Use Case Specification: Create Product Backlog

# Brief Description

This use case allows the Product Owner to create product backlog.

# Actors

Product Owner

# Preconditions

Product owner is logged into the system.

# Flow of Events

# *Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The product owner creates product backlog | The system creates the product backlog. |

# Post-Conditions

Product Backlog is created in the system.

# Business Rules

# Nonfunctional Requirements

Use Case Specification: Add/Display/Update/Delete User Stories in/from Product Backlog

# Brief Description

This use case allows the Product Owner to add/display/update/delete user stories in/from the Product Backlog.

# 2. Actors

Product Owner

# 3. Preconditions

Product Owner is logged into the system.

# 4. Flow of Events

## *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The Product Owner adds/displays/updates/deletes the user stories in/from product backlog | The system updates the product backlog. |

# 5. Post-Conditions

User story is added/displayed/updated/deleted to/from the product backlog.

# 6. Business Rules

# 7. Nonfunctional Requirements

**Use Case Specification**: Create Release Backlog

# Brief Description

This use case allows the Product Owner to create Release Backlog.

# Actors

Product Owner

# Preconditions

* Product Owner is logged into the system.
* The Product Backlog should exist.

# Flow of Events

## *Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The product owner creates release backlog | The system saves the release backlog. |

# Post-Conditions

Release Backlog is created in the system.

# Business Rules

# Nonfunctional RequirementsUse Case Specification: Add/Display/Update/Delete User Stories in/from Release Backlog

# 1. Brief Description

This use case allows the product owner to add/display/update/delete user stories in/from the release backlog.

# 2. Actors

Product Owner

# 3. Preconditions

Product owner is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The product owner adds/displays/updates/deletes the user stories in/from release backlog. | The system updates the release backlog. |

# 5. Post-Conditions

User story is updated in the release backlog.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Respond to the Request to Delete User Story from Release Backlog

# 1. Brief Description

This use case allows the Product Owner to give approval to delete user story from the Release Backlog.

# 2. Actors

Product Owner

# 3. Preconditions

Product owner is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The product owner approves/disapproves the request. | The system notifies of the request response. |

# 5. Post-Conditions

Scrum Master can take action according to the response of the Product Owner.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Assign Scrum Master to Release Backlog

# 1. Brief Description

This use case allows the Product Owner to assign Scrum Master to Release Backlog.

# 2. Actors

Product Owner

# 3. Preconditions

* Product owner is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The product owner selects one employee and assign him/her as scrum master. | The system updates the employee role. |

# 5. Post-Conditions

The assigned employee gets all the authorization of Scrum Master.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Create Sprint

# 1. Brief Description

This use case allows the Scrum Master to create sprint.

# 2. Actors

Scrum Master

# 3. Preconditions

* Scrum Master is logged into the system.
* Release backlog is already created.
* Scrum Master must be assigned to the release backlog.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| Scrum Master creates sprint for the release backlog. | The system creates the sprint. |

# 5. Post-Conditions

New sprint record is created.

# 6. Business Rules

- Sprint name should be unique.

- No sprint should overlap in the same release.

# 7. Nonfunctional Requirements

Use Case Specification: Add/Update User Stories in Sprint

# 1. Brief Description

This use case allows the Scrum Master to add user stories in sprint.

# 2. Actors

Scrum Master

# 3. Preconditions

Scrum Master is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The scrum master adds/updates the user stories in sprint | The system updates the sprint record |

# 5. Post-Conditions

User story is added/updated to the sprint.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Delete User Stories from Sprint

# 1. Brief Description

This use case allows the scrum master to add user stories in sprint.

# 2. Actors

Scrum master

# 3. Preconditions

Scrum master is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The Scrum Master deletes the user stories in release backlog | The system keeps the action in pending for product owner’s response. |

# *4.2 Alternate Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The product owner deletes the user stories in release backlog | The system updates the release backlog. |

# 5. Post-Conditions

Delete request is generated and sent to the Product Owner.

# 6. Business Rules

If actor is a Scrum Master, then Product Owner’s approval is required.

# 7. Nonfunctional Requirements

Use Case Specification: Assign User Stories to Developer and Tester

# 1. Brief Description

This use case allows the Scrum Master to assign the user stories to developer and tester.

# 2. Actors

Scrum Master

# 3. Preconditions

Scrum Master is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The Scrum Master assigns the user stories to developers and testers | The system updates the developer/tester for the user stories. |

# 5. Post-Conditions

Developers and testers are assigned the user stories.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Estimate Development Effort

# 1. Brief Description

This use case allows the developer to estimate time in hours or days.

# 2. Actors

Developer

# 3. Preconditions

Developer is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| Developer estimates the development time for the user stories. | The time is updated in user stories. |

# 5. Post-Conditions

Estimated time is updated in user stories.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Update Development Effort

# 1. Brief Description

This use case allows the developer to update the remaining effort.

# 2. Actors

Developer

# 3. Preconditions

Developer is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The developer updates development time for the user stories. | The system updates the development time of user stories. |

# 5. Post-Conditions

Estimated time is updated.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: CRUD on Employee Record

# 1. Brief Description

This use case allows HR System Admin to create/read/update/delete the employee record.

# 2. Actors

HR System Admin

# 3. Preconditions

HR System Admin is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| HR System Admin create/read/update/delete the employee record. | The system updates the employee record. |

# 5. Post-Conditions

The employee record is added/ updated/ deleted to/from the system.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Assign Employee Roles

# 1. Brief Description

This use case allows HR System Admin to assign employee specific roles.

