|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| <<6/2>> 會議記錄 | | | | | | | | | |
| 會議日期 | | 2014/6/2 | | | | | | | |
| 時間 | | 22:00~24:00 | | | | | | | |
| 地點 | | Skype | | | | | | | |
| 主持人 | | 蔡宗翰 | | | | | | | |
| 記錄者 | | 吳佳倫 | | | | | | | |
| 目的 | | 討論 Project | | | | | | | |
| 參與者 | | | | | | | | | |
| 姓名 | | E-mail | | | | | 角色 | | |
| 蔡宗翰 | | b99902066@ntu.edu.tw | | | | | 組長 | | |
| 林映孜 | | d98944002@ntu.edu.tw | | | | | 組員 | | |
| 范哲誠 | | d02922030@ntu.edu.tw | | | | | 組員 | | |
| 黃奕軻 | | b99902032@ntu.edu.tw | | | | | 組員 | | |
| 劉宗瑋 | | b99902100@ntu.edu.tw | | | | | 組員 | | |
| 呂俊宏 | | enricolu@gmail.com | | | | | 組員 | | |
| 吳佳倫 | | bingo4508@gmail.com | | | | | 組員 | | |
| 會議議程 | | | | | | | | | |
| 1. Discuss project. 2. Discuss process’ pros and cons. | | | | | | | | | |
| 會議討論議題 | | | | | | | | | |
| 5/27 16:20~18:00 @ 德田館R440   1. Review project   We identified some works required to be finished before next meeting including redesign class diagram (with or without patterns) and refining system requirement according to current class diagram.  To make GUI implementation easier, we decided to assign works of dialog design to each member who was in charge of different system components so that our design could be more close to the flow path how clients invoke some functions of our application.   1. Discuss WBS schedule   We reviewed our WBS and found out the necessity of providing a new schedule due to the some changes on work assignment comparing with our original one and delay on our old schedule. The detailed discussion would be continued in the next meeting.  5/29 18:30~22:00 @ 德田館R439  1. Discuss Homework   * + HW1 * A List data structure is implemented with a String array which can contain a series of String objects.      * We can acces List by calling the get() method with an index, and know how many Strings inside the List with a public attribute: length.      * Furthermore, another data structure called SkipList which consists of a series of SkipNodes.      * Each SkipNode can be accessed by invoking the getNode() method in SkipList with an index. And we have the idea about the size of SkipList with its size() method.      * Now we have to traverse both List and SkipList to print out those object items in the two different data structures for some purpose.      * Issue:   + If there’s new kind of list, we need to modify traverse.   + We need to know how to print different objects in lists. * Redesign   + Encapsulate What Varies.      * + Abstract Common Behaviors      * + Compose or Delegate Abstract Behaviors      * + Looks like “Strategy Pattern” * Another Version      * + HW2 * A compiler subsystem contains classes such as Scanner, Parser, ProgramNode, and BytecodeStream.      * The client classes need to use Scanner, Parser, ProgramNode, and BytecodeStream to compile some code.      * Redesign      * + Encapsulate What Varies.   + Abstract Common Behaviors (skip)   + Compose or Delegate Abstract Behaviors      1. Modify WBS of further works    * Implement the first version before 6/5.    * Aggregate and test before 6/9.   6/2 22:00~24:00 @ Skype  1. Importer –  a. 從facebook上的資料抓到allinone table  b. 由於使用multi thread增加速度，遇到thread exception的問題  c. 目前已完成抓FB張貼的文章  d. 繼續將FB的Comment整合在allinone table  2. Layout –  a. 若要把NodeXL的Taskpane用dependency, 需要把裡面用到Excel workbook的地方全部註解  掉, 不知是否可行, 待試  b. TestLayoutControl 可以run, 待測  3. Process pros and cons  Pros:   1. Can track progress, understand (a) which to do in next stage, (b) responsibility of everyone. 