



RAJALAKSHMI
ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

CS23432 – Software Construction

(REGULATION 2023)

RAJALAKSHMI ENGINEERING COLLEGE
Thandalam, Chennai-602015

Name: Arun V

Register No: 231001018

Year / Branch / Section: 2nd /IT/ FA

Semester: IV

Academic Year: 2024 - 2025

INDEX

| S.No. | Date | Title |
|-------|----------|--|
| 1. | 22/01/25 | Azure Devops Environment Setup. |
| 2. | 07/02/25 | Azure Devops Project Setup and User Story Management. |
| 3. | 14/02/25 | Setting Up Epics, Features, And User Stories for Project Planning. |
| 4. | 20/03/25 | Sprint Planning. |
| 5. | 28/03/25 | Poker Estimation. |
| 6. | 04/04/25 | Designing Class and Sequence Diagrams for Project Architecture. |
| 7. | 16/04/25 | Designing Architectural and ER Diagrams for Project Structure. |
| 8. | 18/04/25 | Testing – Test Plans and Test Cases. |
| 9. | 25/04/25 | Load Testing and Pipelines. |
| 10. | 02/05/25 | GitHub: Project Structure & Naming Conventions. |

| | |
|------------------------|---------------------------------------|
| Exp.No: 1 | AZURE DEVOPS ENVIRONMENT SETUP |
| Date:22/01/2025 | |

AIM: To set up and access the Azure DevOps environment by creating an organization through the Azure Portal.

INSTALLATION

1. Open your web browser and go to the Azure website: <https://azure.microsoft.com/en-us/getstarted/azure-portal>.

Sign in using your Microsoft account credentials.

If you don't have a Microsoft account, you can create one here: <https://signup.live.com/?lic=1>

Check out the how-to video series for tips on deploying your cloud workloads from the Azure portal. >

Azure mobile app
Stay connected to your Azure resources—anytime, anywhere. Now available for iOS and Android.
[Learn more >](#)



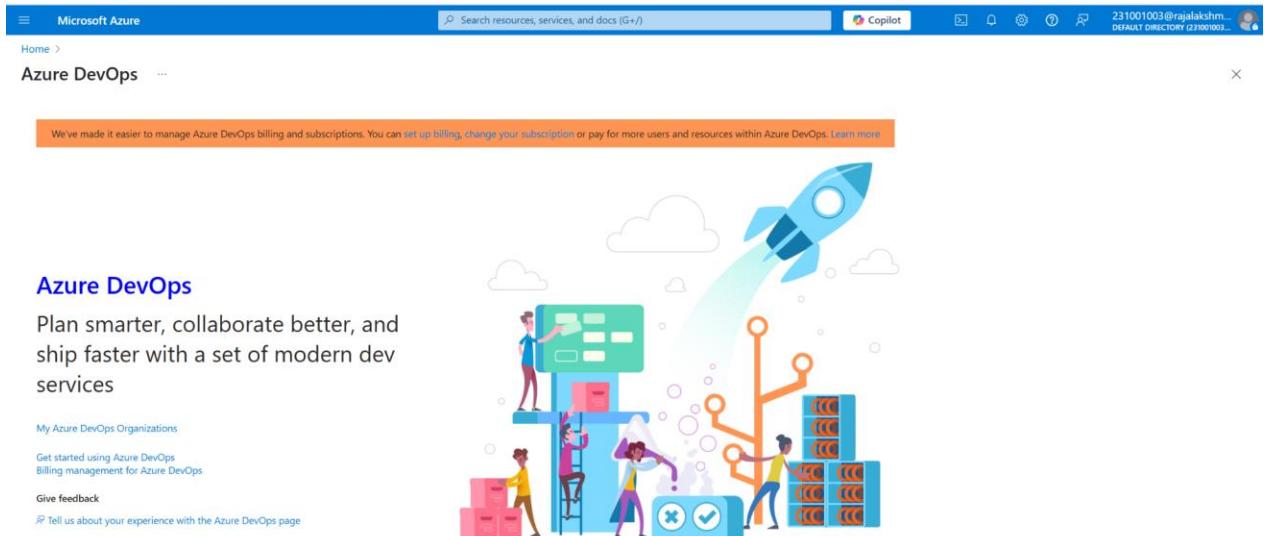
2. Azure home page

The screenshot shows the Microsoft Azure home page. At the top, there's a search bar and a Copilot button. Below the header, there's a section for 'Azure services' with icons for Create a resource, Subscriptions, All resources, Azure DevOps organizations, Quickstart Center, Azure AI services, Kubernetes services, Virtual machines, App Services, and More services. The 'Resources' section shows a table of recent and favorite resources, including 'MYBANKAPP' (Azure Load Testing) and 'BankProject' (Resource group). Below this is a 'Navigate' section with links to Subscriptions, Resource groups, All resources, and Dashboard. The 'Tools' section includes Microsoft Learn, Azure Monitor, Microsoft Defender for Cloud, and Cost Management. The 'Useful links' section has links to Technical Documentation, Azure Services, Recent Azure Updates, and the Azure mobile app. The 'Azure mobile app' section shows download links for the App Store and Google Play.

3. Open DevOps environment in the Azure platform by typing **Azure DevOps Organizations** in the search bar.

This screenshot is similar to the previous one but with a search bar at the top containing the text 'devops'. The search results are displayed on the right side of the page under the 'All' tab, showing items related to Azure DevOps organizations, such as 'Azure Native New Relic Service', 'Managed DevOps Pools', and 'Build Agents for Azure DevOps'. The rest of the page layout remains the same, including the 'Azure services', 'Resources', 'Tools', 'Useful links', and 'Azure mobile app' sections.

4.Click on the ***My Azure DevOps Organization*** link and create an organization and you should be taken to the Azure DevOps Organization Home page.



RESULT: Successfully accessed the Azure DevOps environment and created a new organization through the Azure portal.

Exp.No: 2

AZURE DEVOPS PROJECT SET UP AND USER STORY MANAGEMENT

Date: 07/02/2025

AIM: To set up an Azure DevOps project for efficient collaboration and agile work management.

STEPS:

- 1.Create An Azure Account

The screenshot shows the 'Almost done...' step of creating an Azure DevOps organization. It includes fields for the organization name (dev.azure.com/ 231001033), location (India), and a CAPTCHA challenge (XJW5 GyQ). A 'Continue' button is at the bottom.

Azure DevOps

231001018@rajalakshmi.edu.in [Switch directory](#)

Almost done...

Name your Azure DevOps organization *

dev.azure.com/ 231001033

We'll host your projects in

India

Enter the characters you see

New Audio

XJW5 GyQ

Continue

2.Create the First Project in Your Organization

- a. After the organization is set up, you'll need to create your first **project**. This is where you'll begin to manage code, pipelines, work items, and more.
- b. On the organization's **Home page**, click on the **New Project** button.
- c. Enter the project name, description, and visibility options:
Name: Choose a name for the project (e.g., LMS).
Description: Optionally, add a description to provide more context about the project.
Visibility: Choose whether you want the project to be **Private** (accessible only to those invited) or **Public** (accessible to anyone).
- d. Once you've filled out the details, click **Create** to set up your first project.

Create new project ×

Project name *
Salary-Management System

Description

Visibility

Public
Anyone on the internet can view the project. Certain features like TFVC are not supported.

Private
Only people you give access to will be able to view this project.

Public projects are disabled for your organization. You can turn on public visibility with [organization policies](#).

^ Advanced

Version control ?
Git

Work item process ?
Agile

3. Once logged in, ensure you are in the correct organization. If you're part of multiple organizations, you can switch between them from the top left corner (next to your user profile). Click on the Organization name, and you should be taken to the Azure DevOps Organization Home page.

Azure DevOps Organizations

[Create new organization](#)

dev.azure.com/231001018 (Owner)

Projects Actions

Salary management System Open in Visual Studio

New project

dev.azure.com/2310010180179 (Owner)

Arun V Edit profile

231001018@rajalakshmi.edu.in

Microsoft account

India 231001018@rajalakshmi.edu.in

Visual Studio Dev Essentials

Get everything you need to build and deploy your app on any platform.

Take your benefit

4. Project dashboard

Azure DevOps 231001018 / Salary management System / Overview / Summary

Search

Private Invite

Salary management System

About this project

Project Description

This project is a cloud-based Salary Management System designed to automate and simplify payroll processing using Microsoft Azure. It focuses on delivering a secure, scalable, and efficient solution for handling employee compensation workflows. The application is structured around key functional modules:

Employee Information and Payroll Setup

This module allows HR managers to manage employee profiles, define salary components, set tax rules, and configure allowances and deductions. It ensures all necessary data is captured for accurate payroll computation.

Salary Processing and Payslip Generation

This section automates monthly salary calculations based on employee attendance, leaves, bonuses, and deductions. It also generates digital payslips and enables bulk processing for large employee datasets. Salary slips can be emailed directly or downloaded via the portal.

Compliance and Reporting

This module helps organizations comply with statutory regulations such as PF, ESI, TDS, and more. It provides customizable reports and dashboards for audits, financial planning, and government submissions.

Security and Integration

This section ensures secure access via Azure Active Directory and integrates with third-party applications like ERP and HRMS platforms. Sensitive data is encrypted and managed using Azure Key Vault.

Each module is independently manageable, ensuring high availability, flexibility, and easy maintenance. Built with user experience and security at its core, the system promotes paperless payroll operations and enables real-time, compliant salary management through Azure's robust infrastructure.

5. To manage user stories:

- From the **left-hand navigation menu**, click on **Boards**. This will take you to the main **Boards** page, where you can manage work items, backlogs, and sprints.
- On the **work items** page, you'll see the option to **Add a work item** at the top. Alternatively,

you can find a + button or **Add New Work Item** depending on the view you're in.

From the **Add a work item** dropdown, select **User Story**. This will open a form to enter details for this user story.

