Security_analysis describes how you addressed the relevant security concerns (or why you did not address them) (200-400 words)

Although we were not afraid of security vulnerabilities, certain measures were taken. Due to mostly internal usage, honor code, and limited functionality the threat of a security breach is not as big. Nonetheless, we still know about, and have prevented some, vulnerabilities in our system. Firstly, there is no HTTPS, because we do not have SSL. This means there is a possibility for easy interception, a man in the middle attack. With an attack like this in a POST request, the user can tamper with the request and server does not have any way to ensure the data is not tampered. Nonce? We stop the Cross-Site Scripting by using input sanitation on fields that allow the user to use strings (email and booking title) SQL injection is not possible due to the created prepared statements. We are not really vulnerable to CSRF as all requests that actually change something on the back-end are POST requests that require JSON as body. If this would require a cookie or token, the other tab/webpage will not have access to it, thus there are almost no vulnerabilities to worry about here. The threat this would cause is not the greatest either, the worst that could happen is that a meeting gets edited or deleted. Because we use session so it is very unlikely. Verified via api call, so you cannot hijack, due to the cookies being handled by json. In case of an internal server error, the attacker does not get any extra information of the server as we disabled printing the stack trace to the user. Authentication is handled by Google, which makes the probability of security failures on our end much smaller. The info of the user that is currently logged in is stored in a session, using the standard cookie system employed by Jersey, which contains long random strings as identifier.