Online Leave Management System

(OLMS – LeaveSync)

Bachelor of Technology
Computer Science and Engineering

Submitted By

Ravi Kumar Gupta (13000123069)

September 2025

Techno Main
EM-4/1, Sector-V, Salt Lake
Kolkata- 700091
West Bengal
India

Contents

1. Introduction

The management of employee leave is a critical HR function in any organization. Traditional, paper-based leave management processes are often inefficient, prone to errors, and lack transparency. They can lead to delays in approvals, incorrect leave balance calculations, and difficulties in tracking employee availability, which directly impacts project planning and productivity.

The **LeaveSync** web application is designed to automate and streamline the entire process of employee leave management. The primary goal of this project is to replace the manual system with a centralized, efficient, and user-friendly platform. This system will allow employees to apply for leave, check their leave balances, and view the status of their applications in real-time. It will also empower managers to review and approve or reject leave requests instantly, and enable the HR department to generate comprehensive reports for better workforce management.

The scope of this project includes user authentication, leave application submission, a manager's approval workflow, validation against company holidays, and an automated notification system. By providing a transparent and accessible platform, LeaveSync aims to reduce administrative overhead, minimize paperwork, and improve overall operational efficiency.

2. Life–Cycle Model Used and its Justification

For the development of LeaveSync, the **Iterative Development Model** has been selected. This model breaks down the development of a large application into smaller, more manageable parts or iterations. Each iteration goes through its own cycle of requirements, design, implementation, and testing, resulting in a functional piece of the software.

Justification:

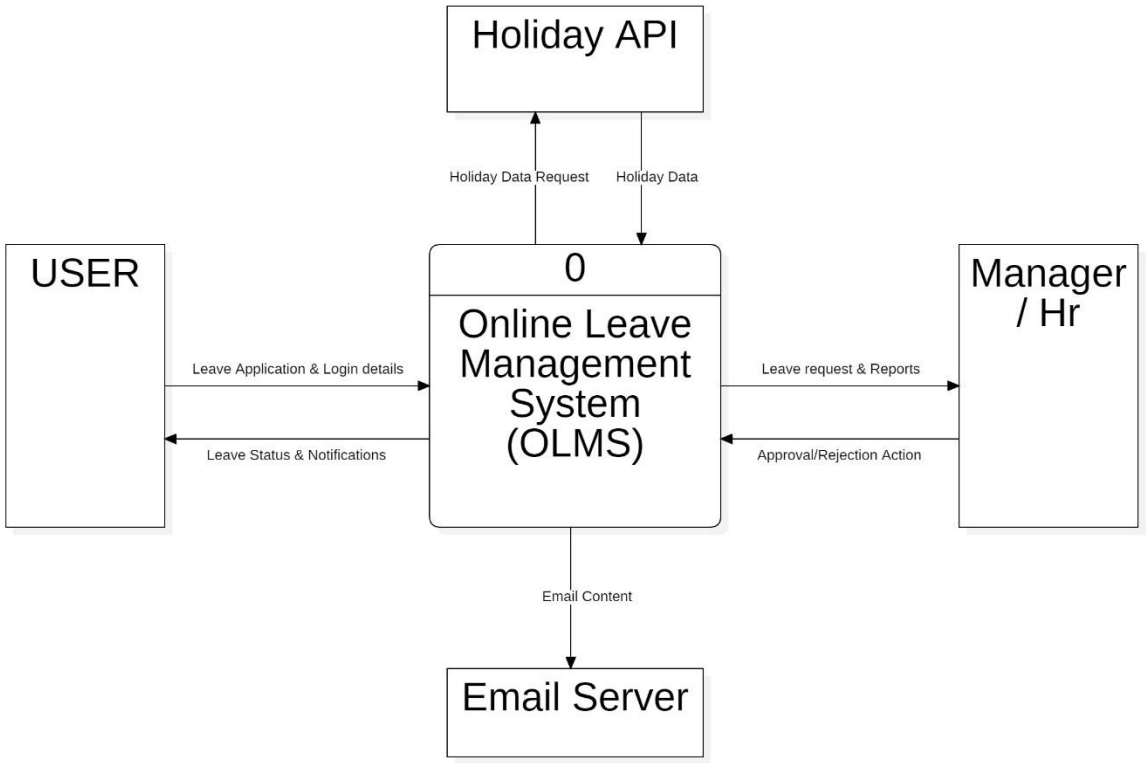
This model is particularly well-suited for the LeaveSync project for the following reasons:

1. **Early Working Prototypes:** Core functionality (user login, leave application, approval) can be developed in the first iteration, allowing stakeholders to provide valuable early feedback.
2. **Flexibility and Adaptability:** Requirements can be refined in subsequent iterations. Features like detailed reporting or a mobile interface can be added later based on user needs.
3. **Risk Management:** Developing the most critical features first significantly reduces the risk of project failure by identifying and fixing major issues early on.
4. **Phased Delivery:** The system can be rolled out in phases, making user adoption smoother and more manageable.

