# 軟體工程 Homework #2

## Part 1.

### a. Describe the major requirements engineering activities

#### 1. Requirements elicitation and analysis (需求誘發與分析)

需求誘發的目的在獲得使用者對於系統的需求，其方法有面談、問卷、使用者觀察、研討會、腦力激盪、使用案例、雛形法，在開發軟體前先透過一些手段與方法來引導、收集使用者真正的需求，並且分析需求是否可行、正確 (Correct)、完整(Complete)、沒有衝突(Conflict-free)。

#### 2. Requirement specification (需求規格化)

經過需求誘發與分析所得到的需求屬於使用者需求，再透過分析後將使用者需求轉為系統需求，而在需求規格化活動中會將需求規格化，寫成具體的文件作為後續設計依據，也作為雙方的合約基礎。

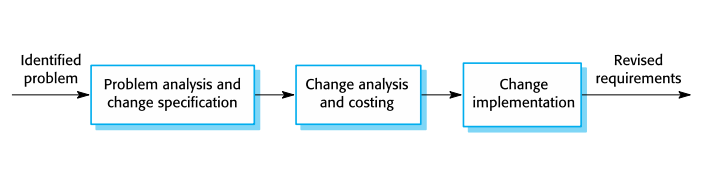
#### 3. Requirements validation (需求驗證)

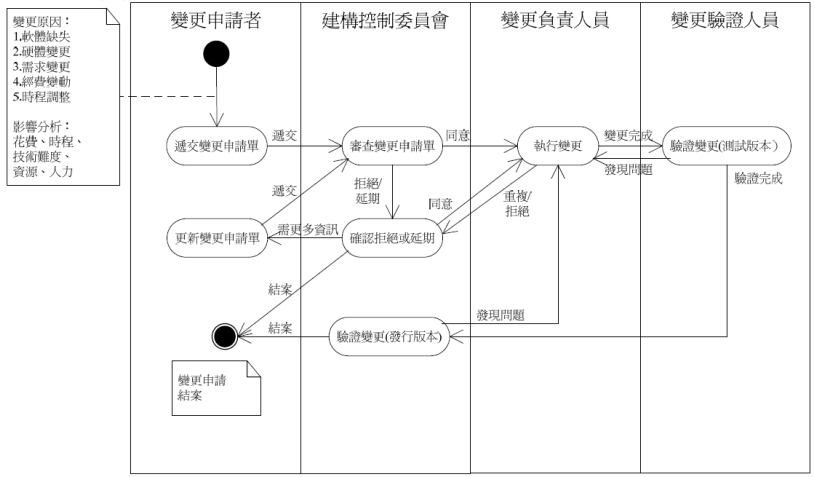
需求驗證主要在確認所定義的需求規格是否正確，是否符合使用者本意，並且確認系統是否為可驗證的。

#### 4. Requirements management (需求管理)

即使需求分析做得很仔細，只要環境持續在變化，那麼使用者就還是有可能會在軟體開發的過程中改變、新增需求，此時就須要對改變或新增的需求進行分析和協調來達到對需求的管理。

### b. Draw an activity diagram to explain the process of requirements change management.





三個步驟來決定新增或改變的需求是否被接受：

#### 1. Problem analysis and change specification

在問題分析及改變規格的環節，將對提出的改變進行分析其是否為有效的，並將分析結果回饋給提出改變的對象，使其可以提出更完善的請求或是放棄請求。

#### 2. Change analysis and costing

以系統需求的基本知識和可追朔的資訊對所提出改變可能造成的影響和時間及成本花費進行評估，分析結束後將決定是否要進行改變。

#### 3. Change implementation

在實作需求改變的階段將對需求文件以及其他必要的系統設計及系統實作進行修改，有組織性的文件才能在改變發生時比較容易修改。

### c. Describe the concept of traceability in requirements management. How can the traceability of requirements be used in the software development process?

