REPORT

PROJECT DEVELOPMENT

CONNECT

Jade Harris | **12SDD**

LAST UPDATED: 2021

CONTENTS

ALGORITHMS

PSEUDOCODE

CONNECT

Jade Harris | **12SDD**

CONTENTS

FRM_LOADINGSCREEN

```
BEGIN timer_Tick
    COUNTDOWN a timer THEN
    DISPLAY Frm_font
    CLOSE this form
END timer_Tick
```

FRM_FONT

```
BEGIN btn_closeWelcome_Click
    DISPLAY loginForm
    CLOSE this form
END btn_closeWelcome_Click
BEGIN btn_send_Click
    DISPLAY userManual.pdf
END btn_send_Click
```

FRM_WELCOMENEWUSER

```
SET cs TO connection string to cloud database
SET user TO username for frm_nav
BEGIN btn_closeWelcome_Click
    DISPLAY hub form
    SetUserNotNewAnymore
    CLOSE this form
END btn_closeWelcome_Click


BEGIN SetUserNotNewAnymore
    OPEN connection to cs
    SET new TO false for user
    CLOSE connection to cs
```

```
END SetUserNotNewAnymore


BEGIN btn_send_Click

        DISPLAY userManual.pdf

END btn_send_Click
```

FRM_LOGIN

```
SET cs to connection string TO cloud database

BEGIN frm_login

        For each textbox, once the user clicks into the textbox SET TO “” and when they leave, if the text is
        still empty return to the original text

        When the user hovers over SHOW button, unconceal password and change the button text to HIDE.
        When their cursor leaves, hide password and change button text to SHOW.

        “ ‘ “ characters or spaces cannot be entered into any textbox

END frm_login


BEGIN cb_EULA_Click

        IF cb_EULA is checked THEN

                DISPLAY the EULA pdf

                DISPLAY “Are you sure you have completely read and agree to EULA?”

                GET DialogResult

                IF DialogResult == OK THEN

                        Leave checkbox checked

                ELSE

                        Uncheck checkbox

                ENDIF

        ENDIF

END cb_EULA_Click
```

```
BEGIN GeneratePassword
    SET passwordGenerated;
    SET allowedCharacters TO
    “abcdefghijkmnopqrstuvwxyzABCDEFGHJKLMNOPQRSTUVWXYZ0123456789!@$?_-“
    SET passwordCharacters TO array of 12 characters
    FOR i = 0 TO 11 STEP 1
        SET passwordCharacters[i] TO random character in allowedCharacters
    NEXT i
    FOREACH character IN passwordCharacters
        passwordGenerated += character
    NEXT character
    SET signup_password TO passwordGenerated
END GeneratePassword


START LogIn
    IF any field is empty THEN
        DISPLAY “Please complete all fields”
    ELSE
        LogInProcess
    ENDIF
END Login


BEGIN LogInProcess
    GetLoginDetails
    CheckLoginDetails
    SetLoggedIn
END LogInProcess
```

```
BEGIN GetLoginDetails

    OPEN connection to cs

    POPULATE dgv_user WITH user record

    CLOSE connection to cs

END GetLoginDetails


BEGIN CheckLoginDetails

    IF a record was found for the user THEN

        IF the entered password matches the stored password THEN

            IF it is stored that the user is not currently logged in  THEN

                RedirectUser

            ELSE

                DISPLAY “Account already logged in on another machine”

            ENDIF

        ELSE
            DISPLAY “Incorrect password”

        ENDIF
    ELSE

        DISPLAY “No user exists”

    ENDIF


END CheckLoginDetails


BEGIN RedirectUser

    SetHubFormsUserVariables

    CheckIfNew

END RedirectUser


BEGIN SetHubFormsUserVariables

    SET frm_nav.username TO username entered
```

```
END SetHubFormsUserVariables


BEGIN CheckIfNew
    SET newUser TO the value of the 'new' column
    IF newUser == true THEN
        SET frm_welcomeNewUser.user TO username entered
        DISPLAY frm_welcomeNew
        HIDE this form
    ELSE
        REM "user must not be new"
        DISPLAY frm_hub
        HIDE this form
    ENDIF
END CheckIfNew


BEGIN Register
    IF any register field is empty THEN
        DISPLAY "Please complete all fields"
    ELSE
        IF cb_EULA is not checked THEN
            DISPLAY "Please agree to EULA to create an account"
        ELSE
            userExists = CheckIfUserExists
            emailExists = CheckIfEmailExists
            IF email is < 7 characters OR username is < 4 characters OR password is < 9
            characters
                DISPLAY "Error, email is < 7 characters OR username is < 4 characters OR
                password is < 9 characters"
            ELSE
                CreateNewUser
            ENDIF

        ENDIF

    ENDIF

END Register


BEGIN CheckIfEmailExists

    OPEN connection to cs

    POPULATE table with records where email matches entered email

    CLOSE connection to cs

    IF a record exists THEN

        RETURN true

    ELSE

        RETURN false

    ENDIF

END CheckIfEmailExists


BEGIN CheckIfUserlExists

    OPEN connection to cs

    POPULATE table with records where username matches entered username

    CLOSE connection to cs

    IF a record exists THEN

        RETURN true

    ELSE

        RETURN false

    ENDIF

END CheckIfUserlExists
```

```
BEGIN CreateNewUser

        OPEN connection to cs

        INSERT entered username, password and email as a new record

        CLOSE connection to cs

        ResetFieldsNowNewUser

END CreateNewUser


BEGIN ResetFieldsNowNewUser

        DISPLAY “User successfully created”

        Reset username, password and email to default

END ResetFieldsNowNewUser
```

FRM_EVENTS

```
SET cs to connection string TO cloud database

SET username to user (public variable set from frm_nav)


BEGIN frm_events

        SET row height of datagridview to 40px

        Refresh

        FOR each row in dgv_events STEP 1

                IF the status column == “Pending” for that row THEN

                        SET second button TO “Wait until hosted”

                ELSE

                        SET second button TO “View Details”

                ENDIF

        NEXT row

END frm_events
```

```
BEGIN Refresh
        PopulateUser
        PopulateWithMyEvents
END Refresh


BEGIN PopulateUser
        OPEN connection to cs
        POPULATE dgv_currentUserInfo with user information
        CLOSE connection to cs
END PopulateUser


BEGIN PopulateWithMyEvents
        REM "to track last time updated"
        UPDATE lbl_lastUpdate TO current time
        GET all of user's events
        OPEN connection to cs
        POPULATE dgv_events with the information for these events from the Event table
        CLOSE connection to cs
END PopulateUser


BEGIN PopulateWithMyEvents
        REM "to track last time updated"
        UPDATE lbl_lastUpdate TO current time
        GET all of user's events
        OPEN connection to cs
        POPULATE dgv_events with the information for these events from the Event table
        CLOSE connection to cs
END PopulateUser
```

```
BEGIN btn_newEvent_click
    DISPLAY frm_myEvents
END btn_newEvent_click


SET selectedRowIndex TO 0


BEGIN dgv_events_Click
    IF user clicks on an event row THEN
        SET selectedRowIndex TO index of event
        SET selectedEvent TO name of the selected event
        SET mostRecent = CheckLastUpdated_Event(selectedEvent)
        IF mostRecent = false THEN
            DISPLAY “Event has been recently updated”
            PopulateWithMyEvents
        ENDIF
    ENDIF
END dgv_events_Click


BEGIN CheckLastUpdated_Event(eventName)
    OPEN connection to cs
    POPULATE dgv_events with the information for this event
    CLOSE connection to cs
    SET lastUpdated TO lastUpdated column
    SET clientsLastUpdated TO lbl_lastUpdate
    IF lastUpdated is later than the clientsLastUpdated THEN
        RETURN false
    ELSE
        RETURN true
    ENDIF
```

```
END CheckLastUpdated_Event



BEGIN ReUpdate_Event(eventName)
    OPEN connection to cs
    UPDATE lastUpdated for eventName with the current time'
    CLOSE connection to cs
END ReUpdate_Event


REM "CellContentClick is triggered when a button is called associated for the event"
BEGIN dgv_events_CellContentClick
    IF user is using the most recent version of the table THEN
        IF user clicks LEAVE button THEN
            DISPLAY "Are you sure you want to leave?"
            GET DialogResult
            IF DialogResult == OK THEN
                LeaveEvent
                ReupdateEvent(selectedEvent)
                Refresh
            ENDIF
        ELSE IF user clicks VIEW button THEN
            IF the event has been hosted THEN
                SET frm_viewingInformation.eventName = selectedEvent
                DISPLAY frm_viewingInformation
            ENDIF
        ELSE
            REM "User must be selecting the VIEW button for an event that hasn't been
            hosted yet"
            DISPLAY "Please wait until the event is hosted"
        ENDIF
    ENDIF
END dgv_events_CellContentClick

BEGIN LeaveEvent
    PopulateSelectedEventDetails
    REM "If event has not yet been hosted then remove the user's votes when they leave (from
    attending)"
    IF event status is pending THEN
        RemoveUsersVotesRecord
    ENDIF
    RemoveUserFromEvent
    RemoveEventFromUser
END LeaveEvent

BEGIN RemoveUsersVotesRecord
    SET Votes as Array of dgv_currentEventSelection.recordVotes split at the '|' character
    SET currentUsersRecord TO ""
    SET currentUser TO ""
    SET idForUsersVote TO 0
    FOR i = 0 TO elements in Votes STEP 1
        IF Votes[i] != "" THEN
            SET currentUsersRecord TO the username part of Votes[i]
            IF currentUsersRecord == username THEN
                idForUsersVote = i
            ENDIF
        ENDIF
    NEXT i
```

```
        SET currentUser TO Votes[idForUsersVote]

        GET the current value of the options priority stored in the [ ] for each location and time/date option

        SET the new value for each option to the current priority – the priority the user voted

        SET updatedVoteRecord TO name of option + [new priority]

        OPEN connection to cs

        UPDATE with updatedVoteRecord, location_newPriorities, dayTime_newPriorities

        CLOSE connection to cs

END RemoveUsersVotesRecord


BEGIN PopulateSelectedEventDetails

        OPEN connection to cs

        POPULATE dgv_currentEventSelection with data for selectedEvent

        CLOSE connection to cs

END PopulateSelectedEventDetails


BEGIN RemoveUserFromEvent_Hosted

        SET attendees to attendees column from dgv_currentEventSelection

        REMOVE username from attendees

        OPEN connection to cs

        UPDATE attendees

        CLOSE connection to cs

END RemoveUserFromEvent_Hosted


BEGIN RemoveEventFromUser_Hosted

        SET events to events column from dgv_currentEventSelection

        REMOVE event from events

        OPEN connection to cs

        UPDATE events

        CLOSE connection to cs
```

```
END RemoveEventFromUser_Hosted
```

FRM_MYEVENTS

```
SET cs to connection string TO cloud database

SET username to user (public variable set from frm_nav)


BEGIN frm_myEvents

    SetupForm

    FOREACH row in dgv_events

        IF the status column of row == “Pending” THEN

            SET that row button TO “Host”

        ELSE

            SET row button TO “Finish”

        ENDIF


END frm_myEvents


BEGIN SetupForm

    SET dgv_events row height to 40pc

    Refresh

END SetupForm


BEGIN Refresh

    PopulateUser

    PopuldateWithMyEvents

END Refresh


BEGIN PopulateUser

    OPEN connection to cs

    POPULATE dgv_userData with information for the user
```

```
        CLOSE connection to cs
END PopulateUser


BEGIN PopulateWithMyEvents
        OPEN connection to cs
        POPULATE dgv_events where owner=username
        CLOSE connection to cs
END PopulateWithMyEvents


BEGIN btn_newEvent_Click
        Open frm_newEvent
END btn_newEvent_Click


INITIALISE selectedEvent
BEGIN dgv_events_CellContentClick
        IF mostRecent == true THEN
                IF cancel button is clicked THEN
                        DISPLAY "Are you sure you want to cancel" + selectedEvent
                        GET DialogResult
                        IF DialogResult == OK THEN
                                CancelEvent
                                Refresh
                                DISPLAY selectedEvent + "successfully cancelled"
                        ENDIF
                ELSE IF host/finish button is clicked THEN
                        IF status column of the event == "Pending" THEN
                                DISPLAY "Are you sure you want to host" + selectedEvent
                                GET DialogResult
                                IF DialogResult == OK THEN
```

```
                    PopulateSelectedEvent

                    HostEvent

                    Refresh

                    ReUpdate_Event(selectedEvent)

                    DISPLAY "Successfully hosted."

                ENDIF

            ELSE

                REM "Otherwise column must already be hosted in which case user
                would be wanting to finish the event"


                DISPLAY "Are you sure event is complete?"

                GET DialogResult

                IF DialogResult == OK THEN

                    CancelEvent

                    Refresh

                    ReUpdate_Event(selectedEvent)

                    DISPLAY "Successfully complete."

                ENDIF

            ENDIF

        ELSE IF manage button is clicked THEN

            Open the managing form

            SET eventname in frm_managing to selectedEvent

            SET username in frm_managing to selectedEvent

END dgv_events_CellContentClick


BEGIN PopulateSelectedEvent

    OPEN connection to cs

    POPULATE dgv_currentEventSelection where eventName = selectedEvent

    CLOSE connection to cs

END PopulateSelectedEvent
```

```
BEGIN HostEvent
    SetAccordingToVotes
    SetHostStatusAndClearInvitees
    AddToHostsEvents
END HostEvent
BEGIN AddToHostsEvents
    SET eventIDs to eventIDs column from dgv_currentUserInfo
    ADD selectedEvent to eventIDs
    OPEN connection to cs
    UPDATE eventIDs
    CLOSE connection to cs
END AddToHostsEvents


BEGIN SetAccordingToVotes
    SET equalVotes TO false
    SET dayTimeCurrentVotes TO array of
    SET lowestIndexes_dayTime TO list of 3 integers
    SET lowestValue_dayTime TO dayTimeCurrentVotes[0]
    ADD 0 TO lowestIndexes_dayTime
    FOR i = 1 TO dayTimeCurrentVotes
        SET vote TO dayTimeCurrentVotes[i]
        IF vote <> “”
            SET value TO the value contained between the [ ] of vote
            IF lowestValue_dayTime > value THEN
                lowestValue_dayTime = value
                CLEAR all elements from lowestIndexes_dayTime
                ADD i TO lowestIndexes_dayTime
            ENDIF
```

```
                IF lowestValue_dayTime == value THEN
                        lowestValue_dayTime = value
                        ADD i TO lowestIndexes_dayTime
                ENDIF
        ENDIF
NEXT i
REPEAT for lowestIndexes_location


REM “Now just get the name of the highest votes options”
SET dayTimeNames TO each option in the eventTime column of dgv_currentEventSelection split at ‘,’
FOR i = 1 TO dayTimeNames.Length – 1 STEP 1
        dayTimeNames[i] = dayTimeNames[i].Substring(0, index of the first [)
NEXT i
SET dayTime_newPriorities TO “”
FOREACH index in lowestIndexes_dayTime
        dayTime_highestVoted += dayTimeNames[index] + “,”
NEXT index


REPEAT for lowestIndexes_location to get location_ highestVoted


IF lowestIndexes_dayTime.Count > 1 THEN
        SET equalVotes TO true
ENDIF
IF lowestIndexes_location.Count > 1 THEN
        SET equalVotes TO true
ENDIF


OPEN connection to cs
UPDATE eventTime=eventTime_highestVoted eventLocation = location_highestVoted for selectedEvent
```

```
    CLOSE connection to cs


    REM “User must choose between the options that got even votes and the event location and time is
    set to this”

    IF equalVotes == true THEN

        Open frm_equalVotes

    ENDIF

END SetAccordingToVotes


INITIALISE invitedUser

BEGIN SetHostStatusAndClearInvitees

    OPEN connection to cs

    UPDATE status=”Hosted” where eventName = selectedEvent

    TRY

        SET invited TO invitees column of dgv_currentEventSelecton split at ,

        FOREACH user IN invited STEP 1

            SET invitedUser TO user

            UpdateInvitees

        NEXT user

    END TRY

    CATCH

    ENDCATCH

    UPDATE invitees=’ ‘ where eventName = selectedEvent

    CLOSE connection to cs

END SetHostStatusAndClearInvitees


INITIALISE attendingUser

BEGIN CancelEvent

    TRY
```

```
        SET invited TO invitees column of dgv_currentEventSelecton split at ‘,’
        FOREACH user IN invited STEP 1
            SET invitedUser TO user
            UpdateInvitees
        NEXT user
    END TRY
    CATCH
    ENDCATCH
    TRY
        SET attendees TO attendees column of dgv_currentEventSelecton split at ‘,’
        FOREACH user IN attendees STEP 1
            SET attendingUser TO user
            UpdateAttendees
        NEXT user
    END TRY
    CATCH
    ENDCATCH
    RemoveEventFromHost
    OPEN connection to cs
    DELETE RECORD where eventName = selectedEvent
    CLOSE connection to cs
END CancelEvent


BEGIN UpdateInvitees
    PopulateFriend(invitedUser)
    SET invitationIDs to invitationIDs column from dgv_friend
    REMOVE event from invitationIDs
    OPEN connection to cs
    UPDATE invitationIDs for invitedUser
    CLOSE connection to cs
END UpdateInvitees

BEGIN UpdateAttendees
    PopulateFriend(invitedUser)
    SET eventIDs to eventIDs column from dgv_friend
    REMOVE event from eventIDs
    OPEN connection to cs
    UPDATE eventIDs for attendingUser
    CLOSE connection to cs
END UpdateAttendees

BEGIN PopulateFriend(friend)
    OPEN connection to cs
    POPULATE dgv_friend where username=friend
    CLOSE connection to cs
END PopulateFriend

BEGIN RemoveEventFromHost
    SET eventIDs to eventIDs column from dgv_currentUserInfo
    REMOVE event from eventIDs
    OPEN connection to cs
    UPDATE eventIDs
    CLOSE connection to cs
END RemoveEventFromHost

BEGIN dgv_events_Click
    mostRecent = checkLastUpdated_Event(username)
    IF mostRecent == false THEN
```

```
        DISPLAY "Your events have been recently changed. Updating"
        Refresh
    ENDIF
END dgv_events_Click

BEGIN CheckLastUpdated_Event(eventName)
    OPEN connection to cs
    POPULATE table2 with the information where eventName=eventName
    CLOSE connection to cs
    SET lastUpdated TO lastUpdated column of table2
    SET clientsLastUpdated TO lbl_lastUpdate
    IF lastUpdated is later than the clientsLastUpdated THEN
        RETURN false
    ELSE
        RETURN true
    ENDIF
END CheckLastUpdated_Event

BEGIN ReUpdate_Event(eventName)
    OPEN connection to cs
    UPDATE lastUpdated for the event where eventName = eventName
    CLOSE connection to cs
END ReUpdate_User
```

