



## **IT314: Software Engineering**

### **G17: StaffGrid**

**User stories, Functional/Non Functional  
Requirements, Use-cases**

## User Stories

Title	Front of the Card	Back of the Card	Priority (MoSCoW)
Secure Login and Logout	As an <b>Employee</b> , I want to securely log in and log out of the system, so that I can access my personal and work-related information safely.	<ul style="list-style-type: none"> <li>• Implement secure authentication methods.</li> <li>• Provide a logout option.</li> <li>• Ensure session management and automatic timeout.</li> </ul>	Must Have (Mo)
Create a New Employee Account	As an <b>Admin</b> , I want to create a new employee account, so that they can use this system.	<ul style="list-style-type: none"> <li>• Enter employee details (e.g., name, email, role, profile Image ).</li> <li>• Assign a unique employee ID and role.</li> <li>• Send account activation email with a new generated ID and password.</li> </ul>	Must Have (Mo)
Change Profile Details	As an <b>Employee</b> , I want to change my profile details, so that I can keep my personal information.	<ul style="list-style-type: none"> <li>• Allow updates to personal details (e.g., address, phone number).</li> <li>• Require authentication to access profile settings.</li> <li>• Save changes after verification.</li> <li>• Notify employees of successful updates via email.</li> </ul>	Should Have (S)
Reset Password	As an <b>Employee</b> , I want to reset my password if I forget it, so that I can regain access to my account without needing admin assistance.	<ul style="list-style-type: none"> <li>• Provide a "Forgot Password" option.</li> <li>• Send a temporary password via email after security verification.</li> <li>• Allow setting a new password with security checks.</li> </ul>	Should Have (S)
Send Announcement	As a <b>Manager/HR/Admin</b> , I want to send notifications to others, so that I can keep them informed about important updates or events.	<ul style="list-style-type: none"> <li>• Compose and send notifications in website</li> <li>• Allow managers/HR/admin to send announcements to: <ul style="list-style-type: none"> <li>→ All employees</li> <li>→ Specific employee groups</li> <li>→ Selected individuals.</li> </ul> </li> </ul>	Could Have (Co)

View Announcement	As an <b>Employee/Manager/HR</b> , I want to view announcements on the website ,so that I can stay informed about important updates or events shared by others.	<ul style="list-style-type: none"> <li>• View all announcements relevant to them</li> <li>• Allow pagination or infinite scrolling for long lists of announcements.</li> </ul>	Could Have (Co)
Upload Project by Manager	As a <b>Manager</b> , I want to upload the project into the system, so that my team can start working on it.	<ul style="list-style-type: none"> <li>• Provide an option for the manager to upload the project details with team member details.</li> <li>• Manager includes the project deadline.</li> <li>• Manager includes the project sub-tasks.</li> </ul>	Should Have (S)
Apply Leave	As an <b>Employee</b> ,I want to apply for leave ,so that I can conveniently request time off and track my applications.	<ul style="list-style-type: none"> <li>• Provide an online form for leave requests.</li> <li>• Select leave type, dates, and reasons.</li> <li>• Route requests to managers.</li> <li>• Track status and update leave balances.</li> </ul>	Should Have (S)
View applied Leave	As an <b>Employee</b> , I want to view my applied leave, so that I can track the status, dates, and history of my leave applications effectively.	<ul style="list-style-type: none"> <li>• Display current leave balance.</li> <li>• Update balances in real-time.</li> <li>• Provide historical leave data and projections.</li> </ul>	Should Have (S)
Manage Leave	As a <b>Manager</b> , I want to approve or deny leave requests submitted by employees, so that I can manage team availability and ensure smooth workflow operations.	<ul style="list-style-type: none"> <li>• I can able to change leave status</li> <li>• I can able to give comment in that applied leave</li> </ul>	Should Have (S)
Mark Attendance	As an <b>Employee</b> , I want to mark my attendance, So that I can record my presence for the day.	<ul style="list-style-type: none"> <li>• Employees can mark attendance using their profile.</li> <li>• Attendance should include date and time stamps.</li> <li>• Prevent duplicate attendance for the same day</li> </ul>	Should Have (S)

