USE CASE DESCRIPTIONS

<table>
<tr><td>USE CASE</td><td>Manage Board Games</td></tr>
<tr><td>SUMMARY</td><td>Add, remove and edit board games; display board games list</td></tr>
<tr><td>ACTOR</td><td>The founder</td></tr>
<tr><td>PRECONDITION</td><td></td></tr>
<tr><td>POSTCONDITION</td><td>Board game has been added, removed or edited; Board games has been displayed on the webpage</td></tr>
<tr><td>BASE SEQUENCE</td><td>
<table>
<tr><td>Scenario A.
ADD</td><td>A.1. System displays the students’ list. [ALT1]
A.1. Enter values (board game’s name, description, number of players).
A.2. Choose an owner from the students’ list or input new guest’s data.
A.3. Approve the data entered.
A.4. System validates data. [ALT2]
A.5. System adds a new board game with the given data to the list of board games and a file.
A.6. The system displays the approval message.</td></tr>
</table>
<table>
<tr><td>Scenario B.
REMOVE</td><td>B.1. System displays the board games’ list. [ALT3]
B.2. Select a board game from the list using the board game name.
B.3. System shows board game details (board game’s title, description, number of players, owner).
B.3. The system removes the board game from the list and updates it in a file.
B.4. The system displays the approval message.</td></tr>
</table>
<table>
<tr><td>Scenario C.
EDIT</td><td>C.1. System displays the board games’ list. [ALT3]
C.2. Select a board game from the list using the board game’s name.
C.3. System shows board game details (board game’s title, description, number of players, owner).
C.4. Enter new data.
C.5. Approve the data entered.
C.6. System validates data. [ALT2]
C.7. The system changes the board game’s data.
C.8. The system displays the approval message.</td></tr>
</table>
1. System updates all files and lists that were used during the process.

Use case ends.
</td></tr>
</table>

ALTERNATE SEQUENCE	[*ALT0] The process can be canceled in all steps with user interaction. Use case ends. [ALT1] If the list does not contain any student, an empty list will be displayed. [ALT2] If the user did not enter any data or did not choose an owner or entered VIA ID did not contain only 6 digits or if it is already used, the process cannot be performed, display a message. [ALT3] If the list does not contain any board game, an empty list will be displayed.
NOTE	This use case covers requirements: 2, 3, 4, 5, 6, 23, 24, 25

USE CASE	Manage Borrowing Board Games
SUMMARY	Borrow, edit borrowing, return, reserve, cancel reservation of a Board Game
ACTOR	The founder
PRECONDITION	The board game to lend is already added to the system
POSTCONDITION	Lend status has been changed
BASE SEQUENCE	<table><tr><td>Scenario A. BORROW</td><td>A.1.System shows a list of all not lended board games, each element with board game name. [ALT1] A.2. Select a board game from the list. A.3. Choose between two user types. <table><tr><td>Scenario A.1. STUDENT</td><td>A.1.1. System lists all members and guests who has not lent a board game [ALT1] A.1.2. Select a student from the list A.1.3. Approve the data entered</td></tr></table> <table><tr><td>Scenario A.2. GUEST</td><td>A.2.1. Enter new guest data A.2.2. Approve the data entered A.2.3. System validates data [ALT2]</td></tr></table> A.4. The system changes lend details.</td></tr></table>

	Scenario B. RETURN	B.1.System shows a list of all lended board games, each element with board game name. [ALT1] B.2. Select a board game from the list. B.3 System shows board game details (board game's title, borrower). B.4.System changes the lend status to not lend. B.5 System checks if a person is a member or a guest; if the game was borrowed by a guest and this person is not an event's participant, an owner, neither a borrower of any game after returning the game, will be removed from the guests list. [ALT5]
	Scenario C. RESERVE	C.1. System lists all members with full name and VIA ID. [ALT1] C.2. Select a member from the list. C.3.System shows a list of all lended board games, each element with board game name. [ALT1] C.4. Select a board game from the list. C.5. Approve the reservation. C.6. System adds the selected member to the list of reservants.
	Scenario D. CANCEL RESERVATION	D.1. System lists all members with full name and VIA ID. [ALT1] D.2. Select a member from the list. D.3.System shows a list of all board games reserved by the selected member, each element with board game name. [ALT1] D.4. Select a board game from the list. D.5.System removes the member from reservants' list.
	Scenario E. EDIT BORROWER	E.1.System shows a list of all lended board games, each element with board game name. [ALT1] E.2. Select a board game from the list. E.3 System shows board game details (board game's title, borrower). E.4. Choose between two user types.