FRM_INVITATIONVOTE

```
SET cs to connection string TO cloud database
SET username to user (public variable set from frm_nav)

BEGIN frm_invitationVote
```

```
    SetupForm

    Set up buttons so that

END frm_invitationVote


BEGIN SetupForm

    PopulateEvent

    PopulateUser

    SetEndTimes

    DISPLAY the duration

    DISPLAY each location and time/date options by creating a substring from 0 to the [ character. Place
    in a textbox beside a button which which the user will toggle to set the priority for the option.

    IF the option does not exist THEN

        Textbox = “NO OPTION SET”

    ENDIF

END SetupForm


BEGIN PopulateEvent

    OPEN connection to cs

    POPULATE dgv_eventData with information for the event

    CLOSE connection to cs

END PopulateEvent


BEGIN PopulateUser

    OPEN connection to cs

    POPULATE dgv_userData with information for the user

    CLOSE connection to cs

END PopulateUser
```

```
BEGIN SetEndTimes

    Add duration to the options for each time/date

END SetEndTimes


BEGIN btn_createEvent_Click

    IF the user has not set any options for location or time/date at the same priority THEN

        ReUpdate

        NewDateTimeVotes

        NewLocationVotes

        InsertVotes

        AddUserToAttendees

        MoveEventToAttending

        RecordUsersVote

        CALL ReturnToEvents FROM frm_invitation

        ReUpdate_Event(eventName)

        Close this form

    ELSE
        DISPLAY "Cannot have two or more buttons at the same priority"

    ENDIF

END btn_createEvent_Click


BEGIN ReUpdate_Event(eventName)

    OPEN connection to cs

    SET lastUpdated TO the time now

    CLOSE connection to cs

END ReUpdate_Event


BEGIN ReUpdate

    PopulateEvent
    PopulateUser
END ReUpdate_Event



INITIALISE dayTime_newPriorities;
BEGIN NewDayTimeVotes
    SET dayTime TO list of dayTimes split at ‘,’
    SET dayTimeNames TO list of dayTimes split at ‘,’
    FOR i = 0 TO dayTimeNames.Length STEP 1
        DayTimeName[i] = dayTimeName[i].Substring(0, ‘[‘)
    NEXT i
    SET dayTime_idOfOption1 to the index of lbl_dayTime_option1.Text
    SET dayTime_currentPriority_option1 TO dayTime[dayTime_idOfOption1].Substring(indexes between
    the [ and ])
    SET dayTime_newPriority_option1 TO dayTime_currentPriority_option1 + number on the button
    beside the option
    SET dayTime_redone_option1 TO lbl_dayTime_option1.Text + dayTime_newPriority_option1

    TRY
        SET dayTime_redone_option2 TO process above for lbl_dayTime_option2
    ENDTRY
    CATCH
    ENDCATCH
    TRY
        SET dayTime_redone_option3 TO process above for lbl_dayTime_option3
    ENDTRY
    CATCH
    ENDCATCH
    SET dayTime_newPriorities = dayTime_redone_option1 + dayTime_redone_option2 +
    dayTime_redone_option3
```

```
END NewDayTimeVotes



INITIALISE location_newPriorities;

BEGIN NewLocationVotes

        SET location TO list of locations split at ‘,’

        SET locationNames TO list of locations split at ‘,’

        FOR i = 0 TO locationNames.Length STEP 1

                LocationName[i] = locationName[i].Substring(0, ‘[‘)

        NEXT i

        SET location_idOfOption1 to the index of lbl_location_option1.Text

        SET location_currentPriority_option1 TO location[location_idOfOption1].Substring(indexes between
        the [ and ])

        SET location_newPriority_option1 TO location_currentPriority_option1 + number on the button
        beside the option

        SET location_redone_option1 TO lbl_location_option1.Text + location_newPriority_option1



        TRY

                SET location_redone_option2 TO process above for lbl_location_option2

        ENDTRY

        CATCH

        ENDCATCH

        TRY

                SET location_redone_option3 TO process above for lbl_location_option3

        ENDTRY

        CATCH

        ENDCATCH

        SET location_newPriorities = location_redone_option1 + location_redone_option2 +
        location_redone_option3

END NewLocationVotes
```

```
BEGIN AddUserToAttendees
    SET invitees to invitees column from dgv_userData
    REMOVE user from invitees
    SET attendees to attendees column from dgv_userData
    ADD user to attendees
    OPEN connection to cs
    UPDATE invitees
    UPDATE attendees
    CLOSE connection to cs
END RemoveUserFromEvent_Hosted
```

```
BEGIN AddUserToAttendees
    SET invitees to invitees column from dgv_userData
    REMOVE user from invitees
    SET attendees to attendees column from dgv_userData
    ADD user to attendees
    OPEN connection to cs
    UPDATE invitees
    UPDATE attendees
    CLOSE connection to cs
END RemoveUserFromEvent_Hosted
```

```
BEGIN MoveEventToAttending
    SET invitationIDs to invitationIDs column from dgv_eventData
    REMOVE event from invitationIDs
    SET eventIDs to eventIDs column from dgv_eventData
```

```
        ADD event to eventIDs

        OPEN connection to cs

        UPDATE invitationIDs

        UPDATE eventIDs

        CLOSE connection to cs

END MoveEventToAttending


BEGIN RecordUsersVotes

        SET votes to recordVotes column from dgv_eventData

        ADD username + “-[“ + record the user’s vote for each time/date option separated by , + “]-[“ + record
        the user’s vote for each location option separated by , + “]|”

        OPEN connection to cs

        UPDATE invitationIDs

        UPDATE recordVotes

        CLOSE connection to cs

END MoveEventToAttending


BEGIN InsertVotes

        OPEN connection to cs

        UPDATE with location_newPriorities and eventTime_newPriorities

        CLOSE connection to cs

END InsertVotes
```

FRM_FRIENDS

```
SET cs to connection string TO cloud database

SET username to user (public variable set from frm_nav)


BEGIN frm_friends

        ReUpdate
```

```
    When user clicks the friend name textbox, set the text to empty and if they leave the textbox and the
    text is still empty return to default text

    Friend name textbox cannot have ‘ or spaces entered

END frm_friends


BEGIN ReUpdate

    PopulateFriends

    PopulateRequests

    PopulateSentRequests

END ReUpdate


BEGIN PopulateFriends

    PopulateUser

    SET lbl_lastUpdate TO the time now

    FOREACH user IN the friends column of dgv_user STEP 1

        ADD user to dgv_friends

    NEXT user

END PopulateFriends


BEGIN PopulateSentRequests

    PopulateUser

    SET lbl_lastUpdate TO the time now

    FOREACH user IN the sentRequests column of dgv_user STEP 1

        ADD user to dgv_sentRequests

    NEXT user

END PopulateSentRequests


BEGIN PopulateRequests

    PopulateUser
```

```
        SET lbl_lastUpdate TO the time now
        FOREACH user IN the friendInvitations column of dgv_user STEP 1
                ADD user to dgv_friendRequests
        NEXT user
END PopulateSentRequests


BEGIN PopulateUser
        OPEN connection to cs
        POPULATE dgv_user with user's data
        CLOSE connection to cs
END PopulateUser


INITIALISE selectedFriendName
BEGIN dgv_friends_CellContentClick
        SET selectedFriendName to the name column of dgv_friends
        IF the remove button is clicked for user in any row THEN
                DISPLAY "Are you sure you want to remove" + selectedFriendName
                GET DialogResult
                IF DialogResult == OK THEN
                        RemoveFriend
                        ReUpdate_User(selectedFriendName)
                        ReUpdate_User(username)
                        PopulateFriends
                ENDIF
        ENDIF
END dgv_friends_CellContentClick


BEGIN RemoveFriend
        PopulateWithSelectedFriend
        RemoveFromUser

        RemoveFromFriend

        DISPLAY selectedFriendName + “successfully removed.”

END RemoveFriend


BEGIN PopulateWithSelectedFriend

        OPEN connection to cs

        POPULATE dgv_friend with user data where username=selectedFriendName

        CLOSE connection to cs

END PopulateWithSelectedFriend


BEGIN RemoveFriendFormUser

        SET friends TO friends column from dgv_userData

        REMOVE selectedFriendName from friends

        OPEN connection to cs

        UPDATE friends for user

        CLOSE connection to cs

END RemoveFormUser


BEGIN RemoveUserFormFriend

        SET friends TO friends column from dgv_friend

        REMOVE user from friends

        OPEN connection to cs

        UPDATE friends for selectedFriendName

        CLOSE connection to cs

END RemoveFormUser


BEGIN btn_requestFriend_Click

        Exists = checkUserExists
```

```
    IF txt_friendName.Text == user logged in THEN
        DISPLAY “You cannot add yourself as a friend”
    ELSE
        IF exists == true THEN
            Message = CheckIfFriendExists
            IF message == “” THEN
                PopulateWithFriendRequest
                AddFriends
                ReUpdate_User(txt_friendName.Text)
                ReUpdate
                DISPLAY “Request successfully sent to ” + txt_friendName.Text”
            ELSE
                DISPLAY message
            ENDIF
        ELSE
            DISPLAY “User does not exist.”
        ENDIF
    ENDIF
END btn_requestFriend_Click


BEGIN PopulateWIthFriendRequest
    OPEN connection to cs
    POPULATE dgv_friend where username = txt_friendName.Text
    CLOSE connection to cs
END PopulateWIthFriendRequest


BEGIN CheckIfFriendExists
    SET message TO “”
    SET frendName = txt_friendName.Text
```

```
        SET friendExists TO if the friends column of dgv_user contains friendname
        SET inviteExists TO if the sentReqeusts column of dgv_user contains friendname
        SET recieved TO if the friendInvitations column of dgv_user contains friendname

        IF friendsExists == true THEN
                SET message TO "You are already friends with this user."
        ELSE IF sentExists == true THEN
                SET message TO "You have already requested this user."
        ELSE IF received == true THEN
                SET message TO "You already have a request from this user."
        ENDIF
        RETURN message
END CheckIfFriendExists

BEGIN AddFriends
        UserToFriendsList
        FriendToSentList
END AddFriends

BEGIN UserToFriends
        SET friendName TO txt_friendName.Text
        SET sentRequests TO sentRequests column from dgv_friend
        ADD user TO sentRequests
        OPEN connection to cs
        UPDATE sentRequests for friendName
        CLOSE connection to cs
END UserToFriends
```

```
BEGIN FriendToSentList
    SET friendName TO txt_friendName.Text
    SET sentRequests TO sentRequests column from dgv_userData
    ADD friendName TO sentRequests
    OPEN connection to cs
    UPDATE sentRequests
    CLOSE connection to cs
END FriendToSentList


BEGIN CheckIfUserExists
    OPEN connection to cs
    POPULATE table2 where username = txt_friendName.Text
    CLOSE connection to cs
    IF number of rows in table2 > 0 THEN
        RETURN true
    ELSE
        RETURN false
    ENDIF
END CheckIfUserExists


INITALISE selectedInviteName
BEGIN dgv_friendInvites_CellContentClick
    REM "public variable set"
    REM "mostRecent is called on the click event"
    IF mostRecent == true
        SET selectedInviteName to username column of selected row
        IF accept friend request button is clicked THEN
            DISPLAY "Are you sure you want to add" + selectedInviteName
            GET DialogResult
            IF DialogResult == OK THEN
                MoveUsers
                ReUpdate_User(selectedInviteName)
                ReUpdate_User(username)
                REM "update the datagridview"
                ReUpdate
            ENDIF
        ELSE IF ignore button is clicked THEN
            DISPLAY "Are you sure you want to ignore request from" + selectedInviteName
            GET DialogResult
            IF DialogResult == OK THEN
                RemoveRequest
                ReUpdate_User(selectedInviteName)
                ReUpdate_User(username)
                REM "update the datagridview"
                ReUpdate
            ENDIF
        ENDIF
    ENDIF
END dgv_friendInvites_CellContentClick


BEGIN RemoveRequest
    PopulateUser
    PopulateWithFriend
    RemoveInviteFormUser
    RemoveFromFriendsSent
END RemoveRequest


BEGIN PopulateWithFriend
```

```
        OPEN connection to cs
        POPULATE dgv_friend with information where username = selectedInviteName
        CLOSE connection to cs
END PopulateWithFriend


BEGIN RemoveInviteFormUser
        SET friendInvitations TO friendInvitations column from dgv_user
        REMOVE selectedInviteName from friendInvitations
        OPEN connection to cs
        UPDATE friendInvitations
        CLOSE connection to cs
END RemoveInviteFromUser


BEGIN RemoveFromFriendsSent
        SET sentRequests TO sentRequests column from dgv_friend
        REMOVE username from sentRequests
        OPEN connection to cs
        UPDATE sentRequests for selectedInviteName
        CLOSE connection to cs
END RemoveFromFriendsSent


BEGIN MoveUsers
        PopulateUser
        PopulateWithFriend
        MoveFriend
        MoveUser
END MoveUsers
BEGIN MoveFriend
        SET friendInvitations TO friendInvitations column from dgv_user
        REMOVE selectedInviteName from friendInvitations

        SET friends TO friends column from dgv_user

        ADD selectedInviteName TO friends

        OPEN connection to cs

        UPDATE friendInvitations

        UPDATE friends

        CLOSE connection to cs

BEGIN MoveFriend


BEGIN MoveUser

        SET friendInvitations TO friendInvitations column from dgv_friend

        REMOVE user from friendInvitations

        SET friends TO friends column from dgv_friend

        ADD user TO friends

        OPEN connection to cs

        UPDATE friendInvitations for selectedInviteName

        UPDATE friends for selectedInviteName

        CLOSE connection to cs

BEGIN MoveFriend


INITALISE mostRecent
```