The screenshot shows the Azure Boards interface for the 'Salary management System Team'. The left sidebar has a 'Backlogs' section selected. The main area displays a backlog of work items:

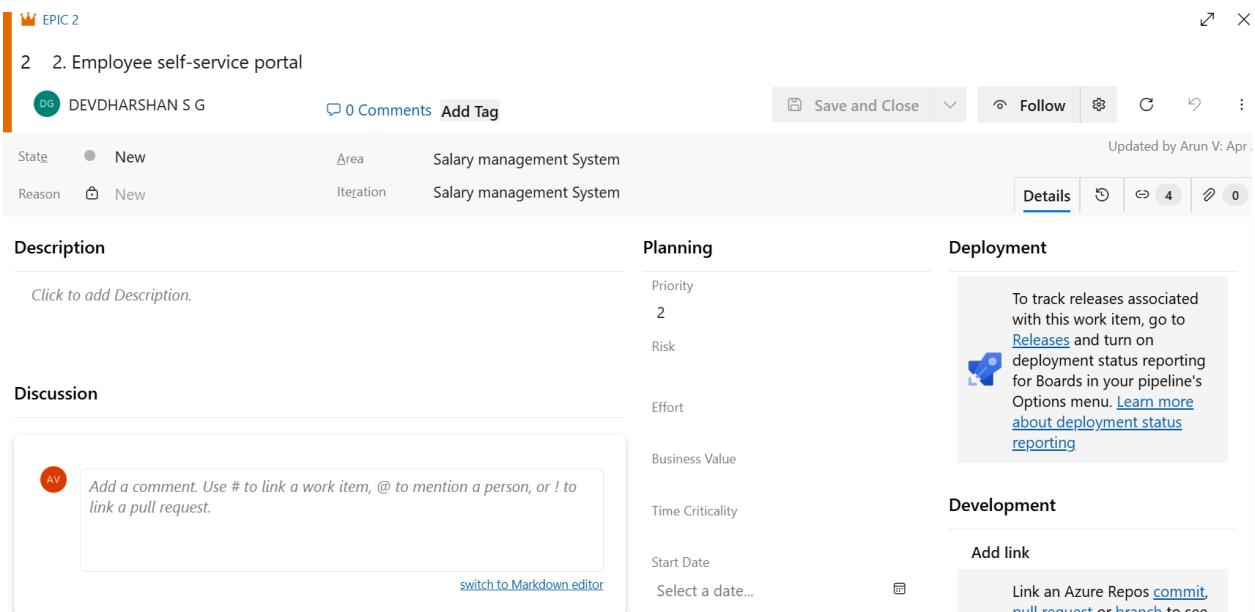
| Order | Work Item Type | Title | State | Effort | Business Area |
|-------|----------------|------------------------------------|-------|--------|---------------|
| 1 | Epic | > 1. Employee Management | New | | Business |
| 2 | Epic | > 2. Employee self-service portal | New | | Business |
| 3 | Epic | > 3. Attendance & Leave Management | New | | Business |
| 4 | Epic | > 4. Payroll Processing | New | | Business |
| 5 | Epic | > 5. Payment And Tax Compliance | New | | Business |

RESULT: Successfully created an Azure DevOps project with user story management and agile workflow setup.

| | |
|-------------------------|--|
| Exp.No: 3 | SETTING UP EPICS FEATURES AND USER STORIES FOR PROJECT PLANNING |
| Date: 14/02/2025 | |

AIM: To learn about how to create epics, user story, features, backlogs for your assigned project.

1. Fill in Epics



The screenshot shows the creation of an Epic titled "EPIC 2" under the "2. Employee self-service portal" section. The Epic is assigned to DEVDHARSHAN S G and has 0 comments and 0 tags. It is set to "New" status and belongs to the "Salary management System" area and iteration. The "Details" tab is selected, showing fields for Description, Planning, Deployment, Discussion, and Development.

| Description | Planning | Deployment |
|--|---|---|
| Click to add Description. | Priority: 2 Risk Effort Business Value Time Criticality | To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting |
| Discussion | Add link Link an Azure Repos commit, pull request or branch to see | |
| Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request. switch to Markdown editor | Start Date Select a date... | |

2.Fill in Features

The screenshot shows a Microsoft Azure DevOps Feature card. At the top, it says "FEATURE 6" and "6 Payment Gateway Integration". Below that, there are sections for "Description" (with placeholder text "Click to add Description."), "Planning" (Priority: 2, Risk), and "Deployment" (with a note about deployment status reporting). The "Discussion" section contains a comment from "AV" and a link to "switch to Markdown editor". On the right, there are sections for "Development" (Add link) and "Details" (with icons for Save, Close, Follow, etc.). The card is associated with the "Salary management System" area and "Iteration 1". It was updated by Arun V. on March 27.

3.Fill in User Story Details

The screenshot shows a Microsoft Azure DevOps User Story card. At the top, it says "USER STORY 9*" and "9 5.1.1 As a Payroll Administrator, I want to integrate a payment gateway so that I can disburse salaries directly to employees' bank accounts.". Below that, there are sections for "Description" (with placeholder text "Click to add Description."), "Acceptance Criteria" (listing requirements like "The system should allow configuration of multiple payment gateways.", "Salary transactions should be processed securely through the selected gateway.", and "A confirmation receipt should be generated for each transaction."), "Planning" (Story Points: 2, Priority: 2, Risk), and "Classification" (Value area: Business, Gleek). The "Development" section contains a note about linking to Azure Repos. The card is associated with the "Salary management System" area and "Iteration 1". It was updated by Arun V. on March 27.

4.Fill in Task Details

The screenshot shows the details of a task named 'TASK 59' with ID 59. The task title is 'Create a Profile page to display the user's personal information'. The task is set to 'New' state and 'Salary management System' area and iteration. It has 0 comments and 0 tags. The task was updated by Arun V just now. The task details are organized into sections: Description, Planning, Deployment, Discussion, Effort (Hours), and Development.

| Description | Planning | Deployment |
|---|---|---|
| Click to add Description. | Priority 2 Activity | To track releases associated with this work item, go to Releases and turn on deployment status reporting for Boards in your pipeline's Options menu. Learn more about deployment status reporting |
| Discussion | Effort (Hours) | Development |
| <p>Add a comment. Use # to link a work item, @ to mention a person, or ! to link a pull request.</p> <p>switch to Markdown editor</p> | Original Estimate Remaining Completed | <p>Add link</p> <p>Link an Azure Repos commit, pull request or</p> |

RESULT: Thus, the creation of epics, features, user story and task has been created successfully.

Exp.No: 4

SPRINT PLANNING

Date: 20/03/2025

AIM: To assign user story to specific sprint for the Online Banking Project.

SPRINT PLANNING

Sprint 1

The screenshot shows the Azure DevOps Taskboard for the 'Salary management System Team' in the 'Sprint 1' cycle. The board has columns for New, Active, Resolved, and Closed work items. Several user stories are visible, each with a detailed description and assigned team members:

- User Story 80: BUG_2 (Assigned to DEVDHARSHAN S G)
- User Story 9: 5.1.1 As a Payroll Administrator, I want to initiate a payment gateway so that I can disburse salaries directly to employee bank accounts. (Assigned to DEVDHARSHAN S G)
- User Story 10: 5.1.2 As an Employee, I want to receive notifications when my salary is credited so that I stay informed about my payments. (Assigned to Arun V)
- User Story 11: 5.1.3 As a Finance Manager, I want to view a payment history log so that I

Sprint 2

The screenshot shows the Azure DevOps Boards interface for the 'Salary management System' project. A modal window titled 'New Sprint' is open, prompting for the sprint name 'SPRINT-2', start date '12/05/2025', end date '11/07/2025', and location 'Salary management System'. The background board displays a backlog of work items, including tasks like 'As a Payroll Administrator, I want to integrate a payment gateway so that I can disburse salaries directly to employees' bank accounts.' and 'As an Employee, I want to receive notifications when my salary is credited so that I stay informed about my payments.'

Sprint 2

The screenshot shows the Azure DevOps Boards interface for the 'Salary management System' project after the sprint has been completed. The backlog now includes a new sprint named 'SPRINT2' with a duration of '35 work days remaining'. The backlog items have been updated to reflect the completion of the previous sprint, such as 'As an HR Manager, I want to upload employee-related documents. So that I can maintain records securely.' and 'As an HR Manager, I want to generate reports on employee data. So that I can analyze workforce trends.'

Sprint 3

The screenshot shows the Azure DevOps Boards interface for the 'Salary management System' project. The left sidebar is collapsed, and the main area displays the 'Sprint 3' backlog under the 'Salary management System Team' board. The backlog is organized into columns: New, Active, Resolved, and Closed. Three work items are visible in the 'New' column:

- 27**: As an HR Manager, I want to mark employees as active or inactive. So that only active employees are processed for payroll.
Person: All
- 35**: As an employee, I want to compare my tax deductions over different months, so that I can plan my finances accordingly.
Person: Unassigned
- 42**: As an HR manager, I want the system to automatically integrate leave requests with salary adjustments. So that employee salaries are accurately adjusted based on the number of leave days taken.
Person: Unassigned

At the top right, there are buttons for 'New Work Item' and 'Column Options'. A status bar at the bottom indicates the sprint duration: '14 July - 12 September 45 work days'.

RESULT: The Sprints are created for the Online Banking Project.

| | |
|-------------------------|-------------------------|
| Exp.No: 5 | POKER ESTIMATION |
| Date: 28/03/2025 | |

AIM: Create Poker Estimation for the user stories - Online Banking System Project.

POKER ESTIMATION

USER STORY 9

9 5.1.1 As a Payroll Administrator, I want to integrate a payment gateway so that I can disburse salaries directly to employees'

No one selected 0 Comments Add Tag Save Follow Updated by Arun V: Just no

| | |
|-------------|-------------------------------------|
| State: New | Area: Salary management System |
| Reason: New | Iteration: Salary management System |

Description

Click to add Description.

