**USER STORIES**

* a key component of agile software development.
* short, simple descriptions of a feature or functionality from the perspective of a user
* to capture requirements in an agile project and help the development team understand the needs and expectations of the users.

**CHARACTERISTICS**

1.User-centric

2.Simple

3.Independent

4.Negotiable

5.Testable

6.Iterative

7.Role-based

8.Traceable

**PATTERN**

* **Role-Feature-Benefit pattern**
* *As a [ type of user], I want [ an action], so that [ some reason]*
* "As a [persona]": Who are we building this for? We’re not just after a job title, we’re after the persona of the person. Max. Our team should have a shared understanding of who Max is. We’ve hopefully interviewed plenty of Max’s. We understand how that person works, how they think and what they feel. We have empathy for Max.
* “Wants to”: Here we’re describing their intent — not the features they use. What is it they’re trying to achieve? This statement should be implementation free — if you’re describing any part of the UI and not what the user goal is you're missing the point.
* “So that”: how does their immediate desire to do something this fit into their bigger picture? What’s the overall benefit they’re trying to achieve? What is the big problem that needs solving?
* EX: As Max, I want to invite my friends, so we can enjoy this service together.
* EX: As a manager, I want to be able to understand my colleagues progress, so I can better report our success and failures.

**WRITING**

->**INVEST** Principle of User story

->INVEST principle which expresses the quality of the user story because in base

a good software product is completely dependent upon a good user story.

1.**I**ndependent – Not dependent on other.

2.**N**egotiable –Includes the important avoid contract.

3.**V**aluable –Provide value to customer.

4.**E**stimable –It should be estimated.

5.**S**mall –It should be simple and small not complex.

6.**T**estable –It should be evaluated by pre-written acceptance criteria.

->**3 Cs** in User Stories

1)Card –Write stories on cards, prioritize, estimate and schedule it accordingly.

2)Conversation –Conduct conversations, Specify the requirements and bring clarity.

3)Confirmation –Meet the acceptance criteria of the software.

**USER STORY EX FOR WEBSITE:**

User story examples for ***login***

1.As a user, I want an option to stay logged in, so that I don’t have to enter my credentials every time.

2.As a user, I want to be able to reset my password if I forget it, so that I can regain access to my account.

3.As a user, I want to see an error message if I enter incorrect login details,

so that I know when my login attempt has failed.

User story examples for ***registration***

1.As a user, I want to log in via my social media accounts, so that I can quickly access the platform

without creating a new account.

2.As a new user, I want to choose my own username and password during registration,

so that I can personalize my login credentials.

3.As a new user, I want to provide my basic information during registration,

such as name and date of birth, so that I can personalize my profile.

**USER STORIES WITH ACCEPTANCE CRITERIA EXAMPLES**

Acceptance criteria and user stories are critical for successful project delivery for any software development project. They are essential for documenting expectations from a product or solution and how the test team can evaluate it for acceptance. Let’s look at the definition of acceptance criteria, their connection to user stories, and their role in project success.

* As a *restaurant customer*, I want to reserve a table online, so I can ensure I have a place to eat at my preferred time.

Acceptance criteria: The system should show available times for reservations. After reservation, the customer should receive a confirmation.

* As *a music app user*, I want to create playlists, so I can organize my favourite songs. Acceptance criteria: The app should provide the option to create a new playlist and add songs to it. Users should be able to name their playlists and view them later.

A list of product with text

Description automatically generated with medium confidence

**USER STORIES WITH REQUIREMENTS EXAMPLES**

Requirements typically refer to the documented expectations and specifications to meet a particular product need. They are a concise description of the feature or functionality that the software must provide.

* As a banking customer, I want to transfer funds between my accounts, so I can manage my money effectively.

Requirements: The system should allow customers to select two accounts, enter an amount, and execute a transfer. The system should update the account balances immediately.

* As a job seeker, I want to save job listings I’m interested in, so I can apply for them later.

Requirements: The job platform should allow users to save job listings to a personal list, which they can view later.

A screenshot of a white and black list

Description automatically generated