Szoftvertechnológia és – technikák
6. gyakorlat
Használati eset diagram, UML összefoglalás

1. A gyakorlat menete

Ez az alkalom nagyrészt önálló munkából áll. Az elején van egy kis vezetett rész a Use Case modellhez, de a többi feladat azt hívatott mérni, hogy mennyire értettétek meg az eddigieket. Ez egy kiváló lehetőség gyakorolni a számonkérésekre, mivel korábbi ZH vagy vizsga feladatokat oldunk meg önállóan. Vezetett feladatok

Használati eset diagram

Feladatod a Megrendelőtől kapott szöveges leírás alapján egy átfogó modellt készíteni, amiben látszanak a rendszer főbb funkciói és résztvevői. Figyelj arra, hogy a leírásból szűrd ki a szoftverfejlesztés szempontjából felesleges részeket!

A CyberPub rendszer a modern technológia csúcsa, ennek megfelelően minden funkció megvalósítása a legmagasabb minőségi követelményeknek is eleget kell, hogy tegyen. A CyberPub rendszer szoftveres részeit a BProf4King Kft készíti el a mellékelt szerződés szerint. A fejlesztés a Vállalkozó számítógépein történik.

A CyberPub rendszer lényege, hogy az étterembe betérő ügyfeleket egy automata robot fogadja, ő veszi fel a rendelést, hozza ki az ételt és rajta keresztül lehet a számlát is intézni. A robot kerekeken gurul, négy karral hozza, ill. viszi az ételt és a tányérokat. A vendég érkezésekor a robot rendeli az asztalt a vendéghez és oda is vezeti az ügyfelet, ahol az ételt el fogja fogyasztani. Az ételrendelés á la carte rendelést jelent az étlapról, amit a robot megmutat a vendégnek, a vendég ez alapján leadja a rendelést, amit a robot továbbít a konyhai alrendszernek. Az ételek felszolgálását a robot végzi, a vendég ebben nem vesz részt. Az étel elfogyasztása a vendég magánügye, a CyberPub rendszer ezt nem felügyeli, szabályozza, vagy figyeli meg. Az étel elfogyasztása után a vendég jelzi a fizetési szándékát a robotnak. A fizetésnél elsőként meg kell adni a fizetés módját, majd ezek után jöhet maga a fizetés. A fizetés során lehetőség van borravaló adására is. A fizetés lebonyolítását a banki alrendszer végzi.

Segítség: a modell a robot szempontjából írja le a rendszert!

2. Önálló feladatok

1. Használati eset diagram (12 pont)

Egy bűvész társulat működését modellezzük. A társulatban vannak bűvészek, segédek, és bűvészmesterek. A mesterek olyan különleges bűvésznek számítanak, akik speciális feladatokat is el tudnak végezni. A társulat legfontosabb tevékenysége a bűvész show-k megtartása egy színház épületében, amin mindig egy vagy két bűvész vesz részt. Egy show-t mindig elő kell készíteni, ami néha a színpad átépítésével jár. Az is előfordulhat, hogy a show részeként a közönségbe a segédek beépített emberként épülnek be. A segédeket a mesterek képzik ki, hogy egyszer ők is bűvészekké válhassanak. A kiképzés minden héten csütörtökön délután történik. A show-k menetének kidolgozása csak és kizárólag a mesterek feladata. Ezen kívül – bár ez nem tartozik szorosan a társulat működéséhez – a mesterek néha külföldi utakon is részt vesznek, hogy más bűvész társulatokkal kölcsönösen megosszák a tapasztalataikat.

2. Osztálydiagram (20 pont)

Egy kollégium struktúráját modellezzük osztálydiagram segítségével. A kollégium épületekből áll, melyeknek tároljuk a nevét. Az épületeken belül szobák találhatók. Egy szobának van szobaszáma (szöveges formátumban megadva), illetve tudjuk, hogy melyik hallgatók laknak épp benne. Egy hallgató egyszerre csak egy szobában lakhat. A hallgatókról tároljuk, hogy hányadik félévben járnak, viszont ez csak lekérdezhető, kívülről nem módosítható. A senior hallgató egy különleges hallgató, aki egy adott hallgatót egy adott gondnoknál fel tud jelenteni. Gondnokból több is lehet a kollégiumban. Minden gondnok több szobához is hozzá lehet rendelve, mint felelős, de egy szobához csak egy gondnok lehet hozzárendelve. Fontos, hogy egy szobánál tudnunk kell, hogy ki a hozzárendelt gondnok, míg a gondnoknak is tudnia kell, hogy mely szobák tartoznak hozzá. Egy gondnok képes egy hallgatót megbüntetni. A büntetésnek kétféle kimenetele lehet: 1) a hallgatót kirúgják a kollégiumból, vagy pedig 2) elzárás lesz a büntetése. Az elzárás helyét és idejét nem kell modelleznünk. Egy gondnoknak opcionálisan lehet egy kedvenc hallgatója, akit sosem fog megbüntetni. A kollégium lakóinak (vagyis a hallgatóknak és gondnokoknak) mind van egy szöveges azonosítója, melyet szeretnénk minden lakóról egyszerre, valamilyen közös módon lekérdezni. Ezen kívül minden lakónak (beleértve a gondnokokat is) központilag megtilthatjuk vagy engedélyezhetjük a belépést egy adott szobába.