REM "Do not confuse dgv_friends which displays all of the user's friends and dgv_friend which displays information for the selected friend"

```
BEGIN dgv_friends_Click

        mostRecent = checkLastUpdated_User(username)

        IF mostRecent == false THEN

                DISPLAY "Your friends have been recently changed. Updating"

                ReUpdate

        ENDIF
```

```
END dgv_friends_Click


BEGIN CheckLastUpdated_User(user)
        OPEN connection to cs
        POPULATE table2 with the information for this event
        CLOSE connection to cs
        SET lastUpdated TO lastUpdated column of table2
        SET clientsLastUpdated TO lbl_lastUpdate
        IF lastUpdated is later than the clientsLastUpdated THEN
                RETURN false
        ELSE
                RETURN true
        ENDIF
END CheckLastUpdated_User


BEGIN ReUpdate_User
        OPEN connection to cs
        UPDATE lastUpdated for the user with the current time
        CLOSE connection to cs
END ReUpdate_User


BEGIN dgv_friendInvites_Click
        mostRecent = checkLastUpdated_User(username)
        IF mostRecent == false THEN
                DISPLAY “Your friends have been recently changed. Updating”
                ReUpdate
        ENDIF
END dgv_friendInvites_Click
```

```
INITIALISE selectedRequestName

BEGIN  dgv_sentRequests_CellContentCLick

        IF mostrecent == true THEN

                IF unrequest button is pressed THEN

                        SET selectedRequestName TO username of row selected

                        DISPLAY "Are you sure you want to remove your friend request to" +
                        selectedRequestName

                        GET DialogResult

                        IF DialogResult == OK THEN

                                PopulateWithSelectedRequestFriend

                                ReUpdate_User(selectedRequestName)

                                ReUpdate_User(username)

                                RemoveSentRequestOfFriend

                                RemoveRequestFromFriend

                                ReUpdate

                                DISPLAY "Removed friend request"

                        ENDIF

                ENDIF
        ENDIF

END dgv_sentRequests_CellContentClick


BEGIN PopulateWithSelectedRequestFriend

        OPEN connection to cs

        POPULATE dgv_requestData with information where username= selectedRequestName

        CLOSE connection to cs

END PopulateWithSelectedRequestFriend


BEGIN dgv_sentRequests_Click

        mostRecent = checkLastUpdated_User(username)

        IF mostRecent == false THEN
```

```
            DISPLAY "Your friends have been recently changed. Updating"
            ReUpdate
        ENDIF
END dgv_sentRequests_Click


BEGIN RemoveSentRequestOfFriend
        SET sentRequests TO sentRequests column from dgv_user
        REMOVE selectedRequestName from sentRequests
        OPEN connection to cs
        UPDATE sentRequests
        CLOSE connection to cs
END RemoveSentRequestOfFriend


BEGIN RemoveRequestFromFriend
        SET friendInvitations TO friendInvitations column from dgv_requestData
        REMOVE user from friendInvitations
        OPEN connection to cs
        UPDATE friendInvitations for selectedRequestName
        CLOSE connection to cs
END RemoveRequestFromFriend
```

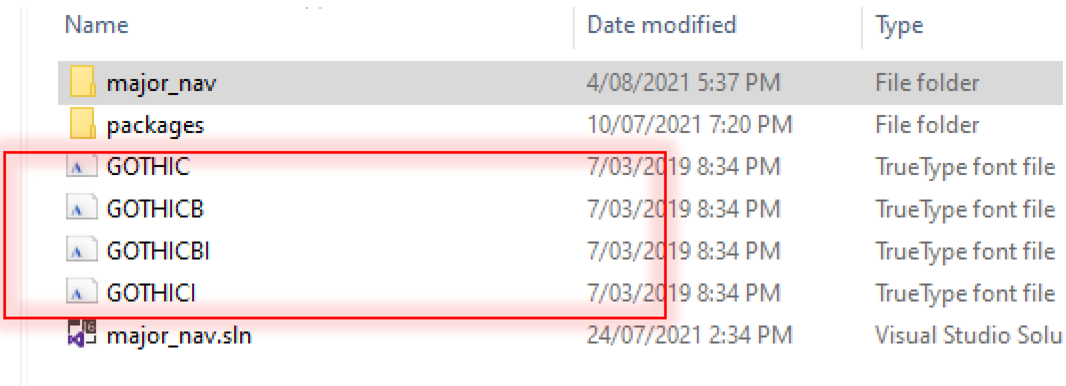
PRE-USE

CENTURY GOTHIC

Jade Harris | **12SDD**

TO INSTALL FONT

I. Locate the CENTURY GOTHIC fonts included in the software package
   a. Else, download off: https://freefontsfamily.com/century-gothic-font-family/
II. Drag into C:\Windows\Fonts
   a. Else, follow these instructions: https://faqs.skillcrush.com/article/275-downloading-installing-a-font-on-your-computer



Name	Date modified	Type
major_nav	4/08/2021 5:37 PM	File folder
packages	10/07/2021 7:20 PM	File folder
GOTHIC	7/03/2019 8:34 PM	TrueType font file
GOTHICB	7/03/2019 8:34 PM	TrueType font file
GOTHICBI	7/03/2019 8:34 PM	TrueType font file
GOTHICI	7/03/2019 8:34 PM	TrueType font file
major_nav.sln	24/07/2021 2:34 PM	Visual Studio Solu

QUICK-START

USER DOCUMENTATION

CONNECT

Jade Harris | **12SDD**

LAST UPDATED: 2021

CONTENTS (Click to navigate to)

NEW USER

MY EVENTS

MY INVITATIONS

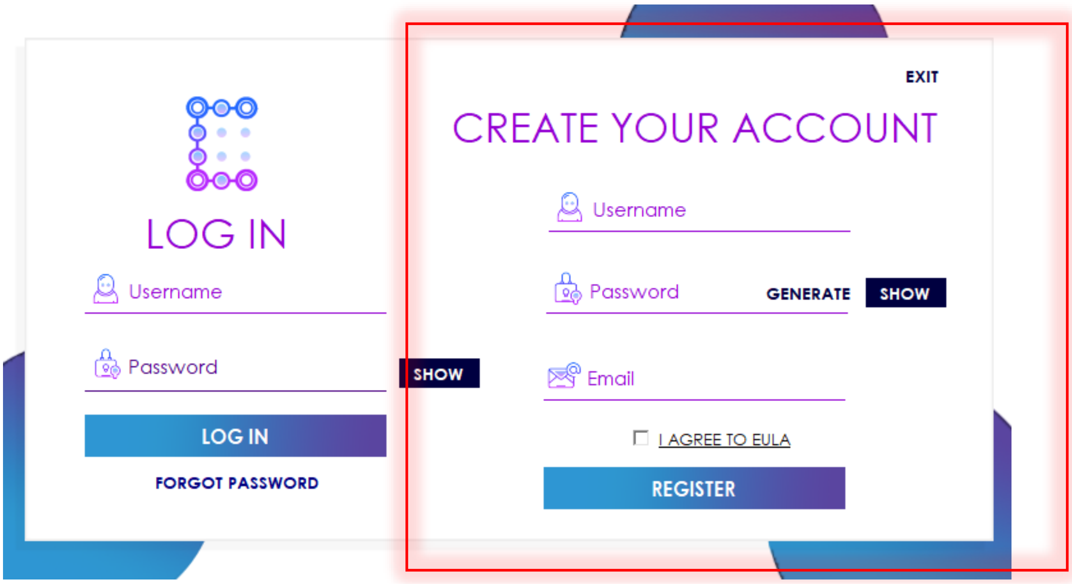
EVENTS

MY FRIENDS

SETTINGS

NEW USER

CREATING A USER



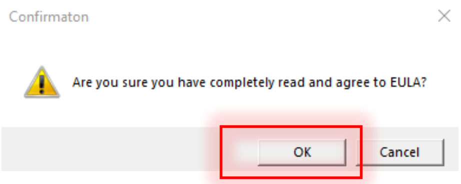
NOTE: No textboxes can have ‘ or spaces entered (including create event)

I. Create a username
   a. Max 9 characters

II. Create a password
   a. Max 12 characters
   b. GENERATE: 12 random alpha-numeric characters
   c. SHOW: When HOVERING, reveal your password

III. Enter your email
   a. Max 25 characters
   b. Ensure you have access to email

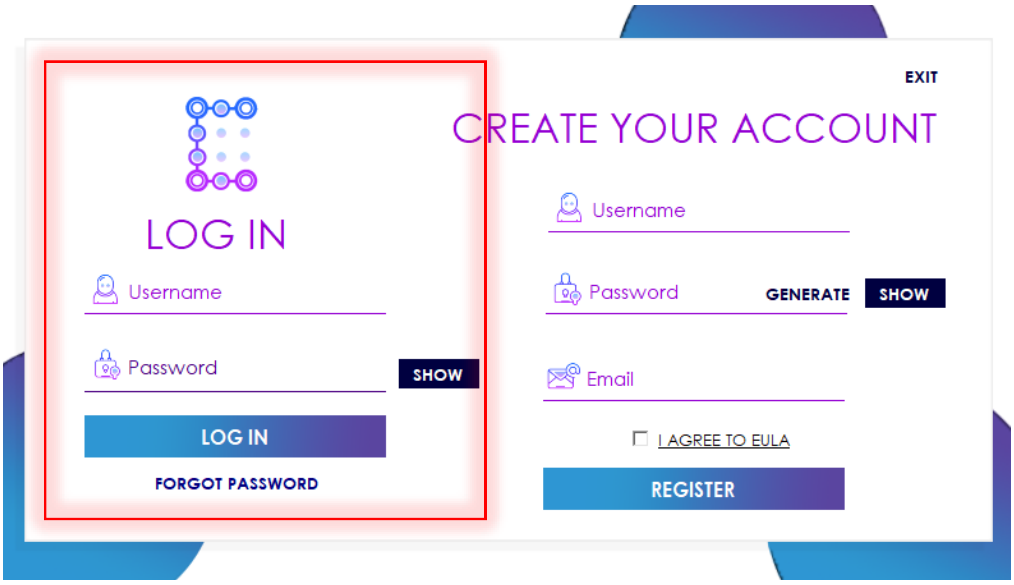
IV. Read and agree to EULA



Are you sure you have completely read and agree to EULA?

NEW USER

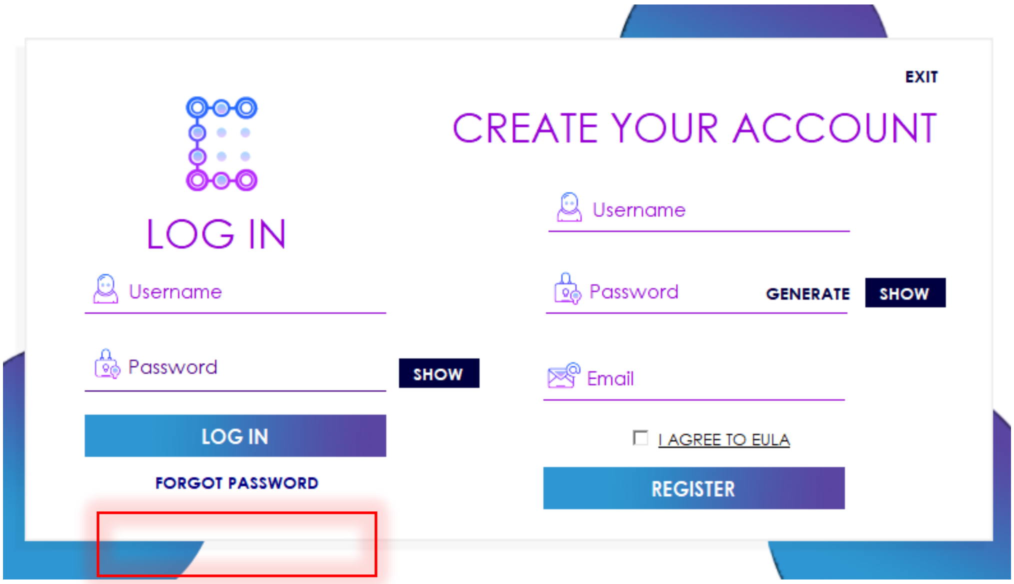
LOG IN



I. Enter username
II. Enter password
    a. SHOW: When HOVERING, reveal your password

NEW USER

FORGET PASSWORD

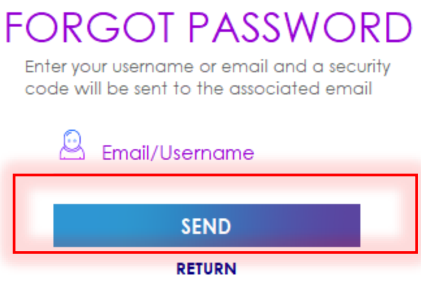


I. Select FORGOT PASSWORD

II. Enter remembered email/username

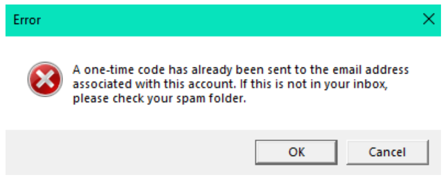
a. Ensure you have access to email

b. If you already have requested a code before, check your emails.



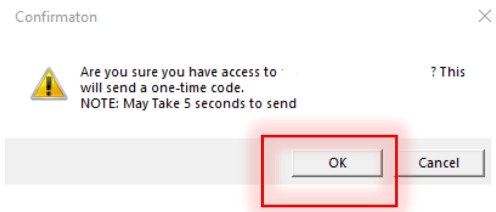
FORGOT PASSWORD

Enter your username or email and a security code will be sent to the associated email



Error

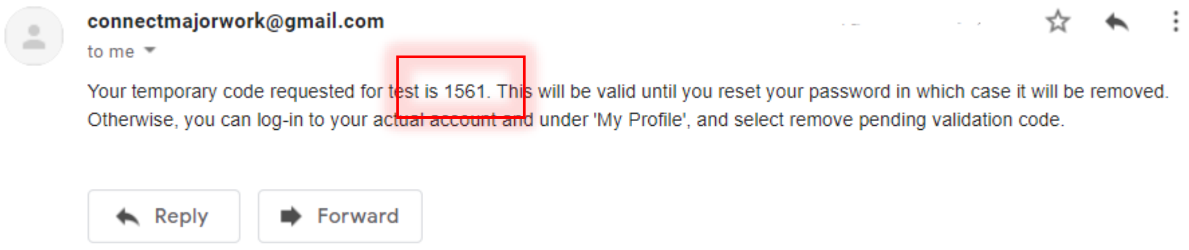
A one-time code has already been sent to the email address associated with this account. If this is not in your inbox, please check your spam folder.



Confirmaton

Are you sure you have access to ? This will send a one-time code.
NOTE: May Take 5 seconds to send

III. Get security code

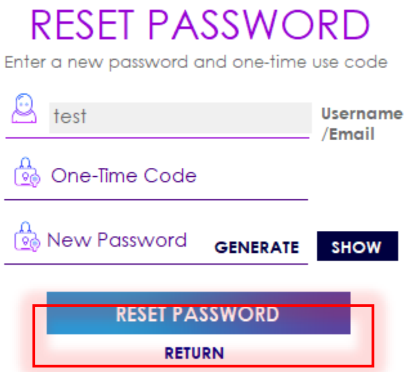


connectmajorwork@gmail.com

to me

Your temporary code requested for test is 1561. This will be valid until you reset your password in which case it will be removed. Otherwise, you can log-in to your actual account and under 'My Profile', and select remove pending validation code.