3. Design

The design of the system is modelled using Data Flow Diagrams (DFDs) to visualize the flow of information. The comprehensive data dictionary below provides a detailed repository of all data elements, processes, and stores used in the system, decomposed to multiple levels.

i) DFD – 0 (Context Diagram)



Overview: The Level 0 DFD, or Context Diagram, presents the entire **LeaveSync** system as a single, high-level process. It defines the boundaries of the system by illustrating its interactions with all external entities. This diagram shows the primary inputs the system receives and the outputs it produces, providing a clear overview of the system's scope without detailing any internal complexity.

Process:

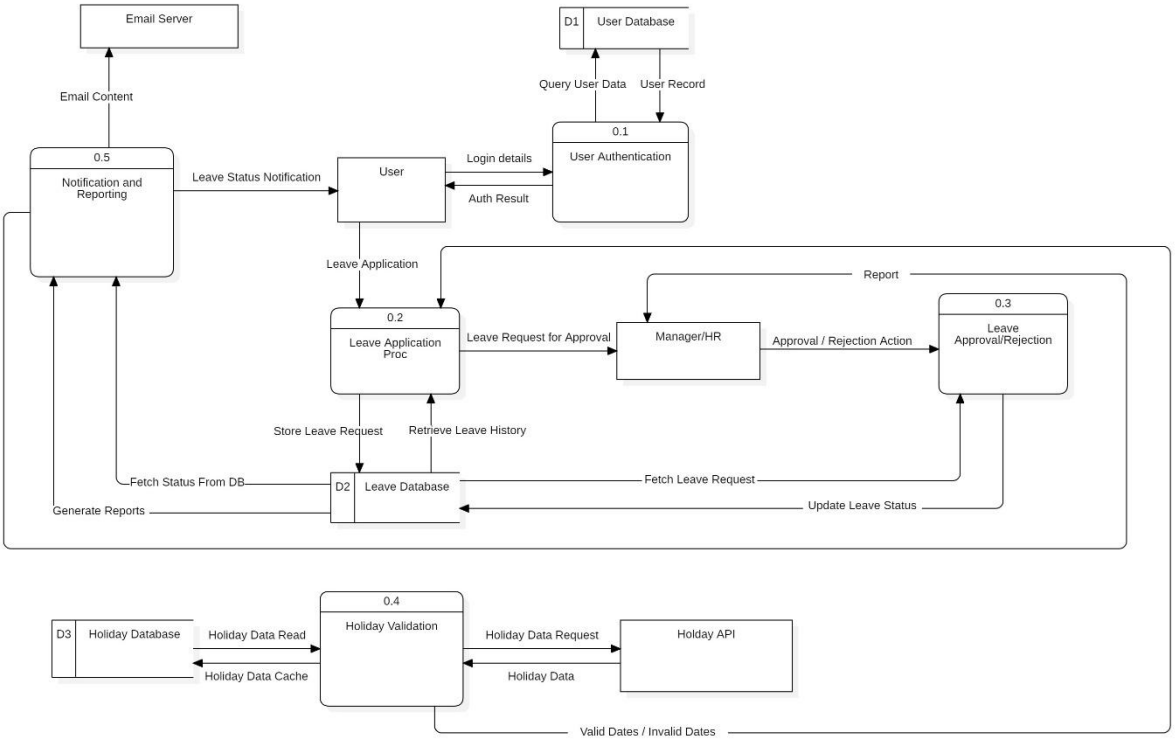
- **0 LeaveSync System:** This represents the complete software application, responsible for handling all functionalities related to leave management.

External Entities: The system interacts with four external entities: **User**, **Manager/HR**, **Holiday API**, and **Email Server**.

Data Stores:

- No internal data stores are shown at this level, as the focus is on the system's external interactions.

ii) DFD – 1 (Detailed Diagram)



Overview: The Level 1 DFD expands on the Level 0 Context Diagram by breaking the single "LeaveSync System" process into its major sub-systems. It provides a detailed view of the primary functions within the application, introduces the internal databases (data stores) where information is kept, and shows how data flows between these internal processes and the external entities.