# 2. Actors

HR System Admin

# 3. Preconditions

HR System Admin is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| HR System Admin assign roles to employees. | The system updates the employee information to the assigned roles. |

# 5. Post-Conditions

Employee is assigned a specific role.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Update Employee Vacation Time

# 1. Brief Description

# This use case allows HR System Admin to update vacation time of employee.

# 2. Actors

HR System Admin

# 3. Preconditions

# HR System Admin is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| HR System Admin selects the employee by name or id. | The system shows the employee details. |
| HR System Admin updates the vacation time of the employee. | The employee information is updated. |

# 5. Post-Conditions

# The vacation time of employee is updated.

# 6. Business Rules

HR System Admin deducts the applied vacation days from the balance days.

# 7. Nonfunctional Requirements

Use Case Specification: Update Employee Benefits

# 1. Brief Description

# This use case allows HR System Admin to update benefits for employee.

# 2. Actors

HR System Admin

# 3. Preconditions

# HR System Admin is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| HR System Admin selects the employee by name or id. | The system shows the employee details. |
| HR System Admin updates the benefits for the employee. | The employee information is updated. |

# 5. Post-Conditions

# The benefits of employee are updated.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Login

# 1. Brief Description

# This use case describes how an Employee logs into the System.

# 2. Actors

# Employee

# 3. Preconditions

# None

# 4. Flow of Events

## *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee opens up a browser and enter the website | The system displays the login page. |
| HR The Employee enters his/her username and password | The system validates the entered username and password and logs the actor into the system |

***4.2 Exceptional Flow***

|  |  |
| --- | --- |
| User Action | System Response |
| The librarian enters invalid username and/or password | The system displays a message that the member ID cannot be found. |

# 5. Post-Conditions

# The actor is now logged into the system.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Create IMR

1. Brief Description

This use case allows the Employee to add user stories to IMR.

1. Actors

Employee

1. Preconditions

Employee is logged into the system

1. Flow of Events

***4.1 Basic Flow***

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The Employee navigates to the IMR dashboard | The system displays the IMR dashboard page |
| The Employee creates user stories and saves to IMR | The system saves the user stories. |

1. Post-Conditions

User stories are saved into database.

1. Business Rules
2. Nonfunctional Requirements

Use Case Specification: Update Profile

1. Brief Description

# This use case allows the Employee to update his/her profile

1. Actors

# Employee

# Preconditions

# Employee is logged into the system

# Flow of Events

# *Basic Flow*

|  |  |
| --- | --- |
| **User Action** | **System Response** |
| The Employee navigates to Edit Profile Page | Edit Profile page is displayed |
| The Employee edits personal information and saves them | The system updates the information |

# 

# Post-Conditions

# The changes are reflected into database

# Business Rules

# Nonfunctional Requirements

Use Case Specification: View Organization Chart

# 1. Brief Description

# This use case allows the Employee to view the organization chart.

# 2. Actors

Employee

# 3. Preconditions

# Employee is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee selects “View Organization Chart” option from the menu | The system retrieves all the information about the organization hierarchy and displays the report |

# 5. Post-Conditions

# The system displays the report.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: View My PayCheck

# 1. Brief Description

# This use case allows the Employee to view his/her paycheck.

# 2. Actors

Employee

# 3. Preconditions

# Employee is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| Employee selects “View PayCheck” option from the menu | System retrieves the information about the employee’s paycheck and displays the report |

# 5. Post-Conditions

# The system displays the report.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: View My Vacation Days

# 1. Brief Description

# This use case allows the Employee to view his/her vacation days.

# 2. Actors

Employee

# 3. Preconditions

# Employee is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee selects “View My Vacation Days” option from the menu | The system retrieves the information about the employee’s vacation days and displays the report |

# 5. Post-Conditions

# The system displays the report.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: Generate Vacations for My Employees

# 1. Brief Description

# This use case allows the Employee to view, approve or disapprove the vacation requests of the employees working under him/her.

# 2. Actors

Employee

# 3. Preconditions

# Employee is logged into the system.

1. Junior Employee has requested for a leave or vacation.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee navigates to vacation request page. | The system displays the list of requests done by his/her juniors. |
| The Employee approves the request. | The system updates the request status. |

# *4.2 Alternate Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee navigates to vacation request page. | The system displays the list of requests done by his/her juniors. |
| The Employee disapproves the request. | The system updates the request status. |

# 5. Post-Conditions

# If the request is approved, the system sends the request to HR department to update the vacation time of the requesting employee.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: View Release or Sprint Staff

# 1. Brief Description

# This use case allows the Employee to view the staffs working under any release or sprints

# 2. Actors

Employee

# 3. Preconditions

# Employee is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee selects “View Release or Sprint Staff” option from the menu | The system retrieves the information about releases and sprints and the employees working on them |

# 5. Post-Conditions

# The system displays the report.

# 6. Business Rules

# 7. Nonfunctional Requirements

Use Case Specification: View Burndown Chart

# 1. Brief Description

# This use case allows the Employee to view the burndown chart.

# 2. Actors

# Employee

# 3. Preconditions

# Employee is logged into the system.

# 4. Flow of Events

# *4.1 Basic Flow*

|  |  |
| --- | --- |
| User Action | System Response |
| The Employee selects “View Burndown Chart” from the menu | The system retrieves the information about user stories development and testing remaining and generates the report. |

# 5. Post-Conditions

# The system displays the report.

# 6. Business Rules

# 7. Nonfunctional Requirements