IV. Enter this code and your new password



RESET PASSWORD

Enter a new password and one-time use code

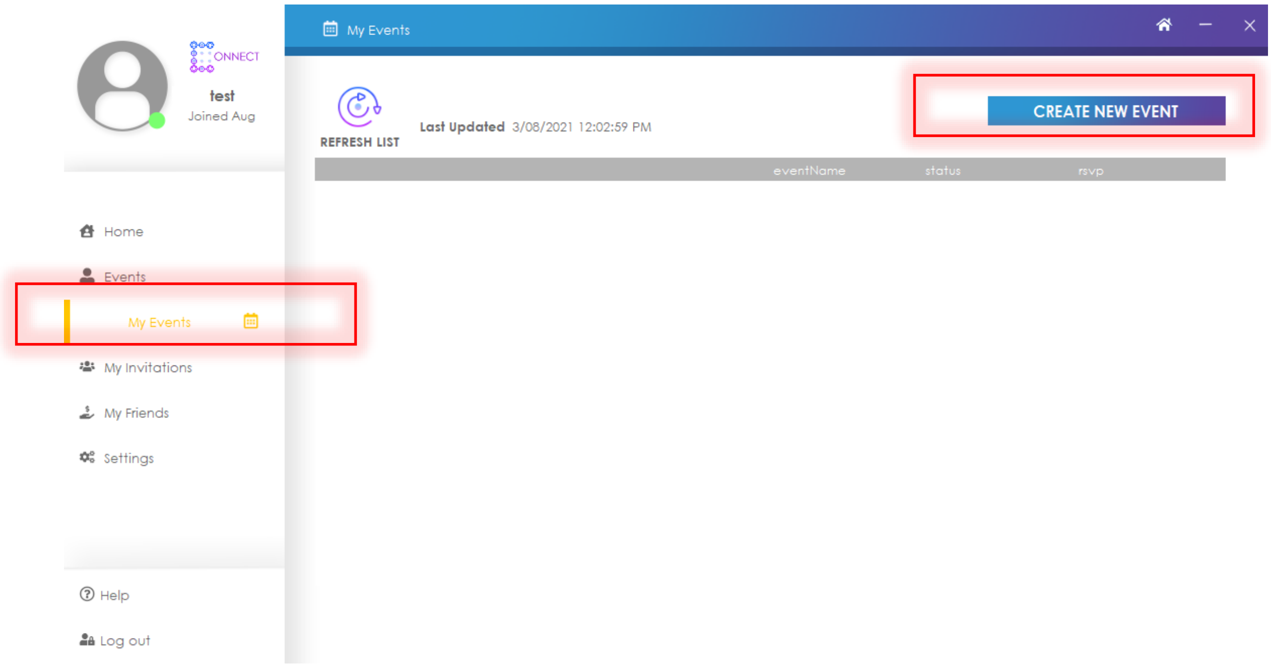
Username /Email: test

MY EVENTS

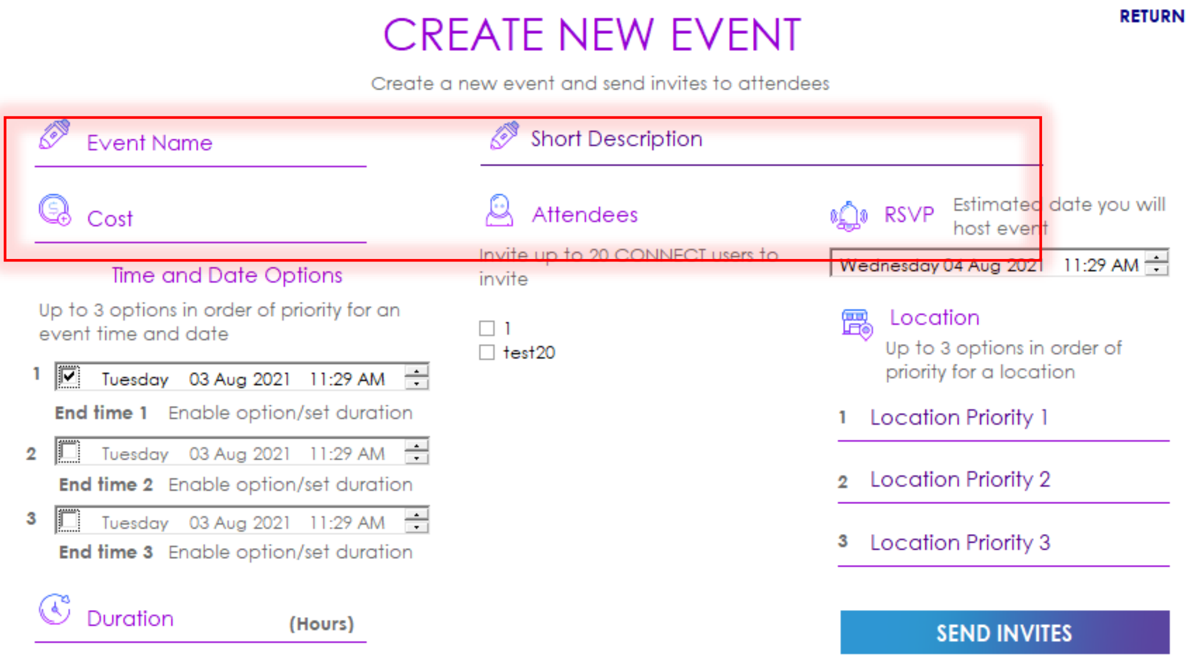
CREATE NEW EVENT

I. Go to MY EVENTS

II. Select CREATE NEW EVENT



Last Updated 3/08/2021 12:02:59 PM



Estimated date you will host event

Wednesday 04 Aug 2021 11:29 AM

Invite up to 20 CONNECT users to invite

☐ 1

☐ test20

Up to 3 options in order of priority for an event time and date

1 ☑ Tuesday 03 Aug 2021 11:29 AM

End time 1 Enable option/set duration

2 ☐ Tuesday 03 Aug 2021 11:29 AM

End time 2 Enable option/set duration

3 ☐ Tuesday 03 Aug 2021 11:29 AM

End time 3 Enable option/set duration

Up to 3 options in order of priority for a location

1 Location Priority 1

2 Location Priority 2

3 Location Priority 3

I. Enter event name
   a. Max 20 characters
   b. NOTE: Your event cannot have the same event as another regardless of who is the host

II. Enter short description
   a. Max 40 characters

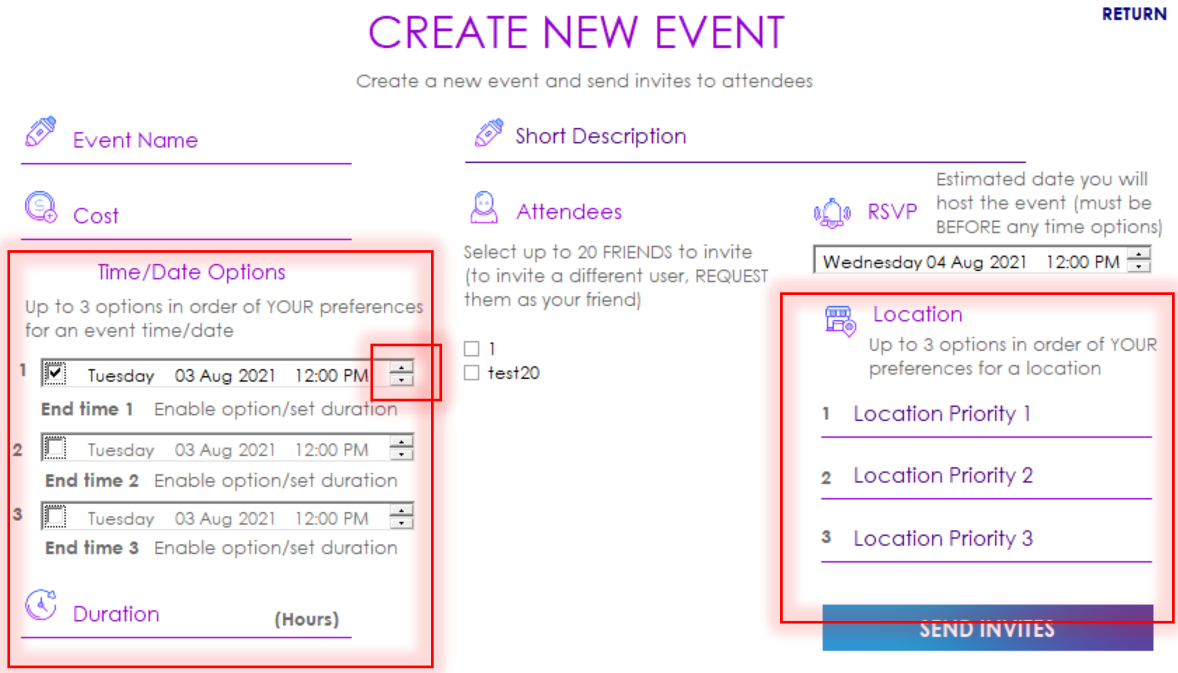
III. Enter cost
   a. To 2 decimal places

IV. Enter short description
   a. Max 40 characters
   b.

SETTING TIME AND LOCATION PREFERENCES

I. Add YOUR preferences for location/time(date) for the event

II. Use ARROWS for time/date
   a. Selecting only ONE time date option will mean USERS CANNOT VOTE and these will automatically be set when you host your event
   b. No time/location can be identical
   c. Once hosted, NO OPTIONS CAN BE ADDED (only removed)
   d. Time and date options are influenced by DURATION



CREATE NEW EVENT

RETURN

Create a new event and send invites to attendees

Event Name

Short Description

Cost

Attendees

Select up to 20 FRIENDS to invite (to invite a different user, REQUEST them as your friend)

□ 1

□ test20

RSVP

Estimated date you will host the event (must be BEFORE any time options)

Wednesday 04 Aug 2021 12:00 PM

Time/Date Options

Up to 3 options in order of YOUR preferences for an event time/date

1 ☑ Tuesday 03 Aug 2021 12:00 PM

End time 1 Enable option/set duration

2 □ Tuesday 03 Aug 2021 12:00 PM

End time 2 Enable option/set duration

3 □ Tuesday 03 Aug 2021 12:00 PM

End time 3 Enable option/set duration

Duration (Hours)

Location

Up to 3 options in order of YOUR preferences for a location

1 Location Priority 1

2 Location Priority 2

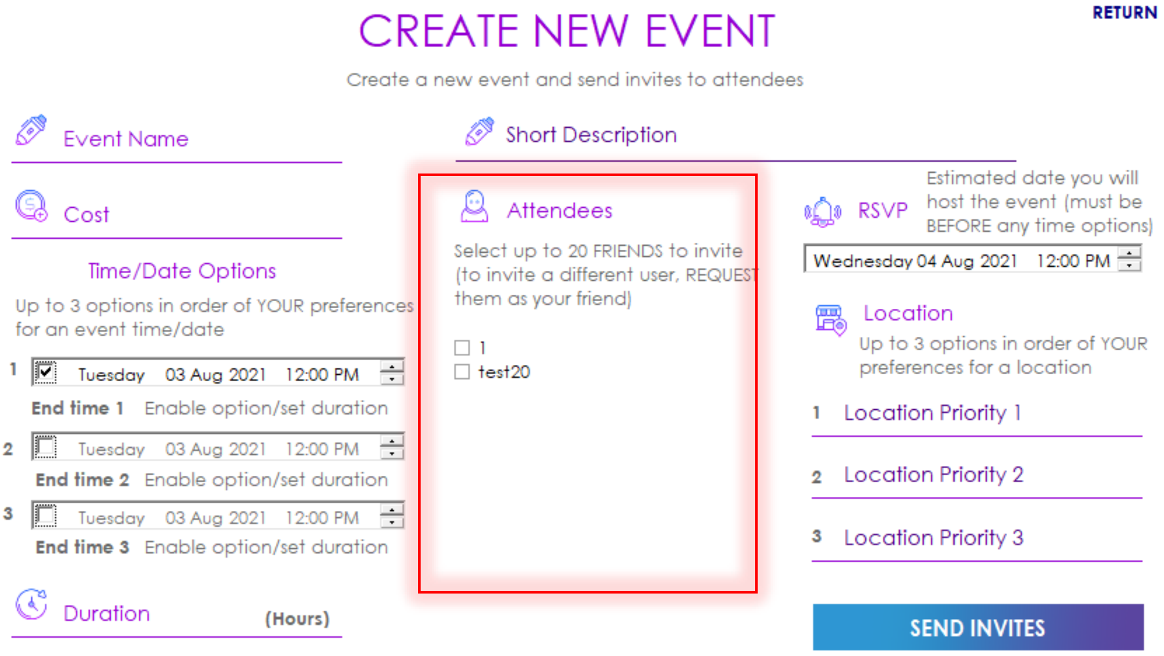
3 Location Priority 3

INVITING FRIENDS (ATTENDEES)

=

NOTE: To invite friends, YOU MUST ADD THEM FIRST (ADD A FRIEND)

I. Use checkboxes to select friends



RETURN

CREATE NEW EVENT

Create a new event and send invites to attendees

Event Name

Short Description

Cost

Attendees

Select up to 20 FRIENDS to invite (to invite a different user, REQUEST them as your friend)

□ 1
□ test20

RSVP

Estimated date you will host the event (must be BEFORE any time options)

Wednesday 04 Aug 2021 12:00 PM

Time/Date Options

Up to 3 options in order of YOUR preferences for an event time/date

1 ☑ Tuesday 03 Aug 2021 12:00 PM
End time 1 Enable option/set duration

2 □ Tuesday 03 Aug 2021 12:00 PM
End time 2 Enable option/set duration

3 □ Tuesday 03 Aug 2021 12:00 PM
End time 3 Enable option/set duration

Location

Up to 3 options in order of YOUR preferences for a location

1 Location Priority 1

2 Location Priority 2

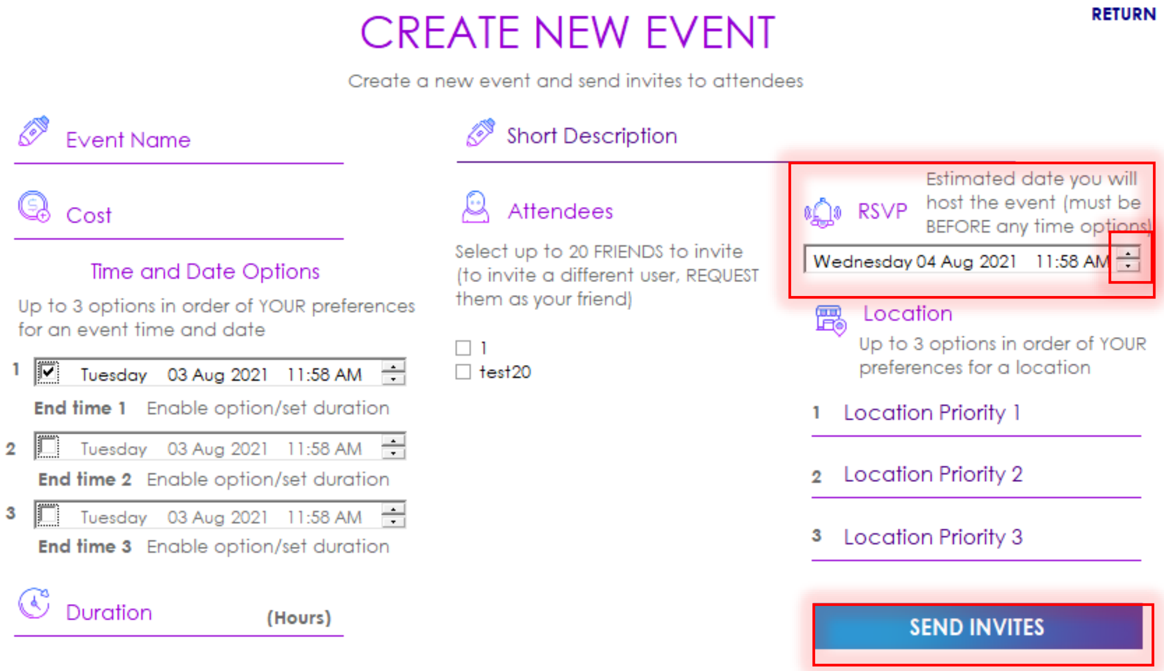
3 Location Priority 3

Duration (Hours)

SEND INVITES

SETTING RSVP

I. Must be BEFORE any time/date option



RETURN

CREATE NEW EVENT

Create a new event and send invites to attendees

Event Name

Short Description

Cost

Attendees

Select up to 20 FRIENDS to invite (to invite a different user, REQUEST them as your friend)

□ 1
□ test20

RSVP

Estimated date you will host the event (must be BEFORE any time options)

Wednesday 04 Aug 2021 11:58 AM

Time and Date Options

Up to 3 options in order of YOUR preferences for an event time and date

1 ☑ Tuesday 03 Aug 2021 11:58 AM
End time 1 Enable option/set duration

2 □ Tuesday 03 Aug 2021 11:58 AM
End time 2 Enable option/set duration

3 □ Tuesday 03 Aug 2021 11:58 AM
End time 3 Enable option/set duration

Location

Up to 3 options in order of YOUR preferences for a location

1 Location Priority 1

2 Location Priority 2

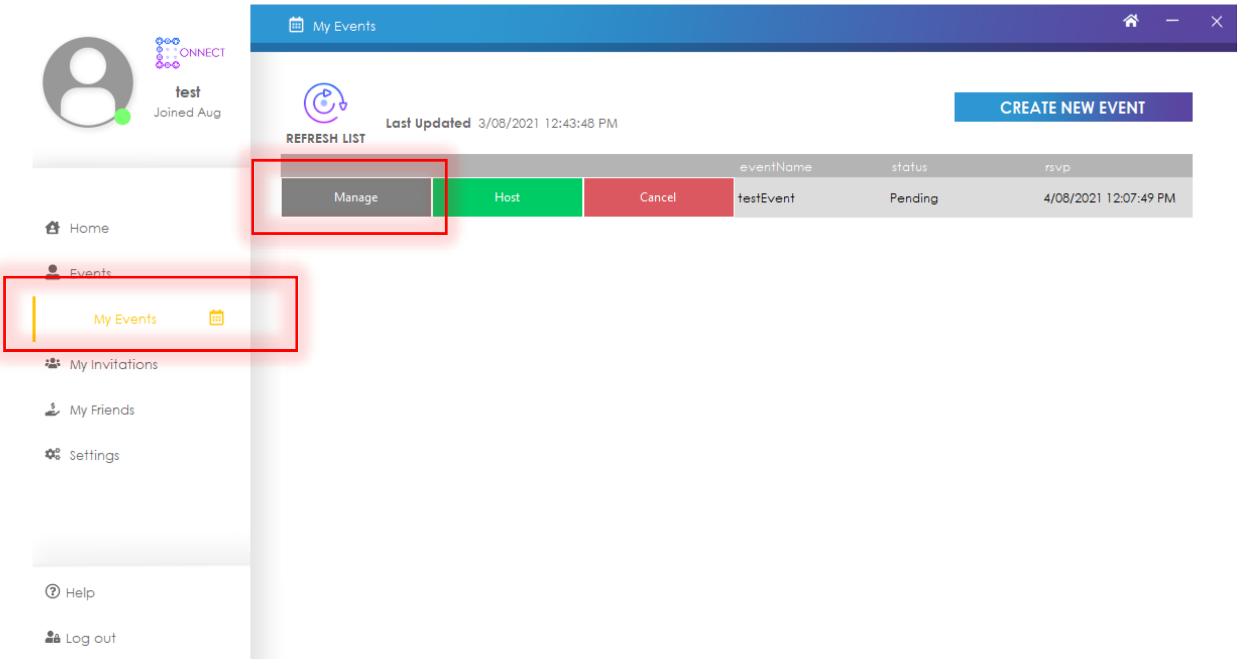
3 Location Priority 3

Duration (Hours)

