**Iteration 1(week 1)**

**Story Narrative Template**

**Story Description**

As an Admin , I want to log in so that I can start using the system.

**Business Context**

The administrator is going to handle all the operations and give rights to other users to Zombie dash.

**Entitlements**

Profile should be secure.

**Assumptions**

Change password option provided. The password strength should be more than 6 char in which one should be number, one in upper case , one in lower case and one special char.

**Out of Scope**

Cannot change username .Remember me option not provided and the forgot password also not provided.

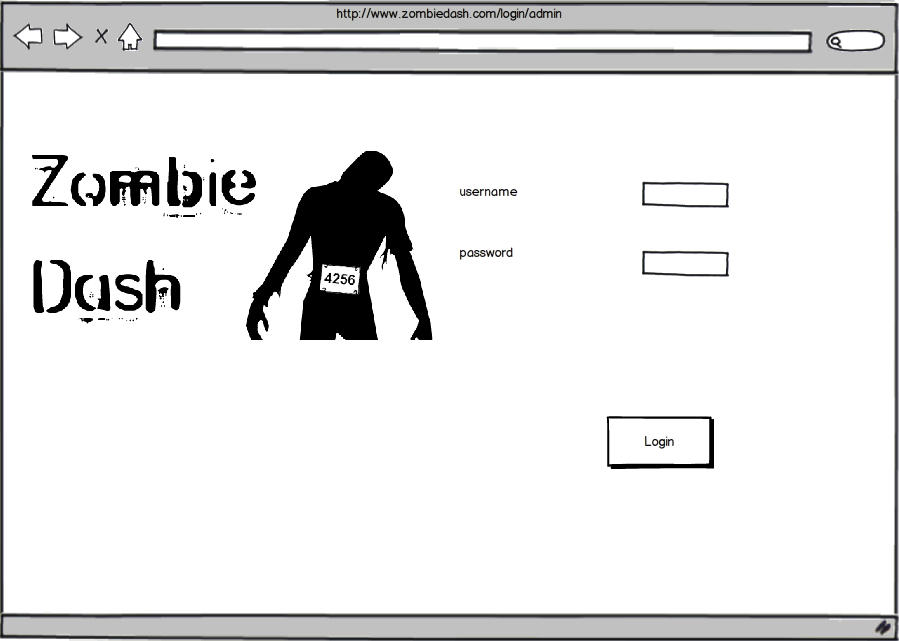
**Dependencies**

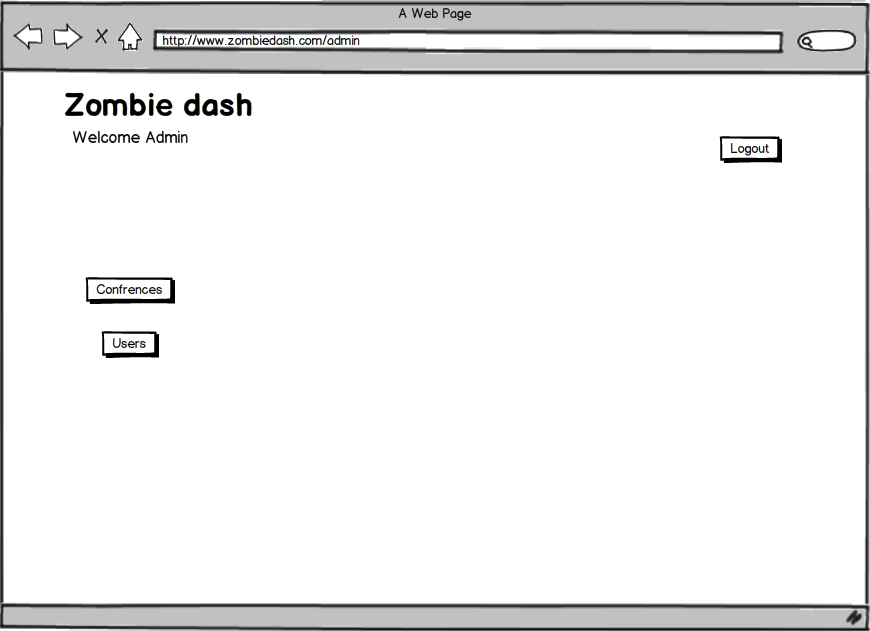
Create admin user.