Acceptance Criteria

- The system should allow configuration of multiple payment gateways.
- Salary transactions should be processed securely through the selected gateway.

Planning

Story Points: 25
Priority: 1
Risk:

Classification

Value area: Business

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

RESULT: The Estimation/Story Points is created for the project using Poker Estimation.

Exp.No: 6

DESIGNING CLASS AND SEQUENCE DIAGRAMS FOR PROJECT ARCHITECTURE

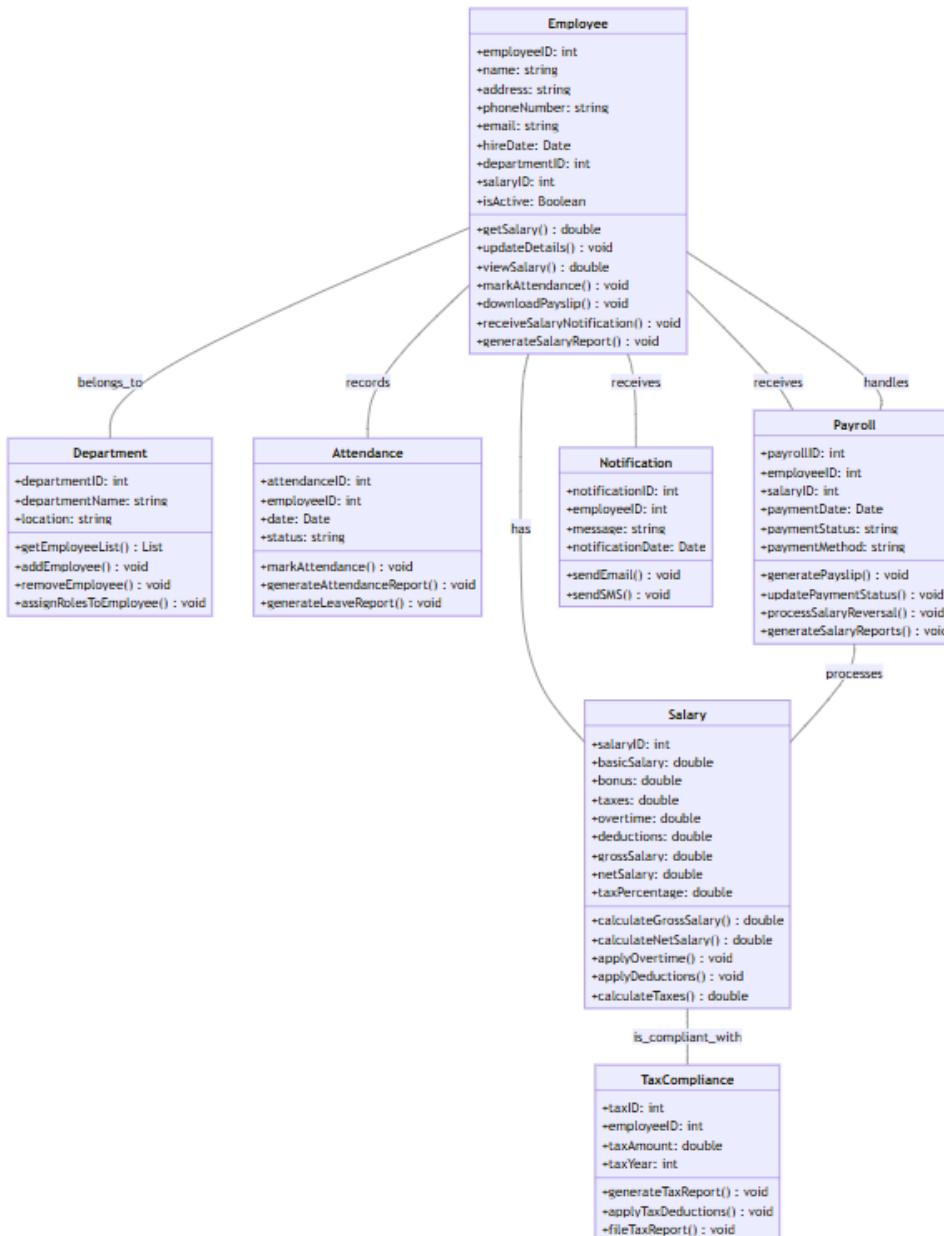
Date: 04/04/2025

AIM: To Design a Class Diagram and Sequence Diagram for the given Project.

6A. CLASS DIAGRAM

Class Diagram

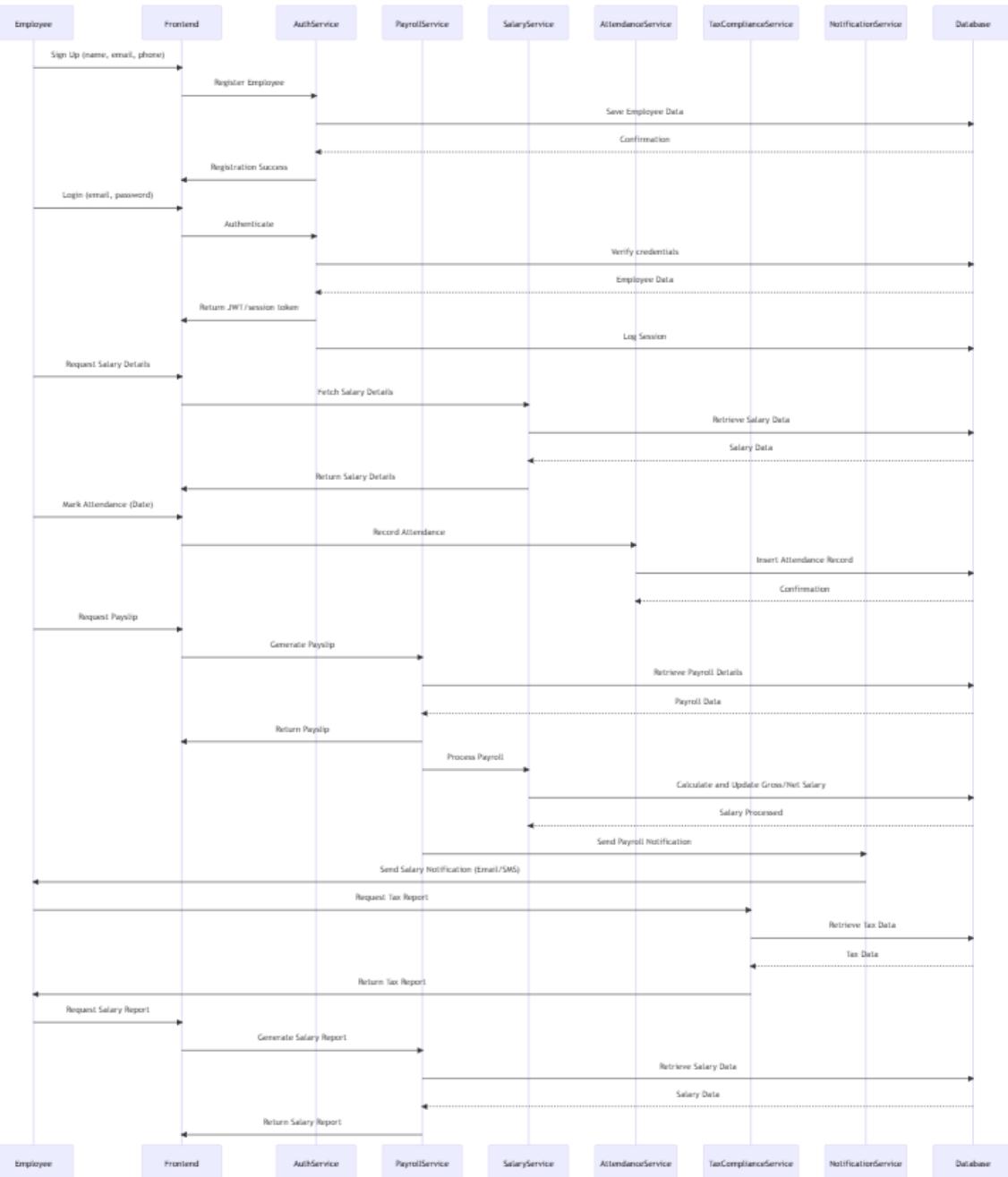
Dhanush Muthukumar 5 May



6B. SEQUENCE DIAGRAM

Sequence Diagram

Dhanush Muthukumar 5 May



RESULT: The Class Diagram and Sequence Diagram is designed Successfully for the Online Banking System.