SEND INVITES

MANAGING MY EVENTS

I. Go to MY EVENTS
II. Click MANAGE

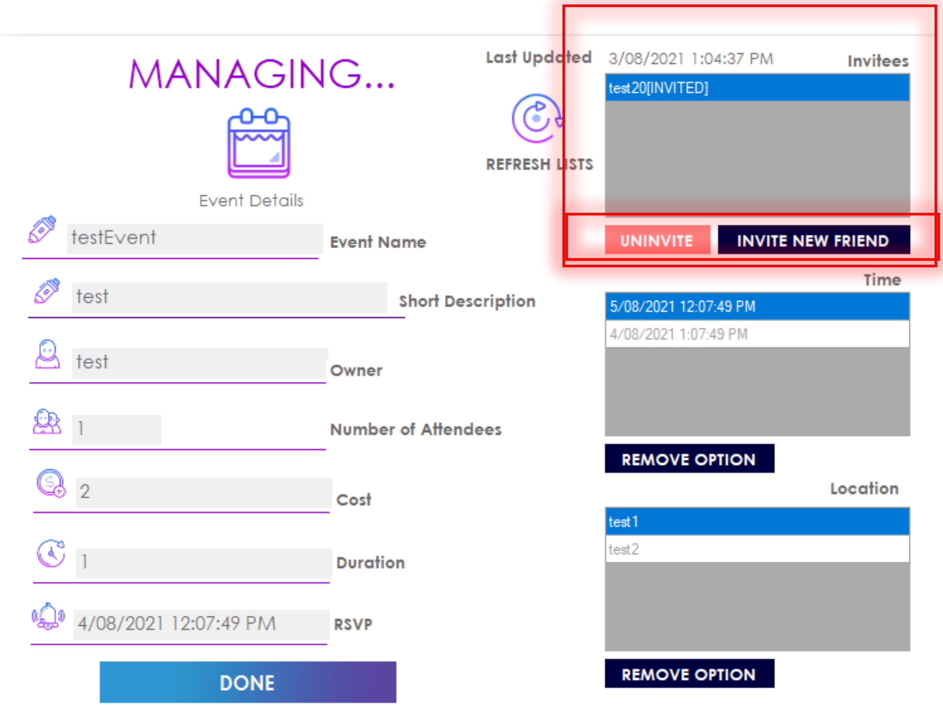


test
Joined Aug

Last Updated 3/08/2021 12:43:48 PM

			eventName	status	rsvp
Manage	Host	Cancel	testEvent	Pending	4/08/2021 12:07:49 PM

MANAGE(ADD/REMOVE) ATTENDEE/INVITEES



Last Updated 3/08/2021 1:04:37 PM

Invitees
test20[INVITED]

Event Name	testEvent
Short Description	test
Owner	test
Number of Attendees	1
Cost	2
Duration	1
RSVP	4/08/2021 12:07:49 PM

Time
5/08/2021 12:07:49 PM
4/08/2021 1:07:49 PM

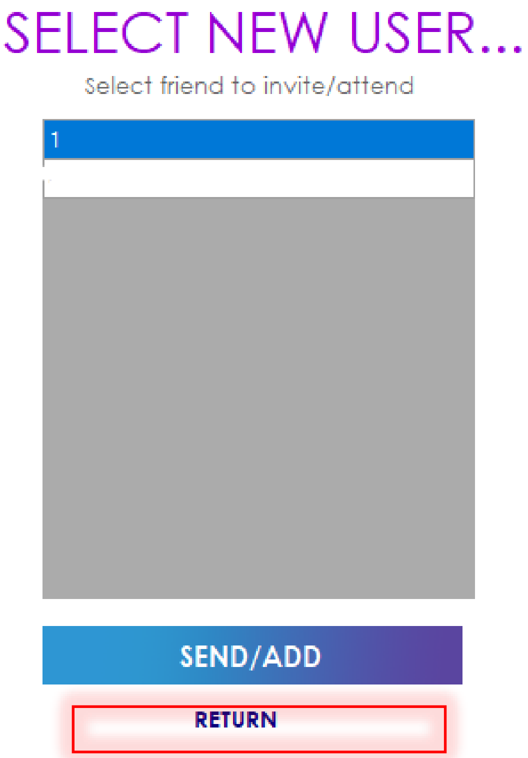
Location
test1
test2

REMOVE ATTENDEE/INVITEE

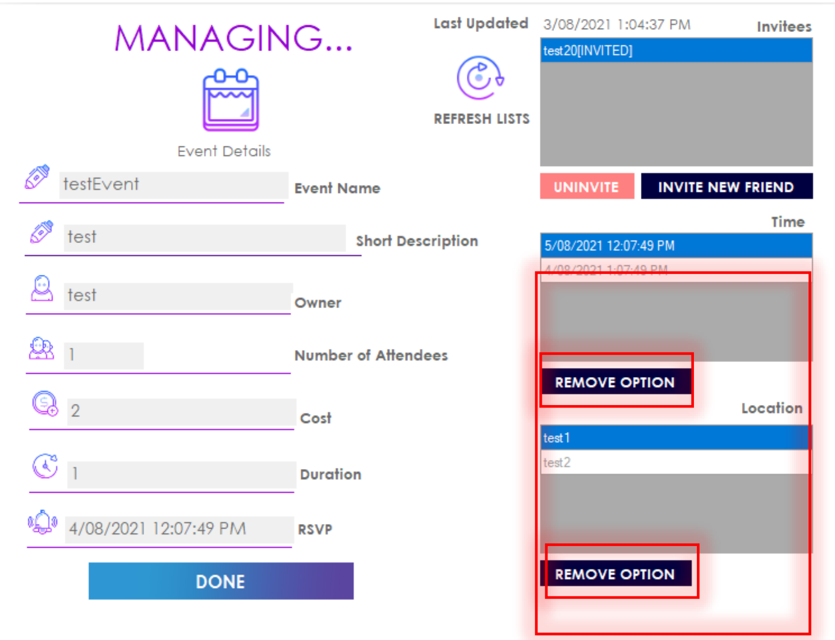
I. Select USER
   a. Users suffixed with whether they have already accepted invite
II. Click UNINVITE

ADD NEW ATTENDEE/INVITEE

III. Click INVITE NEW FRIEND
   a. If EVENT
      i. ALREADY HOSTED: user automatically added (they can leave)
      ii. PENDING: invite sent (for their vote)



REMOVE TIME/LOCATION OPTION



Event Name: testEvent

Short Description: test

Owner: test

Number of Attendees: 1

Cost: 2

Duration: 1

RSVP: 4/08/2021 12:07:49 PM

Last Updated 3/08/2021 1:04:37 PM

Invitees

test20[INVITED]

Time

5/08/2021 12:07:49 PM

4/08/2021 1:07:49 PM

Location

test1

test2

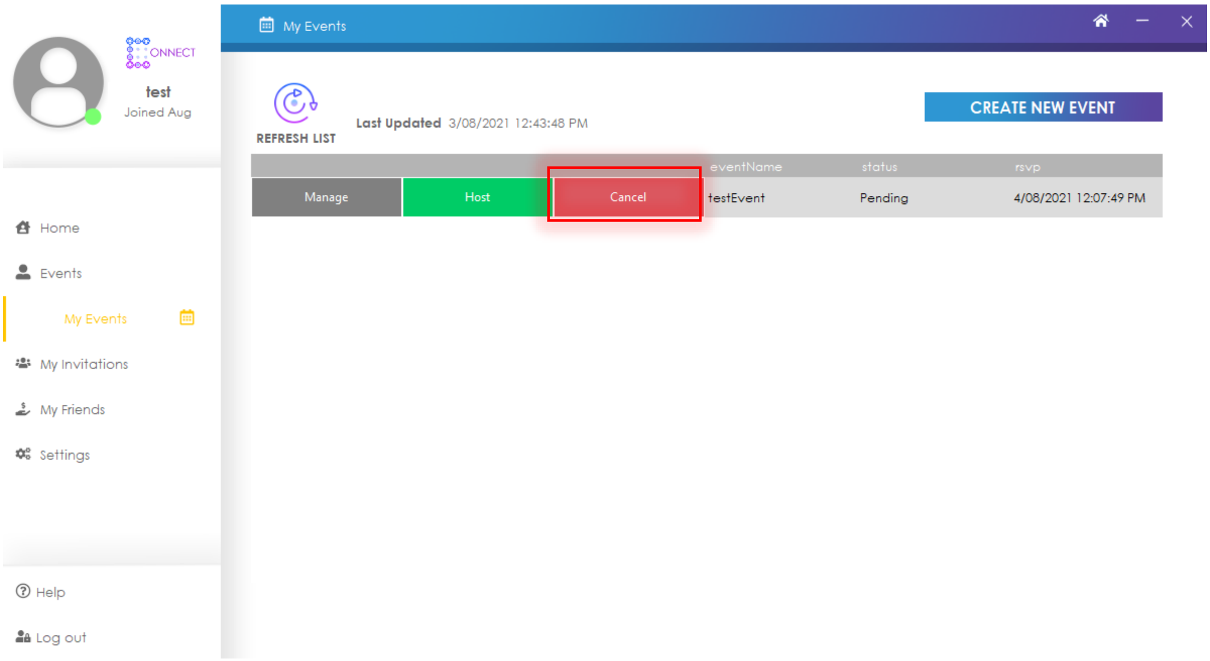
I. Select appropriate option

II. Click REMOVE OPTION

a. Once removed, you CANNOT ADD another option

MY EVENTS

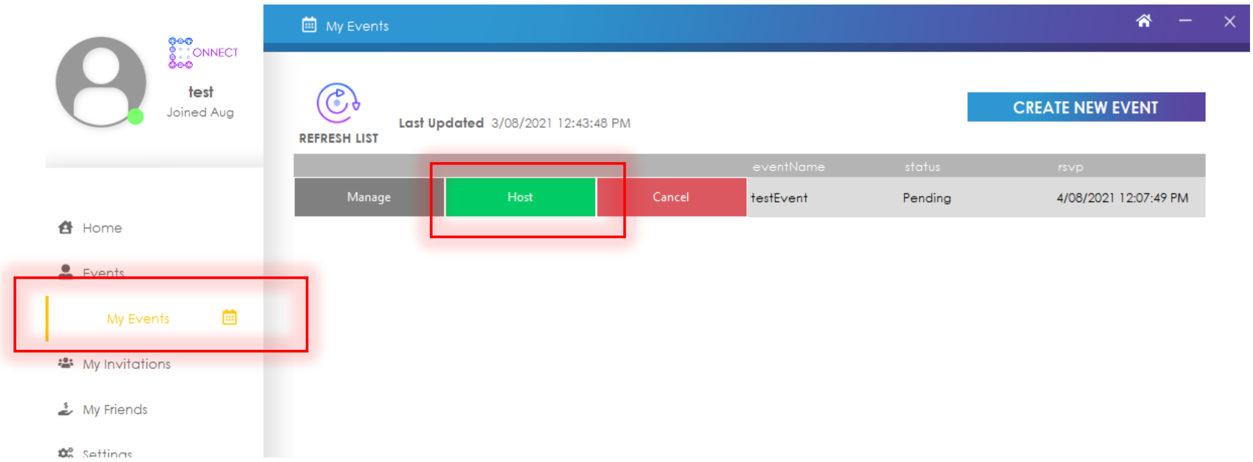
CANCEL AN EVENT



Last Updated 3/08/2021 12:43:48 PM

			eventName	status	rsvp
Manage	Host	Cancel	testEvent	Pending	4/08/2021 12:07:49 PM

HOST AN EVENT



Last Updated 3/08/2021 12:43:48 PM

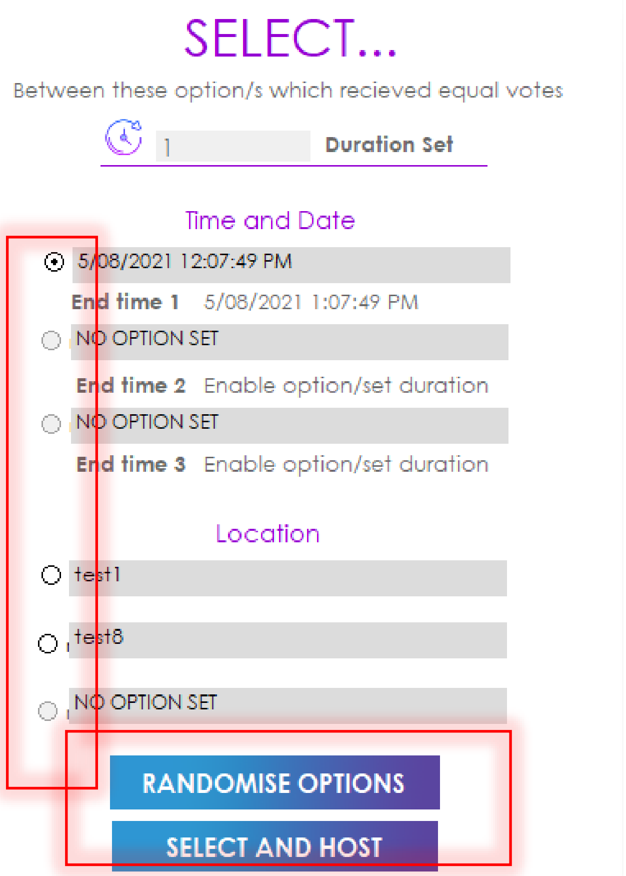
			eventName	status	rsvp
Manage	Host	Cancel	testEvent	Pending	4/08/2021 12:07:49 PM

I. Go to MY EVENTS

II. Click HOST

III. If no options received the same number of votes, then HOSTING the event will automatically set the location and time to the option that received the highest number of votes. The invitees will then become attendees, and once the event is done you can complete it.