**Acceptance Criteria**

Change password can come later.

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * That the admin user is on login page | User enters correct username and password and click on login | The system should login to home page of the admin and show the homepage to the user |
| * That the admin user is on the login page | User enters correct username but incorrect password and click on login | The system should show message stating incorrect username or password and the user should not be able to login. |
| * That the admin user is on the login page | User enters incorrect username but enters correct password and click on login | The system should show message stating incorrect username or password and the user should not be able to login. |
| * That the admin user is on the login page | User enters incorrect username and password and click on login | The system should show message stating incorrect username or password and the user should not be able to login. |

**UI**  




**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As an admin, I want to Create and read a conference in the system, So that I can start to configure a conference.

**Business Context**

The conference page has to be created so that info can be read and made available to the game designer. The game designer can be added to these pages.

**Entitlements**

Include special security requirements here.

**Assumptions**

Admin logged in and has all these options on his home page. Save button is provided.

**Out of Scope**

No autosave option provided on the create page.

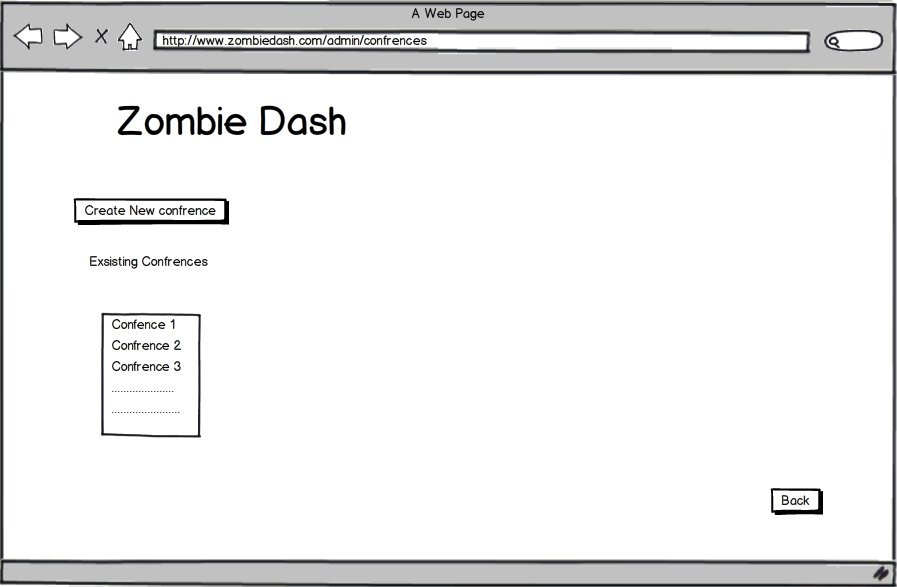
**Dependencies**

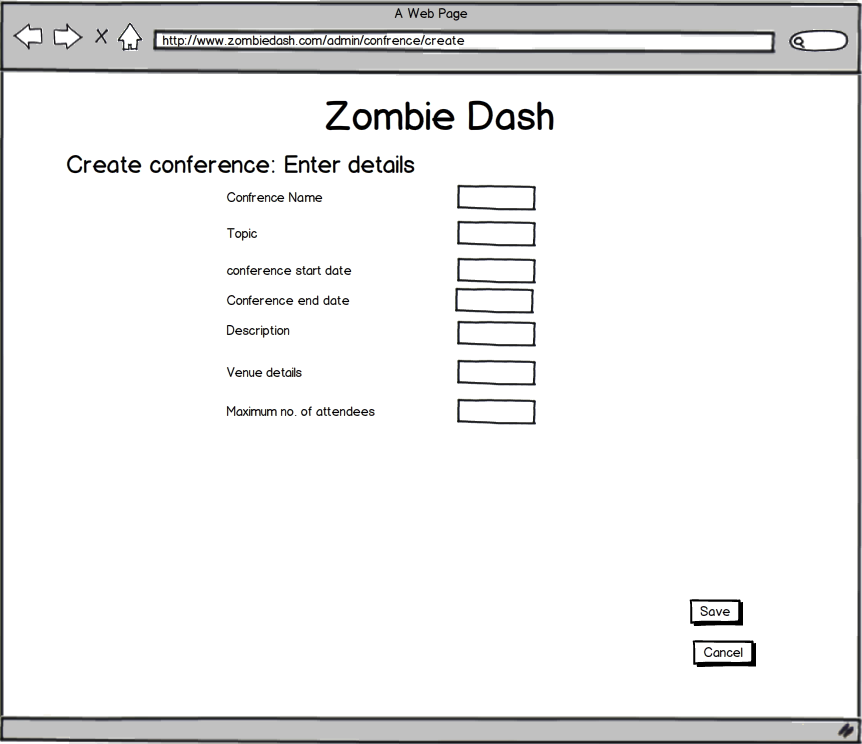
Create admin and Admin login.

