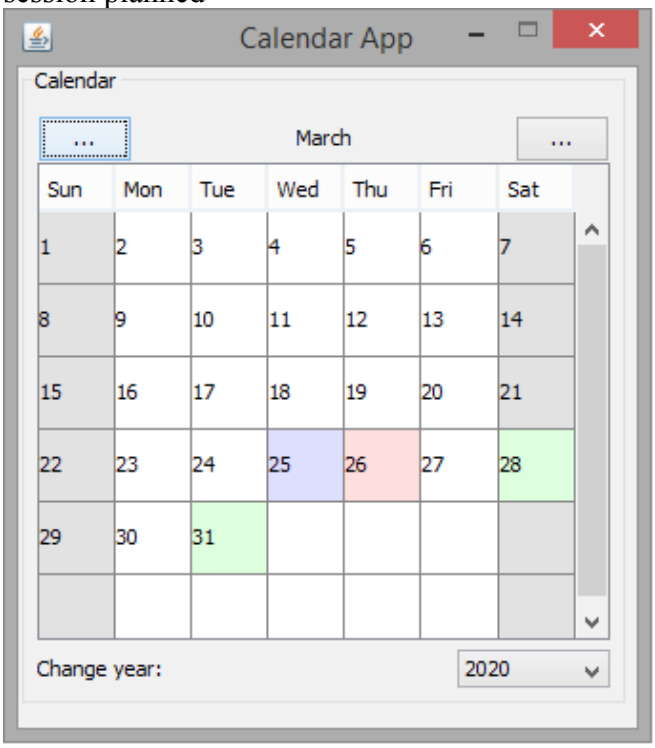
Criterion E: Evaluation

Success Criteria

The program notifies the correct players of when there's a session	Met. The program can notify specific members on the correct date through the Discord API.
The program notifies the correct players of when a session has been canceled	Met. The program differentiates between planned and cancelled sessions and notify accordingly.
DM's are able to add/remove session dates	Met. Integration with Google Sheets allows users to easily access and modify the date database.
DM's are able to add/edit session details such as group	Met. The program is able to read group information and other details in display them.
Application has a user-friendly interface	Met. Interface was found by client to be quite intuitive and easy to use
Interface shows which dates have sessions planned	Met. The calendar UI highlights dates that have a session planned 
Interface shows which dates have canceled sessions	Met. The calendar UI highlights planned sessions green and canceled sessions red.

Client Feedback
The client was quite pleased with the product and asked if they could implement it immediately. They disliked the aesthetic of the application but that it wasn't an issue at all. Their biggest concern was that you couldn't modify session dates from directly within the application, and that you were forced to go through the spreadsheet instead. Overall, the client was quite pleased with the end result, especially with the automatic notifications that will ensure that no miscommunications occur (see appendices for transcript of meeting).

Recommendations for Further Development
The biggest criticism the client had was the fact you couldn't add session dates from within the application itself. This could be done by using Java Swing to create text entry fields where the user could fill in the needed information, and then update the database using the Sheets API

Another improvement would be to allow users to create new groups from within the application without having to code it in. This could be done by getting user-entered group name and group ID from a Java Swing interface, and using it to notify members of this new group.

Lastly, it would make sense to have multiple types of events, such as recurring events. Users would specify if their event is a recurring event and how often it occurs (weekly, monthly, etc.) This could be accomplished by having the application create new events in a loop, each time making the new event occur one interval after the previous event.

Word Count: 397