Project

Örebro Universitet

Maria Alkeswani

Jamal Alaskari

Mohamad Daadoush

Grupp I

Innehållsförteckning

Introduction

Att ge användarna ett schemaverktyg vars uppgift skall vara att automatiskt generera ett effektivt och bra schema av informationen som ges/inmatas till programmet. Schemaverktyget skall ha tagit hänsyn till dubbel bokningar av deltagare och stationer, lunch, fikaraster och raster mellan tävlingar. Schemaverktyget ska ha möjlighet att uppdatera schemat ifall någon form av fördröjning skulle inträffa under deltävlingarna och då även kunna uppdatera det schemat och göra det aktuellt igen.

Requirements

Functional requirements:

- Systemet ska generera ett schema med alla deltävlingar som är både effektivt och rättvist för deltagarna.
- De skall även vara av fungerande typ, inga dubbelbokningar av stationer.
- Systemet kommer att sortera dem, så att deltagarna som är i samma ålder och kön ska spela tillsammans.
- Olika grupper kan samtidigt närvara på olika platser.
- Vi måste få veta hur många spelare som kommer att spela i varje grupp. Detta är för att veta hur lång tid tävlingen kommer att hållas för varje grupp.
- Tävlingarna är i 3 olika sporter, running, jumping, throwing. I finalen, 3 priser kommer att ges ut för var och en av dem.
- Flera grupper kan inte vistas på samma plats samtidigt.
- En spelare kan endast spela en tävling på en plats. Spelaren kan inte vistas på två olika platser samtidigt.
- Vi måste tänka på rasten mellan olika tävlingarna, fika tider, lunchtiden och raster mellan tävlingarna.
- In running disciplines, de som kommer spela i finalen, måste ha löpat 60m, 200m, 800m, 1500m, 3000m, och 60m hurdles. Dessutom så måste spelarna ha fått lägsta löpnings-tiden. De bästa 6 spelare ska spela i finalen.
- Det finns 4 kategorier av jumping, long, triple, high, pole. I ”long” samt ”triple” jumps, varje athlete kan göra upp till 2 trials, det bästa av dem räknas som det finala resultatet. I high jump samt pole, spelaren börjar i samma level, i ”high” rundan. High level kommer att öka efter varje gång. De bästa 6 spelare ska spela i finalen

- I throwing runda, varje spelaren har rätt att göra 4 trials , det bästa av de 4 är räknad som det finala resultatet. Endast shots som är inom gränsen kommer att räknas. De bästa 6 spelare ska spela i finalen

- De som får bästa resultat i tävlingarna, kommer att spela i finalen mot varandra. I slutet av finalen kommer vi att ha 3 vinnare. En för throwing, en för jumping och en för running.

Non-Functional requirements:

- Tävlingen är tidsbegränsad till ett visst antal dagar.

- De tävlande kan vara med i flera olika grenar i tävlingen, således skall verktyget ha detta i hänseende när det genererar schemat.

- Systemet ska kunna hantera tillräckligt många spelare utan prestationsförlust

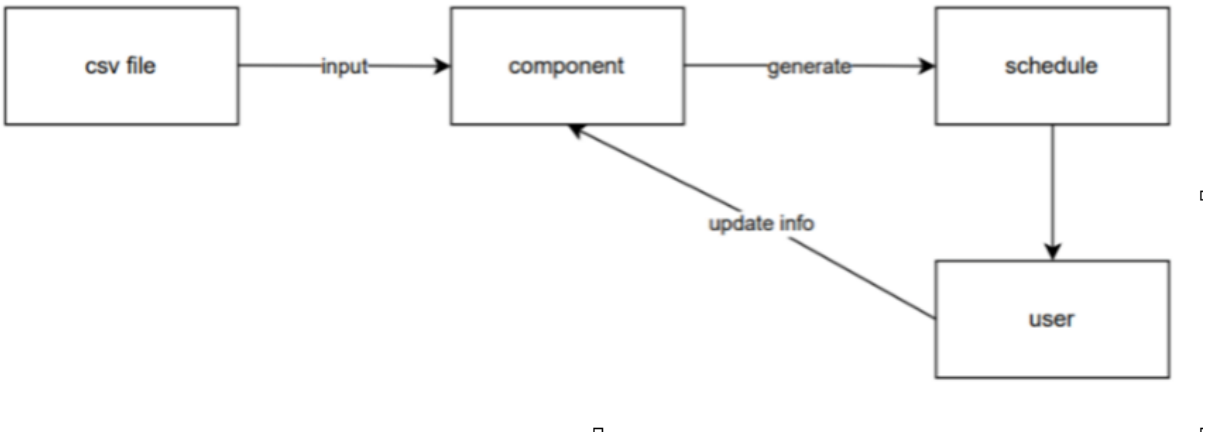
- Input och output ska vara en CSV-fil.