EQUAL VOTES



Between these option/s which recieved equal votes

1 Duration Set

Time and Date

● 5/08/2021 12:07:49 PM

End time 1 5/08/2021 1:07:49 PM

○ NO OPTION SET

End time 2 Enable option/set duration

○ NO OPTION SET

End time 3 Enable option/set duration

Location

○ test1

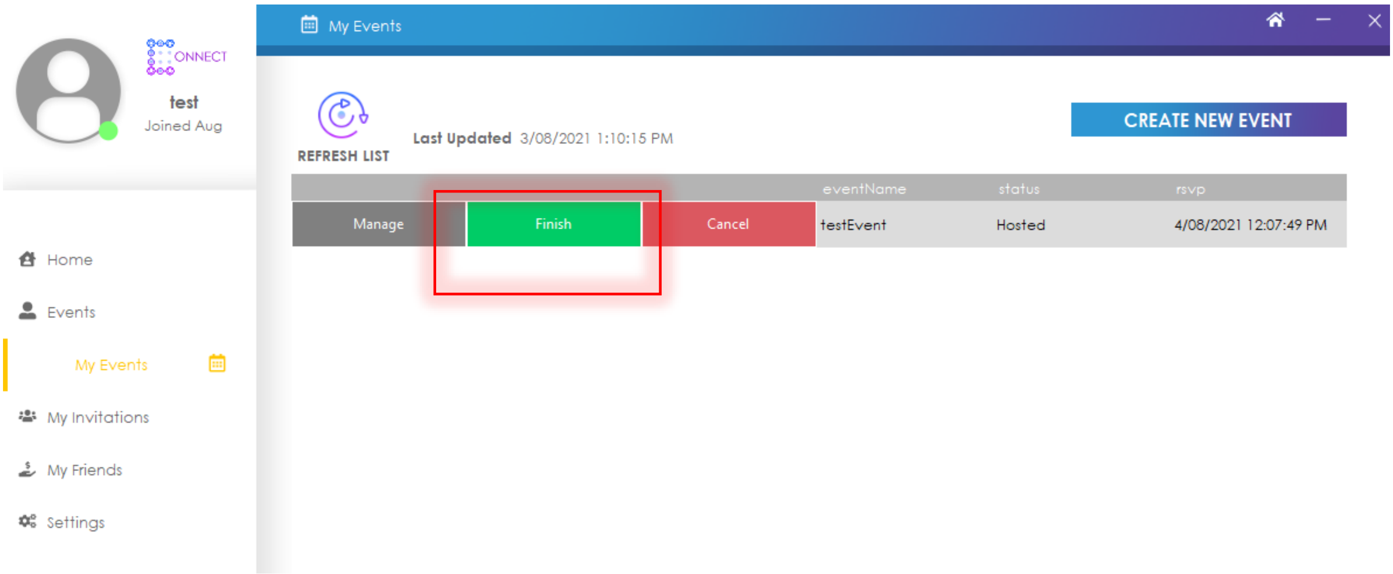
○ test8

○ NO OPTION SET

I. Select RADIO BUTTONS of option that the event will be set to
   a. RANDOMISE OPTIONS will randomly select the radio buttons

EVENTS

COMPLETING EVENT



Last Updated 3/08/2021 1:10:15 PM

			eventName	status	rsvp
Manage	Finish	Cancel	testEvent	Hosted	4/08/2021 12:07:49 PM

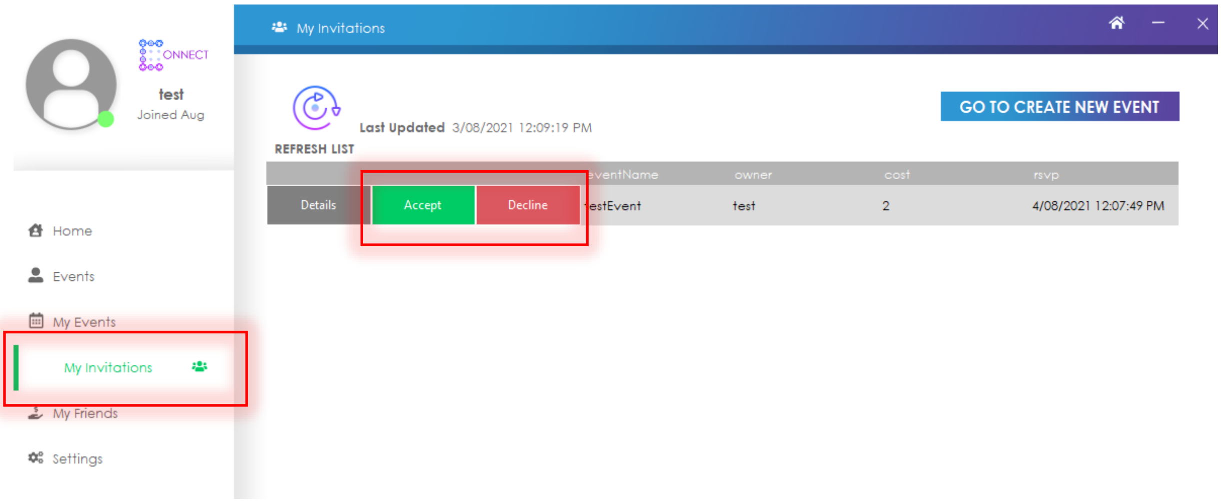
I. If event is hosted, HOST button becomes FINISH
II. Once you are ready to COMPLETE event, CLICK FINISH
   a. This is irreversible. Event will be removed.

MY INVITATIONS

ACCEPT/DECLINE INVITATION

I. Go to MY INVITATIONS

II. Use buttons to ACCEPT/DENY selected invitation



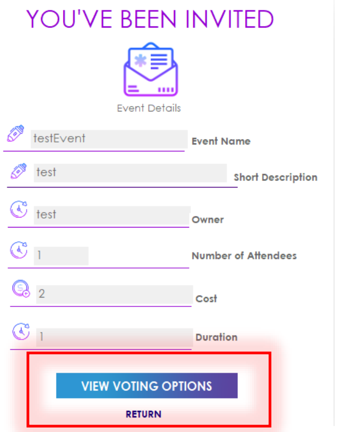
test
Joined Aug

Last Updated 3/08/2021 12:09:19 PM

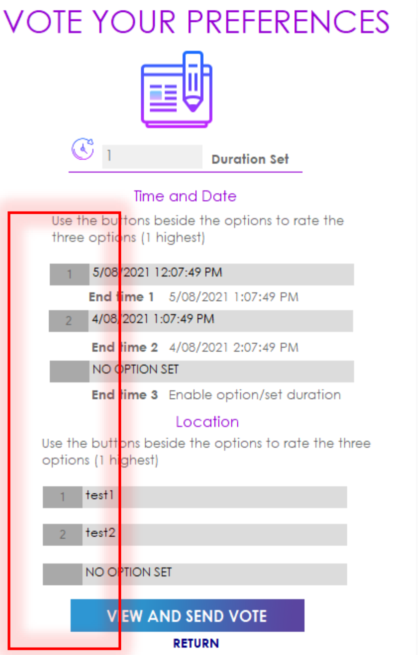
			eventName	owner	cost	rsvp
Details	Accept	Decline	testEvent	test	2	4/08/2021 12:07:49 PM

SETTING VOTES/PREFERENCES

III. Click ACCEPT button



testEvent	Event Name
test	Short Description
test	Owner
1	Number of Attendees
2	Cost
1	Duration



1 Duration Set

Time and Date

Use the buttons beside the options to rate the three options (1 highest)

1 5/08/2021 12:07:49 PM
End time 1 5/08/2021 1:07:49 PM
2 4/08/2021 1:07:49 PM
End time 2 4/08/2021 2:07:49 PM
NO OPTION SET
End time 3 Enable option/set duration

Location

Use the buttons beside the options to rate the three options (1 highest)

1 test1
2 test2
NO OPTION SET

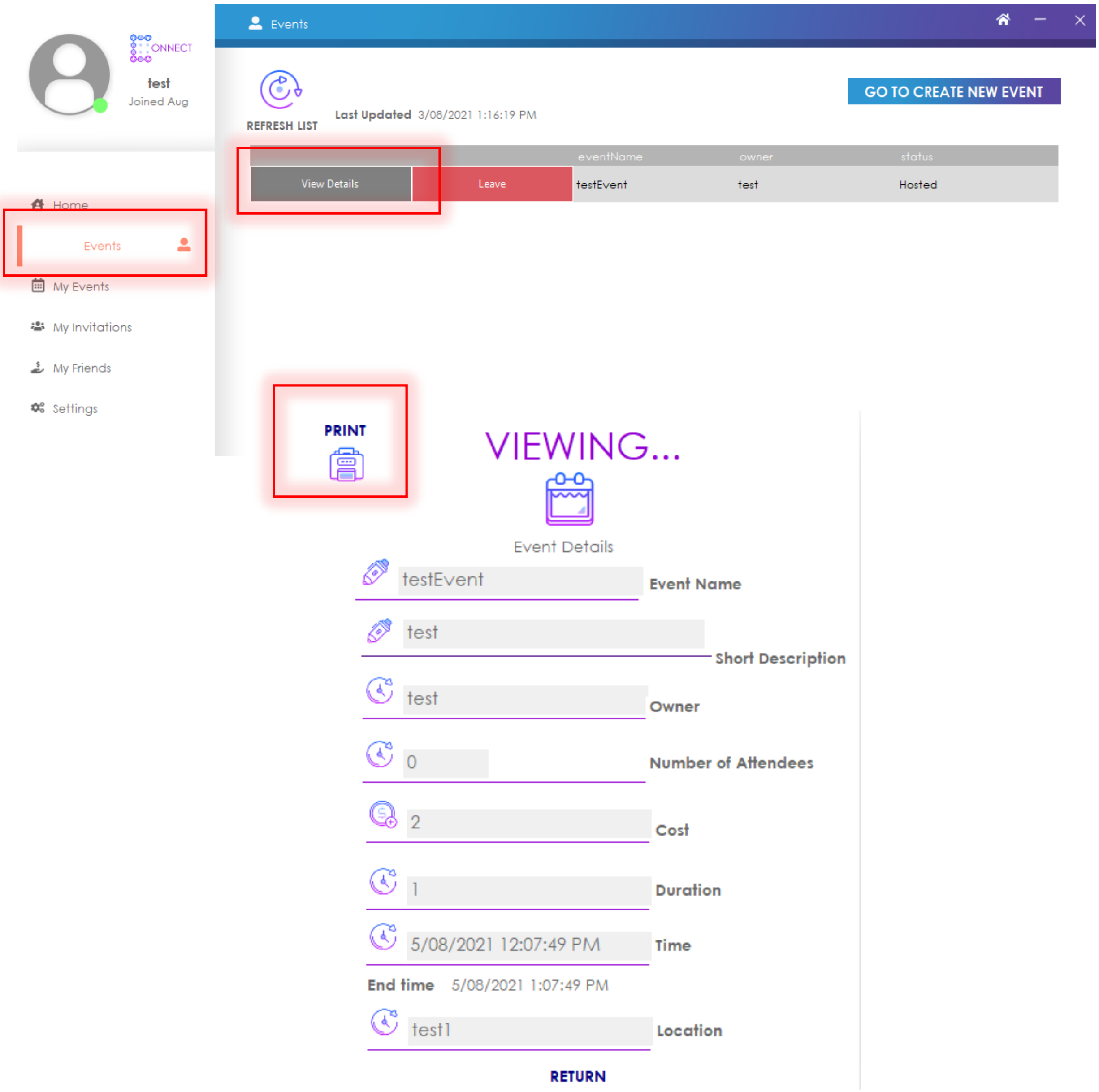
IV. Use the buttons beside each option to toggle in order of preference

EVENTS

VIEW DETAILS

IF EVENT IS HOSTED

I. Go to MY INVITATIONS
II. Click VIEW DETAILS
III. Option to PRINT event details



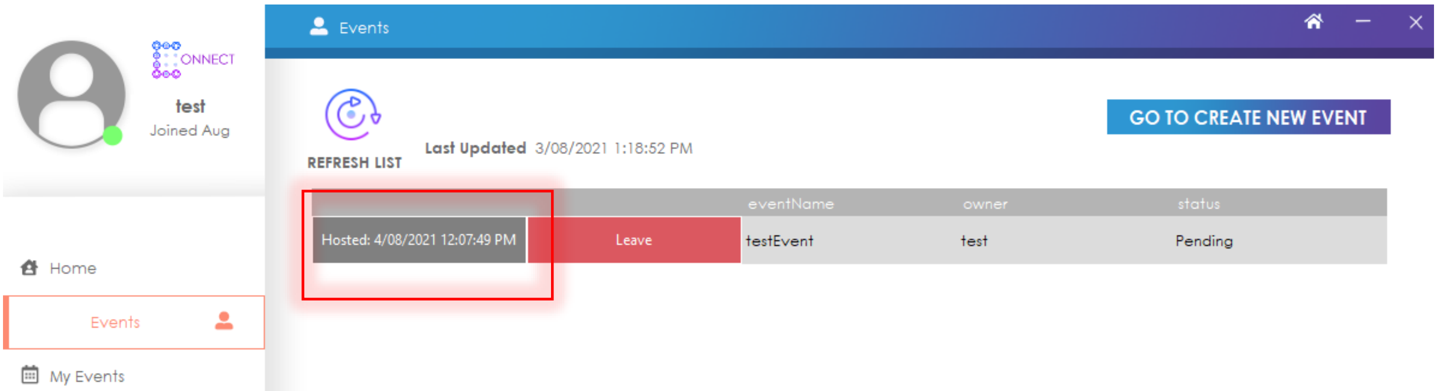
Last Updated 3/08/2021 1:16:19 PM

		eventName	owner	status
View Details	Leave	testEvent	test	Hosted

Event Name: testEvent
Short Description: test
Owner: test
Number of Attendees: 0
Cost: 2
Duration: 1
Time: 5/08/2021 12:07:49 PM
End time 5/08/2021 1:07:49 PM
Location: test1

IF EVENT IS NOT HOSTED:

I. Cannot see details only RSVP date



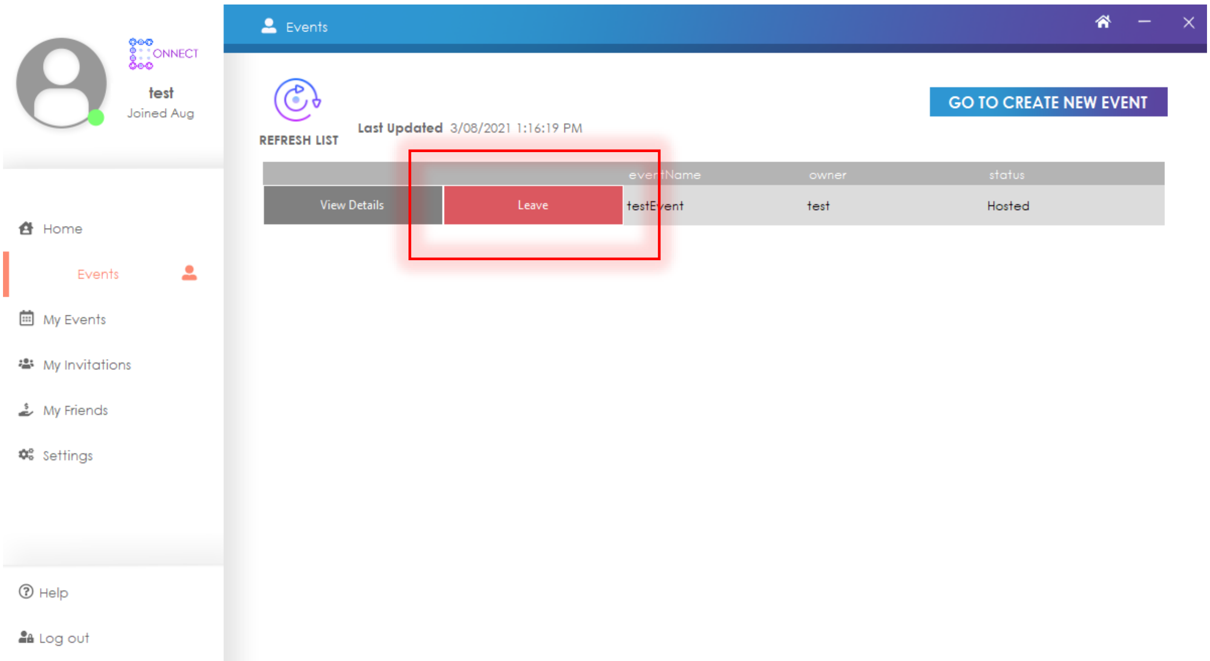
Last Updated 3/08/2021 1:18:52 PM

		eventName	owner	status
Hosted: 4/08/2021 12:07:49 PM	Leave	testEvent	test	Pending

