Task 3 - feedback

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).

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?

**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 🇬🇧).
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?

\[
\begin{align*}
\text{protected} & = \text{protected}_\text{action} \\
\text{role} & = \{\text{actor}, \text{role}\} \\
\text{right} & = \{\text{role}, \text{protected}_\text{action}\}
\end{align*}
\]
Task 3

Explain what RBAC policy concerns are captured in your...

... **SecureUML** model and *not* in the **UMLsec** model

... **UMLsec** model and *not* in the **SecureUML** model
## Construct Semantics

<table>
<thead>
<tr>
<th>RBAC concepts</th>
<th>SecureUML</th>
<th>UMLsec</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>Class stereotype</td>
<td>Actor value of association tag <em>(role)</em></td>
</tr>
<tr>
<td></td>
<td>«secuml.user»</td>
<td></td>
</tr>
<tr>
<td>User assignment</td>
<td>Dependency stereotype</td>
<td>Associated tag <em>(role)</em></td>
</tr>
<tr>
<td></td>
<td>«assignment»</td>
<td></td>
</tr>
<tr>
<td>Roles</td>
<td>Class stereotype</td>
<td>Activity partition</td>
</tr>
<tr>
<td></td>
<td>«secuml.role»</td>
<td>Role value of association tag <em>(role)</em></td>
</tr>
<tr>
<td>Permission assignment</td>
<td>Association class stereotype</td>
<td>Action</td>
</tr>
<tr>
<td></td>
<td>«secuml.permission»</td>
<td>Associated tag <em>(right)</em></td>
</tr>
<tr>
<td>Object</td>
<td>Class stereotype</td>
<td>Activity partition</td>
</tr>
<tr>
<td></td>
<td>«secuml.resource»</td>
<td></td>
</tr>
<tr>
<td>Operation</td>
<td>Operation of</td>
<td>Action</td>
</tr>
<tr>
<td></td>
<td>«secuml.resource»</td>
<td>Associated tag <em>(protected)</em></td>
</tr>
<tr>
<td></td>
<td>class</td>
<td></td>
</tr>
<tr>
<td>Permission</td>
<td>Authorisation</td>
<td>Not defined</td>
</tr>
<tr>
<td></td>
<td>constraints</td>
<td></td>
</tr>
</tbody>
</table>