3. Szekvenciadiagram (12 pont)

Készítsen szekvenciadiagramot az alábbi kódrészlet alapján! A szekvenciadiagram a **SocialMediaService** objektum **userLoggedIn()** metódusának meghívásával kezdődjön! Ügyeljen a helyes jelölések használatára!

```
public class SocialMediaService {
  private WebView view;
  private User user;

  public SocialMediaService(WebView v, User u){
    this.view = v;
    this.user = u;
  }

  public void userLoggedIn() {
    view.init();
    int s = user.getSelection();
    switch (s) {
    case 1:
      writeMessage();
      break;
    case 2:
      createNewStory();
      break;
    }
  }

  public void writeMessage() {
    if (user.getNumOfFriends() == 0)
      view.showAlertMessage();
  }

  public void createNewStory() {}
  public void createNewGroup() {}
}
```

```
public class User {
  private SocialMediaService service;
  public void set(SocialMediaService s) {
    this.service = s;
  }

  public int getSelection() {}
  public int getNumOfFriends() {}
  public void changePhoneNumber() {}
  public void changeName() {}
}

public class WebView {
  private ArrayList<Component> components;
  public Component getComponent() {}

  public void init(){
    this.showComponents();
  }

  public void showAlertMessage() {}
  public void dispose() {}
  public void showComponents() {}
}

public class Component {
  private String id;
  private int x;
  private int y;
}
```

4. Állapotgép (12 pont)

Egy ZH javító rendszer működését modellezzük. Kezdetben a rendszer készenléti állapotban van, majd, ha érkezik egy ZH, megkezdi az előfeldolgozást. Amennyiben az adminisztrációs adatok nincsenek rendben, visszatérünk készenléti állapotba, ha rendben vannak, akkor elkezdődik a ZH javítása. Mindig, amikor a javításból ki vagy belép a rendszer, naplózás történik. A javítás a teszt feladatok javításával kezdődik. Ha a teszt sikertelen, akkor a ZH javítása befejeződik. Ha a teszt sikerült, akkor ezután a rendszer a két nagy feladatot egyszerre javítja ki. Amint ezzel végzett, a pontszámítás következik: először a teszt, majd a nagy feladatok pontszámát számítja ki a rendszer saját belső algoritmusai alapján. A pontszámítás után a ZH javítása befejeződik. A ZH javítása közben bármikor felmerülhet a plágium gyanúja. Ilyenkor a ZH egy plágium ellenőrzésnek lesz alávetve, ennek eredményétől függően vagy pontosan ott folytatódik a javítás, ahol abbamaradt, vagy pedig a ZH javítása befejeződik. Akárhogyan is fejeződik be a ZH javítása, a rendszer a kész állapotba kerül, és emailben kiküldi a hallgatónak az eredményét.

5. Aktivitásdiagram (12 pont)

Egy hivatalnok elektronikus iratok átnézésével kezdi a napot. Az iratokat addig nézi át, ameddig nem talált egy kitöltendő iratot, vagy pedig le nem telt a munkaideje. Ha letelt a munkaideje, akkor hazamegy, és ezzel a munkája véget ér. Ha talált egy kitöltendő iratot, akkor a hivatalnok először előkészíti, majd kitölti azt. Ezután emailben elküldi a főnöknek a kitöltött iratot és folytatja az iratok átnézését. A főnök, miután megkapta a kitöltött iratot, átnézi azt. Amennyiben mindent rendben talál, kinyomtatja az iratot, és ezzel a főnök munkája pillanatnyilag véget ér. Ha valami hibát talál a kitöltött iratban, akkor a főnök először mérgeskedik egyet, majd emailben tájékoztatja a hivatalnokot arról, hogy ki van rúgva. A hivatalnok, amint erről értesül, hazamegy, és ezzel a munkája véget ér.