EVENTS

LEAVING AN EVENT

I. NOTE: Leaving an event not yet hosted will **remove your votes**



Last Updated 3/08/2021 1:16:19 PM

		eventName	owner	status
View Details	Leave	testEvent	test	Hosted

MY FRIENDS

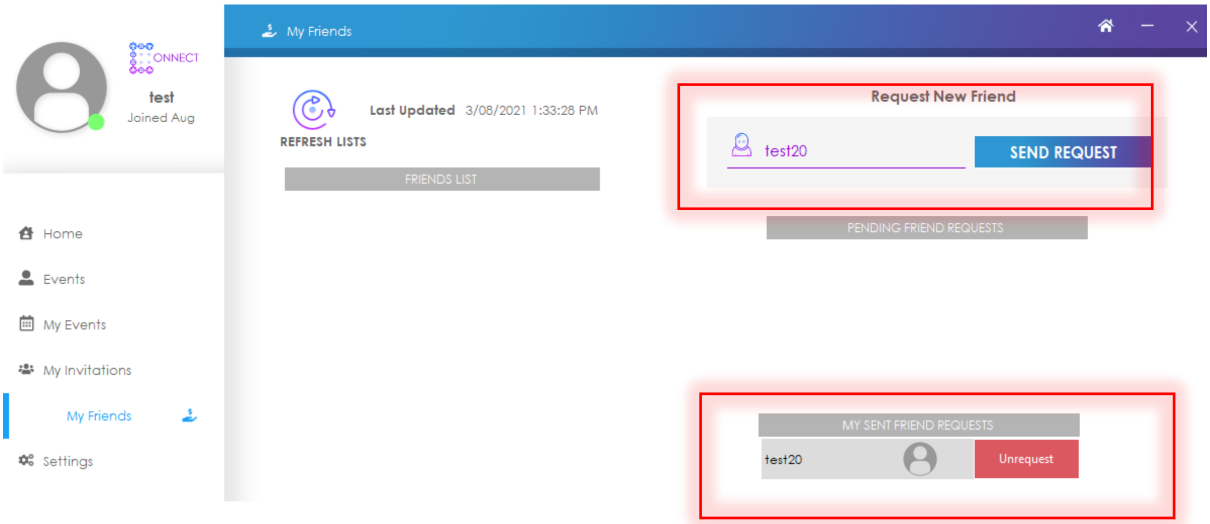
SENT/CANCEL FRIEND REQUEST

SEND

I. Enter username of other account
II. Wait for user to accept

CANCEL REQUEST

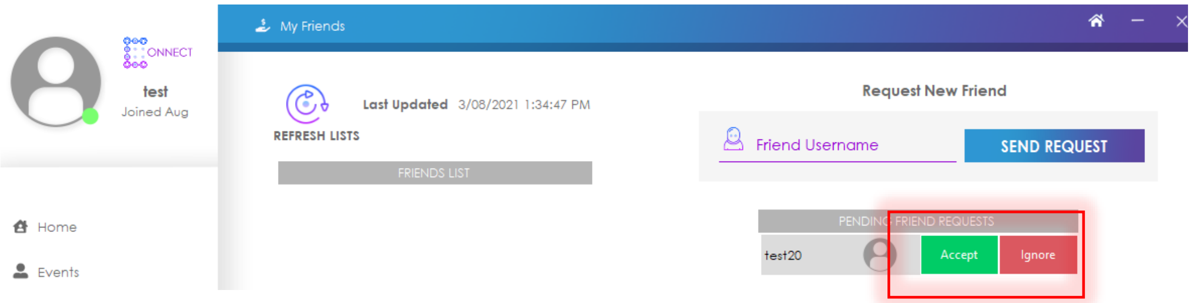
I. (Assuming user has accepted request)
II. Click UNREQUEST



MY FRIENDS

ACCEPT/IGNORE RECEIVED REQUEST

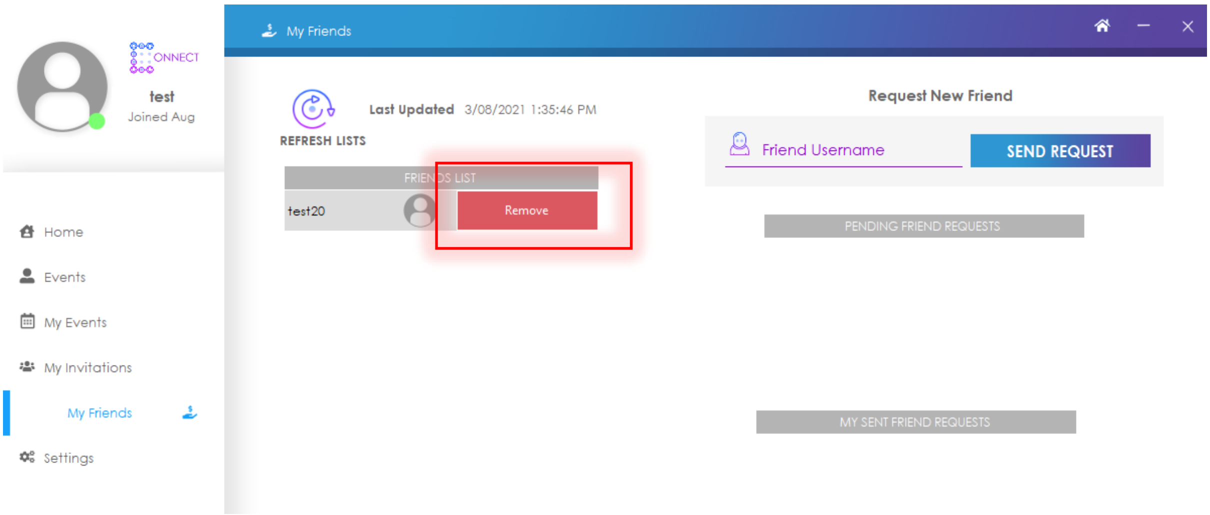
I. Use buttons to ACCEPT or IGNORE request



MY FRIENDS

REMOVE FRIEND

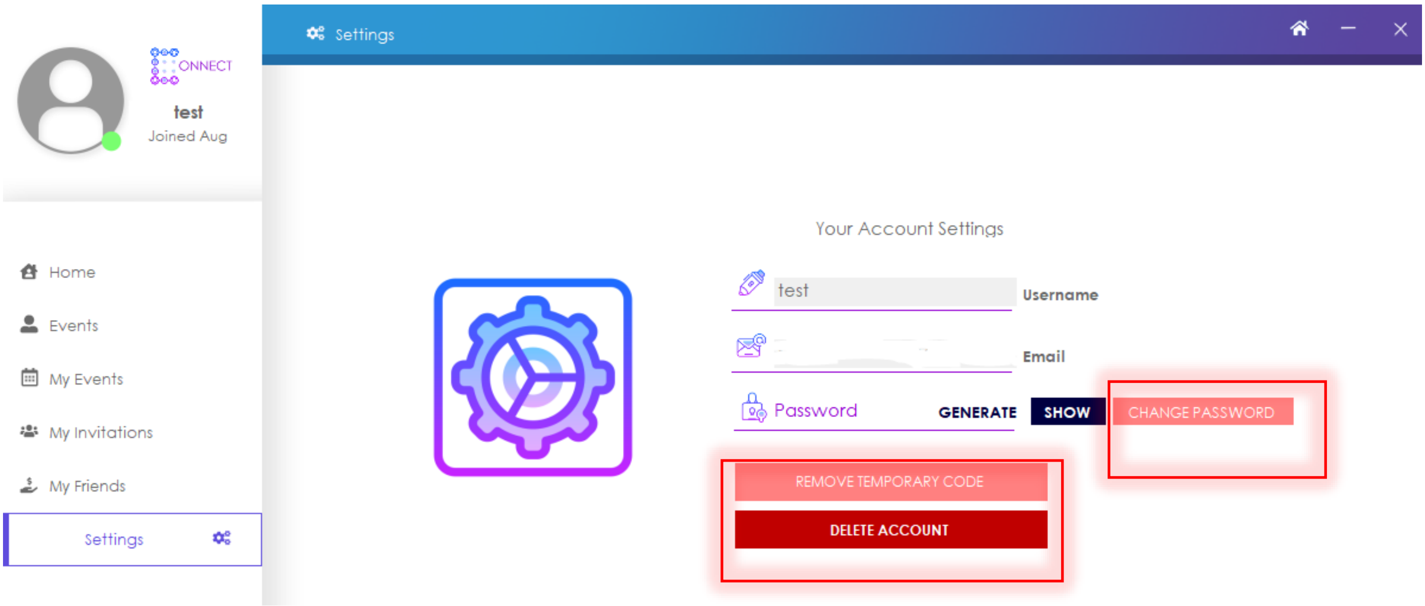
I. Click REMOVE beside friend's name
   a. You must re-request them
   b. Any events of yours they are in will remain



SETTINGS

RESET YOUR PASSWORD

I. Resets password from next log in



Your Account Settings

test Username

Email

SETTINGS

REMOVE TEMPORARY CODE

II. Remove the code sent to your email allowing you to reset password (if any exist)

   a. Next time forgot password, new passcode will be sent

SETTINGS

DELETE ACCOUNT

I. Removes account along with any events invited to and friends

REPORT

TESTING AND EVALUATING

CONNECT

AUTHOR: Jade Harris | **12SDD**

MENTOR: Adam Leserve

CONTENTS

TESTING AND EVALUATING REPORT

TESTING AND EVALUATING EXEUCTION

Testing and evaluating of CONNECT was conducted by combining a variety of effective methods throughout the development process.

During development, testing commenced continuously by constructing and UNIT TESTING each isolated function in a separate program. Through the use of drivers and stubs where necessary, once the envisioned functionality was achieved, the function was implemented into the other functions and INTEGRATION TESTED. Further, I gained results from VOLUME TESTING with 5 simultaneous computers accessing the database. These were evaluated against the QUALITY ASSURANCE and BENCHMARK standards. If runtime or logic errors occurred, I employed breakpoints and debugging output statements to identify the issue. Otherwise, the module was sufficiently tested and evaluated.

Once the functionality of the program was complete and able to be used completely, I utilised BETA TESTING to evaluate the end user's experience with a diverse range of hardware and software, important for the 'general public' target audience of CONNECT. Accompanied by a BETA test survey, this allowed me to evaluate their feedback against the application's requirements, particularly ergonomic and subjective specifications.

TESTING AND EVALUATING REPORT

ANALYSIS OF BETA TESTING RESULTS

Overall, the BETA test results revealed significant interface and reliability issues that occurred due to using a different system configuration, and testing from a fresh perspective. These were resolved immediately.

SECTION 1: FUNCTIONALITY

Overall, the demographic of testers was deliberately diverse with ages ranging from 18-50 and best reflected the 'general' target audience. All responses strongly agreed that CONNECT achieved its purpose successfully.

SECTION 2: INTERFACE

Most responses rated the interface 3, 4 and 5. This rating fulfilled the application's requirement of receiving positive/overall satisfied feedback.

For the response that rated the interface a 3, the main concern was the clarity of the error and success messages. This was overlooked during development, but is extremely impactful in UX, thus it was resolved easily and quickly.

Both the 3 rating and 4 rating noted the poor interface due to the font not transferring over different software configurations, This significantly impacted the information available for the user on the interface. To rectify this, I included the font required for the interface in the software package as well as instructions to install it.

SECTION 3: RELIABILITY

Responses for the level of reliability of the software were between 4 and 5, achieving the programs desired specification of positive/satisfied feedback.

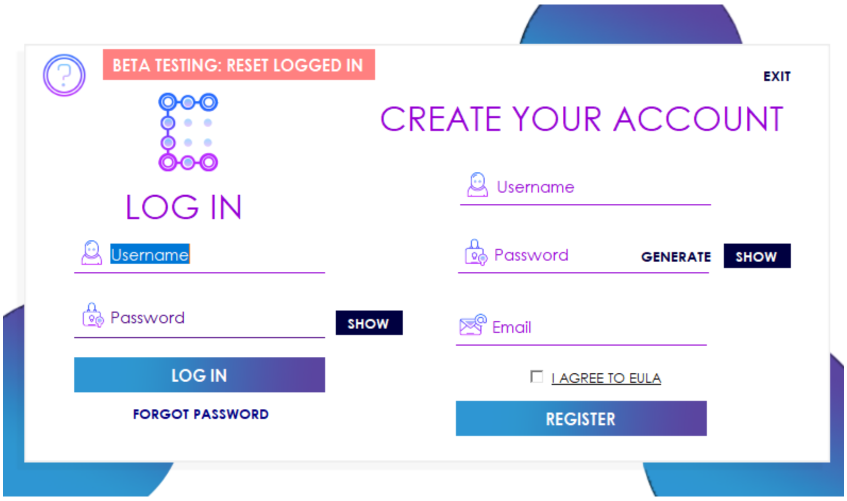
The two ratings of 4 identified bugs where features of the application had not yet been implemented: when both the VIEW button on the frm_invitedEvents and the HOSTED button in frm_myEvents did not cause any action. The other rating of 4 raised the bug of being able to copy-and-paste a longer password into the entry form. This was resolved by disabling copy-and-paste by turning the ShortcutsEnabled property to false.

These intricate responses ensured that the reliability of the database remained intact.

TESTING AND EVALUATING REPORT

TEST DATA TABLES

FRM_LOGIN



This form allows user to both create and log in to program. Use of alphanumeric, and special characters is required for testing. Only valid username, password and email can be entered.

PASSWORD/USERNAME

- 4-9 characters
- Cannot enter ‘ character
- No spaces

Further, EMAIL

- Must contain @ symbol

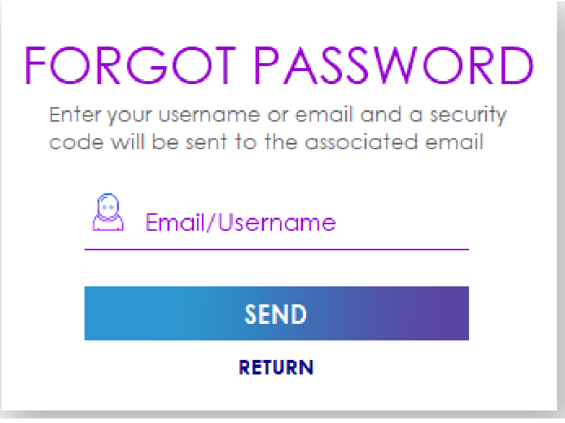
SIGNUP USERNAME (differs from normal username

- 4-9 characters
- Cannot enter ‘ character
- No spaces

INPUT	EXPECTED OUTPUT	OUTPUT
Test user	Testuser	Testuser
Test’user	Testuser	Testuser
Testuserrrrrrr	Testuserr	Testuserr
$Testuser	$Testuser	$Testuser

“” or Username	Message Box Error: Please complete all fields.	Message Box Error: Please complete all fields
T	Message Box Error: Must be at least 4 characters	
Testuser (User doesn’t exist)	Message Box Error: No user exists.	Message Box Error: No user exists.
Testuser (User does exist and password correct)	Open hub form	Open hub form

FRM_FORGOTPASSWORD



FORGOT PASSWORD

Enter your username or email and a security code will be sent to the associated email

This form allows users to enter a remembered email or username (if it exists) to get a temporary reset code. Password can then be reset. Only valid username/email can be entered and searched. NOTE: Email not required to have @ symbol (because it is checked)

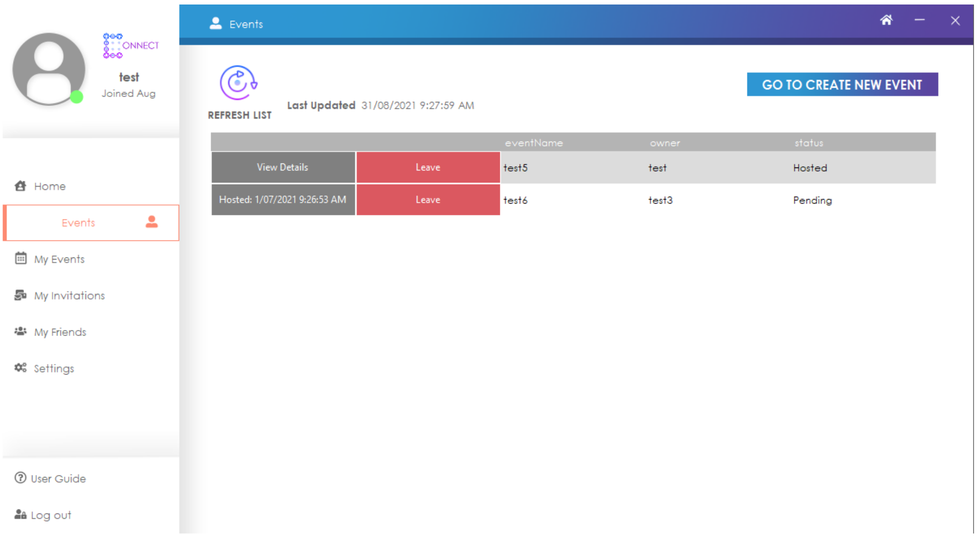
- Max 25 characters
- Cannot enter ‘ character

EMAIL/USERNAME

INPUT	EXPECTED OUTPUT	OUTPUT
Test user	Testuser	Testuser
Test’user	Testuser	Testuser
Testuserrrrrrrrrrrrrrrrrr	Testuserrrrrrrrrrrrrrrrrr	Testuserrrrrrrrrrrrrrrrrr
$Testuser	$Testuser	$Testuser
“” or Email/Username	Message Box Error: Please enter a username or email (cannot be default or empty)	Message Box Error: Please enter a username or email (cannot be default or empty)
T (User doesn’t exist)	Message Box Error: No user exists.	Message Box Error: No user exists.
T (User does exist and error code doesn’t exist)	Message Box: This will send a one-time code to your email Open newPassword form	Message Box: This will send a one-time code to your email Open newPassword form

T (user does exist but error code already exists)	Message Box Error: One time code has already been sent. Open newPassword form	Message Box Error: One time code has already been sent. Open newPassword form

FRM_EVENTS



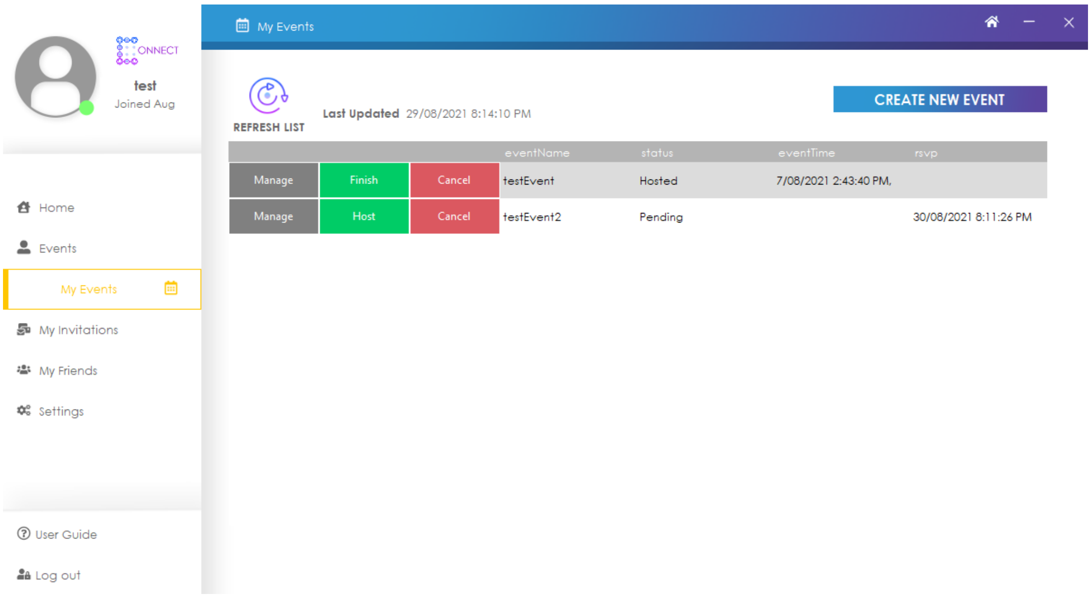
Last Updated 31/08/2021 9:27:59 AM

		eventName	owner	status
View Details	Leave	test5	test	Hosted
Hosted: 1/07/2021 9:26:53 AM	Leave	test6	test3	Pending

The ‘conditional’ buttons embedded in the second column should allow the user to VIEW DETAILS or wait until the event is hosted depending on whether the event is hosted or pending.

