

Wall-D: Demo Company Setup & Usage Guide

Flutter + Firebase Implementation

Version: 1.0 - Complete Demo Company Example

Target Stack: Flutter (Mobile/Web) + Firebase Firestore + Realtime DB

Scope: TechVision Inc. - A complete enterprise workflow example

Date: December 2025

TABLE OF CONTENTS

1. Demo Company Overview
 2. Organization Hierarchy
 3. Designations & Roles System
 4. User Registration Forms
 5. Employee Workflows
 6. Department-Specific Workflows
 7. Forms & Data Models
 8. Task Workflows
 9. Approval Chains
 10. Real-Time Scenarios
 11. Firebase Structure
 12. Flutter Implementation Guide
-

1. DEMO COMPANY OVERVIEW

1.1 Company Profile

COMPANY NAME:	TechVision Inc.
INDUSTRY:	Software Development & Digital Services
HEADQUARTERS:	Bangalore, India
EMPLOYEES:	120 (across 2 offices)
LOCATIONS:	Bangalore HQ, Delhi Office
FOUNDED:	2015
WORKING STYLE:	Agile + Project-Based

DEPARTMENTS:

- Engineering (35 employees)
- Design & UX (15 employees)
- Marketing & Sales (20 employees)
- Operations & HR (10 employees)
- Finance & Admin (8 employees)
- Executive Leadership (2 employees)

1.2 Business Workflow Overview

TYPICAL WORKFLOW:

1. TASK CREATION

Manager creates task (e.g., "Develop payment module")
Task gets assigned to team lead/senior developer
Task contains custom fields (story points, sprint, project)
Team members notified

2. TASK EXECUTION

Assignee marks task IN_PROGRESS
Daily standups tracked in comments
Time logs attached to task
Blockers documented

3. TASK COMPLETION & APPROVAL

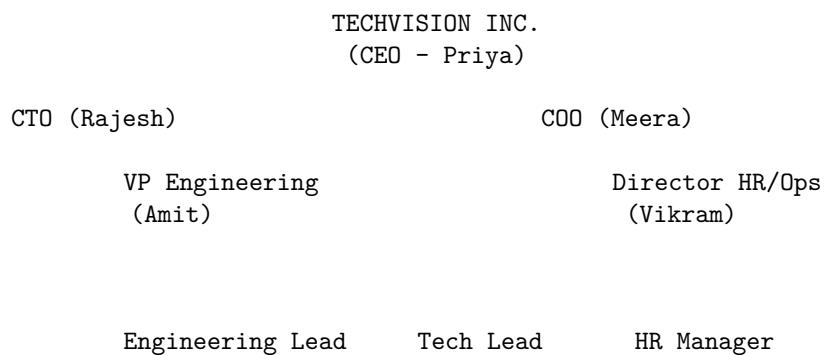
Developer marks task COMPLETED
QA Lead reviews (Code Review Approval - Level 1)
Tech Lead approves (Technical Approval - Level 2)
Manager approves (Delivery Approval - Level 3)
Task marked COMPLETED

4. ANALYTICS & REPORTING

Sprint completion tracked
Team productivity metrics
Project status dashboard
Executive reporting

2. ORGANIZATION HIERARCHY

2.1 Complete Org Chart



Design (Neha)	(Sanjay)	(Priya S.)
Senior Devs (5)	QA Lead (Deepak)	Finance Manager (Ramesh)
Junior Devs (12)	DevOps Engineer (Ravi)	Admin Staff (2)
VP Sales & Marketing (Mohit)		
Sales Manager (Surbhi)	Product Manager (Anil)	
Sales Reps (4)	Marketing Manager (Kavya)	
Business Dev (2)	Content Team (2)	