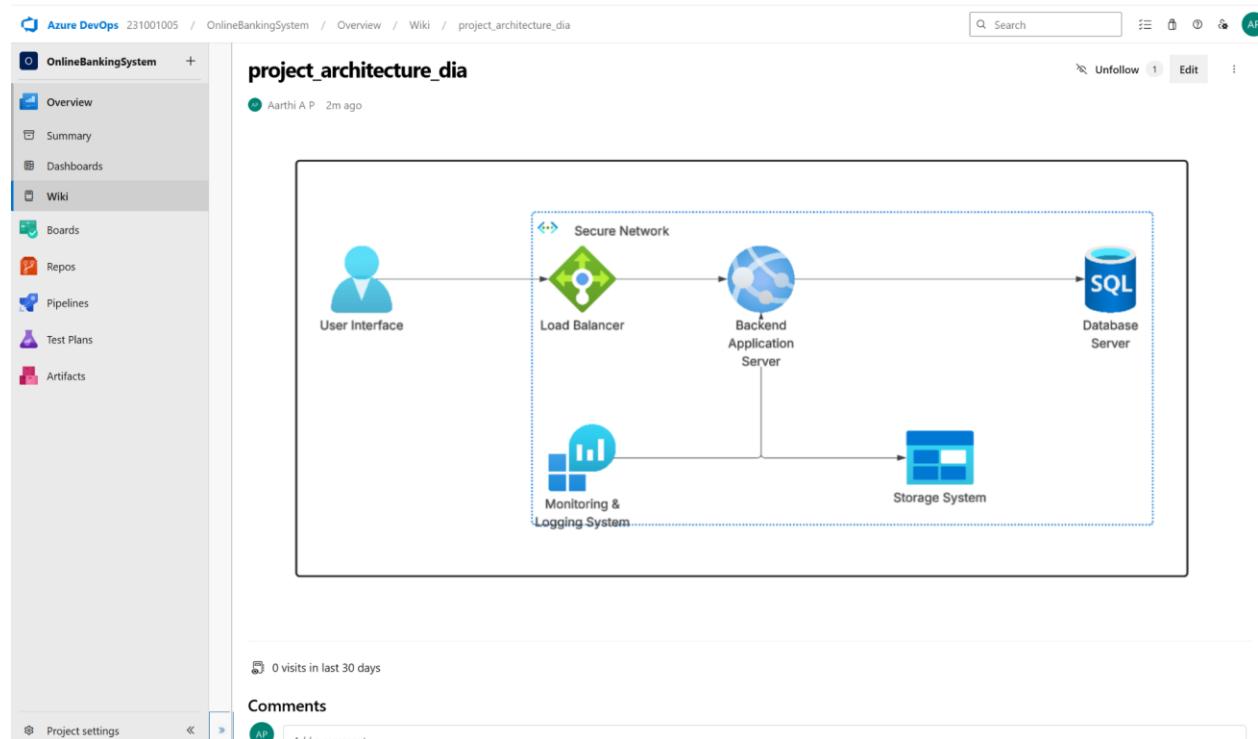
Exp.No: 7

Date: 16/04/2025

DESIGNING ARCHITECTURAL AND ER DIAGRAMS FOR PROJECT STRUCTURE

AIM: To Design an Architectural Diagram and ER Diagram for the given Project.

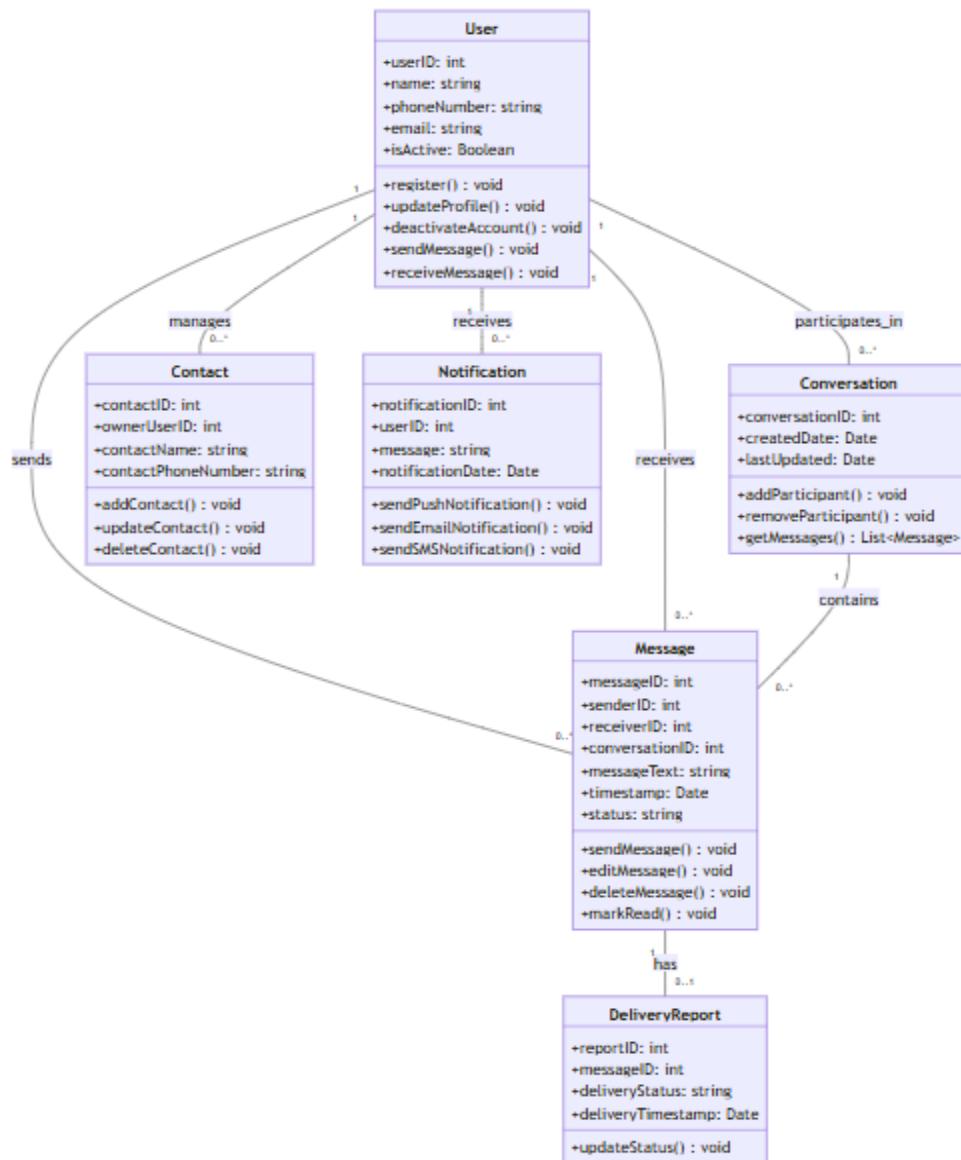
7A. ARCHITECTURAL DIAGRAM



7B. ER DIAGRAM

ER Diagram

Dhanush Muthukumar 5 May



0 visits in last 30 days

Comments



Add a comment...

RESULT: The Architecture Diagram and ER Diagram is designed Successfully for the Online Banking System.

| | |
|-------------------------|--|
| Exp.No: 8 | TESTING-TEST PLANS AND TEST CASES |
| Date: 18/04/2025 | |

AIM: Test Plans and Test Case and write two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

TEST PLANNING AND TEST CASE

Test Case Design Procedure

1. Understand Core Features of the Application

- Employee Management (Add, Update, Delete Employees, Roles)
- Employee Self-Service Portal (View Payslip, Request Leave, Update Profile)
- Attendance & Leave Management (Mark Attendance, Apply/Approve Leave)
- Payroll Processing (Calculate Salary, Generate Payslips)
- Payment and Tax Compliance (Salary Payments, Tax Calculations, Reports)

2. Define User Interactions

Each test case reflects realistic user actions:

- Adding and updating employee records
- Employees viewing and downloading payslips
- Marking daily attendance and applying for leave
- Processing monthly payroll correctly
- Generating tax reports and payment receipts

3. Design Happy Path Test Cases

- Focus on successful flows with valid inputs and expected system behavior
- Examples:
 - Employee successfully added with valid details
 - Employee views payslip for current month
 - Attendance marked successfully for the day
 - Payroll generated correctly for all active employees
 - Tax report generated with correct deduction values

4. Design Error Path Test Cases

- Focus on invalid, missing, or unexpected inputs and system handling
- Examples:
 - Employee addition fails due to missing mandatory fields
 - Payslip viewing fails due to incorrect employee ID
 - Attendance marking fails if already marked for the day
 - Payroll processing fails due to incomplete employee data
 - Tax report generation fails due to invalid tax year input

5. Break Down Steps and Expected Results

Each test case includes:

- Clear step-by-step actions
- Specific expected results for each step
This ensures reproducibility and easy automation later.

6. Use Clear Naming and IDs

- Example test case IDs:

- TC01 – Add Employee with Valid Data
- TC07 – Employee Payslip View Failure (Invalid ID)
- TC12 – Attendance Marking Success
- TC18 – Payroll Processing Blocked (Incomplete Data)
- TC23 – Tax Report Generation Failure (Invalid Year)
- Helps trace test cases to user stories and features in Azure DevOps.

7. Separate Test Suites

Organize test cases based on functional modules:

- TS01 – Employee Management
- TS02 – Employee Self-Service Portal
- TS03 – Attendance & Leave Management
- TS04 – Payroll Processing
- TS05 – Payment and Tax Compliance

1.New test plan

New Test Plan

Name *

Area Path *

Iteration *

Create **Cancel**

2. Test suite

The screenshot shows the Azure DevOps interface for a 'salary management system' project. The left sidebar is filled with various DevOps tools like Boards, Repos, Pipelines, and Test Plans. The main area is titled 'salary management system - Functional Validation (ID: 61)'. On the left, under 'Test Suites', there's a list with one item: 'salary management system - Functional Validation (ID: 61)'. A context menu is open over this item, showing options: 'New Suite', 'Assign configurations', 'Export', 'Assign testers to run all tests', and 'Import test suites'. Below the menu, there's a cartoon illustration of a person writing on a whiteboard with a dog sitting nearby. The text 'Add a test case' is displayed above a button labeled 'New Test Case'.

3. Test case

Give two test cases for at least five user stories showcasing the happy path and error scenarios in azure DevOps platform.

Online Banking System– Test Plans

User Stories – Banking Application

- **US199:** As a user, I want to securely log in to my bank account so I can view and manage my finances.
- **US200:** As a user, I want to transfer funds between my own accounts so that I can manage my money efficiently.
- **US196:** As a user, I want to generate a debit card PIN so that I can activate and use my card securely.
- **US198:** As a user, I want to open a fixed deposit so that I can earn guaranteed returns.
- **US201:** As a user, I want to pay my credit card bill so that I can avoid penalties.

TEST SUITE: TS01 – USER LOGIN (ID: 199)

Test Case: TC01 – Login with Valid Credentials

User Story: As a user, I want to securely log in to my bank account so I can view and manage my

finances.

