Exercise

Role-based Access Control
Goals

• Learn and apply languages for role-based access control modelling
• Understand the limitations of these languages
Consider the Following Scenario
(slide 1)

The football game report is started before the match, continued during the match, and finalised once the match is over. The new report is created by the league secretary, who is employed by the Football Federation. The league secretary fills in initial information such as league name, his own contact data, region, division, and game numbers as well as the names of the competing teams. Once done, the league secretary informs the team representatives, so that they could provide the team composition for the game, including player names and their registration numbers. The team composition should be provided no later than 2 h before the game.
Consider the Following Scenario

(slide 2)

During the match the game report needs to be maintained by the umpire, who needs to register the scored goals and “given” cards. The umpire is also responsible for filling in the match results, including the scores (including final, full time, extra time and penalties). After the match, the umpire needs to invite the team representatives to sign the report and (optionally) provide comments. Once signatures from the team representatives are received, the umpire himself comments and signs the report. After the game the report is sent to the league secretary for confirmation.
## WOLFS FOOTBALL LEAGUE MATCH REPORT FORM

League Secretary  
**CALEX KARUUNA**  
20 STREET ALLEA  
CITY

Each team representative is to sign this form and ensure that all details are correct and that all players are registered with the ERIS system.

<table>
<thead>
<tr>
<th>LEAGUE NAME</th>
<th>WOLFS</th>
<th>DIVISION</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>REGION</td>
<td>AA</td>
<td>GAME</td>
<td>45</td>
</tr>
</tbody>
</table>

**HOME TEAM NAME**: Green Rabbits  
**AWAY TEAM NAME**: Orange Carrots

### Final Scores

<table>
<thead>
<tr>
<th>Full Name</th>
<th>Reg, No</th>
<th>Cards</th>
<th>Goals</th>
<th>Full Name</th>
<th>Reg, No</th>
<th>Cards</th>
<th>Goals</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1] John</td>
<td>1</td>
<td></td>
<td></td>
<td>Joseph</td>
<td>11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[2] Peter</td>
<td>2</td>
<td>YY-&gt;R</td>
<td>1</td>
<td>Boris</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[4] Davy</td>
<td>5</td>
<td></td>
<td></td>
<td>Ferdinand</td>
<td>10</td>
<td>Y</td>
<td></td>
</tr>
<tr>
<td>[6] King</td>
<td>7</td>
<td></td>
<td></td>
<td>Svetoslay</td>
<td>4</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>[7] Shark</td>
<td>9</td>
<td>Y</td>
<td></td>
<td>Jevgenyi</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>[8] Anupras</td>
<td>10</td>
<td></td>
<td></td>
<td>Matheus</td>
<td>6</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>[9] George</td>
<td>11</td>
<td>1</td>
<td></td>
<td>Anis</td>
<td>55</td>
<td>YY-&gt;R</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Full Time Score</th>
<th>Extra Time Score</th>
<th>Penalties</th>
<th>Full Time Score</th>
<th>Extra Time Score</th>
<th>Penalties</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>-</td>
</tr>
</tbody>
</table>

Signature of Home Team Representative: **HomeSignedIt**  
Signature of Away Team Representative: **AwaySignedIt**

I am satisfied with my opponents registration/performance (Please indicate YES or NO above)

Home Team Caution: **No problem, everything is OK**  
Away Team Caution: **No problem, everything is OK**

**Umpire Name**: Billy-Goat the Beard  
**Referee signature**: BilGoatB

**Confirmation by league secretary**: Confirmed  
**Confirmation date**: 2020.02.20
Task 1

• Define the SecureUML model representing role-based access control policy regarding the gathered and registered data
To guide your solution, consider

• What are objects and their concerned attributes?
• What operations do change the values of the attributes?
• What are roles?
• What are the security actions?
• What are the permissions of roles towards the object?
• Who are the users?
To guide your solution, consider:

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
To guide your solution, consider:

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
To guide your solution, consider

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- **What are the security actions?**
- What are the permissions of roles towards the object?
- Who are the users?
To guide your solution, consider

- What are objects and their concerned attributes?
- What operations do change the values of the attributes?
- What are roles?
- What are the security actions?
- What are the permissions of roles towards the object?
- Who are the users?
AC#1: Attribute `makeReport` is associated to the GameReport operation `createWithInitialInformation()`, which **Insert** new GameReport (new attributes `leagueName`, `secretaryContactData`, `region`, `division`, `gameNumber1`, `gameNumber2`, `teamHomeName1`, `teamHomeName2`).

