Scheduling Flutter App – Test Cases Document

By Hannah Jeffers, Matthew Latta, Patrick McManus, Michael Connors, and Pedro Gomez

Table of Contents

[Table of Contents 2](#_Toc373970788)

[1). Introduction 2](#_Toc137328837)

[1.1. Purpose of the Document 2](#_Toc280641526)

[2.Test Design 3](#_Toc1400901956)

[2.1. Summary of Test Types 3](#_Toc1512015099)

[2.2. Test Items and Software Risk Issues 3](#_Toc1836313480)

[3). Features to be Tested 3](#_Toc88829348)

[Appendix A – Abbreviations 13](#_Toc320283376)

1). Introduction

This test plan describes the approach to be taken in testing the scheduling flutter app.

1.1. Purpose of the Document

This document identifies testing pertaining to white box, black box, and user acceptance testing.

2.Test Design

2.1. Summary of Test Types

|  |  |
| --- | --- |
| **Test Type** | **Definition** |
| White Box (WB) | Test cases to check if code is running smoothly in a specific function (ex. getting through for loops, if statements, returning correct values, etc). This assumes that the tester can see all of the code and values within the method. |
| Black Box (BB) | Test cases to check if code is running smoothly overall in general (might overlap with UA). This assumes that the tester is a developer, but can only see the output and not any code. |
| User Acceptance (UA) | Test cases to check that the User can do the things they are designed to do (might overlap BB). This is from the user’s intended point of view. |

2.2. Test Items and Software Risk Issues

This test plan applies to white box, black box, and user acceptance tests. It is possible that even if the software is working correctly, we could get negative results during the testing procedures. This could be a result of poor testing and or improperly used too.