Test Type: Happy Path

Steps and Expected Results:

- **Action:** Navigate to the login page
Expected Result: Login page is displayed with username and password fields
- **Action:** Enter valid username and password
Expected Result: Input is accepted with no errors
- **Action:** Click the "Login" button
Expected Result: User is redirected to the account dashboard

Notes:

Ensure the test user has an active bank account. This test simulates a successful login.

Test Case: TC02 – Login with Invalid Password

User Story: As a user, I want to securely log in to my bank account so I can view and manage my finances.

Test Type: Error Path

Steps and Expected Results:

- **Action:** Navigate to the login page
Expected Result: Login page is displayed
- **Action:** Enter valid username and incorrect password
Expected Result: Password is rejected
- **Action:** Click the "Login" button
Expected Result: Error message “Incorrect password” is shown

Notes:

Ensures failed login attempts are handled gracefully with proper error messages.

TEST SUITE: TS03 – PAYMENT GATEWAY INTEGRATION (ID: 511)

Test Case: TC01 – Successful Payment Gateway Integration

User Story: As a Payroll Administrator, I want to integrate a payment gateway so that I can disburse salaries directly to employees' bank accounts.

Test Type: Integration – Happy Path

Steps and Expected Results:

- **Action:** Navigate to the Payroll System Settings
Expected Result: System settings page is displayed with integration options.
- **Action:** Select the “Add Payment Gateway” option
Expected Result: Integration form for payment gateway appears.
- **Action:** Enter valid API credentials and configuration details
Expected Result: Credentials are accepted; no validation errors.
- **Action:** Click the "Connect" or "Save" button
Expected Result: Payment gateway is successfully linked to the payroll system.

Test Case: TC02 – Disburse Salary via Integrated Payment Gateway

User Story: As a Payroll Administrator, I want to integrate a payment gateway so that I can disburse salaries directly to employees' bank accounts.

Test Type: Transaction – Happy Path

Steps and Expected Results:

- **Action:** Navigate to the “Salary Disbursement” section
Expected Result: Disbursement panel with employee list is displayed.
- **Action:** Select employees and enter salary amounts
Expected Result: Amounts and accounts are validated successfully.
- **Action:** Click the “Disburse via Payment Gateway” button
Expected Result: Salaries are transferred; confirmation message is shown.

Notes:

Ensure employee bank account details are pre-configured.

Check payment gateway balance or limits before testing.

EST SUITE: TS04 – PAYMENT HISTORY LOG (ID: 513)

Test Case: TC01 – View Complete Salary Payment History

User Story: As a Finance Manager, I want to view a payment history log so that I can track all salary transactions.

Test Type: Functional – Happy Path

Steps and Expected Results:

- **Action:** Login as Finance Manager
Expected Result: Finance Manager dashboard is displayed.
- **Action:** Navigate to the “Payment History” or “Transaction Log” section
Expected Result: Payment history page loads successfully.
- **Action:** View list of past salary transactions
Expected Result: List shows employee names, amounts, dates, payment statuses, and reference IDs.

Notes:

Ensure data is accurate, complete, and sorted by latest transactions by default.

Log should reflect real-time updates after each disbursement.

Test Case: TC02 – Filter Payment History by Date and Status

User Story: As a Finance Manager, I want to view a payment history log so that I can track all salary transactions.

Test Type: Functional – Filtering

Steps and Expected Results:

- **Action:** Go to Payment History
Expected Result: Payment records are displayed.

- **Action:** Apply filters by date range and transaction status (e.g., Success, Failed, Pending)
Expected Result: Only matching records are displayed accurately.

Notes:

Filters should support combined criteria (e.g., date + status).

Useful for monthly reviews or auditing specific issues.

TEST SUITE: TS05 – EMPLOYEE ONBOARDING (ID: 111)

Test Case: TC01 – Add New Employee with Personal and Salary Details

User Story: As an HR Manager, I want to add a new employee with personal and salary details, so that the system stores their information for payroll processing.

Test Type: Functional – Happy Path

Steps and Expected Results:

- **Action:** Login as HR Manager
Expected Result: HR dashboard is displayed.
- **Action:** Navigate to “Add New Employee” section
Expected Result: New employee form is displayed.
- **Action:** Enter personal details (name, date of birth, contact, address, joining date)
Expected Result: Data is accepted without validation errors.
- **Action:** Enter salary details (base salary, allowances, deductions, payment method)
Expected Result: Salary fields are validated and accepted.
- **Action:** Click the “Save” or “Submit” button
Expected Result: Employee record is created and stored in the system.

Notes:

Ensure mandatory fields are marked and validated.

Check that the employee appears in the employee list immediately after saving.

Test Case: TC02 – Attempt to Add Employee with Missing Required Fields

User Story: As an HR Manager, I want to add a new employee with personal and salary details, so that the system stores their information for payroll processing.

Test Type: Error Path – Validation

Steps and Expected Results:

- **Action:** Go to Add New Employee form
Expected Result: Form is displayed.
- **Action:** Leave mandatory fields (e.g., name or salary) blank
Expected Result: System shows appropriate validation error messages (e.g., “Name is required”).
- **Action:** Try to submit the form
Expected Result: Submission is blocked until all required fields are completed.

Notes:

Ensures data integrity and prevents incomplete records.

Test Case: TC03 – Duplicate Employee Entry

User Story: As an HR Manager, I want to add a new employee with personal and salary details, so that the system stores their information for payroll processing.

Test Type: Error Path – Duplication Check

Steps and Expected Results:

- **Action:** Attempt to add an employee using the same national ID or email as an existing employee
Expected Result: System detects duplicate and displays error message (e.g., “Employee already exists”).

Notes:

Helps prevent accidental duplication in employee records.

Duplication check should be based on unique identifiers.

Test Cases

TEST CASE 63

63 TC01 – Successful Payment Gateway Integration

Arun V 0 Comments Add Tag

State: Design Area: Salary management System
Reason: New Iteration: Salary management System\Sprint 1

Save and Close Follow Updated by Arun V: Yesterday

Steps Summary Associated Automation Steps 1 0

Steps

Click or type here to add a step

| Steps | Action | Expected result | Attachments |
|-------|---|---|-------------|
| 1. | Navigate to the Payroll System Settings | System settings page is displayed with integration options. | |
| 2. | Select the "Add Payment Gateway" option | Integration form for payment gateway appears. | |
| 3. | Enter valid API credentials and configuration details | Credentials are accepted; no validation errors. | |
| 4. | Click the "Connect" or "Save" button | Payment gateway is successfully linked to the payroll system. | |

Parameter values

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

Add an existing work item as a parent

Tests

9 5.1.1 As a Payroll Administrator, I want to integrate a payment g... Updated Friday New

Status

TEST CASE 70

70 TC05 – View Complete Salary Payment History

Arun V 0 Comments Add Tag

State: Design Area: Salary management System
Reason: New Iteration: Salary management System\Sprint 1

Save and Close Follow Updated by Arun V: Yesterday

Steps Summary Associated Automation Steps 1 0

Steps

Click or type here to add a step

| Steps | Action | Expected result | Attachments |
|-------|--|---|-------------|
| 1. | Login as Finance Manager | Finance Manager dashboard is displayed. | |
| 2. | Navigate to the "Payment History" or "Transaction Log" section | Payment history page loads successfully. | |
| 3. | View list of past salary transactions | List shows employee names, amounts, dates, payment statuses, and reference IDs. | |

Parameter values

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

Add an existing work item as a parent

Tests

11 5.1.3 As a Finance Manager, I want to view a payment history l... Updated Friday New

Status

TEST CASE 74

74 TC08 – Add New Employee with Personal and Salary

Arun V 0 Comments Add Tag

State: Design Area: Salary management System
Reason: New Iteration: Salary management System\Sprint 1

Save and Close Follow Updated by Arun V: Yesterday

Steps Summary Associated Automation 5 1 0

Steps

| Steps | Action | Expected result | Attachments |
|-------|--|--|-------------|
| 1. | Login as HR Manager | HR dashboard is displayed. | |
| 2. | Navigate to "Add New Employee" section | New employee form is displayed. | |
| 3. | Enter personal details (name, date of birth, contact, address, joining date) | Data is accepted without validation errors. | |
| 4. | Enter salary details (base salary, allowances, deductions, payment method) | Salary fields are validated and accepted | |
| 5. | Click the "Save" or "Submit" button | Employee record is created and stored in the system. | |

Click or type here to add a step

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

[Add an existing work item as a parent](#)

Tests

13 1.1.1 As an HR Manager, I want to add a new employee with p...
Updated Friday New

Status

TEST CASE 76

76 TC10 – Duplicate Employee Entry

Arun V 0 Comments Add Tag

State: Design Area: Salary management System
Reason: New Iteration: Salary management System\Sprint 1

Save and Close Follow Updated by Arun V: Yesterday

Steps Summary Associated Automation 5 1 0

Steps

| Steps | Action | Expected result | Attachments |
|-------|--|--|-------------|
| 1. | Attempt to add an employee using the same national ID or email as an existing employee | System detects duplicate and displays error message (e.g., "Employee already exists"). | |

Click or type here to add a step

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

[Add an existing work item as a parent](#)

Tests

13 1.1.1 As an HR Manager, I want to add a new employee with p...
Updated Friday New

Status

TEST CASE 67

67 TC04 – Notification Delivery Failure

Arun V 0 Comments Add Tag

State: Design Area: Salary management System
Reason: New Iteration: Salary management System\Sprint 1

Updated by Arun V: Yesterday

Steps

| Steps | Action | Expected result | Attachments |
|-------|---|---|-------------|
| 1. | Admin disburses salary | Salary is credited to the employee's account. | |
| 2. | Employee's email/SMS service is unavailable | System logs notification failure and retries or alerts admin. | |

Click or type here to add a step

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

Add an existing work item as a parent

Tests

10 5.1.2 As an Employee, I want to receive notifications when my Updated Friday New

Status

TEST CASE 64

64 TC02 – Disburse Salary via Integrated Payment Gateway

Arun V 0 Comments Add Tag

State: Design Area: Salary management System
Reason: New Iteration: Salary management System\Sprint 1

Updated by Arun V: Yesterday

Steps

| Steps | Action | Expected result | Attachments |
|-------|---|--|-------------|
| 1. | Navigate to the "Salary Disbursement" section | Navigate to the "Salary Disbursement" section | |
| 2. | Select employees and enter salary amounts | Amounts and accounts are validated successfully. | |
| 3. | Click the "Disburse via Payment Gateway" button | Salaries are transferred; confirmation message is shown. | |

Click or type here to add a step

Deployment

To track releases associated with this work item, go to [Releases](#) and turn on deployment status reporting for Boards in your pipeline's Options menu. [Learn more about deployment status reporting](#)

Development

Add link

Link an Azure Repos [commit](#), [pull request](#) or [branch](#) to see the status of your development. You can also [create a branch](#) to get started.