View Attendance Records	As an <b>Employee</b> , I want to view my attendance records, so that I can track my attendance history.	<ul style="list-style-type: none"> <li>• Display detailed attendance records with dates and times.</li> </ul>	Should Have (S)
Delete an Employee Profile	As an <b>Admin</b> , I want to delete an employee's profile from the system, so that I can remove records of employees who are no longer with the company.	<ul style="list-style-type: none"> <li>• Remove all employee records.</li> <li>• Archive data for compliance.</li> <li>• Notify employees.</li> </ul>	Should Have (S)
Ensure System Security	As a <b>Technical Department</b> , I want to ensure the EMS is running securely, so that employee data is protected from unauthorized access.	<ul style="list-style-type: none"> <li>• Implement regular security audits.</li> <li>• Ensure encryption of sensitive data.</li> <li>• Apply access controls.</li> <li>• Monitor for suspicious activities.</li> <li>• Update software regularly.</li> </ul>	Must Have (Mo)
Manage Salaries and Bonuses	As an <b>HR</b> , I want to manage employee salaries and bonuses, so that I can ensure accurate payment to employees.	<ul style="list-style-type: none"> <li>• Pay salary to employees</li> <li>• Maintain records of salary structures.</li> <li>• Process salary and bonus calculations.</li> <li>• Distribute payment information.</li> <li>• Update payroll records</li> </ul>	Should Have (Mo)

# Functional Requirements

## General:

- The system should manage and securely store all employee-related information.
- All users should be able to log in, log out, and change their passwords.
- The system should restrict access based on user roles (Admin, Manager, HR, Employee).
- The system should provide role-specific dashboards.
- All users should be able to view and update their profiles, including profile photos.
- The system should allow all users (Admin, Manager, HR, Employee) to mark attendance.
- The system should display a list of all projects uploaded to the platform.
- The system should have the capability to receive and display announcements effectively

## Admin:

- Admin should be able to create new employee accounts with auto-generated credentials and email notifications.
- Admin should be able to delete employee accounts after confirmation.
- Admin should view the list of all employees working in the company.
- Admin should be able to assign roles and permissions to employees.
- Admin should be able to send announcements to all employees or selected groups.
- Admin should upload, update, and delete project details.
- Admin should view attendance records of all employees (daily, monthly, yearly).
- Admin should approve or reject leave requests from HR and Managers.

## HR:

- HR should be able to make an announcement
- HR should be able to process salary payments via a payment gateway.
- HR should be able to mark attendance
- HR should apply for leave and track the status of their applications.

## Manager:

- Manager should upload, update, and delete project details and statuses.
- Managers should view attendance records for their team members.
- Managers should apply for leave and track the status of their applications.
- Manager should be able to make an announcement

## Employee:

- Employees should view their attendance records.
- Employees should mark their attendance.
- Employees should apply for leave and track the status of their applications.
- Employees should view project details assigned to them.
- Employees should view announcements made by Admin, HR, or Managers.

## Non-Functional Requirements

- **System Availability:** The system should be available 99.9% of the time to ensure continuous access.
- **Performance:** The system should handle up to 500 concurrent users without performance degradation.
- **Scalability:** The system should be scalable to accommodate future growth in the number of users and data.
- **Security:** The system must protect sensitive employee data with encryption and role-based access control.
- **Usability:** The user interface should be intuitive and easy to navigate for all types of users.
- **Compliance:** The system must comply with relevant labor laws and data protection regulations.
- **Backup and Recovery:** Regular backups should be performed, and the system should support data recovery within 24 hours in case of failure.
- **Response Time:** The system should respond to user actions within 3 seconds under normal operating conditions.
- **Interoperability:** The system should be compatible with existing HR and payroll software used by the organization.
- **Maintainability:** The system should be designed for easy maintenance, allowing updates and patches without significant downtime.

# Use Cases and their descriptions:

## 1. Use Case: Login/Logout Functionality

**Actor:** User (Admin, Employee, HR, Manager)

**Preconditions:**

- User is on the login page.
- The backend is connected and operational.
- User has valid credentials (for login).