AC#2: Attribute `confirmReport` is associated to the GameReport operation `confirm()`, which **Update** GameReport attribute `confirmation`. 

AC#3: Attribute `signAndComment` is associated to the GameReport operation `signByTeamRep()`, which **Update** GameReport attributes `signature1` and `comments1` if TeamRepresentative.home is true. If TeamRepresentative.away is true then, GameReport attributes `signature1` and `comments2` are **Update**.

AC#4: Attribute `makeTeam` is associated to the Team operation `createTeam()`, which **Insert** new Team (new attributes `name`).

AC#5: Attribute `fillScores` is associated to the Team operation `fillResults()`, which **Update** Team attributes `finalScore`, `fullScore`, `extraTimeScore`, `penalties`.

AC#6: Attribute `makePlayer` is associated to the Player operation `createPlayer()`, which **Insert** new Player (new attributes `name`, `registrationNumber`).

AC#7: Attribute `signAndComment` is associated to the GameReport operation `signByUmpire()`, which **Update** GameReport attributes `umpireSignature` and `umpireComments`.

AC#8: Attribute `giveYellowCard` is associated to the Player operation `giveYellowCards()`, which **Update** Player attributes `yellowCards`.

AC#9: Attribute `giveRedCard` is associated to the Player operation `giveRedCards()`, which **Update** Player attributes `redCard`.

AC#10: Attribute `giveGoal` is associated to the Player operation `giveGoal()`, which **Update** Player attributes `goals`.

AC#11: Execute `createTeam()`, if t<2h (where t is game starting time).

---

**What authorisation constraints should be defined in your model?**

- They can be defined in any **formal**, **semiformal** or **natural** language (in case of natural language, use **English 😊**).
Task 2

• Define the **UMLsec** model representing the role-based access control policy regarding the gathered and registered data
To guide your solution, consider

- What are the objects?
- What are their operations?
- What are the roles?
- What are the role’s rights?
- What are the associated tags?
- Who are the users?
To guide your solution, consider:
- What are the objects?
- What are their operations?
- What are the roles?
- What are the role’s rights?
- What are the associated tags?
- Who are the users?
To guide your solution, consider:
- What are the objects?
- What are their operations?
- What are the roles?
- What are the role’s rights?
- What are the associated tags?
- Who are the users?
To guide your solution, consider
- What are the objects
- What are their operations?
- What are the roles?
- What are the role’s rights?
- What are the associated tags?
- Who are the users?

\{
\text{protected} = \text{protected\_action}
\}

\{
\text{role} = (\text{actor, role})
\}

\{
\text{right} = (\text{role, protected\_action})
\}
{protected = createWithInitialInformation}
{role = (Calex Karuuna, LeagueSecretary)}
{right = (LeagueSecretary, createWithInitialInformation)}

{protected = CreateTeam}
{role = ((Home, Away), TeamRepresentative)}
{right = (TeamRepresentative, CreateTeam)}

{protected = CreatePlayer}
{role = ((Home, Away), TeamRepresentative)}
{right = (TeamRepresentative, CreatePlayer)}

{protected = RegisterPlayerPerformance}
{role = (Billy-Goat the Beard, Umpire)}
{right = (Umpire, RegisterPlayerPerformance)}

{protected = FillResults}
{role = (Billy-Goat the Beard, Umpire)}
{right = (Umpire, FillResults)}

{protected = SignByTeamRep}
{role = ((Home, Away), TeamRepresentative)}
{right = (TeamRepresentative, SignByTeamRep)}

{protected = SignByUmpire}
{role = (Billy-Goat the Beard, Umpire)}
{right = (Umpire, SignByUmpire)}

{protected = Confirm}
{role = (Calex Karuuna, LeagueSecretary)}
{right = (LeagueSecretary, Confirm)}