Related Work

Add link

Add an existing work item as a parent

Tests

9 5.1.1 As a Payroll Administrator, I want to integrate a payment g... Updated Friday New

Status

4. Installation of test

Test and feedback

The screenshot shows the 'Test & Feedback' extension page on the Chrome Web Store. The extension has a rating of 4.2 stars from 175 reviews and over 200,000 users. It is categorized under 'Workflow & Planning'. The main image displays a screenshot of a computer screen with a red annotation overlaying a button, labeled 'Capture & Annotate'. Below this are four smaller thumbnail images showing various features like screenshots and annotations. The right side of the page includes a 'CAPTURE' section with a list of features: Screenshots, Notes, Screen recordings, etc., and a preview window showing a recording interface.

Showing it as an extension

The screenshot shows the Azure DevOps interface for a project titled 'Salary management System'. The left sidebar is open to 'Test Plans'. In the center, a test plan named 'TS01 – PAYMENT GATEWAY INTEGRATION (ID: 62)' is displayed with tabs for 'Define', 'Execute', and 'Chart'. Under the 'Test Points (2 items)' section, there are two entries: 'Title' and 'TC01 – Successful Payment Gateway Integration'. On the right, a floating 'Extensions' pane is open, showing 'Full access' granted to the 'Test & Feedback' extension. The extension is listed with its icon, name, and a 'Manage extensions' button. Other extensions visible in the pane include McAfee® WebAdvisor and Manage extensions.

5. Running the test cases

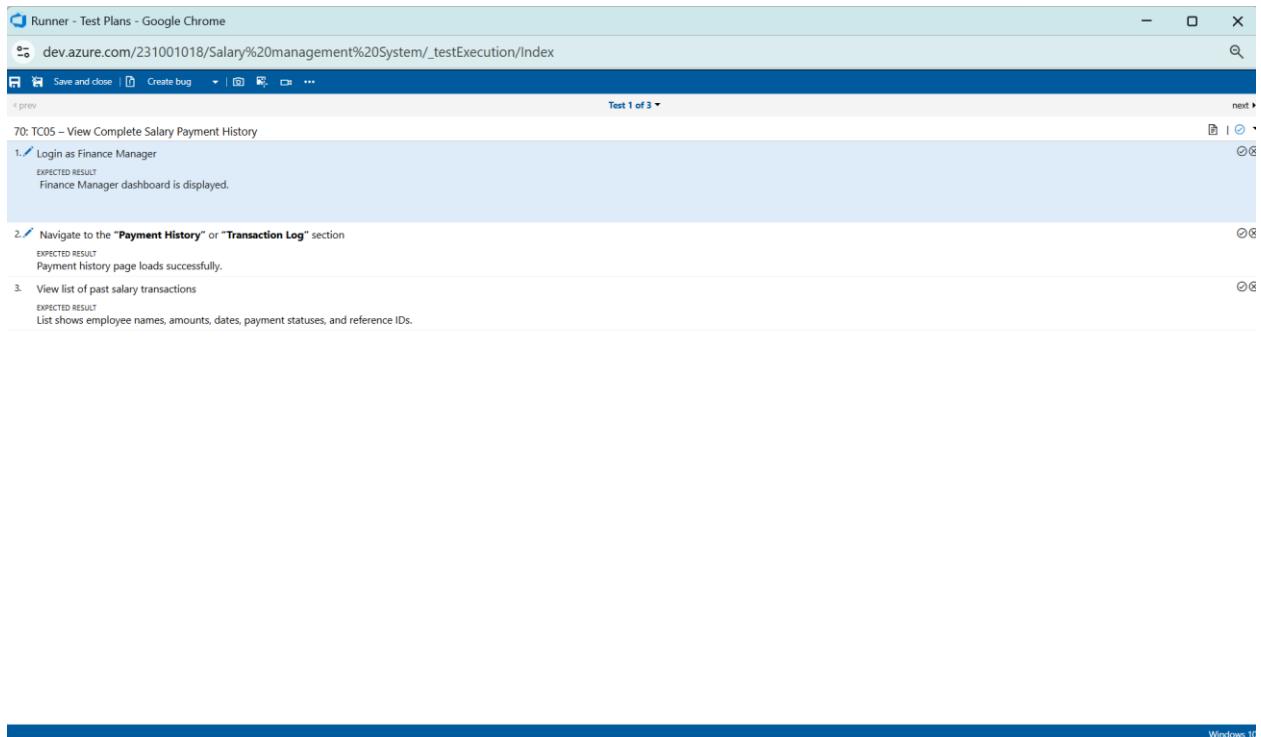
The screenshot shows the Azure DevOps Test Plans interface. On the left, the sidebar navigation includes: Overview, Boards, Repos, Pipelines, Test Plans (selected), Test plans, Progress report, Parameters, Configurations, Runs, and Artifacts. Below the sidebar is a "Project settings" link.

The main area displays a "Test Suites" view for "salary management system - Functional". A specific suite, "TS03 - PAYMENT HISTORY LOG (3)", is selected and expanded. This suite contains three test points: "TC05 - View Complete Salary Payment History", "TC06 - View Complete Salary Payment History", and "TC07 - Export Payment History". All three test points are listed as "Passed" with green status indicators.

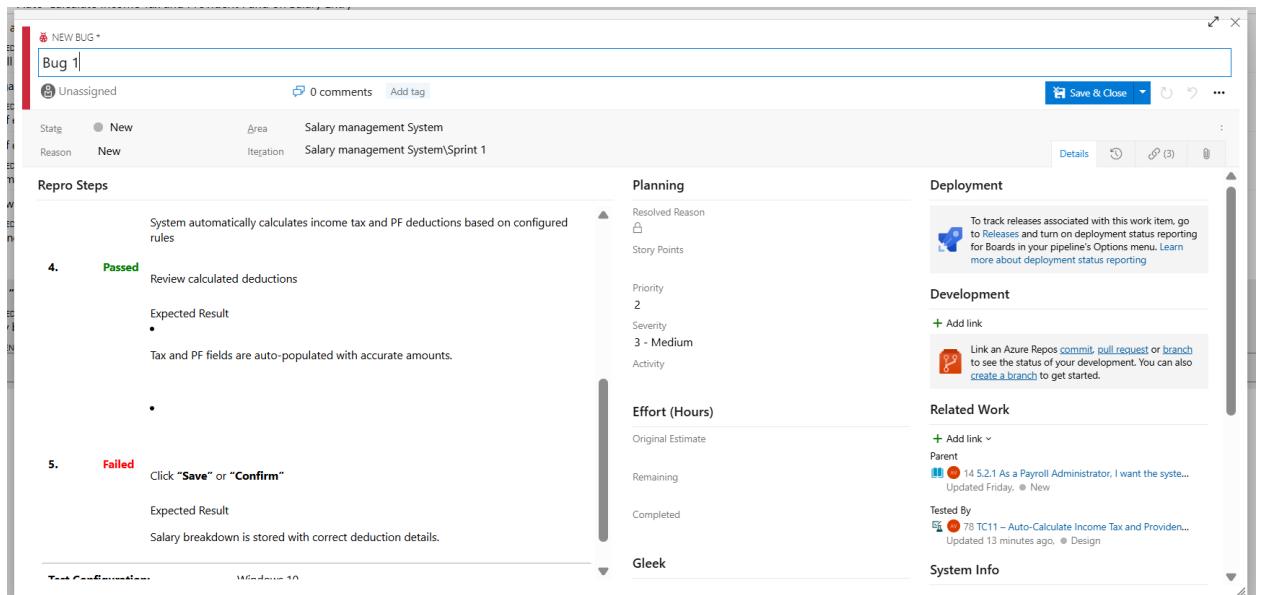
The screenshot shows a browser window titled "Runner - Test Plans - Google Chrome" with the URL "dev.azure.com/231001018/Salary%20management%20System/_testExecution/Index". The page displays a list of test steps for "78: TC11 - Auto-Calculate Income Tax and Provident Fund on Salary Entry".

- 1. Login as Payroll Administrator**
EXPECTED RESULT: Payroll dashboard is displayed.
- 2. Navigate to the "Employee Salary Processing" section**
EXPECTED RESULT: List of employees with salary details is displayed.
- 3. List of employees with salary details is displayed.**
EXPECTED RESULT: System automatically calculates income tax and PF deductions based on configured rules
- 4. Review calculated deductions**
EXPECTED RESULT: Tax and PF fields are auto-populated with accurate amounts.
- 5. Click "Save" or "Confirm"**
EXPECTED RESULT: Salary breakdown is stored with correct deduction details.