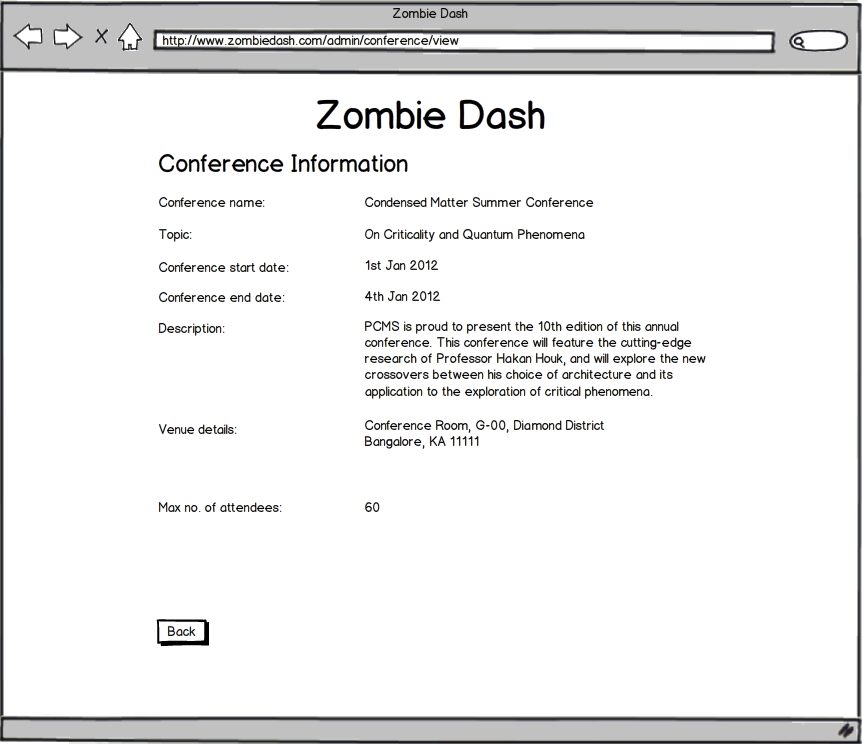
**Acceptance Criteria**

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * That the admin is on the homepage | Admin clicks on conference | The system should show the conference page with Create new conference option and list of existing conferences with their names. |
| * That the admin user is on the conference page | The admin clicks on create new conference | System should show create page where user can enter information about the conference. |
| * Admin is on create conference page | User enters all the information and clicks on save | System should save all the conference information and redirect to the conference page with the newly saved conference name in the list of existing conferences. |
| * Admin is on create conference page | User clicks on cancel at any point of time | The system should show a message box showing message “are you sure you want to cancel” with Cancel and Ok options. |
| * That the admin is on the create conference page and has clicked cancel and is being shown the message box with yes and no option | Admin clicks on Ok | System should redirect to conference page and changes should not persist. |
| * That the admin is on the create conference page and has clicked cancel and is being shown the message box with yes and no option | Admin clicks on Cancel | System should be on the same page with the text entered for the conference information not gone. |
| * Admin is on create conference page | The admin misses some field in the information and clicks on save | The system should show message “\* Marked fields compulsory” and \* marked in front of the missing fields. |
| * Admin is on create conference page | Admin enters some text other than number in the no. of attendees field and moves focus from this field | The system should show error message “Must me a non-negative integer” right next to the no. of attendee’s field. |
| * Admin is on conference page | Admin clicks on a conference name in the existing conference list | The system should show information about that conference in a new page(conference info page). |
| * Admin is on conference info page | Admin clicks on back | The system should redirect the user to conference page. |
| * Admin is on conference page | Admin clicks on back | The system redirects to homepage of admin. |
| * Admin is on create conference page | The admin enters date which is not in proper format , i.e. YYYY-MM-dd and clicks on save. | The system should show a message in front of the date field(start date and end date)”Enter date in YYYY-MM-dd format” |
| * Admin is on create conference page | The admin enters month more than 12 in the date field(Start date and end date) and clicks on save. | The system should show appropriate message in front of that field. |
| * Admin is on create conference page | The admin enters date from the past, i.e. date less than system date in date fields and clicks on save. | The system should show a message “Enter a date from the present or future” in front of the date field. |
| * Admin is on create conference page | The admin enters start date > end date and clicks on save. | The system should give message “must be equal to or greater than start date” in front of end date field. |
| * Admin is on create conference page | The admin tries to enters characters more than allowable limit | The system should not allow entering more than the specified no. of characters and showing message “exceeded max no. of characters” in front of the field. |
| * Admin is on create conference page | The admin clicks on the bottom right corner of some of the fields and drags it | The size of the text boxes should increase. |