- Varje spelare ska ha tillräckligt med tid för paus och den ska beräknas per deltagargrupp.

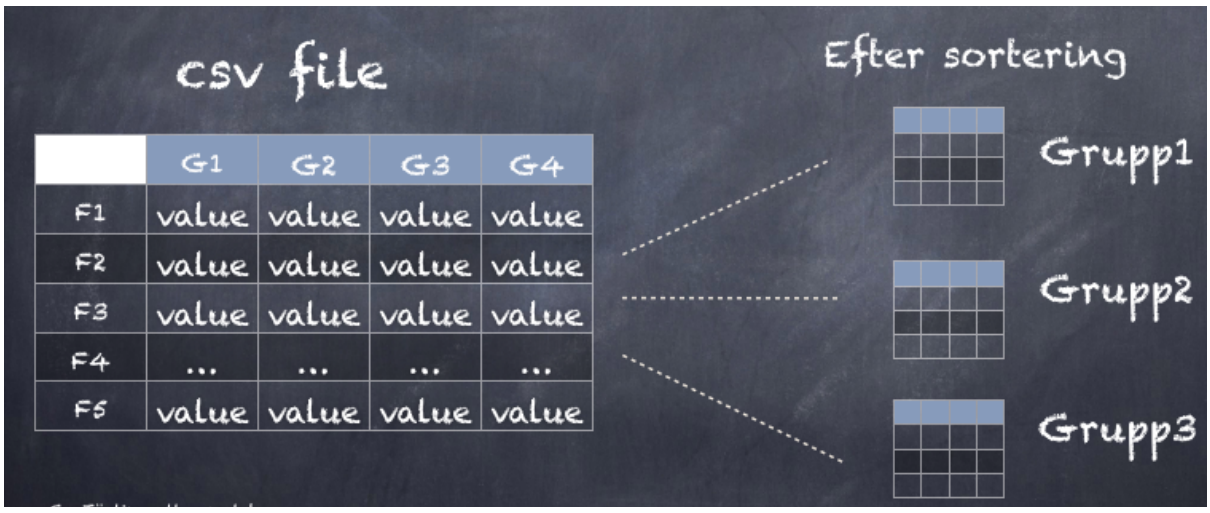
- Systemet bör eliminera alla deltagare som är yngre än 7 år, detta kommer att rapporteras eller nämnas i den genererade CSV-filen.

- Ta hand om korrekt ordning av data i inmatnings CSV-fil.

System Architecture



- Programmet kommer att ta spelarnas information från en CSV-fil och använda denna data i programmet.
- Varje komponent tolkar filen enligt en specifik algoritm så att filen sorteras i grupper efter ålder och kön. T.ex:



- Programmet kommer att kontrollera om spelarna deltar i tävling med sin grupp eller inte, vilket innebär att undersökningen sker efter att spelarnas filer har sorterats i grupper.

- Programmet kommer att gå igenom varje fil och kontrollera om det finns en spelare som inte deltar i ett spel 0 eller deltagare 1. T.ex: där F = spelare och G = tävling.

	G1	G2	G3	G4
F1	1	1	0	1
F2	1	0	1	1
F3	1	1	0	0
F4	1	1	0	1

- Sedan kommer en annan algoritm att använda deltagarmatrisen för att hitta det totala antalet deltagare för varje tävling till varje grupp. T.ex: för grupp 7_8_F är 4 spelare som är köra först tävling och 3 spelare som är köra andra tävling,etc.