Processes:

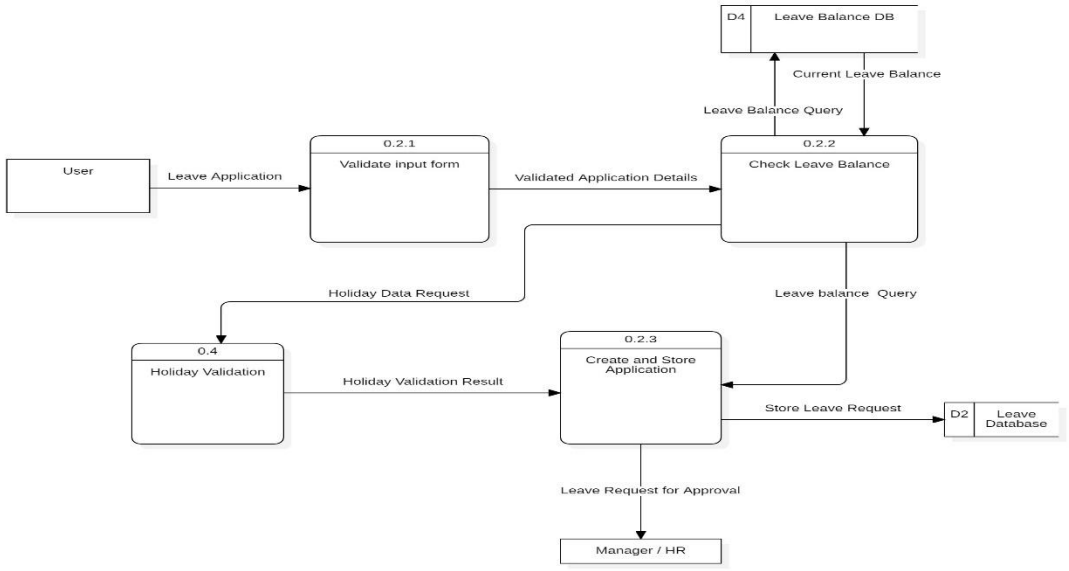
- **0.1 User Authentication:** Verifies user login credentials against the user database.
- **0.2 Leave Application Proc.:** Manages the entire leave application lifecycle, from submission and validation to storage.
- **0.3 Leave Approval/Rejection:** Handles the decision-making workflow for managers to approve or reject leave requests.
- **0.4 Holiday Validation:** Checks if the requested leave dates conflict with any official holidays by communicating with an API or local cache.
- **0.5 Notification & Reporting:** Responsible for sending automated email notifications to users and generating business reports for management.

External Entities: The same external entities from the Level 0 DFD interact with these more detailed internal processes.

Data Stores:

- **D1 User Database:** Stores credentials, roles, and profile information for all users.
- **D2 Leave Database:** Stores all leave application records, including their status and history.
- **D3 Holiday Database:** Acts as a local cache for holiday data fetched from the external API to improve performance.

iii) DFD – 2 (Application & Approval Processes)



Overview: This Level 2 DFD provides a more granular view by expanding the **0.2 Leave Application Proc.** from the Level 1 diagram. It details the specific sequence of sub-processes that occur when a user submits a new leave request, including initial validation, checking balances, and finalizing the application for storage.

Processes:

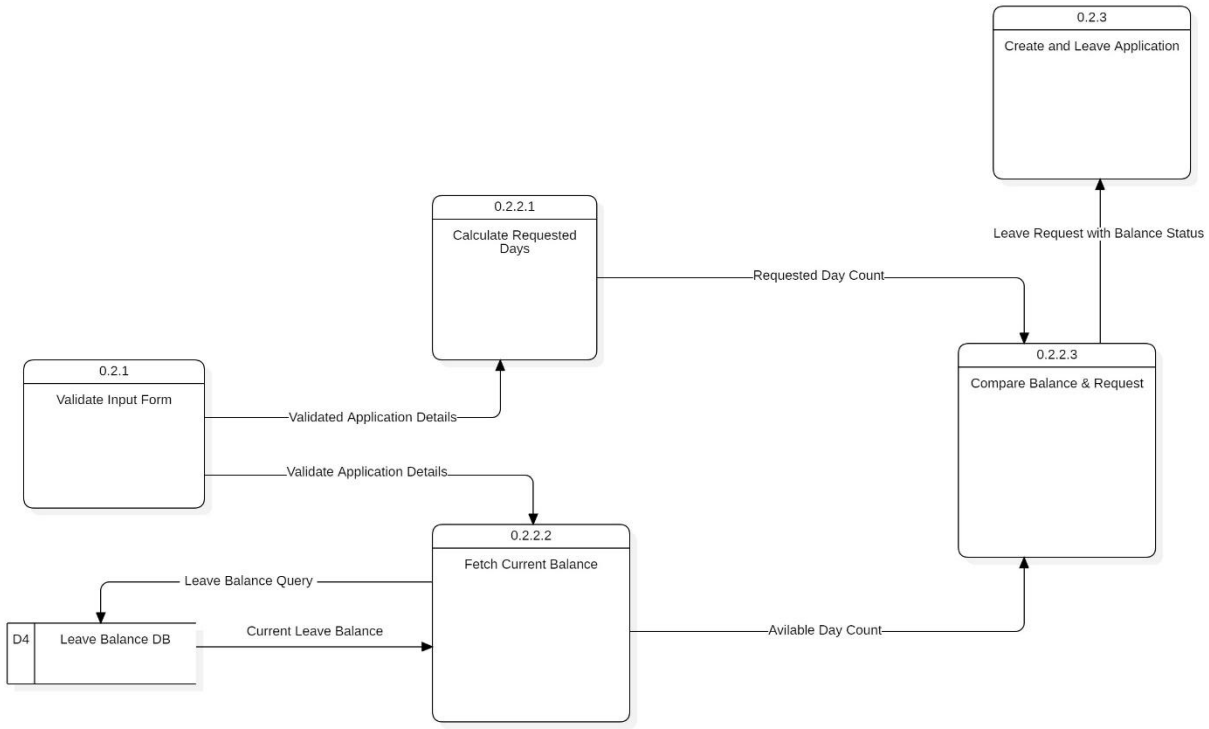
- **0.2.1 Validate Input Form:** Performs initial checks on the submitted leave application for completeness and correct data formats (e.g., end date is after start date).
- **0.2.2 Check Leave Balance:** Verifies if the employee has a sufficient number of available leave days for the requested period.
- **0.2.3 Create & Store Application:** Creates the final, validated leave application record and saves it to the main Leave Database.

External Entities: The boundaries of this sub-system interact with the **User** (providing input) and the **Manager/HR** (receiving the final request for approval). It also interacts with **Process 0.4** for holiday validation.

Data Stores:

- **D2 Leave Database:** Interacts with this store to retrieve leave history for validation.
- **D4 Leave Balance DB:** This new data store is introduced to specifically manage and query the current leave balance for each employee.

iv) DFD – 3 (Check Balance Process)



Overview: The Level 3 DFD offers the most detailed view of the system's logic by expanding the **0.2.2 Check Leave Balance** process. It breaks this single function down into its primitive (atomic) steps, showing exactly how the system calculates and verifies an employee's leave eligibility.

Processes:

- **0.2.2.1 Calculate Requested Days:** A pure calculation process that determines the net number of working days in the user's requested date range.
- **0.2.2.2 Fetch Current Balance:** A data retrieval process that queries the Leave Balance Database for the user's available days for the specific leave type.
- **0.2.2.3 Compare Balance & Request:** A logical process that compares the number of requested days against the available days to produce a final status ("Sufficient" or "Insufficient").

External Entities:

- No new external entities are introduced. All interactions are internal to the parent process.

Data Stores:

- This sub-system's primary interaction is with the **D4 Leave Balance DB** to fetch the necessary data for the comparison.