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype



****

****

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

1.As an attendee I want to play the game selected so that I am involved in it and can win

2. As a customer playing the trivia game, I want to submit my answers and receive my scores, so that I can know my progress.

**Business Context**

The game will keep the players engrossed and interested in the conference. This will increase the customer base for zombie. The whole concept behind zombie dash is the playing of game.

**Entitlements**

Include special security requirements here.

**Assumptions**

Game/s should be selected and available to play. Attendee is logged in.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

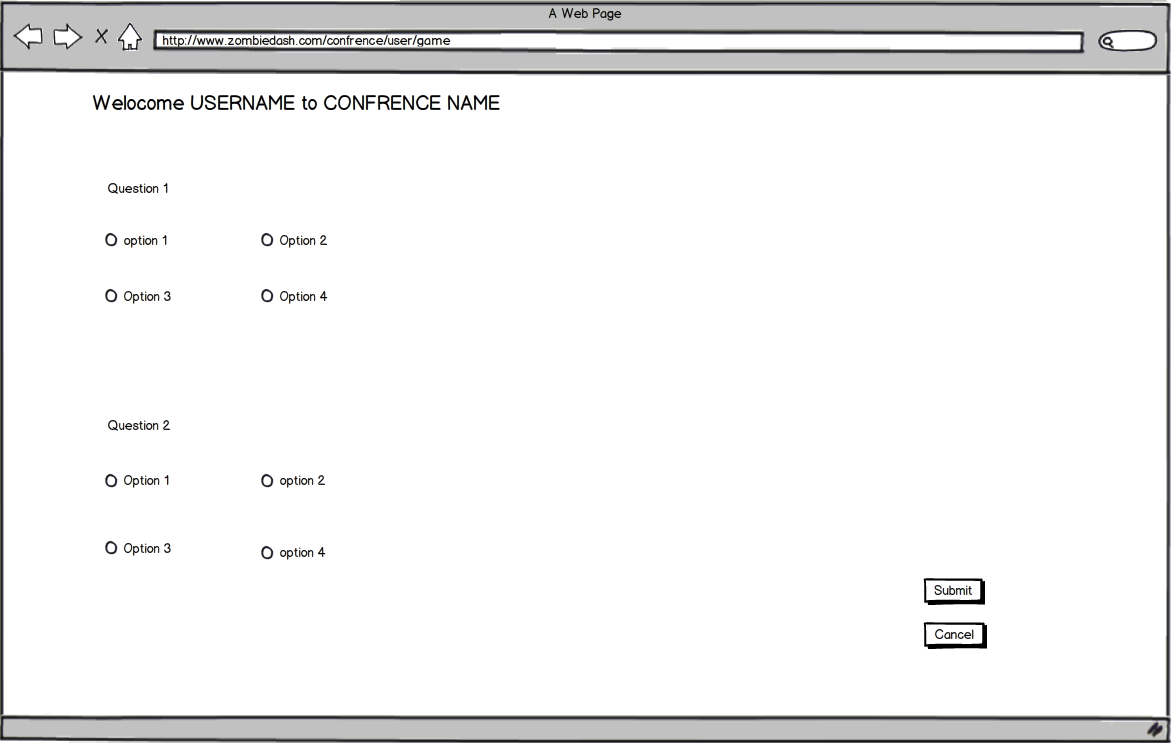
**Dependencies**

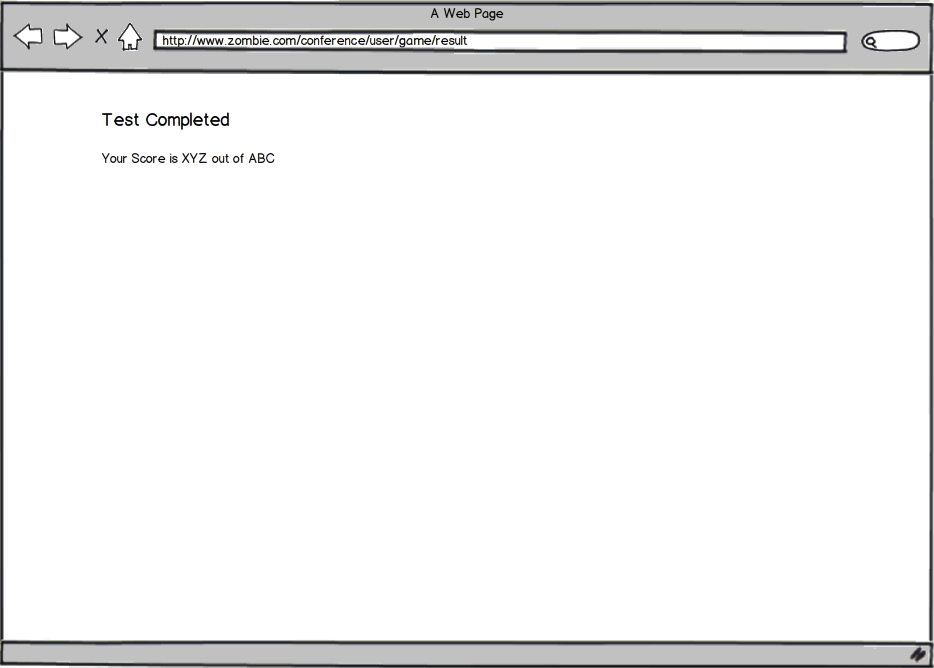
Selection of a Game, Choosing mode of gameplay, Attendee Registration, Attendee Login, View Homepage, Game designer activity.

**Acceptance Criteria**

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * That the customer/attendee is logged in | s/he selects a new trivia game/task from the list of tasks | The system should open new game page where all the answer options are deselected |
| * That the attendee is on the game page | The attendee tries to answer a question by selecting an option | An option should get selected for that question |
| * That the attendee is on the game page | The attendee tries to select another option for an answered question | The new option should get selected and the old option deselected |
| * That the attendee is on the game page | The attendee clicks cancel or back | The system should show message box with message “are you sure you want to cancel” with yes or no option. |