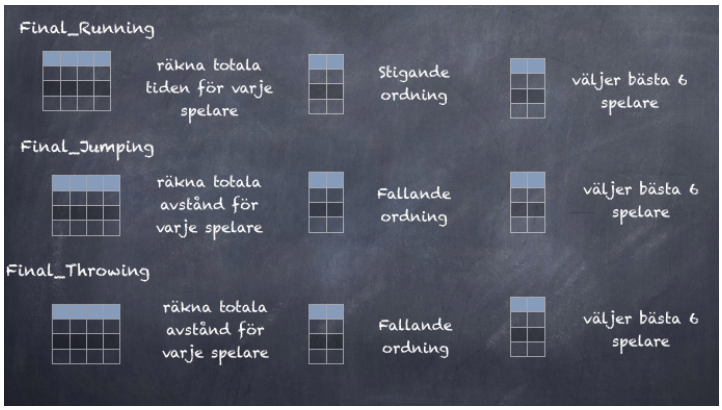
G1	G2	G3	G4
4	3	1	3

- Efter att ha hittat antalet deltagare kommer programmet att beräkna den uppskattade tiden för en grupp på en station. (mer om uppskattade tiden i Implementation and Testing)
- Efter att operation har utförts på alla gruppfiler kommer vi att få två matriser:

  Antalet spelare matrix.

  Den uppskattade tidsmatrisen utan att tiden ska krockas eller att det sker upprepning av samma tävling.

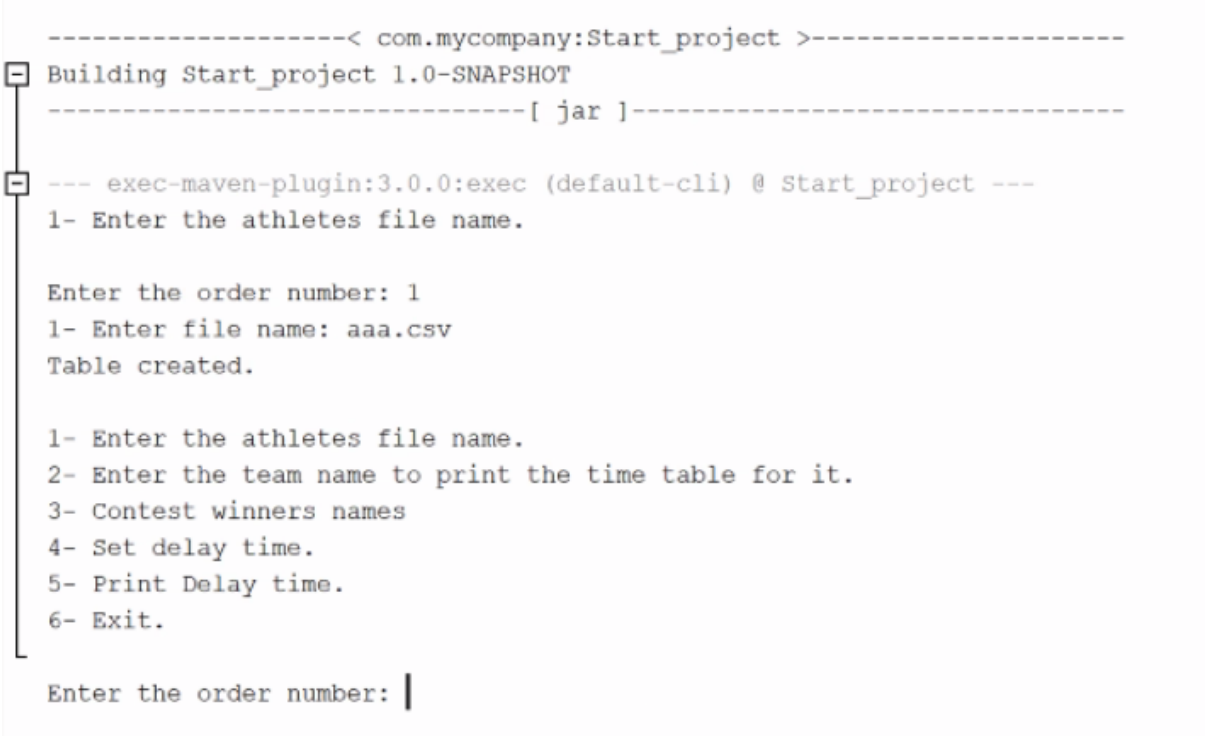
- Dessa processer kommer att upprepas för alla grupper för att få tiden för varje grupp i varje station.
- Programmet kommer att beräkna tiden för finalens tävlingar. Sprint, Jumping och Throwing, endast de 6 bästa spelarna i varje spel kommer att väljas ut för att spela i finalen. (om vi har resultatet till deltagare)



