

User Stories

For your project, you will be using user stories and acceptance criteria for planning your work. There are two main messages a user story should convey:

1. What functionality needs to be developed and who it is for. (who and what)
2. To help readers understand why this important (why)

Developing User Stories

A user story is an explanation of a feature from the perspective of the user using your application. It is a request for a particular feature on your application from the perspective of a user.

The format of the user story outlines the type of user who wants this feature, what they want, and their purpose for wanting it:

"**As a** (INSERT ROLE HERE), **I want to** (INSERT ACTION HERE), **so that I can** (INSERT REASON HERE)".

For example: As a logged in user, I want to be able to post a comment to a photo, so that I can express my thoughts on the photo.

User stories **DO NOT** need to only be worked on by a single developer -- however each story should have a "story owner" whose job it is to breakdown the work sub-tasks for other developers to pick up.

Developing Acceptance Criteria

Acceptance criteria defines the boundaries of the user story. We use acceptance criteria to determine what needs to be true, in order for this story to be considered done. This can help prevent scope creep as you are agreeing before the work begins what is included and what is not.

To come up with a good set of acceptance criteria for a user story, you need to think about what should be included in the user story. One way to define an acceptance criteria is creating a scenario in the format of **Given-When-Then**.

For example: **Given** that I'm a logged in user and I'm on the homepage and **When** I view a photo, **Then** the system will show me a list of comments for that photo and a comment form at the top of the list of comments, and **When** I try to click the "Comment" button or press Enter without filling out the text form, **Then** nothing will happen. **When** I fill out the text form and click the "Comment" button or press Enter, **Then** the system will save my comment and will show my comment with my profile picture at the bottom of the list of comments.

You should come up with an idea about how the user story could be implemented and discuss if that's the right path to take with your team. Before agreeing on a set of acceptance criteria, each team member should come up with a list of questions that will narrow down the scenarios that the acceptance criteria will need to handle. Then, you will discuss as a group the answers to those questions and develop the acceptance criteria together.

Keep in mind that your acceptance criteria should be covering only what is needed for a MVP, minimum viable product.

Breakdown of the Life of a User Story

Create a wiki page on your group's GitHub repository called "User Stories" to hold your user stories, questions/answers, and acceptance criteria for your Project Advisor to review. See example here: [Example User Stories](#)

1. Come up with a user story
2. Take 5 minutes and think about what questions you have about the story -- think about both the "happy path" (i.e nothing goes wrong) as well as the edge cases
3. Go around the group and have each team member list their questions and keep a running list -- group similar questions together -- note, do not answer questions during the portion, you're just gathering all the outstanding questions into one place
4. Discuss / debate / agree on the answers to the questions
5. Use the questions / answers to develop the acceptance criteria for the story
6. Determine who the story owner will be
7. Story owner breaks down the story into sub-tasks for other developers to take
8. During project week, story owner makes sure that acceptance criteria is met for the story in the final product

Example Authorization User Stories

Note: these stories assumes there is a User model / database available, with an email, username and password field -- generally, model creation would be included as part of a story but since the user is such a building block, it may make sense to pair on that and get it done ASAP.

User Story 1: Login

As an **unauthorized user**, I want to **be able to login to the website via a form**, so that I can **access my private information**.

Questions for Developing Acceptance Criteria

- Will the user enter a username or an email address to login?
 - User will login via email and password
- What routes should we use for login?
 - User will login via `/login` route
- Where should the user be redirected after login?
 - User will be redirected to the `/` homepage
- Will we allow OAuth authentication via a third party?
 - Not yet -- maybe in a future story
- What happens if the user doesn't exist yet?
 - Display the message `Invalid Login :(please try again.`
- What happens if the user enters the wrong password?
 - Display the message `Invalid Login :(please try again.`
- Should this story include allowing a user to reset their password?
 - Not yet -- maybe in a future story
- Should logging in use session-based or use token-based authentication?
 - We will use token-based auth for now

Acceptance criteria

1. **Given** that I'm a logged-out user and
 - **When** I'm on the `/login` route
 - **Then** there will be a login form with an email and password field and a "Login" button to submit the form.
2. **When** I try to fill out the form with an invalid email and password combination and press Enter or press the "Login" button
 - **Then** at the top of the form, I will see a red message `Invalid Login :(please try again.`
3. **When** I try to fill out the form with an email that doesn't exist in the system and press Enter or press the "Login" button
 - **Then** at the top of the form, I will see a red message `Invalid Login :(please try again.`
4. **When** I try to fill out the form with a valid email and password and press press Enter or press the "Login" button
 - **Then** I will be redirected to the homepage at the `/` route.
5. **Given** that I am a logged-in user
 - **When** I refresh the homepage at the `/` route
 - **Then** I will still be logged in
6. **Given** that I am a logged-out user
 - **When** I try to navigate to the homepage at the `/` route
 - **Then** I will be redirected to the `/login` route

User Story 2: Sign Up

As an **unauthorized user**, I want to **be able to sign up for the website via a signup form**, so that I can **access this awesome website**.

Questions for Developing Acceptance Criteria

- How long should the user session last?
- What information are we collecting from the user?
- What does the UX like? What does the form look like? How simple is it?
- Where should the user be redirected after signup?
- What happens if the user has signed up previously? (what makes a user unique?)
- What are the password requirements?
- What happens if the user's password isn't strong enough?
- Are we allowing sign up via OAuth?
- Do we need a confirmation email?
- What routes should we use for sign up?