2. If use properly, problem can be solved efficiently. To use process precisely, you have to analyze your problems and model them first. Therefore, process can constrain problem scope and provide information whether it has been use well enough. For example, if a software project is light-weight, using large scale process cannot lower down your cost of management. (This is case of misuse. Should carefully estimate tradeoff issue.) 3. Can be organized in any way, hierarchy align can make responsibility clear. You may seldom change the highest one, since most of time the direction won’t change too far. Therefore the correct (working-well) parts don’t need to be changed. 4. Consistent behavior for solving same problem. (Moreover, the useful one can be principle or rules.)   Cons:   1. “Misuse case”, “Not completely use case” will cause some consequence. 2. Cannot remove risks. Can only decide what to do when it happens. Different process has different risks. 3. 魚與熊掌不可兼得. Tradeoff issue always exist. There is no perfect process without cons. For example, flexibility and complexity and efficiency. | | | | | | | | | |
| Action Item後續處理項目 | | | | | | | | | |
| 編號 | 處理動作 | | | 負責人員 | | 處理期限 | | 狀態 | 備註 |
|  | WBS draft | | | 黃奕軻 | | 4/1 | | **Closed 4/1** | 蔡宗翰 review |
|  | 提供Github public key | | | 全員 | | 4/1 | | **Closed 4/1** |  |
|  | 新版WBS | | | Eric | | 4/4 | | **Closed** |  |
|  | 加入Implementation的詳細內容 | | | 全員 | | 4/7 | | **Closed** | On google doc |
|  | 繪製High Level Architecture | | | Eric | | 4/8 | | **Closed** |  |
|  | 上傳前次的 Meeting Minutes | | | 吳佳倫 | | 4/8 | | **Closed 4/1** |  |
|  | WBS—分析、設計階段認領 | | | 全員 | | 4/7 | | **Closed** |  |
|  | 設定 Github key, 並確認成功上傳至少一次 | | | 全員 | | 4/8 | | **Closed** | 奕軻 review |
|  | Homework6 | | | 吳佳倫 | | 4/8 | | **Closed 4/08** |  |
|  | Check all Commit | | | 黃奕軻 | | 4/8 | | **Closed 4/08** |  |
|  | Review System Architecture | | | 全員 | | 4/14 | | **Closed 4/14** | 映孜review |
|  | Code Survey | | | 全員 | | 4/14 | | **Closed 4/14** |  |
|  | Terminology and Naming | | | 呂俊宏 | | 4/14 | | **Closed 4/14** |  |
|  | Homework 7-1 | | | 蔡宗翰 | | 4/15 | | **Closed 4/15** |  |
|  | Homework 7-2 | | | 呂俊宏 | | 4/15 | | **Closed 4/15** |  |
|  | Homework 8-1 | | | 林映孜 | | 4/21 | | **Closed 4/21** |  |
|  | Homework 8-2 | | | 劉宗瑋 | | 4/21 | | **Closed 4/21** |  |
|  | Homework 9-1 | | | 范哲誠 | | 4/28 | | **Closed 4/28** |  |
|  | Homework 9-2 | | | 廖尉棠 | | 4/28 | | **Closed 4/28** |  |
|  | User requirement | | | 全體 | | 4/28 | | **Closed 4/28** |  |
|  | System requirement | | | 全體 | | 5/4 | | **Closed 5/5** |  |
|  | Interface requirement | | | 全體 | | 5/5 | | **Closed 5/5** |  |
| 1. r | Homework 10 | | | 吳佳倫 | | 5/6 | | **Closed 5/5** |  |
|  | Refine System Architecture | | | 全體 | | 5/13 | | **Closed 5/12** |  |
|  | Presentation of Requirements & Architecture | | | 蔡宗翰 | | 5/6 | | **Closed 5/6** |  |
|  | Modifying requirement | | | 全體 | | 5/12 | | **Closed 5/12** |  |
|  | Homework 11 | | | 黃奕軻 | | 5/13 | | **Closed 5/12** |  |
|  | Re-assign duties for further works | | | 全體 | | 5/13 | | **Closed 5/13** |  |
|  | Homework 12 | | | 劉宗瑋 | | 5/12 | | **Closed 5/19** |  |
|  | Class Diagram | | | 全體 | | 5/19 | | **Closed 5/26** |  |
|  | Homework 13 | | | 林映孜 | | 5/27 | | **Closed 5/26** |  |
|  | Implement | | | 全體 | | 6/7 | | **Ongoing** |  |
|  | Unit-testing | | | 全體 | | 6/7 | | **Ongoing** |  |
|  | Homework14 | | | 蔡宗翰 吳佳倫 | | 6/3 | | **Closed 6/2** |  |
|  | Redesign the class diagram and requirement refine | | | all | | 5/29 | | **Closed 5/29** |  |
|  | WBS modification | | | all | | 6/3 | | **Ongoing** |  |
|  | 整理 process’ pros and cons | | | 黃奕軻 | | 6/3 | | **Open** |  |
| 下次會議 | | | | | | | | | |
| 日期 | | | 時間 | | 地點 | | | | |
| 6/3 Tuesday. | | | 16:30 | | R10? | | | | |