

TEST PLAN DOCUMENT

for
**SWIMMING POOL
MANAGEMENT
SOFTWARE
SPMS**

Instructors Prof. Sourangshu Bhattacharya

TEAM HEAVY CODER

*Parimi Nishnath (22CS10050)
Chiluveru Pranav Vardhan (22CS10019)
Pratapgiri Sathvik (22CS10053)*

Contents

1 Test Plan Outline

- 1.1 Introduction 4**
- 1.2 Features to be tested 4**
 - 1.2.1 Sign up process.....4
 - 1.2.2 Login process.....4
 - 1.2.3 Notices.....4
 - 1.2.4 Non-Member functionalities.....5
 - 1.2.5 Member functionalities.....5
 - 1.2.6 Manager functionalities.....5

2 Test Plans

- 2.1 Test Plans for Home Page6**
 - 2.1.1 Test plan for Login process.....6
 - 2.1.2 Test plan for Sign up process.....6
 - 2.1.3 Test plan for showing Notices.....7
- 2.2 Test Plans for various users.7**
 - 2.2.1 Non-Member Functionalities7**
 - a) Get Membership.....7
 - b) Book Slots.....8
 - c) Book Pool.....8
 - d) Events.....8
 - e) My Profile.....9
 - f) Log Out.....9

2.2.2 Member Functionalities10

a) Book Slots.....10

b) Book Pool.....10

c) Events.....11

d) My Profile.....12

e) Log Out.....12

2.2.3 Manager Functionalities 13

a) Modify Requirements for Membership.....13

b) Approvals.....14

c) Organize Events.....15

d) Add / Delete Slots.....15

e) My Profile.....16

f) Log Out.....16

2.3 Approach 17

2.3.1 Methodology.....17

2.3.2 Test Types.....17

2.4 Environmental Needs 19

2.5 Software Risk Issues 19

1 Test Plan Outline

1.1 Introduction

The purpose of this test plan is to assess the functionality of the [Pool Management System](#). The main aim of this plan is to guarantee the smooth operation of the application and the accurate provision of information to users, minimizing data inconsistencies and loss of information.

1.2 Features to be tested

1.2.1 Sign up Process

1. Appropriate message for invalid password.
2. Test whether redirection working properly or not.
3. Appropriate message for successful sign up.
4. Appropriate error message for unsuccessful sign up.
5. Redirection to home page on successful sign up.

1.2.2 Login Process

1. Redirection to user dashboard depending on type of user.
2. Appropriate error message if login is unsuccessful.
(Account's existence and incorrect password)

1.2.3 Notices

Updated notices by the manager can be read.

1.2.4 Non-Member functionalities

Test all the changeable features / specific functionalities available in non-member home page.

GetMembership()
BookPool()
BookSlots()
Profile()
Events()
PayMembership()

1.2.5 Member functionalities

Test all the changeable features / specific functionalities available in member home page.

BookPool()
BookSlots()
Profile()
Events()
CancelMembership()

1.2.6 Manager functionalities

Test all the changeable features / specific functionalities available in manager home page.

Approvals()
ManageRequirementsPage()
ManagerProfile()
OrganizeEvents()
ManageSlots()
AddNotices()

2 Test Plans

2.1 Test Plans for Home Page

There are three links in the navigation bar, namely -“Sign up”, “Login” & “Notices”. These links are to be tested because they are the ones which can lead us to the user interface. On clicking “Notices”, we get the updated list of notices by the Manager.

Footer of Home Page contains address link, which will lead us to the location of our pool on Google Maps.

2.1.1 Test plan for Login Process

- 1. First we need to go to Login link present in nav bar of home page.*
- 2. User email and password are to be entered in Login Page. After entering, redirection will lead us to the home page of respective user. For example if we login with manager’s credentials, it will lead us to the manager home page.*

Error Handling :

- > If we try to login using a non-registered account, it will throw up an error showing “**Account Doesn’t Exist**”.*
- > If we try to login with valid username and invalid password, it will throw up an error message showing “**Wrong Password**”.*

2.1.2 Test Plan for Signup Process

- 1. We need to click on Sign Up link in nav bar of home page. It will lead us to the sign up page.*

2. We should enter all the details in the Sign up form.

Error Handling :

While entering the password, we should make sure that our password contains at least one upper case alphabet, a special character and a lower case alphabet. If the entered password is not in this manner, it will throw up a message.

3. After entering all the details we can click on signup, which pops up a message - “**Account created successfully**” and on clicking “**OK**” it will redirect us to the home page.

2.1.3 Test Plan for Notices

Notices page will display the updated list of notices which are added by the manager. This should be updated as soon as manager add notice and save the changes.

