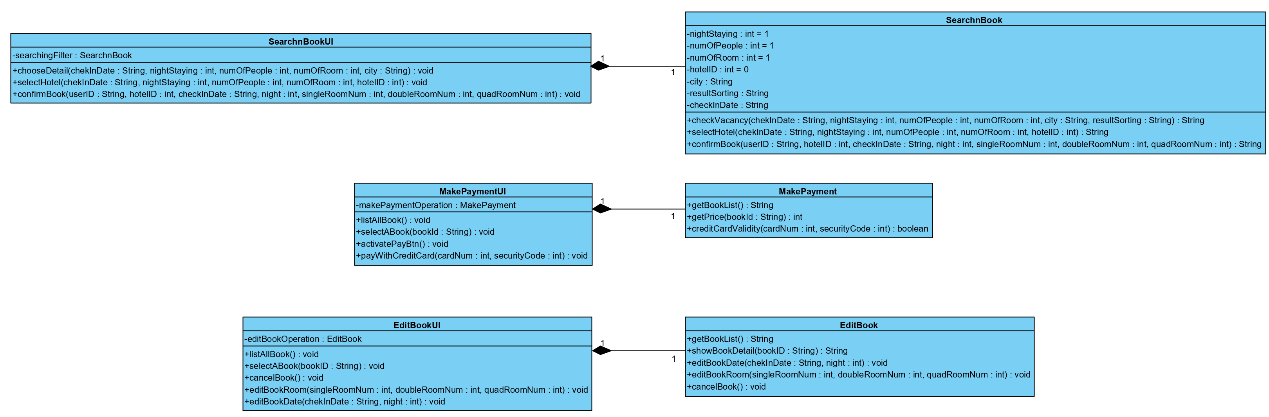
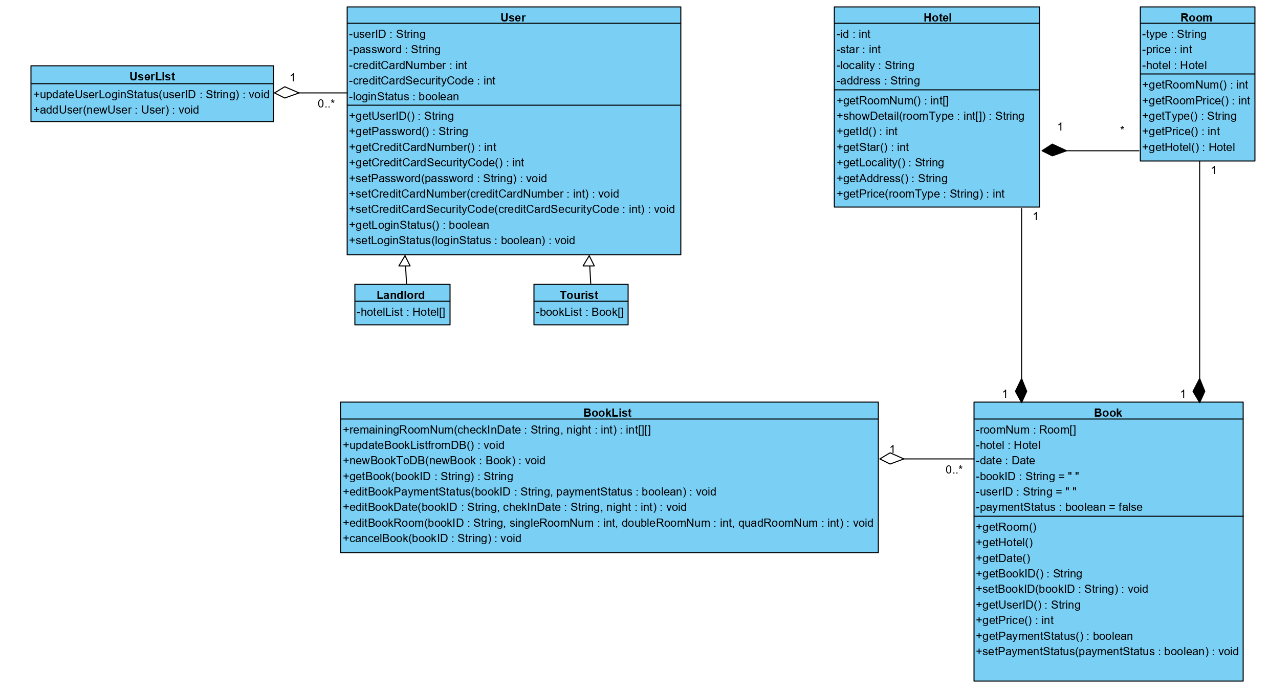
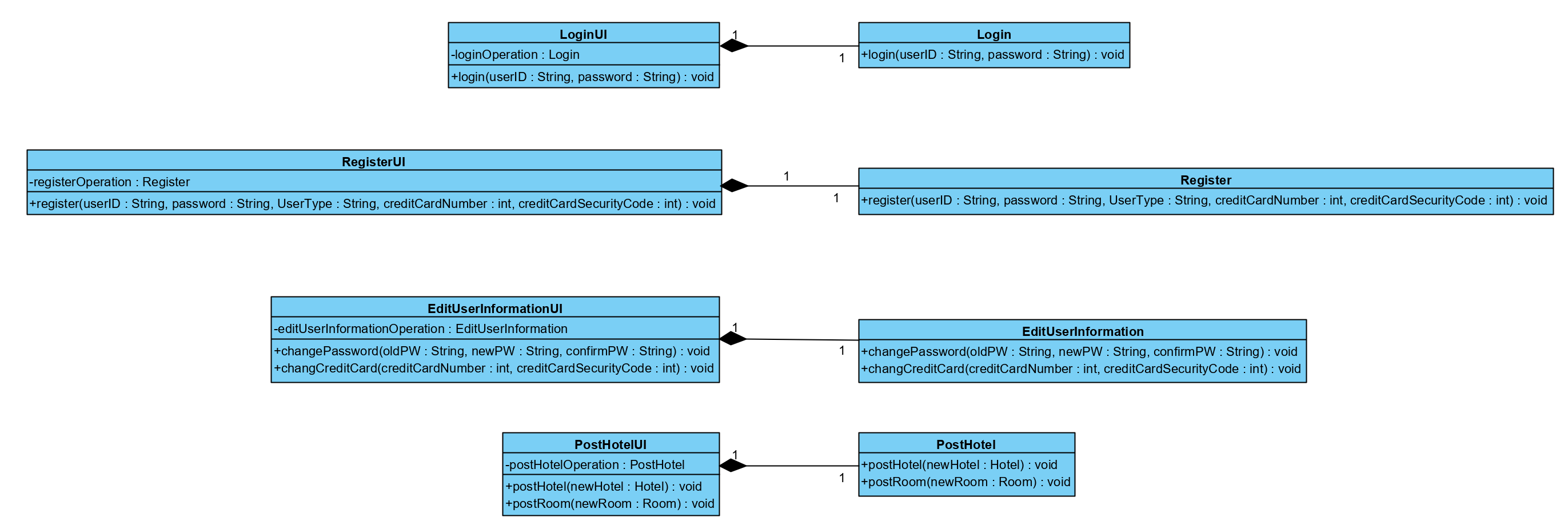
1.





2.

Facade Pattern是讓各個子系統統一使用一個高階的接口系統，並且降低各子系統的關聯性，好處是當子系統中有更改時，較不容易因為有其中一個子系統有更改牽一髮而動全身，進而造成未預期的錯誤，在開發時，不同的子系統因為彼此關聯性低，所以團隊中不同人可以分工負責其中一個子系統，而且可以同時進行coding，也不太需要關心其他子系統的內部設計。

3.

Reuse-Pattern 在user的部分，因為分為landlord跟tourist兩者都是user，所以在login時的userid與password等相關參數與函數是可以reuse的。

Modularization & Componentization 在使用各項功能如login、register、posthotel等功能時，因為是設計成不同的class且有獨立的UI class，所以在新增功能或是修改內部功能時比較不會有互相影響。

Low-Coupling and High Cohesion 我們把hotel、user、booking等不同分類的class分開，且各個行為使用不同的UI class，且盡量讓各class間的互動減少。