

SAMPLE USER STORIES AND ACCEPTANCE CRITERIA
CS 3311 / LMC 3432: JUNIOR DESIGN
SPRING 2021

Here is the template for your user stories and acceptance criteria:

Story: [title]

As a [role]

I want [feature or goal]

So that [benefit or reason]

Acceptance Criteria: (presented as Scenarios)

Scenario 1: [description of the circumstance]

Given [context]

And [some more context]...

When [event]

Then [outcome]

And [another outcome]...

Scenario 2: ...

Some examples of **GOOD** user stories and acceptance criteria:

Story: Account Holder withdraws cash

As an Account Holder

I want to withdraw cash from an ATM

So that I can get money when the bank is closed

Scenario 1: Account has sufficient funds

Given the account balance is \ \$100

And the card is valid

And the machine contains enough money

When the Account Holder requests \ \$20

Then the ATM should dispense \ \$20

And the account balance should be \ \$80

And the card should be returned

Scenario 2: Account has insufficient funds

Given the account balance is \ \$10

And the card is valid

And the machine contains enough money

When the Account Holder requests \ \$20

Then the ATM should not dispense any money

And the ATM should say there are insufficient funds

And the account balance should be \ \$10

And the card should be returned

← Note that there can be multiple pieces of context in the acceptance criteria for scenarios.

← Note that there can also be multiple outcomes for a given scenario.

← Note that the outcomes describe what happens in the system as opposed to the experiences of the user. Students will often times write stories in which the outcome will describe user feelings or actions (e.g.

Scenario 3: Card has been disabled
Given the card is disabled
When the Account Holder requests \ \$20
Then the ATM should retain the card
And the ATM should say the card has been retained

“And the account holder will have \$20”). You can’t control or dictate your users’ emotions or actions; you can only program how your product will behave in the event of a specific event.

To put it another way, acceptance criteria that describe user feelings and actions could never be marked “done” because you would never be able to confirm that the human response you have described actually happens.

Story: User makes playlists

As a user
I want to create/edit playlists
So that I can catalog tracks that I wish to frequently access

Scenario 1: A user finds a playlist to add a track to
Given the playlist that the user wants to modify exists/has been created
And the desired track has been located
And the track is available
When the user selects "Add to Playlist"
Then a menu of the user's existing playlists appears

Scenario 2: A user adds a track to an existing playlist
Given the menu of the user's existing playlists has appeared
And the user has selected a track to be added to a playlist
When the user selects the playlist to which to add the track
Then the desired track is added to the selected playlist
And the user is returned to the original screen/page
And a banner message appears on the bottom of the browser
And the banner message disappears after three seconds

← Acceptance criteria can attempt to describe and detail all the front end phenomena associated with an event.

Some examples of **BAD** user stories and acceptance criteria:

Story:

As a manager
I want to have the items in my possession be found
so I can free up space in the lost-and-found area.

Scenario: A building manager wants to verify that the items in their possession is properly reflected on the solution.

Given the building manager has an account and knows what items are in the building's possession to be recovered by its respective founder;
When the building manager logs into the solution with their account,
Then they are capable of

- editing posts regarding the items in their possessions and
- updating their status of whether its found as well as
- communicating with any individual that lays claim to the item.

← This acceptance criteria can best be described as “too big.” Rather than detailing how the system will react to each of these possibilities, the author has just started to list features.

Think of it this way: how would you ever know these tasks to be “done”? (I don't think you could know given how they are currently described.) What, for example, would have to happen for you to be able to declare an update feature done and coded?

Story:

As a finder
I want to post about an item I've found on campus so that the owner can be reunited with the item and easily find its owner.

Scenario: A guest on campus finds an item and doesn't know what to do with the possession.

Given the guest finds an item and can't hand it off to a student to handle the issue or doesn't trust someone else to,
When the guest finds the item and unsure where to hand it off to,
Then they ask an administrator that knows about the system or knows of the solution, consults it to find the designated location to handle lost items, and either directs the guest to drop it off there or

← This acceptance criteria describes a set of human actions external to the app (i.e., something that someone is doing in real life) rather than outcomes that are programmable within the system/app.

they assume responsibility over delivering the lost item to the appropriate location.

You could never program these and thus declare this task/story “done.”

Story:

As a user, I want to have multiple profiles on one account with varying access to settings, so different profiles can/cannot access certain content

Scenario: A user creates a profile categorized as “main”

Given a new profile is categorized as main
When the user continues to access the application under that profile
Then they are granted the access associated with a main user

← This acceptance criteria is essentially circular – which is to say that it does little more than rehash the scenario. This acceptance criteria could be valid if the scenario or acceptance criteria was far more specific (i.e., it described the front end UI changes associated with profile creation or back end management).