| * The attendee is on the game page and has clicked cancel or back and is being shown the message box with yes and no option | The attendee clicks on yes | System should redirect to attendee home page and changes should not persist. |
| * The attendee is on the game page and has clicked cancel or back and is being shown the message box with yes and no option | The attendee clicks no | System should be on the same page (game page) with the options selected for questions not gone. |
| * That the attendee is on the game page | The attendee selects options for all the questions and clicks on submit | The system should calculate score accordingly and show it in the results page (test completed). |
| * That the attendee is on the game page | The attendee does not select options for some or all the questions and clicks on submit | The system should show a popup message “Answer all questions” with ok option. |
| * That the attendee is on the game page, has left out to mark some or all questions and clicked on submit and is being shown a popup message “Answer all questions” with ok option | The attendee clicks on ok | The system should remove the popup message and all the options that are marked by the attendee should be there. |
| * That the attendee is on the game page with some options selected | The attendee clicks on refresh or F5 | The system should load the current page again with the options marked gone |

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype



****

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As an admin , I want to CRUD a game designer in the system, So that I can start to configure the profile of game designer.

**Business Context**

Game designers are the users who are going to design the games for different conferences. The have to be created in the system and then added to each conference so that conference details are available to them so that they can design a game.

**Entitlements**

Include special security requirements here.