INPUT	EXPECTED OUTPUT	OUTPUT
Status: Hosted	Button displays View Details Open viewEvent form	Button displays View Details Open viewEvent form
Status: Pending	Button displays Hosted: {date hosted} Message Box Error: Please wait until the event is hosted by {user}	Button displays Hosted: {date hosted} Message Box Error: Please wait until the event is hosted by {user}

FRM_MYEVENTS



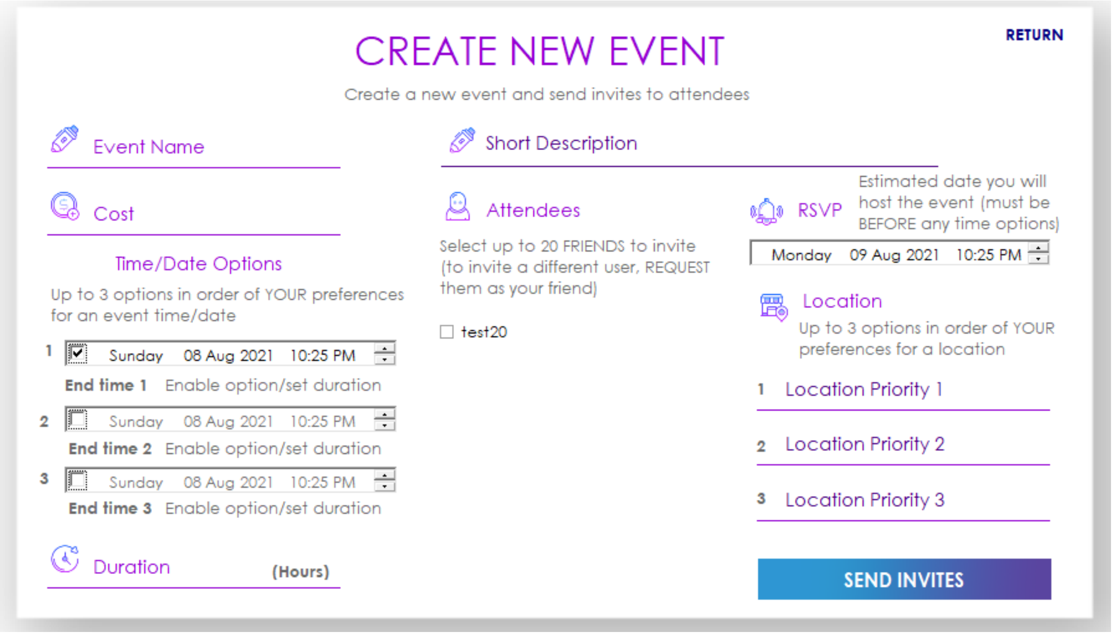
Last Updated 29/08/2021 8:14:10 PM

			eventName	status	eventTime	rsvp
Manage	Finish	Cancel	testEvent	Hosted	7/08/2021 2:43:40 PM,	
Manage	Host	Cancel	testEvent2	Pending		30/08/2021 8:11:26 PM

The ‘conditional’ buttons embedded in the second column should say HOST or FINISH depending on whether the event is hosted or pending.

INPUT	EXPECTED OUTPUT	OUTPUT
Status: Hosted	Button displays Finish Message Box Confirmation: Are you sure you want to finish the event?	Button displays Finish Message Box Confirmation: Are you sure you want to finish the event
Status: Pending	Button displays Host Message Box Confirmation: Are you sure you want to host the event?	Button displays Host Message Box Confirmation: Are you sure you want to host the event?

FRM_NEWEVENT



RETURN

CREATE NEW EVENT

Create a new event and send invites to attendees

Event Name

Short Description

Cost

Attendees

Select up to 20 FRIENDS to invite (to invite a different user, REQUEST them as your friend)

☐ test20

RSVP

Estimated date you will host the event (must be BEFORE any time options)

Monday 09 Aug 2021 10:25 PM

Time/Date Options

Up to 3 options in order of YOUR preferences for an event time/date

1 ☑ Sunday 08 Aug 2021 10:25 PM

End time 1 Enable option/set duration

2 ☐ Sunday 08 Aug 2021 10:25 PM

End time 2 Enable option/set duration

3 ☐ Sunday 08 Aug 2021 10:25 PM

End time 3 Enable option/set duration

Location

Up to 3 options in order of YOUR preferences for a location

1 Location Priority 1

2 Location Priority 2

3 Location Priority 3

Duration (Hours)

SEND INVITES

This form allows users to create a new event. Primarily, data-validating elements have already been used. For the other text inputs:

- Cannot enter ‘ character
- DESCRIPTION: Max 40 characters
- EVENT NAME: Max 20 characters
- LOCATION OPTIONS: Max 10 characters

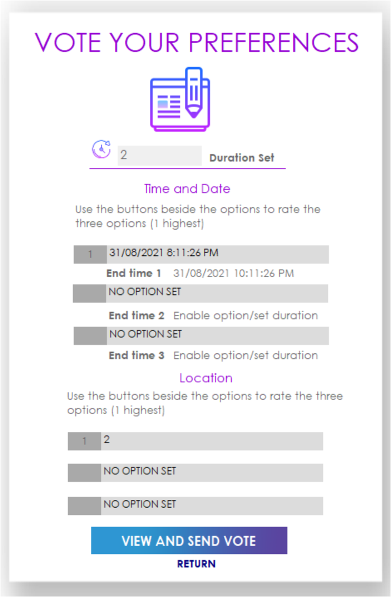
Most prominently, the cost and duration information can only contain numbers and decimals to two places.

COST/DURATION

- COST: Max 20 characters
- DURATION: Max 6 characters
- No spaces
- Numbers only
- Two decimal places

INPUT	EXPECTED OUTPUT	OUTPUT
abcdefg	“”	“”
Abc123	123	123
$123	123	123
1 2 3	123	123
12.ab123	12.12	12.12
12335123123232122222222	12335123123232122222	12335123123232122222
123124124.122222	123124124.12	123124124.12

FRM_INVITATIONVOTES



VOTE YOUR PREFERENCES

2 Duration Set

Time and Date

Use the buttons beside the options to rate the three options (1 highest)

1 31/08/2021 8:11:26 PM

End time 1 31/08/2021 10:11:26 PM

NO OPTION SET

End time 2 Enable option/set duration

NO OPTION SET

End time 3 Enable option/set duration

Location

Use the buttons beside the options to rate the three options (1 highest)

1 2

NO OPTION SET

NO OPTION SET

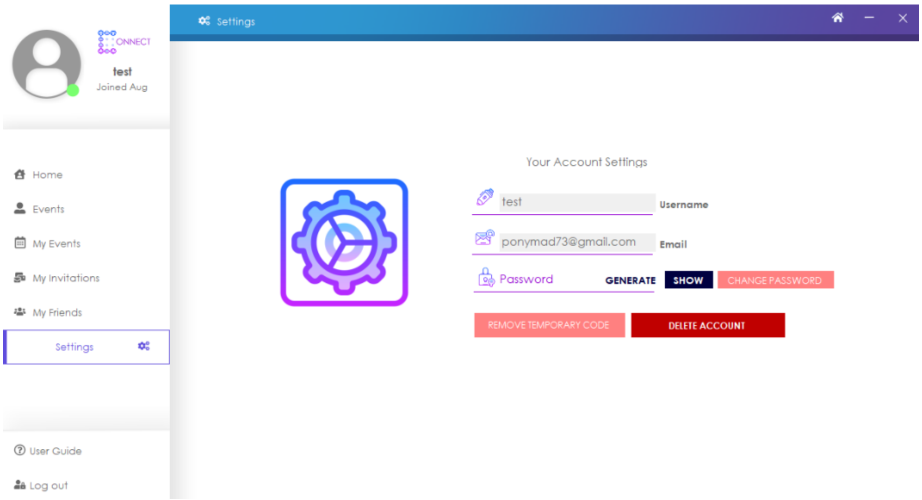
This form allows users to cast their preference on the options.

- No buttons can be the same priority
- Can have toggle options up to number of options

TIME AND DATE OPTIONS

INPUT	EXPECTED OUTPUT	OUTPUT
(3 options) Button 1: 3 Button 2: 3 Button 3: 3	Message Box Error: Cannot have two or more options at the same priority.	Message Box Error: Cannot have two or more options at the same priority.
Button 1: 3 Button 2: 2 Button 3: 3	Message Box Error: Cannot have two or more options at the same priority.	Message Box Error: Cannot have two or more options at the same priority.
Button 1: 3 Button 2: 2 Button 3: 1	Message Box Confirmation: Successfully accepted invitation and sent votes.	Message Box Confirmation: Successfully accepted invitation and sent votes.
(2 options) Button 1: 1 Button 2: 2 Button 3: 3	Impossible (button 3 should be blank/disabled)	Impossible (button 3 is blank/disabled)

FRM_SETTINGS



Your Account Settings

test Username

ponymad73@gmail.com Email

Password

Allow users to adjust their account settings. Users can input a new password (this testing applies to the textbox used in login as well).

- 9-12 characters
- No spaces
- No ' character

INPUT	EXPECTED OUTPUT	OUTPUT
TestPassword	TestPassword Message Box Confirmation: Do you want to reset your password?	TestPassword Message Box Confirmation: Do you want to reset your password?
Abc123	Abc123 Message Box Error: Must be greater than 8 characters	Abc123 Message Box Error: Must be greater than 8 characters
$123 12!12'2	$12312!122 Message Box Confirmation: Do you want to reset your password?	$12312!122 Message Box Confirmation: Do you want to reset your password?
Abcesfsda'	Abcesfsda Message Box Confirmation: Do you want to reset your password?	Message Box Confirmation: Do you want to reset your password?
TestPass11111111	TestPass1111 Message Box Confirmation: Do you want to reset your password?	TestPass1111 Message Box Confirmation: Do you want to reset your password?

REPORT ON FINDINGS

Overall, my program successfully handles majority of the testing cases (especially assuming the user is following the user guide) through the self-validating elements of buttons, date-time pickers, and radio buttons. The main issues were ' character inputs which caused an error with the MySQL string as it is recognised as the end of a parameter. However, these illegal inputs were prevented from input.

One relevant finding, though, is that scaling up would be inefficient with the current use of strings as IDs for events and users. This is because searching (for a distinct username, for example) checks against other strings, which would take an immense amount of time with a lot of records. Due to this search occurring at a database level within the mySQL queries (SELECT * FROM x WHERE condition), changing programming languages and implementing string-matching algorithms would not increase its speed.

Although tedious to alter the current code, this could be overcome by using an autoincrementing ID to identify each record (thus search with numerical values). However, a search of strings would still be required to check if usernames and event names are unique. Thus, mySQL optimisation techniques could be combined with these numerical IDs, such as specifying the exact columns to search rather than *. Alternatively, the database could incorporate in-memory caching using Redis or Memcached, which is compatible with Amazon services and C#. The working principle for these efficient database structures is that if 3000 people searched for the profile each month, the first person would retrieve the information from the database (disk) and the remaining 2999 would retrieve from cache (memory).

BENCHMARK TESTING AND QUALITY ASSURANCE

CRITERIA	ANALYSIS OF END SOFTWARE
Smooth/Quick-Responding Interface Manually test responds to buttons within 1 second	The software meets this requirement aside from an approximately 4 second delay when sending an email to the user (with their forgotten password). This is because the SMTP client must communicate with external services. To mitigate this, the wait cursor and a warning in the confirmation messageBox visually indicates the interface is processing. Another consideration is if the number of users and events increase, search functions may take longer. However, considering the scope of this project, the program accurately fulfills this criteria.
Communicates smoothly with server Manually test server-related interactions complete within 1 second.	As addressed before, within the scope of this project, load testing with numerous events and users was not necessary. Thus, CONNECT successfully communicates with the cloud server within 1 second to meet this specification. However, as more records and users accumulate, this criteria may not be met because there are more records to search through. Instead, it may be more effective to identify events and users by their ID's (as opposed to string of their names).
Organise events in objective matter Software calculates priority and uses randomNumbers to create events.	The end application successfully calculates the priority of time and event options with frm_invitationVote. Further, it incorporates the random number class if numerous options receive the same number of votes, allowing the user to randomise which radio button is selected.
User-friendly Distribute 10+ program prototypes with a survey and receive positive/satisfied feedback. Interface uses consistent buttons and messageboxes.	Due to the pandemic and the security risk of the database connection string, testing with 10+ program prototypes was unachievable. However, from the 5 versions which were distributed, positive feedback was received, suggesting minor adjustments which were immediately incorporated. This is especially achieved through incorporating the user manual.

	The interface successfully uses a consistent colour scheme, the same Flat button style, and the Windows Message Box to fulfill this criteria.
Robust Test each input with a variety of illegal data. Use checkboxes/comboboxes where possible Distribute 10+ program prototypes and ensure no errors occur.	Overall, the end program was sufficiently robust and fulfilled this criteria. The above test data tables illuminate how the application employs message boxes and handling events to prevent the input of most illegal data. Further, SQL attacks are significantly deterred by the inability to copy and paste, and limit of 40 characters in any input box. Throughout, the program successfully uses data-validating checkboxes, combo boxes and radio buttons to enhance robustness. While the program prototype could not be distributed to 10 users, the minor errors identified by the feedback was immediately resolved, and overall, the development process of the program meant that there were minor opportunities to enter dangerous information. However, one reduction of the end product’s robustness is the MySQL connection string which could potentially result in a security breach with malicious intent.
Customisable Distribute 10 program prototypes with a survey and receive positive/satisfied feedback. 4+ different settings.	As mentioned, regarding the user interface requirement, distributing 10+ program prototypes was unachievable. However, from the 5 prototypes that were tested, the feedback was satisfied with the UI irrespective of customisability. Thus, although the program does not incorporate 4+ different settings to achieve this quality assurance criteria, it was found as unnecessary.
Functions on Windows OS devices Test program works on range of 5+ windowsOS systems (with internet).	The end application achieves this specification through successful distribution on 5 different hardware and software configurations running WindowsOS and with internet connection.
Connects users anywhere Information is stored and accessed in a Cloud-server using the Internet. Test program works on range of 5+ systems (with internet).	Through the cloud MySQL database and CRUD manipulation, this feature was successfully achieved. The varying setup configuration and internet of the 5 BETA testers did not affect the performance of CONNECT.