**Postconditions:**

- Users are logged in and redirected to their respective dashboard.
- Users can log out successfully, ending their session.

**Main Success Scenario:**

1. User visits the login page.
2. User enters their username and password.
3. The system verifies the user's credentials.
4. If correct, the system logs the user in and redirects them to their dashboard.
5. Users can click on the logout button at any time.
6. The system logs the user out and redirects them to the login page.

**Extensions:**

- 3a. If login fails (incorrect credentials), the system displays an error message.
- 6a. If the logout process fails, the system alerts the user, and they try logging out again.

## 2. Use Case: Forgot Password Functionality

**Actor:** User (Admin, Employee,HR,Manager)

**Preconditions:**

- User has forgotten their password.
- User has a valid email registered in the system.

**Postconditions:**

- User receives a temporary password via email.
- Users can log in using the temporary password and reset it.

**Main Success Scenario:**

1. User clicks on the "Forgot Password" link on the login page.
2. User enters their registered email address.
3. The system generates a temporary password.
4. The system sends the temporary password to the user's email address.
5. User logs in with the temporary password.
6. The system prompts the user to set a new password.
7. User enters and confirms a new password.
8. The system securely stores the new password after hashing.
9. User is redirected to the dashboard and can now use the new password.

**Extensions:**

- 3a. If the email is not registered, the system notifies the user and offers the option to contact support.
- 6a. If the new password does not meet security requirements, the system rejects it and prompts the user to choose a stronger password.
- 9a. If the password update fails, the system shows an error message, and the user is prompted to retry.

**3. Use Case: Create a New Employee Account**

**Actor:** Admin

**Preconditions:**

- Admin is logged into the system.
- Admin has appropriate permissions to create new employee accounts.

**Postconditions:**

- A new employee account is created with valid details.
- Employee data is stored in the system.
- An email is sent to the employee with their company ID and auto-generated password.



**Main Success Scenario:**

1. Admin navigates to the "Create New Employee" page.
2. Admin enters the employee's details (e.g., name, email, role, etc.).
3. System validates the entered data (e.g., checks for valid email, ensures required fields are filled).
4. Admin submits the form.
5. System creates the employee account, stores the data, and generates an auto-generated password.
6. System sends an email to the employee's provided email address containing their company ID and auto-generated password.
7. System displays a success message, and the new employee appears in the employee list.

**Extensions:**

- 3a. If the data is invalid (e.g., invalid email or missing fields), the system displays an error message and prompts the admin to correct the information.
- 5a. If account creation fails, the system displays an error message, and the admin is prompted to retry.
- 6a. If the email fails to send, the system displays an error message, and the admin is prompted to retry the email send process.

**4. Use Case: Delete Employee Profile**

**Actor:** Admin

**Preconditions:**

- Admin is logged into the system.
- Admin has appropriate permissions to delete employee profiles.
- The employee profile exists in the system.

**Postconditions:**

- The employee profile is deleted from the system.
- Any associated data (e.g., attendance, leave records) is either deleted or flagged.

**Main Success Scenario:**

1. Admin navigates to the employee profile that they wish to delete.
2. Admin clicks the "Delete" button.

3. System prompts the admin to confirm the deletion action (e.g., "Are you sure you want to delete this employee profile?").
4. Admin confirms the deletion.
5. System deletes the employee's profile and associated data (if applicable).
6. System displays a success message confirming the deletion.

**Extensions:**

- 3a. If the admin cancels the deletion, the system returns the admin to the employee profile page without making any changes.
- 5a. If the deletion fails (e.g., due to network issues or database errors), the system displays an error message, and the admin is prompted to retry.

## **5. Use Case: View Profile**

**Actor:** User (Admin, HR, Manager, Employee)

**Preconditions:**

- User is logged into the system.
- Users have valid access permissions to view their profile (Admin can view all profiles, others can view their own profile).

**Postconditions:**

- User profile is displayed, showing the details of the user's information (name, role, email, etc.).
- Users can see their or others' profile depending on their role.