對於需求變更的分析並不是件容易的事情，當系統需求、設計、程式碼很多很複雜時，就會需要可追朔的資訊協助開發端知道變更會影響到的範圍，通常會建立需求追朔表來記錄需求與需求之間的關係、並可以追朔需求的來源以及它相對應的設計與實作。

### d. Discuss the associated Requirements Engineering (RE) activities within Scrum framework. What kinds of Scrum meeting and artifacts are related to RE activities? What kinds of RE techniques or practices, such as prototyping, can be used in Scrum? and Why?

#### 1. Requirements elicitation and analysis (需求誘發與分析)

需求誘發的目的在獲得使用者對於系統的需求，在Scrum中，Product Owner會去了解Stakeholders的想法以及他們的需求，並且Product Owner也會參與user story的過程及討論。

#### 2. Requirement specification (需求規格化)

在需求規格化活動中會將需求規格化，寫成具體的文件作為後續設計依據，在Scrum中，會進行Sprint Planning Meeting，將討論和定義如何實作user story，並把實作的方法和設計加入Sprint Backlog。

#### 3. Requirements validation (需求驗證)

需求驗證主要在確認所定義的需求規格是否正確，在Scrum中會進行Sprint Review Meeting，Product Owner將參與會議並且確認該Sprint的potentially shippable product increment，是否有符合需求，並且在Product Backlog Refinement來進行調整。

#### 4. Requirements management (需求管理)

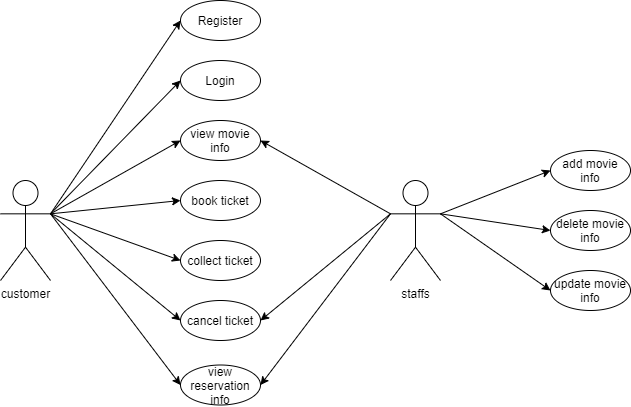
需求會因環境變化而改變，因此需要對需求進行管理，在Scrum中，需求將被視為Product Goal，而達成Product Goal所要進行的任務會被列在Product Backlog裡，Product Owner需要對Product Backlog進行ordering，並且若有任何想要更改的想法都要與先Product Owner提議，最後由Product Owner決定是否採取變更。

#### Using RE techniques in Scrum

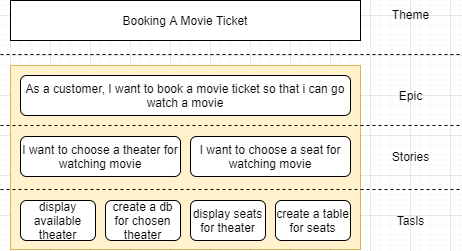
在Scrum中引用Requirements elicitation作法，將能夠幫助Product Owner更有效率地確認以及理清使用者需求，以此來建立更完善的、更可以代表使用者的Product Backlog。使用Prototyping技術將有利於使用者了解當前進度以及更快速地給予開發端使用者回饋，以利開發端更清楚確認需求。引用Requirements validation作法，將利於開發端驗證需求是否正確、完整。

## Part 2.

### a. Draw the use case diagram for the system. The use case diagram must have at least two actors, such as customer and staff.



### b. Write the user story for booking a movie ticket with the user story template.



### c. Write the user case for booking a movie ticket with the use case template.

|  |  |
| --- | --- |
| ID : | 0001 |
| Title : | Booking a movie ticket. |
| Description : | Register an account and starting to book a ticket. |
| Primary Actor : | Registered member |
| Preconditions : | Registered member, and login the system to view the information of movies. |
| Post conditions : | Registered member, and have received notification for the reservation and payment info. |
| Main Success Scenario : | 1. Select a movie 2. Click “book tickets” button to make a reservation 3. Choose a date 4. Choose a theater 5. Choose the time and preferred seats 6. Confirm the selections and make a payment 7. System show the payment and ticket info |
| Fail Scenario : | The payment was failed and led to a page of notification to tell customer that the payment was failed. |
| Frequency of use : | 20000 reservations per day |
| Status : | Doing |
| Owner : | Booking system |
| Priority : | 1 |