v) Complete Data Dictionary

Name	Type	Description	Composition	Source	Destination	Format
User	External Entity	Employee who logs in, applies for leave, and checks leave status.	-	-	-	-
Manager/HR	External Entity	Reviews leave requests, approves/rejects them, and checks reports.	-	-	-	-
Holiday API	External Entity	Provides holiday details for validation of leave requests.	-	-	-	-
Email Server	External Entity	Sends notification emails about leave status to users and managers.	-	-	-	-
0.0 OLMS	Process	Main system that processes login, leave application, approval, holiday validation, and reports.	-	-	-	-
0.1 User Authentication	Process	Verifies login credentials of users.	-	UserID, Password	D1 User Database	Text
0.2 Leave Application Proc.	Process	Handles leave application submission and stores in DB.	-	User	D2 Leave Database	Text/Numeric
0.2.1 Validate Input Form	Process	Checks completeness and valid formats of leave request.	UserID, Dates, Reason	User	0.2.2	Text
0.2.2 Check Leave Balance	Process	Verifies if user has enough leave balance.	UserID, LeaveType, RequestedDays	0.2.1	0.2.3	Numeric
0.2.2.1 Calculate Requested Days	Process	Calculates requested days excluding weekends.	StartDate, EndDate	0.2.1	0.2.2.3	Numeric
0.2.2.2 Fetch Current Balance	Process	Retrieves current leave balance from DB.	UserID, LeaveType	D4 Leave Balance DB	0.2.2.3	Numeric
0.2.2.3 Compare Balance & Request	Process	Compares requested vs available balance.	RequestedDays, AvailableDays	0.2.2.1, 0.2.2.2	0.2.3	Status
0.2.3 Create & Store Application	Process	Finalizes and stores leave application in DB; forwards to Manager.	AppID, UserID, Dates, Status	0.2.2, 0.4	D2, Manager/HR	Text/Numeric
0.3 Leave Approval/Rejection	Process	Manages leave approval/rejection by Manager/HR.	AppID, Action, Remarks	Manager/HR, D2	D2	Text
0.4 Holiday Validation	Process	Validates leave dates against holidays using API/DB.	DateRange	Holiday API, D3	0.2.3	JSON/XML
0.5 Notification & Reporting	Process	Sends notifications and generates reports.	AppID, Status, ReportCriteria	D2	User, Manager/HR, Email Server	Text/PDF
0.5.1 Email Notification	Process	Sends status update to user via Email Server.	Recipient, Subject, Body	0.5	Email Server	SMTP
0.5.2 Report Generation	Process	Generates leave history reports for Manager/HR.	UserID, Dept, History	D2	Manager/HR	PDF/Text

Name	Type	Description	Composition	Source	Destination	Format
D1 User Database	Data Store	Stores user credentials and profile.	UserID, Password, Role, Dept	0.1	0.1	Text/Numeric
D2 Leave Database	Data Store	Stores leave requests, status, and history.	AppID, UserID, Dates, Status	0.2, 0.3, 0.5	0.2, 0.3, 0.5	Text/Numeric
D3 Holiday Database	Data Store	Caches holiday data from API.	HolidayID, Date, Type, Desc	0.4	0.4	JSON/XML
D4 Leave Balance DB	Data Store	Stores leave balances per employee.	UserID, LeaveType, AvailableDays	HR System	0.2.2	Numeric
Login Details	Data Flow	Credentials entered by user.	UserID, Password	User	0.1	Text
Query User Data	Data Flow	Query to fetch user record.	UserID, Password	0.1	D1	Text
User Record	Data Flow	User details returned from DB.	UserID, Role, Dept	D1	0.1	Text/Numeric
Auth Result	Data Flow	Success/failure of login.	UserID, AuthStatus	0.1	User	Text
Leave Application	Data Flow	Application submitted by user.	UserID, Dates, Reason	User	0.2.1	Text
Validated Application Details	Data Flow	Application after basic validation.	UserID, ValidDates, Reason	0.2.1	0.2.2	Text
Leave Balance Query	Data Flow	Query for user's balance.	UserID, LeaveType	0.2.2.2	D4	Text/Numeric
Current Leave Balance	Data Flow	Available balance from DB.	UserID, AvailableDays	D4	0.2.2.2	Numeric
Requested Day Count	Data Flow	Number of requested days.	UserID, NumberOfDays	0.2.2.1	0.2.2.3	Numeric
Available Day Count	Data Flow	Number of days available.	UserID, NumberOfDays	0.2.2.2	0.2.2.3	Numeric
Leave Request with Balance Status	Data Flow	Leave with balance decision.	UserID, Dates, BalanceStatus	0.2.2.3	0.2.3	Text/Status
Holiday Data Request	Data Flow	Request for holiday validation.	DateRange	0.2.2	0.4	JSON/XML
Holiday Data	Data Flow	Holiday list returned.	HolidayName, Date	Holiday API	0.4	JSON/XML
Holiday Data Cache	Data Flow	Save holidays in DB.	HolidayID, Date, Type	0.4	D3	JSON/XML
Holiday Data Read	Data Flow	Cached holidays fetched.	HolidayList	D3	0.4	JSON/XML
Holiday Validation Result	Data Flow	Valid/invalid status.	Dates, Status	0.4	0.2.3	Text
Store Leave Request	Data Flow	Save finalized application.	AppID, UserID, Dates, Status	0.2.3	D2	Text/Numeric
Leave Request for Approval	Data Flow	Request sent to Manager/HR.	AppID, UserID, Dates	0.2.3	Manager/HR	Text
Fetch Leave Request	Data Flow	Pending requests for Manager.	AppID	D2	0.3	Text/Numeric
Approval/Rejection Action	Data Flow	Manager/HR's decision.	AppID, Action, Remarks	Manager/HR	0.3	Text
Update Leave Status	Data Flow	Update status in DB.	AppID, Status, Remarks	0.3	D2	Text/Numeric
Fetch Status from DB	Data Flow	Status retrieved for notifications.	AppID, Status	D2	0.5	Text/Numeric
Leave Status Notification	Data Flow	Notification to user.	AppID, Status, Remarks	0.5.1	User	Text
Email Content	Data Flow	Email notification data.	Recipient, Subject, Body	0.5.1	Email Server	SMTP
Reports	Data Flow	Generated reports for HR/Manager.	UserID, Leave History	0.5.2	Manager/HR	PDF/Text
Generate Reports	Data Flow	Query to DB for reports.	ReportCriteria	0.5.2	D2	Text/Numeric