**Main Success Scenario:**

1. User logs into the system.
2. User navigates to the "View Profile" page.
3. The system retrieves the profile data from the database.
4. The system displays the user's profile information, including name, role, email, and any other relevant details.
5. Users can view the information without being able to edit (if they are not an Admin or HR).

**Extensions:**

- 3a. If the profile is incomplete or data is missing, the system displays a message indicating that the profile needs to be updated.
- 5a. If the profile data fails to load (e.g., due to network issues), the system shows an error message and prompts the user to try again.

## **6. Use Case: Show All Employees Working for the Company**

**Actor:** Admin

### **Preconditions:**

- Admin is logged into the system.
- Admin has the necessary permissions to view the list of all employees.

### **Postconditions:**

- The system retrieves and displays a list of all employees currently working for the company.
- Employees' basic details are displayed in the list (e.g., name, role, email).

### **Main Success Scenario:**

1. Admin logs into the system.
2. Admin navigates to the "Employee List" page.
3. The system retrieves a list of all employees from the database.
4. The system displays the list of employees, showing basic details like name, role, and email.
5. Admin can search/filter through the employee list by name, role, or other attributes.

### **Extensions:**

- **3a. If the employee list fails to load**, the system shows an error message, and the user is prompted to retry.
- **5a. If the user searches with no results**, the system displays a message saying "No results found."

## **7. Use Case: View Attendance Records**

**Actor:** Employee, Manager, HR

**Preconditions:**

- Employees are logged into the system.
- Attendance records exist for the employee.

**Postconditions:**

- Employees can view their attendance records.

**Main Success Scenario:**

1. Employee logs into the system.
2. Employee navigates to the "Attendance Records" page.
3. The system retrieves the employee's attendance data.
4. The system displays attendance records, including dates, status (Present, Absent, etc.), and any additional information (e.g., leave taken).
5. Employees can view records for the day, month, or year.

**Extensions:**

- 3a. If no attendance data is found, the system displays a message stating "No attendance records available."
- 4a. If there is an error retrieving the records, the system displays an error message and prompts the employee to try again.

**8. Use Case: Mark Attendance**

**Actor:** Employee, Manager, HR

**Preconditions:**

- Employees are logged into the system.
- The system has an active attendance marking process (e.g., face recognition, manual entry).

**Postconditions:**

- Employee's attendance is marked for the day.

**Main Success Scenario:**

1. Employee logs into the system.
2. Employee navigates to the "Mark Attendance" page.

3. The system prompts the employee to mark attendance (e.g., via face recognition or button click).
4. The employee successfully marks attendance.
5. The system records the attendance for the employee for that day and confirms the action.

**Extensions:**

- 3a. If face recognition fails, the system prompts the employee to manually mark attendance.
- 4a. If attendance marking fails, the system displays an error message and prompts the employee to retry.

## **9. Use Case: Apply for Leave**

**Actor:** Employee, Manager, HR

**Preconditions:**

- Employees are logged into the system.
- Employees have an available leave balance.
- The employee has a valid reason for applying for leave.

**Postconditions:**

- Employee's leave request is recorded and pending approval.
- Leave balance is not updated until the request is approved.

**Main Success Scenario:**

1. Employee logs into the system.
2. Employee navigates to the "Apply for Leave" page.
3. Employee selects the leave type (e.g., sick leave, vacation, etc.).
4. Employee enters the leave start and end dates and any additional notes (e.g., reason for leave).
5. Employee submits the leave request.
6. System records the leave request and updates the employee's leave status to "Pending."
7. The system notifies the employee that their leave request has been submitted successfully.

**Extensions:**

- 4a. If the employee has insufficient leave balance, the system displays an error message and prevents submission.
- 5a. If the leave request fails (e.g., due to network issues), the system displays an error message, and the employee is prompted to retry.

**10. Use Case:Approve/Deny Leave Requests**

**Actor:** Admin

**Preconditions:**

- Manager is logged into the system.
- Manager has access to view and manage leave requests.
- Leave requests are pending approval/denial.

**Postconditions:**

- Leave request status is updated to "Approved" or "Denied."
- Employee's leave balance is updated if the leave is approved.