# 3). Features to be Tested

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Test #** | **Requirement Key** | **Test** | **Expected Behavior** | **Test Type** |
| 1  User | login | ~ enter valid email and password in respective user input text boxes and hit submit  ~ enter either invalid email and/or password in respective user input text boxes and hit submit | ~ successful login; see main page  ~ error logging in; unable to access main page; prompted to enter a valid email and/or password | BB/UA  BB/UA |
| 2  User | signup | ~ enter valid email, password, first name, last name, and username in respective user input text boxes and hit submit  ~ clicks “Yes” for “Are you a Coordinator?” in the dropdown box  ~ clicks “No” for “Are you a Coordinator?” in the dropdown box  ~ enter either invalid email and/or username in respective user input text boxes and hit submit | ~ successful signup; see main page  ~ becomes a Coordinator object; can see Coordinator pages and has Coordinator permissions  ~ becomes a User object; cannot see Coordinator pages nor use Coordinator permissions  ~ error signing up; unable to see main page; prompted to enter a valid email and/or username or log in with either | BB/UA  BB/UA  BB/UA  BB/UA |
| 3  User | forget\_reset\_ password | ~ click on “Forgot password?” in log in page  ~ enter valid email in respective user input text box and hit submit | ~ can see fr\_password\_page  ~ link to reset password is sent to given email | BB/UA  BB/UA |
| 4  User | Create new User object | ~ take information from user input in the signup page and save them to User object’s variables  ~ is given an invalid user input | ~ new User object created with given variables from user input and added to List users; assign random number as userID  ~ cannot create User object until given valid user input | WB/BB  WB/BB |
| 5  User | Deactivate User | ~ User "deletes” their profile | ~ sets variable isActive to False | WB |
| 6  User | addClub | ~ User adds a new Club object  ~ User adds an already existing Club object | ~ successful addition to List clubs  ~ tells User that given Club is already added; does not add duplicate | WB/BB  WB/BB |
| 7  User | removeClub | ~ User removes given Club object that is in List clubs  ~ User removes given Club object that is not in List clubs | ~ successful removal of given Club object from List clubs  ~ tells User that given Club is not in List clubs | WB/BB  WB/BB |
| 8  User | addPreference | ~ User adds a new preference (string)  ~ User adds an already existing preference (string) | ~ successful addition to List preferences  ~ tells User that given preference is already added; does not add duplicate | WB/BB  WB/BB |
| 9  User | removePreference | ~ User removes given preference (string) that is in List preferences  ~ User removes given preference (string) that is not in List preferences | ~ successful removal of given preference (string) from List preferences  ~ tells User that given preference (string) is not in List preferences | WB/BB  WB/BB |
| 10 User | isEmailWithUser | ~ valid email connected to an existing active User  ~ invalid email not connected to an existing active User  ~ valid email connected to a nonactive User | ~ returns True  ~ returns False  ~ returns False | WB  WB  WB |
| 11 User | isUsernameWithUser | ~ valid username connected to an existing active User  ~ invalid username not connected to an existing active User  ~ valid username connected to a nonactive User | ~ returns True  ~ returns False  ~ returns False | WB  WB  WB |
| 12 Coord-inator | setCoordinatedClub | ~ the club is not already in the coordinatedClubs array  ~ the club is already in the coordinatedClubs array | ~ add the club to the Coordinator’s list of coordinatedClubs and make them an admin in said club.  ~ Tell the user that they are already an admin for this club | WB  BB/UA |
|  |  |  |  |  |
| 13 Coord-inator | setCoordinator | ~ the user is not a coordinator  ~ the user is already a coordinator | ~ the user’s isCoordinator boolean is set to true  ~ the user is told that they are already a coordinator | WB  BB |
| 14 Coord-inator | hostEvent | ~ the coordinator does not have admin permissions for the club  ~ the coordinator is an admin for the club; the event is not being managed by this coordinator and isn’t in the club’s event list  ~ the coordinator is an admin for the club; the event is already being managed by this coordinator but isn’t in the club’s event list  ~ the coordinator is an admin for the club; the event is not being managed by this coordinator, but it already is in the club’s event list  ~ the coordinator is an admin for the club; the event is being managed by this coordinator and is in the club’s event list | ~ Tell user they have invalid permissions to host an event in a club they don’t coordinate  ~ Add the event to the managedEvents list for the coordinator and call the addEvent method for the specific club  ~ Only call the addEvent method for the specific club  ~ Only add the event to the managedEvents list for the coordinator  ~ The event is already being hosted by them, so nothing needs to be updated. Tell that to the user | BB/UA  WB  WB  WB  BB/UA |
| 15 Coord-inator | endEvent | ~ the coordinator does not have admin permissions for the club  ~ the coordinator is an admin for the club; the event is being managed by this coordinator and is in the club’s event list  ~ the coordinator is an admin for the club; the event is already being managed by this coordinator but isn’t in the club’s event list  ~ the coordinator is an admin for the club; the event is not being managed by this coordinator, but it already is in the club’s event list  ~ the coordinator is an admin for the club; the event is not being managed by this coordinator and isn’t in the club’s event list | ~ Tell user they have invalid permissions to end an event in a club they don’t coordinate  ~ Remove the event from the managedEvents list for the coordinator and call the removeEvent method for the specific club  ~ Only remove the event from the managedEvents list for the coordinator  ~ Only call the removeEvent method for the specific club, but let the user know that they are ending an event hosted by another admin in the same club  ~ the event does not exist for this coordinator or club, so nothing needs to happen. Let the user know about this. | BB/UA  WB  WB  WB/UA  BB/UA |