6.Recording the test case



7.Creating the bug



8. Test case results

The screenshot shows the Azure DevOps interface for a test plan. On the left, the navigation bar includes 'OnlineBankingSystem', 'Test Plans', and 'Test plans'. The main area displays a test suite named 'TS03 : As a user, I want to generate' with two test points: 'Successful Pin Generation' (Passed) and 'PIN mismatch during generation' (Not Run). A modal window titled 'Successful Pin Generation' shows the 'Test Case Results' table:

| Outcome | TimeSta... | Configuration | Run by | Tester | Test |
|---------|------------|---------------|------------------|------------------|---------|
| Passed | Just now | Windows 10 | Abinaya Maniv... | Abinaya Maniv... | Onli... |
| Failed | Just now | Windows 10 | Abinaya Maniv... | Abinaya Maniv... | Onli... |
| Passed | 21m ago | Windows 10 | Abinaya Maniv... | Abinaya Maniv... | Onli... |
| Passed | 26m ago | Windows 10 | Abinaya Maniv... | Abinaya Maniv... | Onli... |

9. Test report summary

The screenshot shows a bug report for 'BUG_2'. The top section includes fields for 'State' (New), 'Reason' (New), 'Area' (Salary management System), and 'Iteration' (Salary management System\Sprint 1). The 'Repro Steps' section details three steps, all of which failed. The 'Planning' section shows 'Story Points' (2), 'Priority' (2), 'Severity' (3 - Medium), and 'Activity'. The 'Deployment' section provides instructions for tracking releases. The 'Development' section includes a link to a branch. The 'Related Work' section lists a parent work item and a child work item. The 'Effort (Hours)' section shows original estimate, remaining, and completed hours. The 'Gleek' section is empty. The 'System Info' section shows the last update was on 23/05/2025.

Assigning bug to the developer and changing state

The screenshot shows the 'BUG_3' work item details in Azure DevOps. The 'State' is set to 'New' and 'Reason' is 'New'. The 'Area' is 'Salary management System' and 'Iteration' is 'Salary management System\Sprint 1'. The 'Repro Steps' section contains three steps, all of which have failed. The first step describes a payroll admin processing salary disbursement, with an expected result of a successful transaction. The second step describes a system checking employee notification settings, with an expected result of notifications being enabled. The third step describes a salary being credited to an employee's bank account, with an expected result of a notification being triggered. The 'Planning' section shows a priority of 2, severity of 3 - Medium, and activity 'Gleek'. The 'Deployment' section provides instructions on tracking releases. The 'Development' section links to an Azure Repos commit. The 'Related Work' section lists a parent work item and a test case. The 'System Info' section shows the last 14 days of test results.

10. Progress report

The screenshot shows the 'Progress report' for the 'salary management system - Functional Validation' test plan. The left sidebar shows the project navigation with 'Test Plans' selected. The main area displays a summary of 1 test plan and 13 test points, with 10 out of 13 run, resulting in a 76% pass rate. A pie chart shows 7 Passed and 3 Failed. To the right, an 'Outcome trend' chart tracks tests over time from May 11 to May 25, showing a significant increase in tests run after May 22. The legend indicates that grey areas represent 'Not run', green areas represent 'Passed', and red areas represent 'Failed'.

11.Changing the test template

The screenshot shows the 'Organization Settings' page for a workspace named '231001005'. On the left, there's a sidebar with 'General' and 'Security' sections, and a 'Boards' link. The main area is titled 'All processes' and lists several process templates:

| Name | Description | Team projects |
|-----------------|--|---------------|
| Basic (default) | This template is flexible for any process and great for teams getting started with Az... | 0 |
| Agile | This template is flexible and will work great for most teams using Agile planning me... | 0 |
| 231001005 Agile | ... | 1 |
| Scrum | This template is for teams who follow the Scrum framework. | 0 |
| CMMI | This template is for more formal projects requiring a framework for process improv... | 0 |

12.View the new test case template

The screenshot shows the 'Add a field to Test Case' dialog. The left sidebar has tabs for 'Definition', 'Options', and 'Layout'. The 'Definition' tab is active, showing the following fields:

Add a field to store custom, queryable data about your work items.

Use an existing field
Field: Acceptance Criteria

Create a field

| | |
|-------------|--|
| Name | Test Type |
| Type | Text (single line) |
| Description | Optionally provide a description for the field |

Learn more [\[?\]](#)

Add field Cancel

The screenshot shows the 'Layout' tab selected in the 'Test Plan' settings. The 'Steps' section is currently active. The interface includes a toolbar with 'New field', 'New group', 'New page', 'Get extensions', and a refresh icon. Below the toolbar, there are three tabs: 'Steps' (selected), 'Summary', and 'Associated Aut...'. The main area displays a 'Text (multiple lines)' field labeled 'Steps' and a 'Custom' section labeled 'Test Type' (Text (single line)). To the right, there are sections for 'Recent test results', 'Deployment', 'Development', and 'Related Work', each with a 'Links' button. A vertical sidebar on the left contains navigation links for General, Security, Boards, and Process.

RESULT: The test plans and test cases for the user stories is created in Azure DevOps with Happy Path and Error Path

| | |
|-------------------------|---|
| Exp.No: 9 | LOAD TESTING AND PERFORMANCE TESTING |
| Date: 25/04/2025 | |

AIM: To create an Azure Load Testing resource and run a load test to evaluate the performance of a target endpoint.

LOAD TESTING

Steps to Create an Azure Load Testing Resource:

Before you run your first test, you need to create the Azure Load Testing resource:

1. Sign in to Azure Portal
Go to <https://portal.azure.com> and log in.
2. Create the Resource
 - o Go to *Create a resource* → Search for “Azure Load Testing”.
 - o Select Azure Load Testing and click Create.
3. Fill in the Configuration Details
 - o *Subscription*: Choose your Azure subscription
 - o *Resource Group*: Create new or select an existing one.
 - o *Name*: Provide a unique name (no special characters).
 - o *Location*: Choose the region for hosting the resource.
4. (Optional) Configure tags for categorization and billing.
5. Click Review + Create, then Create.
6. Once deployment is complete, click Go to resource.

Microsoft Azure

Home > Create a resource > Marketplace > Azure Load Testing > Create a load testing resource ...

Search resources, services, and docs (G+/)

Copilot

231001003@rajalakshm... DEFAULT DIRECTORY (231001003...)

Basics Encryption Tags Review + create

Azure Load Testing is a fully managed load-testing service that makes it easy to generate high-scale load and identify performance bottlenecks. [Learn more!](#)

Project details
Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

Subscription *

Resource group *

Create new

Instance details
Name * ✓

Region *

Previous Next **Review + create**

Microsoft Azure

Home > Create a resource > Marketplace > Azure Load Testing > Create a load testing resource ...

Search resources, services, and docs (G+/)

Copilot

231001003@rajalakshm... DEFAULT DIRECTORY (231001003...)

Basics Encryption Tags **Review + create**

Validation passed.

Basics

| | |
|----------------|--------------------|
| Subscription | Azure for Students |
| Resource group | BankProject |
| Name | MYBANKAPP |
| Region | East Asia |

Encryption

| | |
|-----------------|-----|
| Encryption type | MMK |
|-----------------|-----|

Microsoft Azure

Home > Microsoft.CloudNativeTesting1746461401491 | Overview ...

Search Delete Cancel Redeploy Download Refresh

Overview

Your deployment is complete

Deployment name : Microsoft.CloudNativeTesting1746461401491
Subscription : Azure for Students
Resource group : BankProject

Start time : 5/5/2025, 9:40:23 PM
Correlation ID : ab036e0a-0f70-4d67-9550-f59c511ecdf6

Deployment details

Next steps

Go to resource

Give feedback

Tell us about your experience with deployment

Cost management
Get notified to stay within your budget and prevent unexpected charges on your bill.
[Set up cost alerts >](#)

Microsoft Defender for Cloud
Secure your apps and infrastructure
[Go to Microsoft Defender for Cloud >](#)

Free Microsoft tutorials
[Start learning today >](#)

Work with an expert
Azure experts are service provider partners who can help manage your assets on Azure and be your first line of support.
[Find an Azure expert >](#)

Steps to Create and Run a Load Test:

Once your resource is ready:

1. Go to your Azure Load Testing resource and click Add HTTP requests > Create.
2. Basics Tab o *Test Name*: Provide a unique name.
 - o *Description*: (Optional) Add test purpose.
 - o *Run After Creation*: Keep checked.
3. Load Settings o *Test URL*: Enter the target endpoint (e.g., https://yourapi.com/products).
4. Click Review + Create → Create to start the test.

Microsoft Azure

Home > Microsoft.CloudNativeTesting1746461401491 | Overview > MYBANKAPP

Create a URL-based test ...

Basics Test plan Parameters Load Monitoring Test criteria Review + create

Get started by creating a test for a URL, or configure an advanced load test for multiple URLs with additional options. [Learn more](#)

Test details

Provide a test name and a description. Test name and description will help you identify a test in the list of tests created in this resource.

Test name *

Test description

Run test after creation

Debug mode
If this is enabled, the first test run will run in debug mode providing debug logs along with request and response data for failed requests. The test run in debug mode will run with only one engine and up to 10 minutes duration. [Learn more](#)

Enable advanced settings

Configure a test with multiple requests in the 'Test plan' tab.

Previous Next Review + create

Create a URL-based test ...

Validation passed.