- Programmet fungerar genom att dela upp tävlingstidarna i tre delar från 8:00 till 10:00 följt av en 15 minuters kafferast. Därefter börjar deltagarna att tävla igen från 10:15 till 12:00 följt av en lunchrast i en timme och sedan börjar de tävla igen till 18:00. Samma processer kommer att upprepas nästa dag.
- Nu är allt redo för schemaläggning, CSV_fil sorteras efter ålder och kön, koden har räknat antal deltagare för varje grupp i varje station, tiden har beräknat till varje grupp i varje station och start-sluttid och rasterna har bestämt.
- Efter detta processer koden börjar att placeras grupperna i dem olika stations från klockan 8. De yngsta åldern prioriteras till start t.ex: koden kommer först till grupp 7_8_F i första station(sprint(60-200)) och kontrollerar att tävlingen avslutas innan kafferast, om den avslutar innan kafferast då kör dem den här tävling, om inte då litar koden efter en grupp som kan köra och sluta före kafferast. Därefter kommer grupp 7_8_M i andra station (Running Circal(800-1600-3000-60)) och kontrollerar att tävlingen avslutas innan kafferast. Detta operationer upprepas för alla grupper i alla stations.Tiderna ska begränsas från start tid till kafferast och sen från kafferast till lunch tid och sen från lunch tid till sluttiden.
- Delay funktionen: En specifik grupps tävling kan skjutas upp i tiden ett tag, när något oväntat händer eller om det finns externa händelser. Programmet kommer i så fall att ta hänsyn till detta och reagera efter det, genom att uppdatera schemat.

Interface Description

Program interface :



Programmets interface ser ut som följande: först frågar programmet efter “athletes file name”, där bestämmer man vilken order nummer man ska ha, som i vårt fall är nummer “1” och därefter kommer programmet fråga efter filens namn. Programmets användare skriver namnet på filen som är i form av en CSV_fil. Till exempel “aaa.csv”.
En table kommer att skapas i form av en CSV_fil och flera alternativ kommer att skrivas ut i programmet, som man kan välja mellan. I vårt program så är det 6 alternativ:

1 Enter the athletes file name.
2 Enter the team name to print the time table for it
3 Contest winners names
4 Set delay time.
5 Print delay time
6 Exit

Programmet kommer att vänta på ett svar från användaren i form av ett nummer, användaren skriver ett nummer som kommer att bestämma vad programmets nästa svar är.

Vill användaren därefter få ett tidsschema för en grupp, så väljer man andra alternativet i listan, då kommer programmet att ställa frågan om vilken grupp man vill ha schemat för, och då kan användaren skriva namnet på det gruppen som man frågar efter. Namnet ska se ut som till exempel "9_10_F" där 9_10 förklarar åldern på deltagare i gruppen samt F eller M om deltagarna är male eller female.
Efter att man har svarat med gruppens namn, så kommer programmet att svara med gruppens namn som titel på svaret, samt veckans schema för gruppen som gäller. Därefter så återgår programmet till huvudmenyn med de 6 alternativ som man kan välja mellan, beroende på vad man vill göra med programmet.

Om användaren väljer nummer "3" så kommer programmet att importera data som förklarar vinnarnas information i de 3 olika tävlingarna (om det skulle finnas resultat för de deltävlingarna), detta kommer i sin tur skriva ut namnet på vinnaren, åldern, könen den tillhör samt vilket tävling det gäller.
Efter att programmet har get ut svaret på datan som användaren har frågat efter, så kommer programmet att återgå till "menyn" med de 6 alternativ, för att vänta på ett order nummer från användaren igen.

Om man har delay time som man vill sätta in i schemat, så borde man svara med nummer 4, som förklarar "Set delay time."
Som ett svar så kommer programmet att skriva ut en lista av alla grupper som är med i programmet. Varje grupp har ett nummer så att användaren har möjlighet att välja grupp som har orsakat fördröjningen (delay) (svaret läggs till i form av ett nummer).
Efter att användaren har valt vilken grupp som har delay time, så kommer programmet att ge listan över de stationer så att man väljer vilken station som gäller.
Efter att man har valt stationen som gäller, så frågar programmet efter hur många minuter fördröjningen har blivit. Så fort man har svarat, så kommer programmet att skapa en tabell i form av en CSV_fil som har de nya tiderna som gäller för varje tävling samt radera det gamla filen med de gamla tiderna innan fördröjningen.