2.2 Firebase Organization Structure

```
{
  "tenantId": "techvision_inc",

  "organizations": {
    "root_node": {
      "id": "root_node",
      "name": "TechVision Inc.",
      "type": "company_root",
      "parent_id": null,
      "manager_id": "user_ceo_priya",
      "level": 0,
      "location": "Bangalore HQ",
      "children": ["executive_team", "engineering_div", "sales_marketing_div", "operations_div"],
      "metadata": {
        "founded_year": 2015,
        "industry": "Software Development",
        "total_employees": 120
      }
    },
    "executive_team": {
      "id": "executive_team",
      "name": "Executive Leadership",
      "type": "executive",
      "parent_id": "root_node"
    }
  }
}
```

```

    "parent_id": "root_node",
    "manager_id": "user_ceo_priya",
    "level": 1,
    "children": ["cto_office", "coo_office"],
    "members": ["user_ceo_priya", "user_cto_rajesh", "user_coo_meera"],
    "budget": 0,
    "metadata": {}
  },
  "engineering_div": {
    "id": "engineering_div",
    "name": "Engineering Division",
    "type": "division",
    "parent_id": "root_node",
    "manager_id": "user_cto_rajesh",
    "level": 1,
    "location": "Bangalore HQ",
    "children": ["backend_team", "frontend_team", "qa_team", "devops_team"],
    "metadata": {
      "employees": 35,
      "projects_active": 8,
      "sprint_cycle": "2_weeks"
    }
  },
  "backend_team": {
    "id": "backend_team",
    "name": "Backend Engineering Team",
    "type": "team",
    "parent_id": "engineering_div",
    "manager_id": "user_lead_amit",
    "level": 2,
    "location": "Bangalore HQ",
    "children": ["backend_seniors", "backend_juniors"],
    "members": ["user_lead_amit", "user_lead_sanjay", "user_senior_dev_1", "user_senior_de"],
    "metadata": {
      "tech_stack": ["Node.js", "Python", "PostgreSQL"],
      "projects": ["Payment System", "API Gateway", "User Service"]
    }
  },
  "frontend_team": {
    "id": "frontend_team",
    "name": "Frontend & Design Team",
    "type": "team",
    "parent_id": "engineering_div",

```

```

    "manager_id": "user_lead_neha",
    "level": 2,
    "location": "Bangalore HQ",
    "members": ["user_lead_neha", "user_designer_1", "user_designer_2"],
    "metadata": {
        "tech_stack": ["Flutter", "React", "Figma"],
        "projects": ["Mobile App", "Web Portal"]
    }
},
{
    "qa_team": {
        "id": "qa_team",
        "name": "QA & Testing Team",
        "type": "team",
        "parent_id": "engineering_div",
        "manager_id": "user_lead_deepak",
        "level": 2,
        "location": "Bangalore HQ",
        "members": ["user_lead_deepak", "user_qa_engineer_1", "user_qa_engineer_2"],
        "metadata": {
            "focus": ["Functional", "Performance", "Security"],
            "automation_coverage": "65%"
        }
    },
    "sales_marketing_div": {
        "id": "sales_marketing_div",
        "name": "Sales & Marketing Division",
        "type": "division",
        "parent_id": "root_node",
        "manager_id": "user_vp_sales_mohit",
        "level": 1,
        "location": "Bangalore HQ",
        "children": ["sales_team", "marketing_team"],
        "metadata": {
            "employees": 20,
            "quarterly_revenue_target": "500L",
            "new_leads_monthly": 50
        }
    },
    "sales_team": {
        "id": "sales_team",
        "name": "Sales Team",
        "type": "team",
        "parent_id": "sales_marketing_div",

```

```

    "manager_id": "user_manager_surbhi",
    "level": 2,
    "location": "Bangalore HQ",
    "members": ["user_manager_surbhi", "user_sales_rep_1", "user_sales_rep_2"],
    "metadata": {
        "regions": ["South India", "West India"],
        "pipeline_value": "2Cr"
    }
},
{
    "operations_div": {
        "id": "operations_div",
        "name": "Operations & HR Division",
        "type": "division",
        "parent_id": "root_node",
        "manager_id": "user_coo_meera",
        "level": 1,
        "location": "Bangalore HQ",
        "children": ["hr_team", "finance_team", "admin_team"],
        "metadata": {
            "employees": 18,
            "focus": "Operational Excellence"
        }
    },
    "hr_team": {
        "id": "hr_team",
        "name": "HR & People Operations",
        "type": "team",
        "parent_id": "operations_div",
        "manager_id": "user_manager_vikram",
        "level": 2,
        "location": "Bangalore HQ",
        "members": ["user_manager_vikram", "user_hr_staff_1", "user_hr_staff_2"],
        "metadata": {
            "attrition_rate": "8%",
            "hiring_active_roles": 5
        }
    }
}