4. Size Estimation using FP Metric

Function Point (FP) analysis is a standardized method used to measure the size of a software system based on its functionality from a user's perspective. It is independent of the programming language or technology used, making it a reliable metric for estimation.

a) Estimate the size of each module

The analysis for the LeaveSync system was conducted by identifying key user functions from the Data Flow Diagrams and categorizing them into four logical modules. The final adjusted Function Point (FP) size for each module is summarized below.

Module Name	Unadjusted FP (UFP)	Adjusted FP
1. User & Authentication Module	17	18.7
2. Leave Application Module	48	52.8
3. Manager Approval Module	7	7.7
4. Notification & Reporting Module	15	16.5
Total Project Size	**87**	**95.7**

The detailed, step-by-step calculations for this Function Point analysis are available in the following **spreadsheet**

5. COCOMO Estimations

The Constructive Cost Model (COCOMO) is used to provide estimations for the project's effort, schedule, and staffing requirements based on the estimated size of the software. The detailed, step-by-step formulas and calculations for this analysis are available in the following **spreadsheet**

Below is a summary of the methodology and final results derived from those calculations.

a) Justify the choice of COCOMO Model

The **Basic COCOMO** model is chosen because this is an early-stage estimation for a small-to-medium-sized project. It provides a quick and straightforward estimate based on the software's estimated size.

b) Justify the choice of Product Class for each module

The **Organic** product class is selected for all modules.

- **Justification:** The project is a business application developed by a small, experienced team with well-understood requirements and no rigid constraints. All modules fall under this category of standard application development.

c) Effort

- Total LOC = 95.7 FP * 60 LOC/FP = 5742 LOC = **5.742 KLOC**
- **Effort (E)** = 2.4 * $(5.742)^{1.05}$ ≈ **15.03 Person-Months**

d) Development Time

- **Time (D)** = 2.5 * $(15.03)^{0.38}$ ≈ **7.003 Months**

e) Development Cost

- Assuming an average loaded cost of **₹80,000 per month**.
- **Cost** = 7.003 * 80,000 ≈ **₹5,60,243**

f) Staff Requirement

- **Staff (S)** = 15.03 / 7.003 ≈ **2.14 Persons**
- This suggests a core development team of **2 people**.

6. Conclusion

The LeaveSync project is designed to provide a robust and efficient solution to a common business challenge. The choice of an Iterative Model ensures a flexible development process. The system's functionality, detailed in the DFDs and data dictionary, covers all critical aspects of the leave management workflow.

Based on Function Point and COCOMO estimations, the project requires approximately 15 person-months of effort over a 7-month timeline with a core team of two. These estimations provide a solid baseline for project planning, resource allocation, and timeline management, ensuring a successful delivery of the final product.

7. References

1) Pressman, R. S., & Maxim, B. R. (2019). *Software Engineering: A Practitioner's Approach*. McGraw-Hill Education.
2) Sommerville, I. (2015). *Software Engineering*. Pearson Education.
3) Boehm, B. W. (1981). *Software Engineering Economics*. Prentice-Hall.