Det 5.te valet i listan som förklarar "print delay time" kommer att räkna det totala fördröjnings tiden i hela schemat i alla stationerna och för alla grupper. Så fort man väljer nummer "5" så ger programmet en siffra som förklarar hur många minuter den totala fördröjningstiden gäller i schemat.

Schemat interface :

Tidtabellen är som nämnts ovan en CSV_fil, som kan förklaras lättare och ser tydligare ut om man väljer att visa det som xlsx fil(Excel fil)

Mon		
Sprint(60-200)	Running Circal(800-1600-3000-60)	Jumping long/triple I
7_8_F(08:00-08:05)(08:05-08:10)	7_8_M(08:00-08:56)(08:56-09:01)	9_10_F(08:00-08:57)(08:57-09:02)
15_16_M(08:05-08:08)(08:08-08:13)	17_18_F(08:56-09:24)(09:24-09:29)	17_18_M(08:57-09:12)(09:12-09:17)
11_12_M(08:23-08:27)(08:27-08:32)		13_14_F(09:12-09:18)(09:18-09:23)
17_18_M(09:17-09:19)(09:19-09:24)		
7_8_M(09:27-09:31)(09:31-09:36)		
9_10_F(09:31-09:37)(09:37-09:42)		
coffe break from 10:00 to 10:15		
9_10_M(10:15-10:20)(10:20-10:25)	11_12_F(10:15-11:39)(11:39-11:44)	11_12_M(10:15-10:42)(10:42-10:47)
17_18_F(10:20-10:22)(10:22-10:27)		7_8_F(10:47-11:32)(11:32-11:37)
13_14_F(10:29-10:31)(10:31-10:36)		
11_12_F(11:44-11:48)(11:48-11:53)		
13_14_M(11:48-11:53)(11:53-11:58)		15_16_F(11:32-11:50)(11:50-11:55)
lunch from 12:00 to 13:00		
15_16_F(13:00-13:03)(13:03-13:08)	15_16_M(13:00-13:53)(13:53-13:58)	17_18_F(13:00-13:15)(13:15-13:20)

Här har vi ett exempel på tidtabellen som innehåller datan om tävlingar för olika spel på måndagen som man ser på första linjen av tabellen.
I andra linjen i tabellen så förklaras det vilka stationer det gäller, “Sprint(60-200)” är till exempel sprint stationen som är 60 och 200 meter.
I tredje linjen förklaras det vilken grupp som kommer att spela på denna station, så att namnet på gruppen är 7_8_F . Nummerna 7_8 förklarar åldern för deltagarna i gruppen samt bokstaven, M eller F som förkortning till male och female.
Efter namnet på gruppen så kommer tiden där den här gruppen kommer att starta och sluta i denna station. Samt tiden för pausen för den enskilda gruppen. På måndagen så kommer gruppen 7_8_F att spela klockan 08:00 – 08:05 samt ha paus 08:05 – 08:10. D.V.S. första parentesen är tiden för själva spelet, och andra parentesen är för paustiden för själva gruppen.

kafferasten för alla grupper börjar 10:00 till 10:15, den 9.de linjen i tabellen visar att lunch tiden är satt till 12:00 till 13:00 som den 15.de linjen i tabellen visar.

På onsdagen har man tänkt att det kommer att bli en final om det skulle behövas. Finalen kommer att börja klockan 08:00 om det inte händer något försening som skulle orsaka fördröjningstid (delay time)
Efter finalen så blir det awards ceremony som är tänkt att hållas i ungefär en timme.

CSV_fil interface :

Den huvudsakliga indatakällan för systemet är en lista över de deltagare som kommer att tävla samt de tävlingar som deltagarna kommer att tävla i och den presonliga rekorden om det skulle finnas.
Listan kommer att tillhandahållas i form av en CSV_fil som kommer att innehålla rader och kolumner, varje rad innehåller följande kolumner:
Clup, Name, surname, sex, age, Running, Jumping, Throwing.
Inom kolumnen Sex kommer systemet att använda sig av antingen bokstaven F eller M som motsvarar Female eller Male.
Inom Running kommer det innehålla underkolumner som följande: Sprint, Middle, Long och Hurdles.
Inom Jumping kommer det innehålla underkolumner som följande: Long,Triple, High, Pole.
Inom Throwing kommer det innehålla underkolumner som följande: Shot kolumnen