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>
<table>
<tr><td>Scenario E.5. STUDENT</td><td>A.5.1. System lists all members and guests who has not lent a board game [ALT1]
A.5.2. Select a student from the list
A.5.3. Approve the data entered</td></tr>
</table>
<table>
<tr><td>Scenario E.6. GUEST</td><td>A.6.1. Edit guest data.
A.6.2. Approve the data entered.
A.6.3. System validates data. [ALT2]</td></tr>
</table>
E.5. The system changes lend details.
</td></tr>
</table>
5. Lend details shown along with an approved message.

Use case ends.
</td></tr>
<tr><td>ALTERNATE SEQUENCE</td><td>[*ALT0] The process can be canceled in all steps with user interaction. Use case ends.
[ALT1]If the list does not contain any elements, an empty list will be displayed.
[ALT2] If the data is empty or not valid, the process cannot be performed, display message.
[ALT4] If a person isn’t a member, then the system checks if the person has already borrowed a board game (checks if the person is a guest) and if not - allows them to borrow a game, if yes - process cannot be performed, display message.
[ALT5] If a person wants to rate a board game, they can rate it in a scale from 1 to 5 (inserted by the founder) and the average of the rates is recalculated and updated.</td></tr>
<tr><td>NOTE</td><td>This use case covers requirements: 1, 4, 5, 6, 7, 10, 11, 14, 15, 21, 22, 23, 24, 25</td></tr>
</table>

<table>
<tr><td>USE CASE</td><td>Manage Events</td></tr>
<tr><td>SUMMARY</td><td>Create an event and display it on the website</td></tr>
<tr><td>ACTOR</td><td>The founder</td></tr>
<tr><td>PRECONDITION</td><td></td></tr>
<tr><td>POSTCONDITION</td><td>New event has been added and displayed on the website</td></tr>
<tr><td>BASE SEQUENCE</td><td>
<table>
<tr><td>Scenario A. ADD</td><td>A.1. Enter values (event’s name, date and description)
A.2. Approve the data entered.
A.3. System validates data. [ALT1]
A.4. System adds a new event with the given data to the list of events and a file.
A.5. System shows a proper message.</td></tr>
</table>
<table>
<tr><td>Scenario B. REMOVE</td><td>B.1. System displays the events’ list. [ALT2]
B.2. Select an event from the list or search for it using the event’s information.
B.3. System shows the event's details (event’s name, date and description).
B.4. System removes the event from the list and updates it in a file.
B.5. System shows a proper message.</td></tr>
</table>
<table>
<tr><td>Scenario C. EDIT</td><td>C.1. System displays the events’ list. [ALT2]
C.2. Select an event from the list or search for it using the event’s information.
C.3. System shows the event's details (event’s name, date and description).
C.4. User inputs new data (event’s name, date and description).
C.5. Approve the data entered.
C.6. System validates data. [ALT1]
C.7. The system changes the data of the event.
C.8. System shows a proper message.</td></tr>
</table>
<table>
<tr><td>Scenario D. REGISTER</td><td>D.1. System displays the events’ list.
D.2. System displays the students’ list.
D.3. Select an event from the list or search for it using the event’s information.
D.4. Select a student from the students’ list or search for him/her using their information. [ALT3]
D.5. Approve the data entered.
D.6. System validates data. [ALT4]</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>D.7. System adds the person to the list of participants registered for the chosen event and adds it to the students' list.
D.8. System shows a proper message.</td></tr>
</table>

2. System updates all files and lists that were used during the process.

Use case ends.</td></tr>
<tr><td>ALTERNATE SEQUENCE</td><td>[*ALT0] The process can be canceled in all steps with user interaction.
Use case ends.
[ALT1] System checks if the user entered the event's name, date and description.
[ALT2] If the list does not contain any event, an empty list will be displayed.
[ALT3] If the student is not in the system, the user can input data (first name, last name and VIA ID).
[ALT4] System checks if the user entered the event's name, date and description and if any student is selected from the students' list or if the new student's data is entered.</td></tr>
<tr><td>NOTE</td><td>This use case covers requirements: 4, 16, 17, 18, 23, 24, 25</td></tr>
</table>