**Assumptions**

Admin logged in and has all these options on his home page.No autosave option.  
Username and password for the game designer is created by the admin. The game designer will have the ability to change the password after login for the first time.

**Out of Scope**

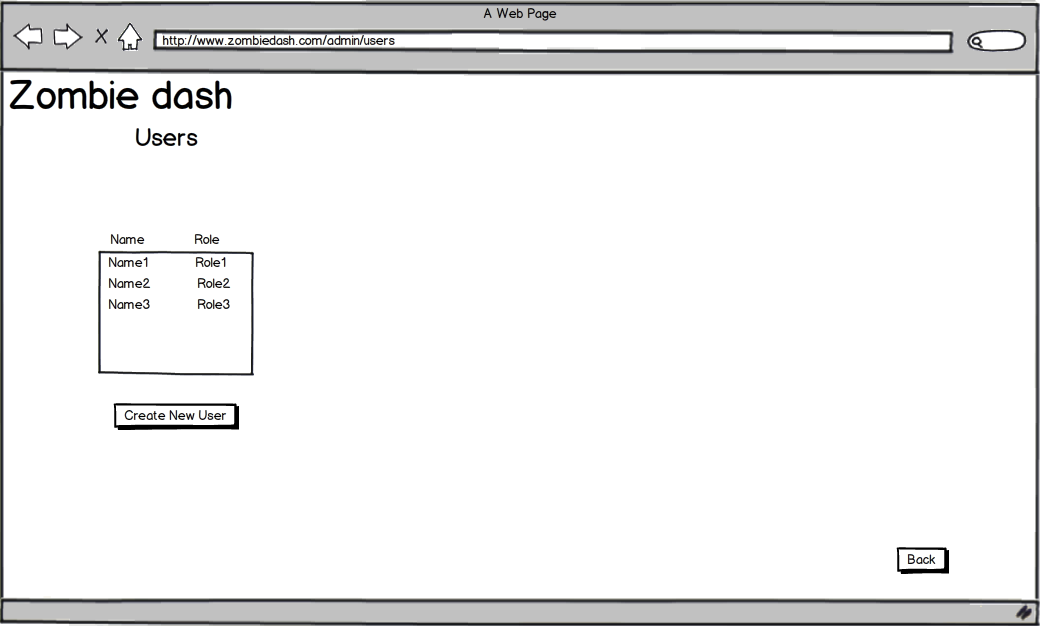
No autosave option.