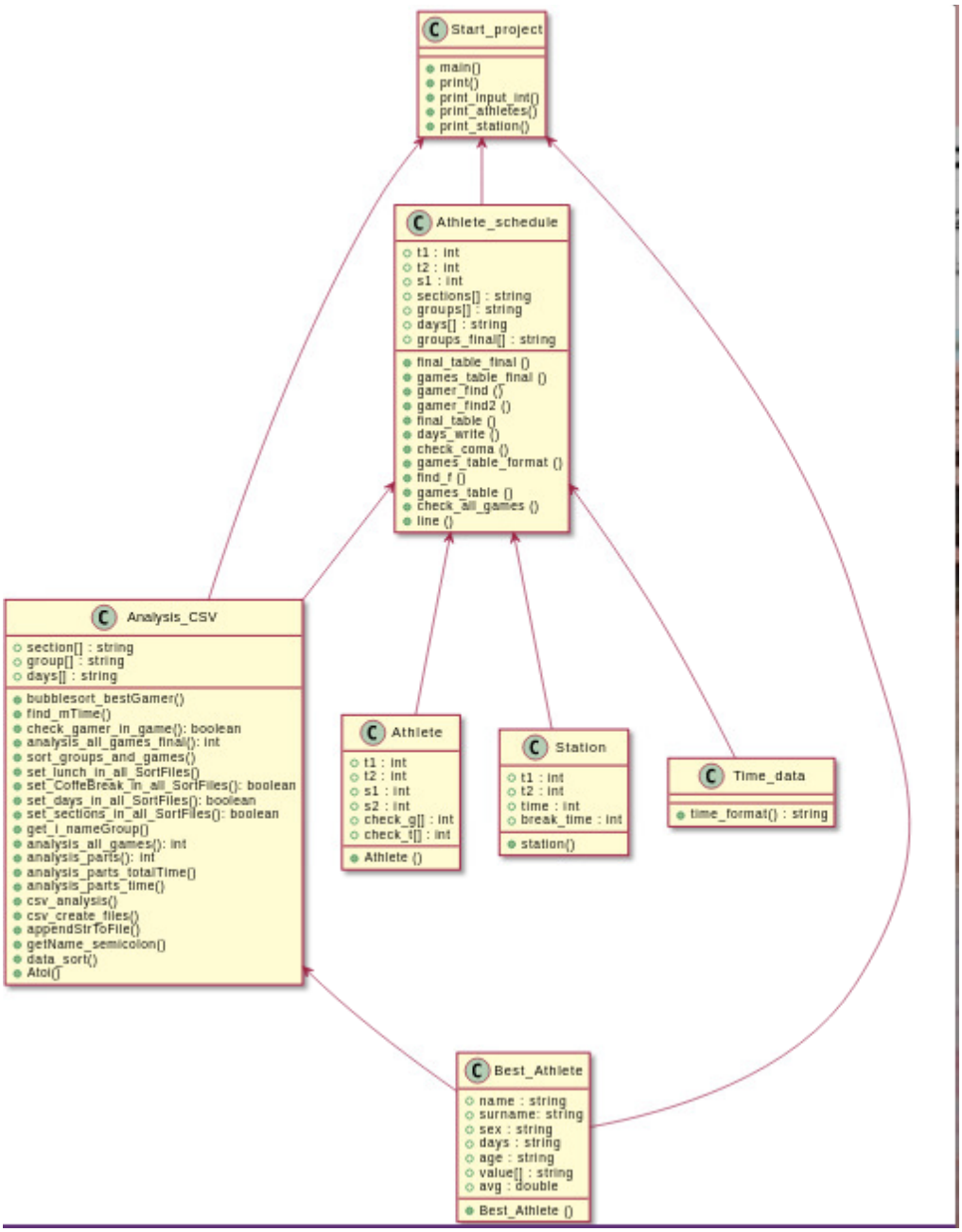
Inom Running, Sprint kolumnen innehåller 60 samt 200 meter, och då har de underkolumner som beskriver tiden i m:s.ms, d.v.s. tiden i minuter, sekunder, millisekunder. Det kan se ut som följande: 1:55.82, detta betyder att den idrottarenn tog sig tiden att löpa i 1 minut och 55 sekunder och 82 millisekunder.
Middle har 800m samt 1500m, Long har kolumnen 3000m och Hurdles har 60m

Inom Jumping samt Throwing så beskriver siffrorna hur många meter kastet eller hoppet är, till exempel, ett hopp är 2,08 meter. När en idrottare inte kommer delta i en specifik tävling av de som nämndes ovan, så kan kolumnen vara tom.