```

3. DESIGNATIONS & ROLES SYSTEM

3.1 Complete Designation Hierarchy

```
{  
    "tenantId": "techvision_inc",  
  
    "designations": {  
        "ceo": {  
            "id": "ceo",  
            "name": "Chief Executive Officer",  
            "hierarchy_level": 1,  
            "reports_to": [],  
            "responsibilities": [  
                "Company strategic direction",  
                "Executive decision making",  
                "Stakeholder management",  
                "Board interactions"  
            ],  
            "permissions": [  
                "create_organization",  
                "edit_organization",  
                "delete_organization",  
                "create_task",  
                "assign_task",  
                "approve_task",  
                "complete_task",  
                "view_all_tasks",  
                "view_analytics",  
                "export_data",  
                "manage_users",  
                "manage_designations",  
                "manage_forms",  
                "manage_workflows",  
                "delete_tasks",  
                "bulk_operations",  
                "system_settings"  
            ],  
            "screen_access": ["developer", "admin", "manager", "employee"],  
            "default_roles": ["admin", "manager", "developer"],  
            "can_approve": true,  
            "approval_authority_level": 99,  
            "requires_approval_to_create": false,  
            "sla_task_completion_days": 1  
        },  
    },  
}
```

```

"cto": {
    "id": "cto",
    "name": "Chief Technology Officer",
    "hierarchy_level": 1,
    "reports_to": ["ceo"],
    "responsibilities": [
        "Engineering strategy",
        "Technology decisions",
        "Engineering team leadership",
        "Technical roadmap"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "approve_task",
        "view_all_tasks",
        "view_team_tasks",
        "view_analytics",
        "export_data",
        "manage_engineering_resources",
        "manage_technology_stack",
        "technical_approval_override"
    ],
    "screen_access": ["admin", "manager", "developer"],
    "default_roles": ["admin", "manager", "developer"],
    "can_approve": true,
    "approval_authority_level": 90,
    "requires_approval_to_create": false,
    "sla_task_completion_days": 2
},

"coo": {
    "id": "coo",
    "name": "Chief Operations Officer",
    "hierarchy_level": 1,
    "reports_to": ["ceo"],
    "responsibilities": [
        "Operations management",
        "HR strategy",
        "Finance oversight",
        "Administrative operations"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "approve_task",

```

```

        "view_all_tasks",
        "view_analytics",
        "manage_hr",
        "manage_finance",
        "export_data"
    ],
    "screen_access": ["admin", "manager", "employee"],
    "default_roles": ["admin", "manager"],
    "can_approve": true,
    "approval_authority_level": 85,
    "requires_approval_to_create": false,
    "sla_task_completion_days": 2
},
"vp": {
    "id": "vp",
    "name": "Vice President / Division Head",
    "hierarchy_level": 2,
    "reports_to": ["ceo", "cto", "coo"],
    "responsibilities": [
        "Division management",
        "Team leadership",
        "Division strategy",
        "Cross-team coordination"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "approve_task",
        "view_team_tasks",
        "view_division_analytics",
        "manage_team_structure",
        "create_sub_team",
        "export_division_data"
    ],
    "screen_access": ["admin", "manager"],
    "default_roles": ["admin", "manager"],
    "can_approve": true,
    "approval_authority_level": 70,
    "requires_approval_to_create": false,
    "sla_task_completion_days": 3,
    "delegates_to": ["manager", "team_lead"]
},
"engineering_lead": {
    "id": "engineering_lead",

```

```

    "name": "Engineering Team Lead / Manager",
    "hierarchy_level": 3,
    "reports_to": ["vp", "cto"],
    "responsibilities": [
        "Team management",
        "Code review oversight",
        "Task assignment",
        "Technical guidance",
        "Team performance tracking"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "approve_task_level_1",
        "view_team_tasks",
        "view_team_analytics",
        "create_sprint",
        "code_review_approval",
        "technical_approval"
    ],
    "screen_access": ["manager", "employee"],
    "default_roles": ["manager"],
    "can_approve": true,
    "approval_authority_level": 50,
    "requires_approval_to_create": false,
    "sla_task_completion_days": 5,
    "approval_chain_position": "level_2_technical",
    "delegates_to": ["senior_engineer", "qa_lead"]
},
"senior_engineer": {
    "id": "senior_engineer",
    "name": "Senior Software Engineer",
    "hierarchy_level": 4,
    "reports_to": ["engineering_lead", "team_lead"],
    "responsibilities": [
        "Complex feature development",
        "Code quality assurance",
        "Mentoring junior developers",
        "Architectural decisions",
        "Performance optimization"
    ],
    "permissions": [
        "create_task",
        "complete_task",
        "view_assigned_tasks",

