

Workshop 1

Requirements, User Stories, and Story Mapping

Students:

Juan David Castaneda Cardenas
Laura Vanesa Alvarez Lafont
Nicolas Andres Bolanos Fernandez
Santiago Andres Fetequa Pulgarin
Sergio Nicolas Correa Escobar

Professor:

Carlos Andres Sierra Virguez

Software Engineering II
School of Engineering
Universidad Nacional de Colombia

Functional Requirements

1. An organizer user can create an account in the application through a form (requesting name, username, password, email).
2. The application restricts access to organizers through a login system.
3. The application must allow a user to change their password.
4. An organizer user can view the events they have created and their related information (such as name, date, link, and number of confirmed guests), displayed in a list.
5. An organizer user can view detailed information about the guests confirmed for each event.
6. An organizer user can create a new event by providing basic information (name, location, date and time, capacity) through a form.
7. An organizer user can delete an event from their list, which will also remove all confirmed attendances.
8. An organizer user can copy the link associated with each event to share for attendance confirmation.
9. The application can send warnings to the organizer user if the number of confirmed attendances exceeds the capacity of the event.
10. A guest user can confirm their attendance to a specific event through a form accessed via a link, even if the capacity of the event was exceeded.
11. A guest user can modify their contact information after confirming attendance.

Non-Functional Requirements

1. The application must be a web application that uses a REST API.
2. The application must generate unique links for each event.
3. The application must ensure that each attendee cannot confirm attendance more than once for the same event.
4. The application must include a relational database.
5. The application must differentiate between confirmed guests and guests in the waiting list using colors.
6. The application must send a warning email to the organizer user once the event has reached its capacity.

User stories

As a new organizer user, I want to create a new account so that I can create events and manage attendance.

Acceptance criteria: Given the user is on the login page, When they click on “Create new account” and fill the form, Then the system creates the account in the database and sends the user a confirmation email.

As an organizer user, I want to access my account so that I can check and manage the events that I have created.

Acceptance criteria: Given the user is on the login page, When they type their username and password and click “Continue”, Then the system verifies their credentials and redirects them to the main page.

As an organizer user, I want to reset my password so that I can regain access if I forget it.

Acceptance Criteria: Given the user is on the login page, When they click "Forgot Password" and enter their email, Then the system sends a password reset link to the user's email.

As an organizer user, I want to check the detailed information of confirmed guests so that I can copy their contact information.

Acceptance criteria: Given the user is on the main page, When they click on an event, Then the system redirects them to a new page with the personal information of the confirmed guests.

As an organizer user, I want to create a new event so that I can invite people and manage attendance to that event.

Acceptance criteria: Given the user is on the main page, When they click on the “Create event” button and fill the form, Then the system saves this information in the database and updates the event list.

As an organizer user, I want to delete an event that I have created so that it does not display any more in my main page.

Acceptance criteria: Given the user is on the main page, When they click on the “Delete event” button, Then the system deletes the event and the information of their guests after confirming the action with the user.

As an organizer user, I want to copy the link associated with any event so that I can share it.

Acceptance criteria: Given the user is on the main page, When they click on the “Share event” button, Then the system copies the associated link to the clipboard.

As a guest user, I want to confirm my attendance to an event that I have been invited to via link so that I appear on the confirmed guests list.

Acceptance criteria: Given the user has clicked on the link, When they fill the form and click on “Confirm attendance” button, Then the system saves the information in the database and sends a confirmation email to the guest.

As a guest user, I want to modify my personal information so that the organizer has my correct contact information.

Acceptance criteria: Given the user has clicked on the link, When they type their personal id and click on “Confirm attendance” button, Then the system retrieves the guest information and allows the user to change it updating this information in the database.

User Story Mapping

The full user story mapping chart is presented in the next page.

References

A Guide to User Story Mapping: Templates and Examples (How to Map User Stories) | Planio.
(s. f.). Planio. <https://plan.io/blog/user-story-mapping>

Organizer account management	Event management	Guest interaction
<div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Log in (11) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Create a new account (10) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Change password (8) </div>	<div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> View created events (8.5) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> View guests information by event (8.5) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Create new events (6) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Delete a existing event (4) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Copy link (7) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Send warning (2.5) </div>	<div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Confirm attendance (6.5) </div> <div style="background-color: #f2e0b7; padding: 5px; border-radius: 5px;"> Modify contact information (3) </div>
<div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design a user module in the database (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement ORM (3) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the login interface (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the login interface (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to process the form fields (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to verify credentials (3) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the insertion logic of a new account to the database (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to send the confirmation email (3) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the email template (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the sign up form (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the sign up button (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the sign up form (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the connection between the button and the form (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to process the form fields (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to send the security email (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the email template (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the "Forgot password" button (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the "Forgot password" button (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the change password form (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the change password form (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the reading logic of events in the database filtered by user (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the connection between the email code and the form (3) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the main page (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the main page (3) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the event module in the database (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement ORM (3) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the reading logic of attendance in the database filtered by event (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the insertion logic of a new event to the database (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the creation form (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the creation button (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement logic to delete an event from the database (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the connection between the button and the form (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to process the form fields (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to send the email (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the email template (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Add a "Delete" button for each event at main page (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Add a "Share" button for each event at main page (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement logic to generate unique links for each event (5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design a confirmation pop up (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the confirmation pop up (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to process the form fields (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to send the email (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the email templates (0.5) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to check existing attendances (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the insertion logic for the attendance into the database (1) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement logic for waiting list (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Implement the logic to send the email (2) </div> <div style="background-color: #d9ead3; padding: 5px; border-radius: 5px;"> Design the email template (1) </div>		