Varje rad kommer att ha ordningen:
Clup, Name, Surname, Sex, Age, Running (sprint 60), Running (sprint 200), Running (middle 800), Running (middle 1500), Running (long 3000), Running (Hurdles 60), Jumping (long), Jumping (triple), Jumping (High), Jumping (pole), Throwing(shot)

Implementation and Testing

Class Diagram



I koden finns sju classes:

1. *Analysis_CSV*:

Den här classen ska sortera spelare som finns i CSV_filen efter kön och ålder. Efter sortering får vi 14 grupper, varje grupp har sin egen CSV_fil.

1. Under_9 F
2. 9 - 10 F
3. 11 - 12 F
4. 13 - 14 F
5. 15 - 16 F
6. 17 - 18 F
7. Adult F
8. Under_9 M
9. 9 - 10 M
10. 11 - 12 M
11. 13 - 14 M
12. 15 - 16 M
13. 17 - 18 M
14. Adult M

För varje grupp ska det bestämmas i vilka tävlingar varje deltagare ska delta. Därefter kommer det att räknas antal deltagare för varje tävling i en grupp. Sedan beräknar tiden för varje station. Upprepar dessa processer för alla grupper för att få tiden för varje grupp i varje station.

När resultaten anges, identifierar den här classen deltagarna som deltar i alla running tävlingar och dessutom lägger dem i CSV_fil. Därefter upprepas den här processen för jumping och throwing. (UTAN HANSYN PÅ ÅLDER OCH KÖN)

Med hjälp av en viss algoritm beräknas resultaten för de sex bästa deltagarna för running, jumping och throwing för att tävla i finalen. I slutet ges de 3 bästa deltagarna en för running, en för jumping och en för throwing.

Om vi har inte resultaten för deltagare, så kommer koden att endast beräkna tiden till sex deltagare för varje tävling i finalen.

Mellan Analysis_CSV class och Best_Athlete class finns association (Relationship).

2. Athlete:

I den har classen finns endast vissa attributes som starttid/sluttid för deltagare, registrera paustiden från början till slut, tävlingsnummer för att verifiera dess inträde, tävlingsnummer för att kontrollera tiden och en constructor.
Athlete class används i Analysis_CSV och i Start_project med association (Relationship).

3. Athletes_schedule:

Den här classen koordinerar schemat i allmänhet och ordnar grupper från starttid 8:00 till sluttid 18:00 , med hänsyn till rast mellan tävlingar, fikatid och lunchtid. Athletes_schedule class använder 4 classes, Analysis_CSV, Station, Athlete och Time_data med association (Relationship).

4. Best_Athlete:

I den har classen finns också bara vissa attributes som name, age och sex för de bästa spelare och en constructor. Best_Athlete class används i Athletes_schedule med association (Relationship).

5. Start_project:

Här finns main funktion och samordnas output och andra alternativ som delay time, print table for one grupp,etc.

6. Station:

Attributes som start/slut tid för varje station, tid för station i minuter, paustid efter varje station och en constructor. Station class används i Athletes_schedule med association (Relationship).

7. Time_data:

I den här classen kommer tiden att koordineras. Starttid i timmar dvs vi börjar klockan 08:00 medan vi räknar tid i minuter. Här ger vi minuter som ska koordineras med starttiden för att ger tiden i timma : minuter (t.ex 9 - 10 F grupp tar 70 minuter i en station, om starttid är 8:00 då sluttid ska vara 9:10).

Uppskattade tiden:

Beräkna tiden för varje tävling/station finns i Analysis_CSV class:
Tid för sprint line (60m, 8 tracks) för n deltagare: (n/8) * 1 (en minut)
Tid för sprint line (200m, 8 tracks) för n deltagare: (n/8) * 1 (en minut)

Total tid för en grupp i sprint line station är: (n/8) * 1 + (n/8) * 1 = (n/8) * (1+1) = (n/8) * 2 (två minuter varje gång)

Tid för Running Circle (400m,6 tracks)för n deltagare (Middle 800m): (n/6) * 3
Tid för Running Circle (400m, 6tracks)för n deltagare (Middle 1500m): (n/6) * 5
Tid för Running Circle (400m, 6tracks)för n deltagare (long 3000m): (n/6) * 10
Tid för Running Circle (400m, 6tracks)för n deltagare (Hurdles, 60 m): (n/6) *10

