Software Requirements Specification

for

I'll B There

Table of Contents

1. Use Cases

1.1. Create Account

1.1.1. **Actors**: NoAccountUser

1.1.2. **Flow of Events**:

1.1.2.1. NoAccountUser connects to illbthere.com

1.1.2.2. NoAccountUser clicks link to create an account

1.1.2.3. Server redirects page to account creation form

1.1.2.4. NoAccountUser fills out form

1.1.2.5. NoAccountUser submits form

1.1.2.6. Server displays message that the user has successfully created an account

1.1.2.7. Server redirects user to the activity creation page

1.1.3. **Entry Conditions**:

1.1.3.1. None

1.1.4. **Exit Conditions**:

1.1.4.1. NoAccountUser has successfully created an account

1.2. Sign in

1.2.1. **Actors**: SignedOutUser

1.2.2. **Flow of Events**:

1.2.2.1. SignedOutUser connects to illbthere.com

1.2.2.2. SignedOutUser enters email address

1.2.2.3. SignedOutUser enters password

1.2.2.4. SignedOutUser selects sign in button

1.2.2.5. SignedOutUser is taken to the activity creation page

1.2.3. **Entry Conditions**:

1.2.3.1. SignedOutUser must have an account

1.2.4. **Exit Conditions**:

1.2.4.1. SignedOutUser must have signed in successfully or

1.3. Create Activity

1.3.1. **Actors**: SignedInUser

1.3.2. **Flow of Events**:

1.3.2.1. SignedInUser selects option to create a new activity

1.3.2.2. SignedInUser chooses kind of activity

1.3.2.3. SignedInUser chooses location where activity will take place

1.3.2.4. SignedInUser fills out optional description for their activity

1.3.2.5. SignedInUser selects date/time for their activity
1.3.2.6. SignedInUser confirms that entered activity details are correct
1.3.2.7. SignedInUser clicks button to create activity
1.3.2.8. SignedInUser is redirected to details page for the newly-created activity

1.3.3. **Entry Conditions**:
1.3.3.1. SignedInUser is signed into their account

1.3.4. **Exit Conditions**:
1.3.4.1. SignedInUser has created an activity

1.4. View Activity

1.4.1. **Actors**: SignedInUser

1.4.2. **Flow of Events**:
1.4.2.1. User visits a link to the activity (in-app, or in a separate application)

1.4.3. **Entry Conditions**:
1.4.3.1. SignedInUser is signed into their account

1.4.4. **Exit Conditions**:
1.4.4.1. SignedInUser is viewing the details screen for the activity

1.5. Join Activity

1.5.1. **Actors**: SignedInUser

1.5.2. **Flow of Events**:
1.5.2.1. User clicks button to join the shown activity

1.5.3. **Entry Conditions**:
1.5.3.1. SignedInUser is signed into their account
1.5.3.2. Activity does not require moderation for new participants
1.5.3.3. SignedInUser has found activity that they would like to join

1.5.4. **Exit Conditions**:
1.5.4.1. SignedInUser has joined an activity

1.6. Leave Activity

1.6.1. **Actors**: SignedInUser

1.6.2. **Flow of Events**:
1.6.2.1. User clicks button to leave the shown activity

1.6.3. **Entry Conditions**:
1.6.3.1. SignedInUser is signed into their account.
1.6.3.2. SignedInUser has found activity that they would like to leave

1.6.4. **Exit Conditions**:
1.6.4.1. SignedInUser has left an activity.

1.7. Create Community

1.7.1. **Actors**: CommunityMod

1.7.2. **Flow of Events**:

1.7.2.1. CommunityMod selects option to create a community

1.7.2.2. CommunityMod enters community name

1.7.2.3. CommunityMod enters community description

1.7.2.4. CommunityMod selects if they want their community to have moderation

1.7.2.5. CommunityMod invites prospective members

1.7.2.6. Server saves community to database

1.7.3. **Entry Conditions**:

1.7.3.1. CommunityMod is signed in

1.7.4. **Exit Conditions**:

1.7.4.1. Community has been created

1.7.4.2. Community is available to be joined

1.8. Join Community

1.8.1. **Actors**: SignedInUser, CommunityMod

1.8.2. **Flow of Events**:

1.8.2.1. SignedInUser selects Join option on community

1.8.2.2. If community requires moderation server sends message to CommunityMod for approval.

1.8.2.3. If community requires moderation SignedInUser is shown a message that their request to join the community has been sent to the moderator.

1.8.2.4. If community does not require moderation then SignedInUser is shown the community page with upcoming events

1.8.3. **Entry Conditions**:

1.8.3.1. SignedInUser has found a community that they would like to join

1.8.4. **Exit Conditions**:

1.8.4.1. If community does require moderation CommunityMod has either approved or denied the join request.

1.8.4.2. If community does not require moderation SignedInUser has joined community.

1.9. Update Account Details

1.9.1. **Actors**: SignedInUser

1.9.2. **Flow of Events:**

1.9.2.1. SignedInUser signs into account

1.9.2.2. SignedInUser clicks on update account

1.9.2.3. SignedInUser edits any of their profile fields that they wish to.

1.9.2.4. SignedInUser selects option to "Save Changes"

1.9.2.5. Server commits changes to the database

1.9.3. **Entry Conditions**:

1.9.3.1. SignedInUser is signed in

1.9.4. **Exit Conditions**:

1.9.4.1. SignedInUser's account is updated to reflect the changes that they made.

1.10. Sign Out

1.10.1. **Actors**: SignedInUser

1.10.2. **Flow of Events**:

1.10.2.1. SignedInUser selects option to sign out

1.10.2.2. Server deletes session

1.10.2.3. Server redirects user to signed out page

1.10.3. **Entry Conditions**:

1.10.3.1. SignedInUser is signed in

1.10.4. **Exit Conditions**:

1.10.4.1. SignedInUser is signed out

2. Test Plan

2.1. Manual Testing

2.1.1. **Objective:** The objective of manual testing is to ensure proper functionality when initially developing functions.

2.1.2. **Environment:** Default development environment (Apache, MySQL, PHP 7.0)

2.1.3. **Workload:** Time as necessary depending on complexity of function

2.1.4. **Pass Criteria:** Function output matches expected.

2.1.5. **Failure Criteria:** Function throws compiler error or incorrect output

2.2. Unit Testing

2.2.1. **Objective:** The objective of manual testing is to ensure proper functionality when initially developing functions.

2.2.2. **Environment:** Default development environment (Apache, MySQL, PHP 7.0)

2.2.3. **Workload:** Time as necessary depending on complexity of function

2.2.4. **Pass Criteria:** Function output matches expected.
2.2.5. **Failure Criteria:** Function throws compiler error or incorrect output

2.3. Integration Testing

2.3.1. **Objective:** ensure feature functionality.
2.3.2. **Environment:** default development environment with mock data.
2.3.3. **Workload:** ~15% of feature development time depending on feature.
2.3.4. **Pass Criteria:** Feature outputs expected JSON output.
2.3.5. **Failure Criteria:** Feature outputs incorrect data or HTTP error code response from API.