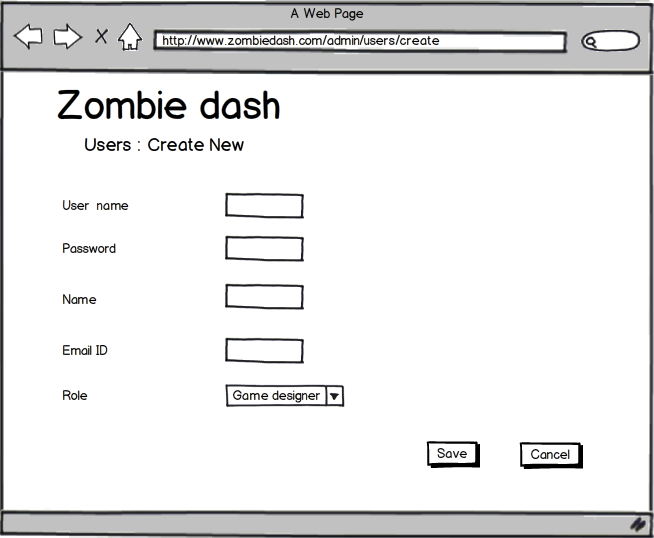
**Dependencies**

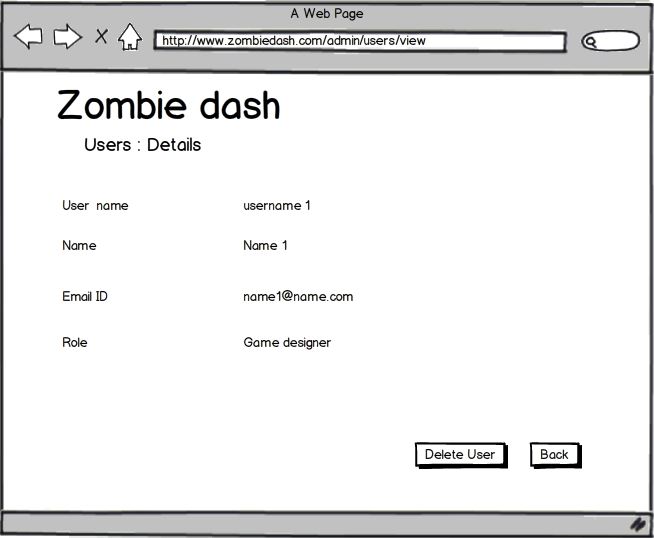
Create admin user, Admin logs in

**Acceptance Criteria**

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * That the admin is on the home page | The admin clicks on Users | Then the system should direct to the users page with list of existing users and option to create a new user. |
| * That the admin is on the users page | The admin clicks on create new user | Then the system should direct to the create page where new users can be created. |
| * That the admin is on the users page | The admin clicks on refresh or F5 | The system should reload the same page , i.e. users |
| * That the admin is on the users page | The admin clicks on back | The system should redirect to the homepage of admin |
| * That the admin is on the users page | The admin clicks on a name from the list of existing users | The system should direct to view page which has details of the user whose name is clicked by the admin |
| * That the admin is on the create page | The admin fills all the details correctly i.e. no conflicting username , password , name ,Email and selects a role and clicks on save | The system should redirect to the users page with the name and role of newly user created present in the existing users list. |
| * That the admin is on the create page | The admin misses out some detail such as name or role or username or password and clicks on save | The system should show a message ” All fields compulsory ” |
| * That the admin is on the create page | The admin enters something in password field | Then the admin should be able to see what s/he is entering, i.e. the password field should not be using some special characters to display password. |
| * That the admin is on the create page | The admin enters some conflicting username or a username that is less than 5 characters in the username field and clicks on save | Then the system should show a message “conflicting username , enter something else ” |
| * That the admin is on the create page | The admin enters some username that is less than 5 characters in the username field and clicks on save | Then the system should show a message “invalid username , should be more than 5 characters ”. |
| * That the admin is on the create page | The admin enters details in the fields and enters a password which is less than 6 characters or/and is not alpha numeric , and clicks on save | The system should give a message “Password should be alpha numeric and more than six characters”. |
| * That the admin is on the create page with some data entered in different fields | The admin clicks on cancel | The system should show a popup message “Are you sure you want to cancel , all you data will be lost” with OK and Cancel option. |
| * That the admin is on the create page with some data entered in different fields, has clicked on cancel and is being shown the popup message “Are you sure you want to cancel , all you data will be lost” with yes and no option | The admin clicks on OK | The system should redirect to the users page with all the data entered by the admin on the create page being lost. |
| * That the admin is on the create page with some data entered in different fields, has clicked on cancel and is being shown the popup message “Are you sure you want to cancel , all you data will be lost” with yes and no option | The admin clicks on Cancel | The system should remove the popup message and be on the same page(create) with the data entered in different fields being there. |
| * That the admin is on the view page | The admin clicks on refresh or F5 | The system should reload the same page |
| * That the admin is on the view page | The admin clicks on Back | The system should redirect to the users page |
| * That the admin is on the view page | The admin clicks on delete | The system should show a popup message “Are you sure you want to delete this user” with OK and Cancel options. |
| * That the admin is on the view page, has clicked delete and is being shown a popup message “Are you sure you want to delete this user” with yes or no option | The admin clicks on OK | The system should delete that user from database and should redirect to the user page the name of the deleted user gone from the list of existing users |
| * That the admin is on the view page, has clicked delete and is being shown a popup message “Are you sure you want to delete this user” with yes or no option | The admin clicks on Cancel | The system should remove the popup ,message and should be on the same page i.e. view page |

**UI**  


****

****

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As an admin , I want to manually log out from my profile , So that my account is secure.

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Logout should be successful and complete so that admin profile is secure.