Total tid för en grupp i Running Circle station är:
(n/6) * 3 + (n/6) * 5 + (n/6) * 10 + (n/6) * 3 =(n/6)*(3+5+10+3)= (n/6) * 21 (21 minuter varje gång)

Tid för Long Jump för n deltagare: n1 * 3 (tre minuter)
Tid för Triple Jump för n deltagare: n2 * 3 (tre minuter)

Total tid för en grupp i två station Long/Triple Jump är: ((n1+n2)/2) * 3 (3 minuter varje gång)

Tid för High Jump – I för n deltagare: n * 3
Tid för High Jump – I I för n deltagare: n * 3

Total tid för en grupp i två station High Jump är: (n/2) * 3 (3 minuter varje gång)

Tid för Pole Vault for för n deltagare: n * 3 (3 minuter varje gång)

Tid för Shot Throwing – I för n deltagare: n * 5
Tid för Shot Throwing – II för n deltagare: n * 5

Total tid för en grupp i två station Shot Throwing är: (n/2) * 5
(5 minuter varje gång)

Koden kommer att välja vilka grupper som kommer att befinna sig i vilka stations med hänseende till att tävlingarna måste ta slut innan fika/lunch rasten eller sluttiden. På detta sätt kommer tiden inte att krockas mellan tävlingarna och rasttiden eller sluttiden.

Testing:

Första testen har skett efter att vi har sorterat csv-filen efter kön och ålder. Testen gick bra, vi fick 14 grupper med dess information utan problem.

Vi gjorde test till de här 2 Arrays den första var för antal deltagare i en tävling för varje grupp, den andra Array var för tiden för grupperna i en station. Den här testen gick bra också.

Vår algoritm går på att om en grupp har en tävling eller deltävling, där tiden kommer krocka med lunchrasten eller fikarasten, så kommer systemet att skjuta fram tävlingen till efter rasten. Detta har vi testat, det fungerade utan problem.

Testerna har visat att det skulle vara bättre att tävlingar som kommer gå över tid som är satt till klockan 18:00, kommer att skjutas fram till nästa dag. Så därför har vi gjort lite ändringar efter testet, så att tävlingstiderna inte får gå över klockan 18:00.

Vi har testat oss fram många algoritmer för att bestämma bästa schemat till tävlingarna och för att inte tiderna ska krockas med varandra samt med rasterna och sluttiden.

Vi har testat delay time algoritmen, testet visade att algoritmen fungerar som den ska, när det blir en fördröjningstid, så uppdaterar schemat sig automatiskt så att tiderna inte krockas med varandra.

Evaluation and Outlook

Vi tycker att vi har fått ett bra resultat och är glada över det. Vårt system har bra kvalitet för att skriva ut data om olika tävlingar. Att systemet visar tiden för tävlingen samt pausen för gruppen räcker för användaren, schemat behöver inte att innehålla mer information än det som redan finns där, detta för att inte förvirra användaren.

Det som kanske kan vara lite “mindre bra” på systemet men ändå kan vara en utvecklingsfråga är att ha ett till alternativ i menyn där man kan söka efter vilka deltagare som finns i varje grupp, eller vilken grupp tillhör en viss deltagare. Dessutom ett alternativ som skriver ut vilka som kom på andra och tredje platsen och inte endast första platsen på tävlingen (om resultaten skulle finnas i csv filen efter tävlingarna). Men då blir det ett allmänt system som är mer utvecklat för hela tävlingen och inte bara ett schema.

En till bra utvecklingsområde är att utveckla systemet på så sätt att det skapar schemat utan många håll, för närvarande så har schemat många hålltider i sig. Det skulle bli bättre om vi har lunch och fikaraster i olika tider för olika grupper. Detta gör schemat mer anpassade till varje grupp i sig, det minskar hålltider där stationer tvingas bli lediga bara för att tiderna inte ska krockas.

Om vi har haft 3 månader att utveckla vårt system så skulle vi utveckla det på det sättet som vi nämnt tidigare. Att söka efter en specifik deltagare, vilken grupp den kommer att tävla i, samt att få upp dens schema. Vi kanske skulle ha lite mer detaljerat information med en bild på varje deltagare, samt en flik för historiken för varje deltagare, till exempel, vilka tävlingar den har vunnit tidigare osv.
information med en bild på varje deltagare, samt en flik för historiken för varje deltagare, till exempel, vilka tävlingar den har vunnit tidigare osv.