2.2 Test Plans for various users

2.2.1 Non-Member Functionalities

a) Get Membership {GetMembership() }

1. We can click on Get Membership button in nav bar and it will lead us to a page containing a details-form.

2. We should fill all the details and click on apply for membership.

Alert Message :

If all the details are not entered, it will throw up an alert message.

3. After getting approved by the manager, payment option will be opened and it is visible in the nav-bar { PayMembership() }.

4. After processing the payment, the non member will be promoted to a member and can access functionalities which were restricted before (such as [Book Slots](#), [Book Pool](#)). Now, he can also see [Cancel Membership](#) option in nav bar.

b) Book Slots {BookSlots() }

c) Book Pool {BookPool() }

Even though these functionalities can be seen in the nav-bar on home page, they cannot be accessed by a non-member. In order to access them, he should be approved as a member.

Alert Message :

On clicking on these links on the dashboard, it will throw up a message box saying “**You need to be a member to Book Slots.**”

d) Events {Events() }

1. If there are any events scheduled, then the user can take participation or can buy tickets for the event. Upcoming events can be seen on the screen.

2. These events are added by the manager.

3. After buying tickets for the event, it shows “**x tickets are booked**” and this action can be reverted by cancelling tickets in My Profile.

e) My Profile {Profile() }

Edit Profile

1. On clicking My profile, we can view our details which we entered during our sign up. Name, email, gender, phone number can be changed and the changed credentials are saved by clicking on “**Save**” button.
2. The applied changes can be seen in User Profile.
3. If we want to change our password, we need to enter the current password, new password and confirm new password.

Error Handling :

Current password should be entered correctly or else it would throw up an error message.

After entering the correct current password, we can enter new password that we are changing to (the rules must be the same as password in signup). If we enter non-matching password in confirm new password, it will throw up an error message.

Your Participations & Your Tickets

1. Participations and Tickets for any event can be seen by the non-member.
2. When tickets are booked, we can see “**x tickets are booked**” in the events page. We can cancel our participation or tickets. If we do so front end **shows nothing** (initially it showed – “**x tickets are booked**”).

f) Log Out

Clicking on log out link will redirect us to the home page and previous pages cannot be reverted.

2.2.2 Member Functionalities

a) Book Slots {BookSlots()}

1. Book Slots link can be selected from the navbar of the dash board. On clicking it, it will take us to Book Slots page.
2. We can select date and check for the available slots. It will show us all the slots that are available. On clicking “Update Slots” we can book the available slot and a message gets popped “Your Slot has been reserved”.
3. If no slots are available on a particular day, it will show nothing.
4. A member can book at most 5 slots per week and if he tries to book 6th slot, it show an alert message.

b) Book Pool {BookPool()}

1. Book Pool link can be selected from the navbar of the dash board. On clicking it, it will take us to Book Pool page.
2. We can select date and check for the available slots. It will show us all the slots that are available. On clicking “Update” we can book the available slot and a message gets popped “Pool in this slot has been reserved”.
3. If no slots are available on a particular day, it will show nothing.
4. A member can book at most 5 times per month and if he tries to book 6th time, it show an alert message.

c) Events {Events() }

1. If there are any events scheduled, then the user can take participation or can buy tickets for the event. Upcoming events can be seen on the screen.
2. These events are added by the manager.
3. After buying tickets for the event, it shows “x tickets are booked” and this action can be reverted by cancelling tickets in My Profile.

d) My Profile {Profile() }

Edit Profile

1. On clicking My profile, we can view our details which we entered during our sign up. Name, email, gender, phone number can be changed and the changed credentials are saved by clicking on “Save” button.
2. The applied changes can be seen in User Profile.
3. If we want to change our password, we need to enter the current password, new password and confirm new password.

Error Handling :

Current password should be entered correctly or else it would throw up an error message.

After entering the correct current password, we can enter new password that we are changing to (the rules must be the same as password in signup). If we enter non-matching password in confirm new password, it will throw up an error message.

Your Participations & Your Tickets

- 1. Participations and Tickets for any event can be seen by the non-member.*
- 2. When tickets are booked, we can see “x tickets are booked” in the events page. We can cancel our participation or tickets. If we do so front end **shows nothing** (initially it showed – “x tickets are booked”).*

f) Log Out

Clicking on log out link will redirect us to the home page and previous pages cannot be reverted.

2.2.3 Manager Functionalities

Manager functionalities be accessed only after logging through manager's credentials. After clicking on Login it will redirect us to manager home page.

a) Modify Requirements for Membership

`{ManageRequirementsPage()}`

1. Modify Membership requirements link can be seen in nav-bar of manager's home page.
2. Manager can change requirements for membership form by clicking on add requirements. To add new requirements, we have to give a name to the requirement and add new-requirement's input details. If we want to add some **file** (**aadhar** or **birth certificate.....**) or **text** (**birthday** or **height** or **weight.....**)
3. After adding requirements, we can still edit those requirements before clicking on update.
4. On clicking "**Update Requirements**" a message is shown, saying "**Membership requirements has been updated**".
5. This updated requirements are updated in the "**Get Membership**" form of a non-member.
6. Text is taken as default requirement.