**Main Success Scenario:**

1. Manager logs into the system.
2. Manager navigates to the "Leave Requests" page.
3. The system displays a list of pending leave requests.
4. Manager selects a leave request to review.
5. Manager reviews the details of the leave request (dates, reason, etc.).
6. Manager approves or denies the leave request.
7. If the leave is approved, the system updates the employee's leave balance accordingly and changes the leave status to "Approved."
8. If the leave is denied, the system updates the leave status to "Denied."
9. The system sends a notification to the employee regarding the status of their leave request.

**Extensions:**

- 6a. If the leave balance is insufficient when approving the leave, the system displays an error and prevents approval until the balance is sufficient.

- 8a. If the leave request cannot be processed (e.g., due to system error), the system displays an error message, and the manager is prompted to retry.

## **11. Use Case: Send Announcements**

**Actor:** Manager, HR, Admin

**Preconditions:**

- Manager, HR, or Admin is logged into the system.
- The sender has permission to create and send announcements.
- Announcement content is ready and recipients are selected.

**Postconditions:**

- Announcement is sent to the selected employee(s).
- The announcement is stored in the system and available for recipients to view.

**Main Success Scenario:**

1. Manager, HR, or Admin logs into the system.
2. User navigates to the "Send Announcements" page.
3. The system allows the user to compose the announcement (title, message, etc.).
4. The user selects one or more employees to send the announcement to.
5. The user submits the announcement.
6. The system sends the announcement to the selected employees and stores it in the database.
7. The system notifies the user that the announcement has been sent successfully.

**Extensions:**

- 3a. If the announcement content is empty, the system displays an error message prompting the user to enter the content before submitting.
- 4a. If there is an issue with selecting recipients, the system displays an error message prompting the user to check the recipient list.

## **12. Use Case: View Announcements**

**Actor:** Employee, Manager, HR

**Preconditions:**

- Employees are logged into the system.
- Employees have received announcements.

**Postconditions:**

- Employees can view all the announcements they have received.

**Main Success Scenario:**

1. Employee logs into the system.
2. Employee navigates to the "View Announcements" page.
3. The system retrieves a list of announcements that have been sent to the employee.
4. The system displays the announcements with details such as title, message, and date.
5. Employees can view the details of each announcement.

**Extensions:**

- 3a. If there are no announcements, the system displays a message stating "No announcements found."
- 4a. If there is an error in fetching announcements, the system displays an error message and prompts the employee to try again.

**13. Use Case: Upload Project**

**Actor:** Manager, Admin

**Preconditions:**

- Manager or Admin is logged into the system.
- The user has the necessary permissions to upload a project.

**Postconditions:**

- The project is uploaded to the system and available for viewing by all roles.
- Project details such as title, description, and status are stored.

**Main Success Scenario:**

1. Manager or Admin logs into the system.



2. User navigates to the "Upload Project" page.
3. The system prompts the user to enter project details (e.g., title, description, team members, tasks, etc.).
4. User submits the project details.
5. The system uploads the project and stores the data in the database.
6. The system confirms the successful upload and notifies the user.

**Extensions:**

- 3a. If the project details are incomplete or invalid, the system displays an error message prompting the user to complete all required fields.
- 5a. If there is an issue with uploading the project, the system displays an error message and prompts the user to retry.

#### **14. Use-cases: Update Project details/status**

**Actor:** Manager,Admin

**Preconditions:**

- Manager is logged into the system.
- Manager has the necessary permissions to change the project status.
- The project exists and is visible to the manager.

**Postconditions:**

- The status of the project is updated to the new status (e.g., "In Progress," "Completed," "On Hold") and details updated.

**Main Success Scenario:**

1. Manager logs into the system.
2. Manager navigates to the "Manage Projects" page.
3. Manager selects a project to update its details/status.
4. The system displays the current project details and allows the manager to select a new status.
5. Manager selects the new status and submits the change.
6. The system updates the project status and confirms the update.

**Extensions:**

- 4a. If the selected status is invalid, the system displays an error message and prompts the manager to select a valid status.
- 5a. If the status update fails, the system displays an error message and prompts the manager to retry.