

# USport

Three risks and mitigations:

Risks:

- The server serving the requests for the front end goes down.
- API requests not achieving the agreed SLO for the product.
- SQL injection attacks

Mitigations:

- Another server could be spun to serve the requests to frontend
- Using alert management systems to ping endpoints in a fixed interval of time to check if the SLO is satisfied. If not a messaging system could be integrated with the alert management system to receive instant messages for action.
- Improve database integration and make it more secure against cyberattacks.

Three test cases that reflect's system's features/functionality:

Test Cases:

Test 1: Create Game

Input:

- Field location
- Sport type
- Add players
- Date and time

Tests:

- User searches for fields.
- User enters sports
- Require minimum players during creating a game
- Scheduling conflict is checked before creating a game

Test 2: Communication between two users

Input:

- View previous chats
- Message

Tests:

- Test that the message is sent successfully from one user to another.
- Test that a user receives the message from the sender.
- Test that the message appears in correct order in the chat history for both users
- Test that the message is delivered in a timely manner and not delayed.

### Test 3: Notifications

Input:

- Message accept notification
- Message reject notifications
- Game accept notification
- Game reject notification

Tests:

- Test the notification state is updated successfully based on different actions written in the input.
- Test that the accepted game is in the upcoming game list.
- Test that the notification is displayed correctly