**Assumptions**

Logout button present on the homepage of the admin and only on the homepage.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

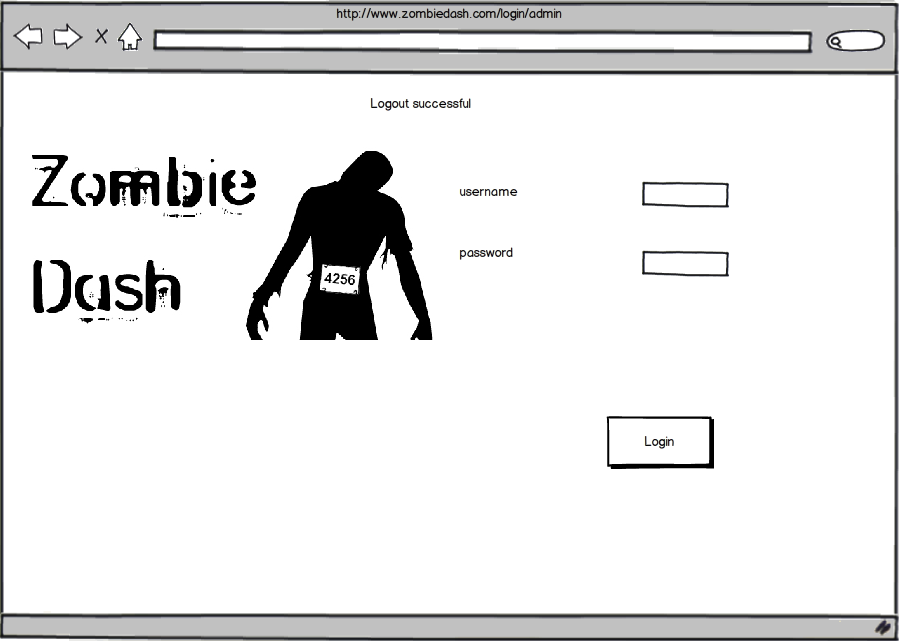
Create admin user, Admin logs in.

**Acceptance Criteria**

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * That the admin is on the homepage | The admin clicks on logout | The system should logout from the admin profile and get directed to login and show message ”Successful logout”. |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype



**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Iteration 2(week 2)**

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

*Caveat: This is a template of what one might typically capture in a story narrative. Every domain is different and every team has different needs. Take what is useful and remember the principle of simplicity – “the art of maximizing the work not done”*

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |

**Story Narrative Template**

**Story Description**

As a [type of user], I want to [goal], so that [reason].

A good story description should always answer Who, What, and Why

**Business Context**

Explain where and how the story fits in the big picture.

**Entitlements**

Include special security requirements here.

**Assumptions**

Preconditions, or things assumed to be true in order for this story to fire successfully.

**Out of Scope**

Call out anything that is explicitly not included in the story that might be inferred by the team.

**Dependencies**

List other story cards that need to be played first

**Data Definitions**

Data definitions for the business entities introduced by the story. This can be a simple link to a Data Dictionary

**Acceptance Criteria**

A list of the acceptance criteria the BA/PO will verify during card signoff. Developers and testers use acceptance criteria as the starting point for creating automated tests.

Acceptance Criteria are conventionally written using the Given, When, Then (GWT) template. Think of each Given, When, Then as a micro use case.

Business rules and validations are typically documented in the Then clause, often using one of the five rule words: ‘must’ (or ‘should’), ‘not’, ‘no’, ‘only if’ (see [RulesSpeak](http://www.rulespeak.com/en/)).

|  |  |  |
| --- | --- | --- |
| **Given** | **When** | **Then** |
| * Precondition * Precondition * etc | Event/Trigger | All the following (post conditions) are true:   * System **must/should**… * Do **only if**… * **No**… * Some entity **may not**…   Note: Here post conditions are AND’d together |
| * Precondition * Precondition * Etc | Event/Trigger | At least one of the following (post conditions) is true:   * System **must/should**… * Do **only if**… * **No**… * Some entity may **not**…   Note: Here post conditions are OR’d together |

*Tip: Try to keep Acceptance Criteria UI agnostic; that way Acceptance Criteria won’t break as the UI evolves. Focus on What, not How. Avoid phrases like “Click the button”.*

**UI**  
Usually a low-fi prototype or hyperlink to low-fi prototype

**Story Questions / Issues** A history of the issues raised and resolved during story narrative development. Sometimes helpful as an audit trail of the evolution of the story requirements

|  |  |
| --- | --- |
| Question / Issue | Response |
| Question for SME or PO | Answer from SME or PO |