<table>
<tr><td>USE CASE</td><td>Manage Guests’ and Members’ Data</td></tr>
<tr><td>SUMMARY</td><td>Adding a guest on the members list or removing a member from the list</td></tr>
<tr><td>ACTOR</td><td>The founder</td></tr>
<tr><td>PRECONDITION</td><td></td></tr>
<tr><td>POSTCONDITION</td><td>The information has been added, modified or removed.</td></tr>
<tr><td>BASE SEQUENCE</td><td>
<table>
<tr><td>Scenario A. ADD</td><td>A.1. System displays the lists with guests. [ALT1]
A.2. Enter new member’s data. [ALT2]
A.3. Approve the data entered.
A.3. System validates data. [ALT3]
A.4. System adds a new member with the given data to the list of members and a file.
A.5. The system displays a proper message.</td></tr>
</table>
<table>
<tr><td>Scenario B. REMOVE</td><td>B.1. System displays the list with members. [ALT4]
B.1. Select a member from the list using the person’s name or VIA ID.
B.2. System shows the member's details (name and VIA ID).
B.3. System removes the member from the members list.
B.4. The system displays a proper message.</td></tr>
</table>
<table>
<tr><td>Scenario C. EDIT</td><td>C.1. Select a student from the list using the person’s name or VIA ID.
C.2. System shows the student's details (name and VIA ID).
C.3. Enter new data (name, VIA ID)
C.4. Approve the data entered.
C.5. System validates data. [ALT3]
C.6. Display changed data.
C.7. The system displays a proper message.</td></tr>
</table>
Use case ends.
</td></tr>
<tr><td>ALTERNATE SEQUENCE</td><td>[*ALT0] The process can be canceled in all steps with user interaction. Use case ends
[ALT1] If the list doesn’t contain any member, then the system shows a message that the list of members is empty..</td></tr>
</table>

	[ALT2] If the person is already a guest, the person's status can be changed from guest to member, so the guest’s data will be added to the members’ list and deleted from the guests’ list. [ALT3] If the data (VIA ID) is empty or not valid (has more than 6 digits, less than 6 digits or it is a String), process cannot be performed, display message [ALT4] If there are no members on the members’ list, process cannot be performed , display message.
NOTE	This use case covers requirements: 7, 8, 9, 12, 13, 23, 24, 25

<table>
<tr><td>USE CASE</td><td>Manage Upcoming Board Games</td></tr>
<tr><td>SUMMARY</td><td>Creating and displaying a list of board games that can be bought next; adding, removing board games from the list; voting for one of the games</td></tr>
<tr><td>ACTOR</td><td>The founder</td></tr>
<tr><td>PRECONDITION</td><td></td></tr>
<tr><td>POSTCONDITION</td><td>Data for a board game has been shown or the list of board games has either been edited or a board game has been added/removed. A vote for one of the games has been added.</td></tr>
<tr><td>BASE SEQUENCE</td><td>
<table>
<tr><td>Scenario A.
ADD</td><td>A.1. Enter values (board game’s title, number of players and description).
A.2. Approve the data entered.
A.3. System validates data. [ALT1]
A.4. System adds a new board game with the given data to the list of board games and a file.
A.5. The system displays the approval message.</td></tr>
</table>
<table>
<tr><td>Scenario B.
REMOVE</td><td>B.1. System displays the list of upcoming board games. [ALT2]
B.2. Select a board game from the list by typing board game’s data. [ALT3]
B.3. System shows board game details (board game’s name, number of players, description).
B.3. The system removes the board game from the list and updates it in a file.
B.4. The system displays the approval message.</td></tr>
</table>
<table>
<tr><td>Scenario C.
EDIT</td><td>C.1. System displays the list of upcoming board games. [ALT2]
C.2. Select a board game from the list by typing board game’s data. [ALT3]</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><td></td><td>
<table>
<tr><td></td><td>C.3. System shows board game details (board game’s name, number of players, description).
C.4. Enter new data.
C.5. Approve the data entered.
C.6. System validates data. [ALT1]
C.7. The system changes the board game’s data.
C.8. The system displays the approval message.</td></tr>
</table>
<table>
<tr><td>Scenario D. ADD VOTE</td><td>D.1. System displays the list of upcoming board games. [ALT2]
D.2. Select a board game from the list by typing board game’s data. [ALT3]
D.3. Add a vote to the selected game.
D.4. System updates the number of votes a game has received.
D.5. System shows a message.</td></tr>
</table>
2.System displays an updated list of board games.
Use case ends.</td></tr>
<tr><td>ALTERNATE SEQUENCE</td><td>[*ALT0] The process can be canceled in all steps with user interaction. Use case ends.
[ALT1] If the entered data is empty, the process cannot be performed, and an alert will be displayed.
[ALT2] If the list does not contain any board game, an empty list will be displayed.
[ALT3] If no game has been selected the process cannot be performed, and an alert will be displayed.</td></tr>
<tr><td>NOTE</td><td>This use case covers requirements: 4, 19, 20, 23, 24, 25</td></tr>
</table>