### d. List three essential non-functional user requirements for the system. Make proper assumptions if necessary

#### 使用性：

使用介面應該利於使用者操作、閱讀，且排版應乾淨整齊

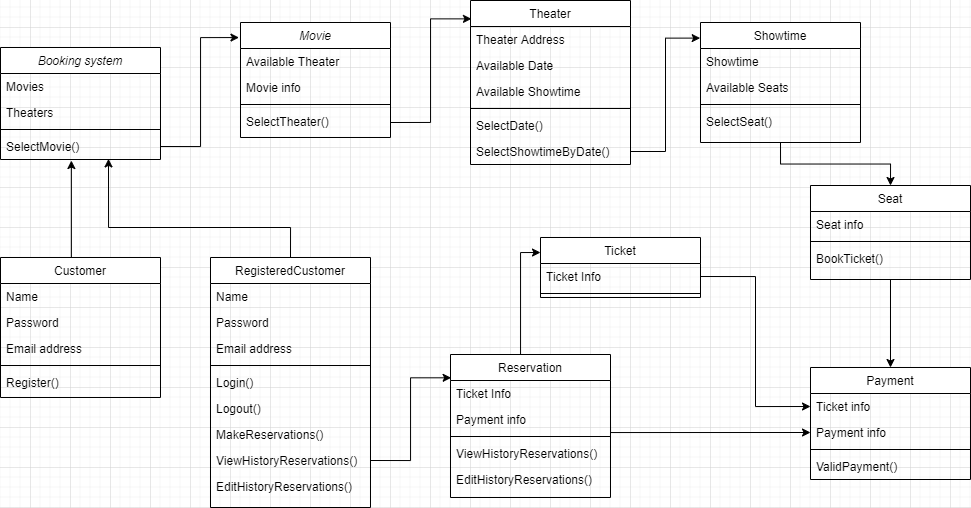
#### 安全性：

對用戶個人資料以及金融資訊應進行高度保密

#### 反應速度：

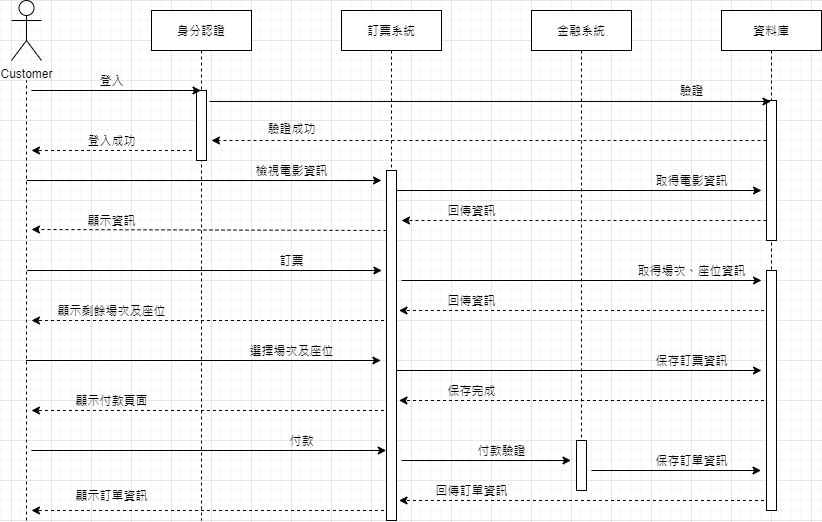
使用者在操作時，系統反應速度應在6秒內

### e. Identified the possible domain classes, such as Customer, Movie, Theater, Showtime, Seat, Ticket, Payment, Reservation, and draw the UML class diagram for the system. Please give essential attributes and methods for each class and specify their visibility. Show appropriate relationships between the classes and specify the multiplicity for each relationship.



### f. Draw the UML sequence diagrams based on your class diagram for the scenarios.

#### a customer successfully books a movie ticket.



#### a customer cancels the booked tickets.