| 16 Coord-inator | demote | ~ the user is a coordinator  ~ the user isn’t a coordinator | ~ set the isCoordinator boolean to be false, demoting themselves back to user  ~ Not possible since it’s a coordinator method, but the isCoordinator boolean would be false anyways so theres nothing to change | BB  WB |
| 17 Coord-inator | getManagedEvents | ~ request the getManagedEvents list | ~ return the list of events managed by this coordinator | BB |
| 18 Coord-inator | getCoordinatedClubs | ~ request what clubs the user coordinates | ~ return the list of clubs that the user is an admin for | BB |
| 19 Coord-inator | relegateAdmin | ~ the club is in the coordinator’s coordinatedClubs list  ~ the club isn’t in the coordinator’s coordinatedClubs list | ~ remove the club from the list and call removeAdmin on it  ~ print error for user that they weren’t an admin in the first place | WB  BB/UA |
| 20  Calen-dar | createCalendar | ~ user gives valid name to calendar. Requests to create one | ~ A new blank calendar with the name they gave now exists in the calendar list | BB |
| 21 Calen-dar | deleteCalendar | ~ user selects an already existing calendar and requests to delete it. | ~ the calendar that they requested is removed from the calendar list | BB |
| 22 Calen-dar | addEventCalendar | ~ user request to add event. They give a valid name, time, and date. | ~ this event now exists on the event list and is viewable | BB |
| 23 Calen-dar | uploadCalendar | ~ user request to upload an already existing calendar. | ~ calendar now exists on the calendar list. Able to see all events that existed on the calendar. The user can do all the actions on this calendar like normal. | BB |
| 24 Calen-dar | shareCalendar | ~ user requests to share a calendar to another user. They give a valid destination and are sending a valid calendar | ~ the calendar arrives at the destination. The user who received it can see all the existing events on the calendar. They cannot make changes to the calendar. | BB |
| 25 Calen-dar | combineCalendar | ~ user requests to combine two calendars into one. Can only combine two at once. The user requests two existing different calendars. Cannot combine calendar with itself. | ~ a new calendar is now added to the calendar. The events from the two calendars are gracefully transferred. User can see all the events on the calendar and also make changes | BB |
| 26 Event | setTime | ~ Valid time, not overlapping  ~ Valid time, overlapping  ~ Invalid time | ~ Change startTime/endTime of selected event  ~ Warn user that times overlap (can change anyway)  ~ Inform user of invalid time. | WB  WB/BB  BB |
| 27 Event | setDate | ~ Valid day/month/year, not overlapping time  ~ Valid day/month/year, overlapping time  ~ Invalid day/month/year | ~ Change dateDay/Month/Year of selected event.  ~ Warn user that times overlap (can change anyway)  ~ Inform user of invalid date. | WB  WB/BB  BB |
| 28 Event | setDesc | ~ Valid desc (no special characters)  ~ invalid name | ~ Change the description of selected event  ~ Inform user, no special characters |  |
| 29 Event | setName | ~ Valid name (no special characters)  ~ Invalid name | ~ Change the name of selected event  ~ Inform user, no special characters | Wb  BB |
| 30 Club | addAdmin | ~ User not an admin but is a member  ~ User is an admin  ~ User not an admin or member | ~ Add specified UserID to admin list  ~ Inform user that specified user already an admin  ~ Warn user that specified user not a member (can add anyway, both member and admin) | WB  BB  WB/BB |
| 31 Club | removeAdmin | ~ User is an admin  ~ User is not an admin  ~ User is not a member | ~ Remove specified UserID from admin list  ~ Inform user that specified user is not an admin  ~ Inform user that specified user is not a member | WB  BB  BB |
| 32 Club | addMember | ~ User not a member  ~ User a member | ~ Add userID to member list  ~ Inform user that specified user already a member | WB  BB |
| 33 Club | removeMember | ~ User in club member list and admin list  ~ User in club member list but not in admin list  ~User not in club member list | ~ Remove specified userID from members list and admin list  ~ Remove specified userID from members list  ~ Inform user that specified user not a member | WB  WB  BB |
| 34 Club | changeDesc | ~ User requests changing club description | ~ Set club description to the new description specified by user | BB |
| 35 Club | addEvent | ~ Event free to add  ~ Event overlaps | ~ Add event to club events list  ~ Warn user that event time overlaps with another club event (can add anyway) | UA  UA/BB |
| 36 Club | removeEvent | ~ Event in events list  ~ Event not in events list | ~ Return nothing, remove event from club events list  ~ Inform user that event ID is not in events list | WB  BB |
| 37 Club | isAdmin | ~ Request true/false statement on if userID is in admin list for club | ~ Return true if userID in member list, return false if userID not in member list | WB |
| 38 Club | isMember | ~ Request true/false statement on if userID is in member list for club | ~ Return true if userID in member list, return false if userID not in member list. | WB |

# Appendix A – Abbreviations

WB – white box

BB – black box

UA – user acceptance

fr – forget\_reset