

User Story Template

User Story

Write user story text description.

Note

User stories should be written according to the following pattern:

As a `<type of user>`, I want `<some goal>` so that `<some reason>`.

Important: a user story should consider needs of particular type (most often role) of user.

User stories should follow INVEST principles:

- Independent. Reduced dependencies = easier to plan;
- Negotiable. Details added via collaboration;
- Valuable. Provides value to the customer;
- Estimable. Too big or too vague = not estimable;
- Small. Can be done in less than a week by the team;
- Testable. Good acceptance criteria.

Note: large (those which won't fit into a single sprint) user stories, also called epics, should be eventually split into smaller user stories (if it's possible at the moment).

Each user story will go through the following lifecycle before implementation begins:

1. Writing a user story (anyone is allowed to write a user story, most often it'll be Product Owner);
2. Discussion (all teams should take part in the final discussion):
 1. [Optional] Breaking an epic into smaller user stories;
 2. Splitting into sub-tasks (with tech specifics);
 3. Verification of acceptance criteria completeness;
 4. Update user story contents if needed;
3. Mark as ready. Goes to the implementation pipeline when it's time.

Activity

Specify the related activity.

[Optional] provide a short description of the activity if necessary.

Note

Activities (also known as themes) are common groups (usually features) that combine user stories together. As an example, "Account" activity may contain user stories describing registration, password recovery, profile updates, etc. Granularity of activities should be chosen in a way that doesn't lead to activities with too many user stories (in such cases these activities should be split into smaller ones).

When combined with user stories, activities present a good visualization of the backlog, called user story map.

Type

Epic or User story. Epics should only be used as a temporary stub for user story(ies). See definition of an epic above.

Status

Status of user story readiness in terms of the lifecycle. A user story should only be marked as ready when all sections were filled in and reviewed by team members.

An example:

- In icebox (new);
- In backlog (discussion in progress);
- Ready to go (discussed and verified);
- In progress, in testing, etc.

Components

Provide a list of affected components (e.g. back-end, front-end, etc.).

Sketches/wireframes/mockups

All graphical data goes here (with some captions if necessary).

An example:

Sketch 1
Image not found or type unknown

Workflow

Describe the whole workflow from user's perspective, step by step, with all necessary details.

All stakeholders should be able to understand contents of this section, so no sophisticated technical details should be placed here.

An example:

1. User opens web app link.
2. User enters credentials.
3. User logs in and sees the dashboard.

Acceptance Criteria

Describe details that are required to validate the software once it's implemented: UI elements behavior, validation logic, messages of all types (validation errors, hints, etc.).

An example:

- Security
 - No other web pages except Sign In should be available for non-authenticated users.
 - When a non-authenticated user opens a link leading to one of internal pages and then logs in, the app should redirect the user from Sign In to the page corresponding to the original link
- UI elements
 - "Login" input field
 - Required field
 - Maximum 255 characters

Tech Notes

Describe all low level technical details here, like API specification, error handling conventions, timeout values, etc. May also contain some diagrams, (pseudo) code snippets, etc.

Links

Provide links to related epics/user stories, mockups, API specification, other high-level design documents, etc.

Revision #2

Created 18 July 2022 23:18:40 by Bryce Holloway

Updated 9 February 2023 20:20:50 by Bryce Holloway