. And after the resource request originated from the application is met, the status of the application can be set to in progress. Finally, after the clients needs are realized, the status of the application can be set to close or achieved.

> Time estimate: GUI: 1h, logic: 1h

Manage staff recruitment story

1. Create staff resource request

The PM/SM can request a staff resource request if needed. The user creates a request form, filling it with required working experience and skills.

> Time estimate: GUI: 1h, logic: 1h

2. Review staff resource request

The HR manager can review the details of the resource request.

> Time estimate: GUI: 1h, logic: 1h

3. Organize recruitment

After HR manager review the staff resource request, he/she can create a recruitment event. Based on the request, HR can select a long-term or outsourcing recruitment.

> Time estimate: GUI: 1h, logic: 1h

4. Update resource request status

After a successful recruitment, the HR manager can set the status of the resource request to complete.

> Time estimate: GUI: 1h, logic: 1h

5. Add staff information

The HR manager can add the information of new staff to the system. The HR manager create a new staff form, filling it with the details of the staff information.

> Time estimate: GUI: 1h, logic: 1h

6. Update staff information

The HR can update the staff information if the status of the staff has changed.

> Time estimate: GUI: 1h, logic: 1h

Manage financial request story

1. Create budget request

The PM/SM can request a extra budget request if needed. The user creates a request form, filling it with required amount of budget and the description.

> Time estimate: GUI: 1h, logic: 1h

2. Review budget request

The FM can review the details of the budget request.

> Time estimate: GUI: 1h, logic: 1h

3. Update the status of budget request

The FM can set the status of budget request to complete if he/she approve to provide more budget or he/she makes a agreement on budget with clients.

> Time estimate: GUI: 1h, logic: 1h

Release plan

User Story Name	Value	Risk
Login Story	High	Medium
Initialize client request	High	Low
Redirect client request	Medium	High
View the list of client requests	High	Medium
View the client request details	High	Medium
Write feedback on request	Low	Low
Make decision on request	High	Low
Organize business meeting	Low	Low
View business meeting	Low	Medium
Update the client request	High	Medium
Create an application	High	Low
Review the history of applications	High	Medium
Check the staff availability	Medium	Medium
Create task	High	Low
Review the list of tasks	High	Medium
Review the task details	High	Medium
Update the task	Medium	Low
Update the status of the application	Medium	Low
Create staff resource request	Medium	Low
Review staff resource request	Medium	Low
Organize recruitment	Medium	Low

User Story Name	Value	Risk
Update resource request status	Medium	Low
Add staff information	High	Low
Update staff information	Medium	Medium
Create budget request	Medium	Low
Review budget request	Medium	Low
Update the status of budget request	Medium	Low

Number of stories for each combination:

	High Value	Medium Value	Low Value
High Risk	0	1	0
Medium Risk	7	2	1
Low Risk	5	9	2

Release plan

- Alpha Release - All high value stores and the combination of (medium, high), (medium, medium)

User story included in Alpha release	Entities involved
Login Story	Staff
Initialize client request	ClientRequest
Redirect client request	ClientRequest
View the list of client requests	ClientRequestList
View the client request details	ClientRequest
Make decision on request	ClientRequest

User story included in Alpha release	Entities involved
Update the client request	ClientRequest
Create an application	Application
Review the history of applications	ApplicationHistory
Check the staff availability	Staff
Create task	Task
Review the list of tasks	TaskList
Review the task details	Task
Add staff information	Staff
Update staff information	Staff

- Beta Release - The other stores.
- v1.0 - Optimize Non-functional requirement.

Iteration Planning of the Alpha Release

1st Iteration

- Login Story

2nd Iteration

- Initialize Client Request
- View the list of client requests
- View the client request details
- Make decision on request
- Update the client request

3rd Iteration

- Create an application
- Review the history of applications

- Create task
- Review the list of tasks
- Review the task details

4th Iteration

- Add staff information
- Update staff information
- Check the staff availability

5th Iteration

- Redirect client request

Metaphor

Metaphor	System
The shopping mall	The SEP company
The client ask for an item	The client request
The staff check the stock	The SCS, FM, AM review the request
The bargains between salesperson and client	The business meeting
Mailing item to client's apartment	The PM and SM handle client request
Dispatch delivery task to courier	The PM and SM assign tasks to sub-teams
The client receives the delivery	The client request is set to close

Programming and Refactoring Process

Pair Programming

Basically we were performing **Driver-Navigator** style of pair programming in the project. After

clarifying the user stories that should be implemented in a iteration and the interface of the class and methods, the **Driver** focused on the details of the implementation, and the **Navigator** reviewed the codes, observed whether there existed any bug or obstacle. During the project, the role switched frequently among two of us, thus both of us have chance to experience how a project is built. Below is how the workflow goes:

- Define user stories that should be implemented and divide the tasks
- Make an agreement on the object model and specify the interface
- The **Driver** starts coding and the **Navigator** keeps observing the screen of **Driver**
- If **Navigator** finds a potential bug or issue, the **Navigator** writes a test to verify and the **Driver** refactors the codes until passes the test

Refactoring

After Alpha release completed, we inspected the codes and match them against following checklist:

- if there is replicated code, then we investigate whether it is possible to encapsulate it into a method to improve reusability.
- if there is redundant class, then we collapse it can dispatch its responsibility to the class that is interested in it.
- if there is any attribute or method that are not supposed to be visible to class user, we set the visibility to private.
- check if there if any hierarchical relationship can be built between objects
- check if the name of class, attribute or method can be made more understandable

Unit Test with TDD

To develop in eXtreme, under the Test-Driver developing, we use the Unit Test Tool to help us test the code the push our developing.

However, our goal is developing a RESTful server with the MVC architecture, the JUnit test tool is not convernient. Finally we use the Http request tools(list http script or Postman) to instead the JUnit to driven our developing.

Daily Meeting

10.4 meeting

Went through the material again, and generated the user stories.

10.15 meeting

First iteration, Alpha release started, discussed how to implement and test Login feature.

10.20 meeting

Third iteration, Alpha release started, discussed how to implement application management feature.

Comparison between eXtreme and Object-Oriented Programming

The extreme programming is much more efficient than object oriented programming. In object oriented programming approach, there is not much room for tolerating requirement and system change, thus it is required to go through the material over and over again to ensure there is no requirement or object model missing. Moreover, it is difficult to debug due to the fact that implementation always happens before test in object oriented approach. On the contrary, extreme programming allows for early testing thus it is easier and more smooth to debug and refactor. However, extreme programming does have drawbacks. The system architecture is not intuitive during the initial iteration, and after several iterations and when we wanted to identify the system hierarchy and decompose the system, we found the objects are highly coupled and it was hard to do the decomposition.