Error Handling :

1. If the requirement label is left unfilled, it will throw up an alert message – "**fill out this field**".

b) Approvals {Approvals() }

1. Approvals link can be seen in nav-bar of manger's home page.
2. Manager can check the membership documents and details submitted by the non-members from "Get Membership".
3. Based on verification, he can approve or disapprove membership requests. If a non-member is approved as a member, payment option { PayMembership() } gets opened for him.
4. After making payment, the non-member will become a member.
5. After approving this non-member, a messaged is shown.

Alert Messages:

The message box is thrown up saying – "You have approved xxxxx" as a member.

c) Organize Events {OrganizeEvents()}

1. Organize-Events link can be seen in nav-bar of manger's home page.
2. Manager can create a new event, by filling the event-form. He needs to enter details like 'Name of the event', 'Duration', 'Gender', 'Date', 'Distance' and 'Start time'.

Alert Messages :

If any field is left unfilled, it will show a message "Please fill all fields" in the message box when we click on "make changes" button.

3. If all the fields are filled and make changes button is clicked, the form will get reset and the data gets saved. This Notification will be sent to all the members and non members (It can be seen in Notifications link in nav bar).

d) Add / Delete Slots {ManageSlots()}

1. Manage Slots link can be seen in nav-bar of manger's home page.
2. On clicking Manage Slots, we get redirected to that page containing four options – Add slot, Add Slot on a particular day, Delete Slot, Delete Slot on a particular day.
3. Add Slots/Delete Slots deal with weekly slots. That is, manager can add/delete a slot which makes changes in permanent weekly slots.
4. Slot timings need to be selected. After entering all the fields, message pops up saying "slot has updated successfully". On clicking "OK", it will redirect us to the Manage Slots page.
5. Add slot on a particular day/ Delete slot on a particular day deals with special slots. That is the manager can add one special slot on a particular day (date, duration & start time has to be selected).

Alert Messages :

If any field is left unfilled, it will show a message “**Please fill all requirements**” in the message box when we click on “**add slot**”/“**delete slot**” button. We can revert the action by clicking on cancel.

e) My Profile {ManagerProfile() }

1. On clicking My profile, we can view Manager details. Name, email, gender, phone number can be changed and the changed credentials are saved by clicking on “**Save**” button.
2. The applied changes can be seen in User Profile.
3. If we want to change the password, we need to enter the current password, new password and confirm new password.

Error Handling :

Current password should be entered correctly or else it would throw up an error message.

After entering the correct current password, we can enter new password that we are changing to (the rules must be the same as password in signup). If we enter non-matching password in confirm new password, it will throw up an error message.

f) Log Out

Clicking on log out link will redirect us to the home page and previous pages cannot be reverted.

2.3 Approach

2.3.1 Methodology

We approach the project with continual planning, learning, improvement, team collaboration, evolutionary development, and timely completion. Encouraging flexible responses to change. We emphasize on individual and team interactions over processes and tools.

2.3.2 Test Types

Unit Testing

Unit Testing is done at the source or code level for language-specific programming errors such as bad syntax, logic errors, or to test particular functions or code modules. The unit test cases shall be designed to test the validity of the programs correctness. Some of these functions are:

GetMembership()
BookPool()
BookSlots()
Profile()
Events()
PayMembership()
CancelMembership()
Approvals()
ManageRequirementsPage()
ManagerProfile()
OrganizeEvents()
ManageSlots()
AddNotices()

White Box Testing

In white box testing, the user interface is not considered. Instead, the focus is on testing inputs and outputs directly within the code, comparing the results against predetermined specifications. This approach disregards the functionality of the program and concentrates solely on examining the code itself and its structure. Test cases are formulated to ensure that every condition is tested with all possible values and executed at least once during testing.

Black Box Testing / Performance Testing

Black box testing typically involves running through every possible input to verify that it results in the right outputs using the software as an end-user would. We have decided to perform Error guessing and Boundary Value Analysis testing on our application.

System Testing

The software will undergo testing to ensure compatibility with various operating systems such as Linux and Windows.

2.4 Environmental Needs

- 1. There are no special hardware requirements for running the website.*
- 2. Each part of a multipart feature should be tested at least once by creating test cases.*
- 3. There are no specific requirements apart from a working internet connection while testing the software.*
- 4. This software is a web app, every user's activity is handled separately by the server via MySQL. So other users will not be affected when testing the software.*

2.5 Software Risk Issues

We are using the latest versions of npm packages, using older versions might give some trouble.