Basics Test plan Parameters Load Monitoring Test criteria Review + create

Basics

| | |
|------------------|--------------------------|
| Test tool | JMeter |
| Test name | Test_5/5/2025_9:41:40 PM |
| Test description | |
| Debug mode | Disabled |

Test plan

| | |
|------------------|----------|
| Test method | URL |
| Requests | Request1 |
| Input data files | |

Load

| | |
|-----------------------------|--------|
| Engine instances | 1 |
| Load pattern | Linear |
| Concurrent users per engine | 50 |
| Test duration (minutes) | 20 |
| Ramp-up time (minutes) | 1 |

Load distribution

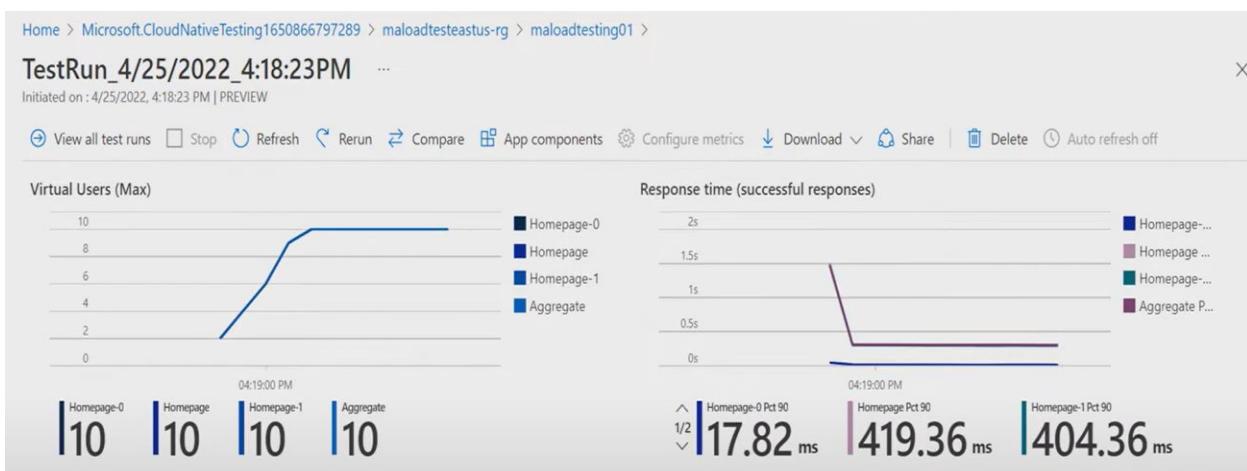
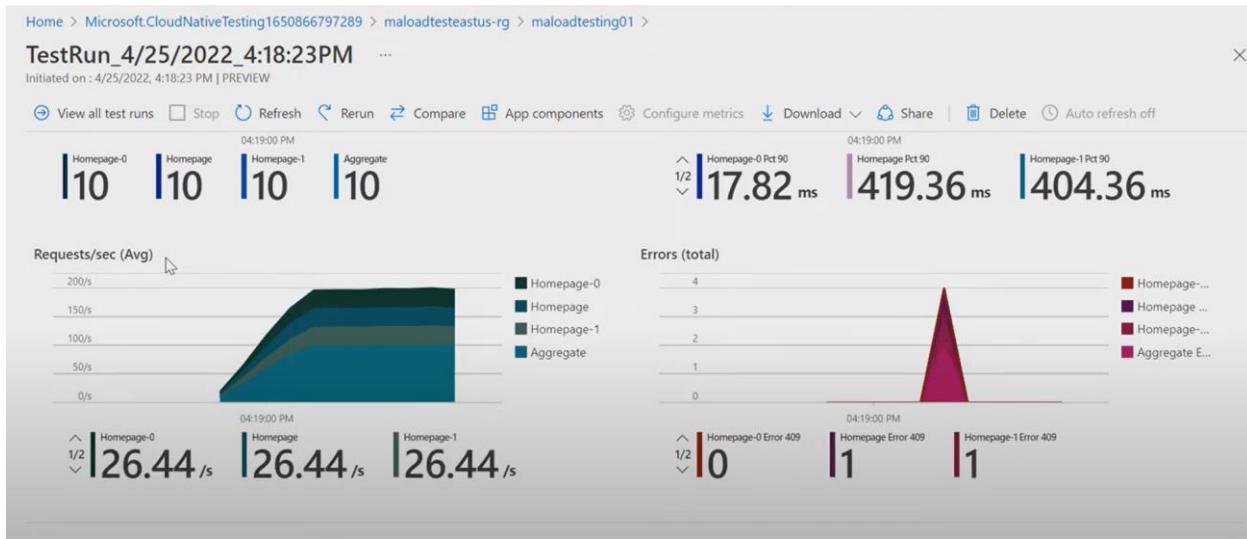
| | |
|-----------|---|
| East Asia | 1 |
|-----------|---|

Resources

| | |
|-----------|---|
| Resources | 0 |
|-----------|---|

Previous Next Create

Load Testing



RESULT: Successfully created the Azure Load Testing resource and executed a load test to assess the performance of the specified endpoint.

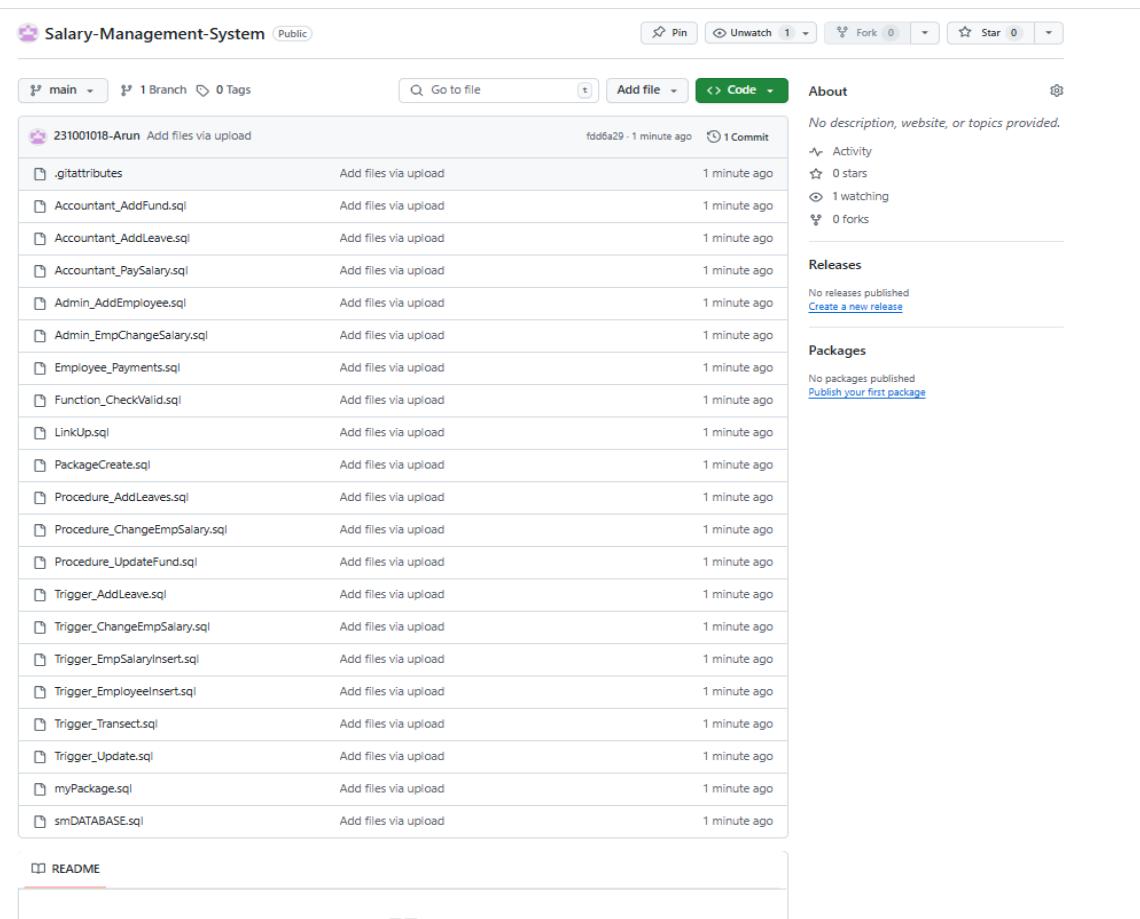
Exp.No: 10

Date: 02/05/2025

GITHUB PROJECT STRUCTURE & NAMING CONVENTIONS

AIM: To provide a clear and organized view of the project's folder structure and file naming conventions, helping contributors and users easily understand, navigate, and extend the Online Banking System project.

GITHUB PROJECT STRUCTURE



The screenshot shows a GitHub repository page for 'Salary-Management-System'. The repository is public and has 1 branch and 0 tags. It contains 1 commit from user 231001018-Arun made 1 minute ago. The commit message is 'Add files via upload'. The repository has 0 stars, 1 watching, and 0 forks. The files listed are all SQL scripts related to banking operations like accountants, admins, and triggers. The repository has 0 releases published.

| File | Description | Time Ago |
|-------------------------------|----------------------|--------------|
| .gitattributes | Add files via upload | 1 minute ago |
| Accountant_AddFund.sql | Add files via upload | 1 minute ago |
| Accountant_AddLeave.sql | Add files via upload | 1 minute ago |
| Accountant_PaySalary.sql | Add files via upload | 1 minute ago |
| Admin_AddEmployee.sql | Add files via upload | 1 minute ago |
| Admin_EmpChangeSalary.sql | Add files via upload | 1 minute ago |
| Employee_Payments.sql | Add files via upload | 1 minute ago |
| Function_CheckValid.sql | Add files via upload | 1 minute ago |
| LinkUp.sql | Add files via upload | 1 minute ago |
| PackageCreate.sql | Add files via upload | 1 minute ago |
| Procedure_AddLeaves.sql | Add files via upload | 1 minute ago |
| Procedure_ChangeEmpSalary.sql | Add files via upload | 1 minute ago |
| Procedure_UpdateFund.sql | Add files via upload | 1 minute ago |
| Trigger_AddLeave.sql | Add files via upload | 1 minute ago |
| Trigger_ChangeEmpSalary.sql | Add files via upload | 1 minute ago |
| Trigger_EmpSalaryInsert.sql | Add files via upload | 1 minute ago |
| Trigger_EmployeeInsert.sql | Add files via upload | 1 minute ago |
| Trigger_Transact.sql | Add files via upload | 1 minute ago |
| Trigger_Update.sql | Add files via upload | 1 minute ago |
| myPackage.sql | Add files via upload | 1 minute ago |
| smDATABASE.sql | Add files via upload | 1 minute ago |

RESULT: The GitHub repository clearly displays the organized project structure and consistent naming conventions, making it easy for users and contributors to understand and navigate the codebase.