```

```

    "code_review",
    "mentor_junior_developers",
    "technical_decision_making",
    "architecture_design",
    "view_team_analytics"
],
"screen_access": ["employee", "developer"],
"default_roles": ["developer"],
"can_approve": false,
"requires_approval_to_create": false,
"sla_task_completion_days": 7,
"delegates_to": ["junior_engineer"]
},

"junior_engineer": {
    "id": "junior_engineer",
    "name": "Junior Software Engineer",
    "hierarchy_level": 5,
    "reports_to": ["senior_engineer", "engineering_lead"],
    "responsibilities": [
        "Feature development",
        "Bug fixes",
        "Test writing",
        "Learning & development",
        "Code collaboration"
    ],
    "permissions": [
        "view_assigned_tasks",
        "complete_task",
        "create_code_review",
        "comment_on_tasks",
        "view_own_analytics"
    ],
    "screen_access": ["employee"],
    "default_roles": ["employee"],
    "can_approve": false,
    "requires_approval_to_create": false,
    "sla_task_completion_days": 10
},

"qa_lead": {
    "id": "qa_lead",
    "name": "QA Lead / Test Manager",
    "hierarchy_level": 3,
    "reports_to": ["engineering_lead", "vp"],
    "responsibilities": [

```

```

        "QA team management",
        "Test case creation",
        "Bug tracking",
        "Quality assurance",
        "Test automation strategy"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "view_team_tasks",
        "qa_approval",
        "create_bug_report",
        "manage_test_cases",
        "quality_metrics"
    ],
    "screen_access": ["manager", "employee"],
    "default_roles": ["manager"],
    "can_approve": true,
    "approval_authority_level": 40,
    "requires_approval_to_create": false,
    "approval_chain_position": "level_1_qa",
    "sla_task_completion_days": 5
},
{
    "qa_engineer": {
        "id": "qa_engineer",
        "name": "QA Engineer / Tester",
        "hierarchy_level": 4,
        "reports_to": ["qa_lead", "engineering_lead"],
        "responsibilities": [
            "Test execution",
            "Bug identification",
            "Test case documentation",
            "Regression testing",
            "Performance testing"
        ],
        "permissions": [
            "view_assigned_tasks",
            "complete_task",
            "create_bug_report",
            "view_team_analytics"
        ],
        "screen_access": ["employee"],
        "default_roles": ["employee"],
        "can_approve": false,
        "requires_approval_to_create": false,
        "approval_authority_level": 20
    }
}

```

```

    "sla_task_completion_days": 7
  },
  "product_manager": {
    "id": "product_manager",
    "name": "Product Manager",
    "hierarchy_level": 3,
    "reports_to": ["vp", "cto"],
    "responsibilities": [
      "Product strategy",
      "Feature prioritization",
      "Customer requirements",
      "Roadmap planning",
      "Stakeholder management"
    ],
    "permissions": [
      "create_task",
      "assign_task",
      "view_all_tasks",
      "product_decision",
      "roadmap_management",
      "export_data"
    ],
    "screen_access": ["manager", "employee"],
    "default_roles": ["manager"],
    "can_approve": true,
    "approval_authority_level": 45,
    "requires_approval_to_create": false,
    "approval_chain_position": "level_1_product"
  },
  "designer": {
    "id": "designer",
    "name": "UX/UI Designer",
    "hierarchy_level": 4,
    "reports_to": ["engineering_lead", "product_manager"],
    "responsibilities": [
      "UI/UX design",
      "Design systems",
      "User research",
      "Prototyping",
      "Design documentation"
    ],
    "permissions": [
      "create_task",
      "view_assigned_tasks",

```

```

        "complete_task",
        "design_approval",
        "view_design_system"
    ],
    "screen_access": ["employee", "developer"],
    "default_roles": ["employee"],
    "can_approve": false,
    "requires_approval_to_create": false
},

"sales_manager": {
    "id": "sales_manager",
    "name": "Sales Manager",
    "hierarchy_level": 3,
    "reports_to": ["vp"],
    "responsibilities": [
        "Sales team management",
        "Pipeline management",
        "Deal closure",
        "Customer relationships",
        "Sales strategy"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "view_team_tasks",
        "view_sales_analytics",
        "customer_management",
        "deal_approval"
    ],
    "screen_access": ["manager", "employee"],
    "default_roles": ["manager"],
    "can_approve": true,
    "approval_authority_level": 50
},

"sales_rep": {
    "id": "sales_rep",
    "name": "Sales Representative",
    "hierarchy_level": 4,
    "reports_to": ["sales_manager", "vp"],
    "responsibilities": [
        "Lead generation",
        "Customer acquisition",
        "Deal management",
        "Customer support",
    ]
}

```

```

        "Sales reporting"
    ],
    "permissions": [
        "view_assigned_tasks",
        "complete_task",
        "customer_interaction",
        "create_lead",
        "pipeline_update"
    ],
    "screen_access": ["employee"],
    "default_roles": ["employee"],
    "can_approve": false,
    "requires_approval_to_create": false
},

"hr_manager": {
    "id": "hr_manager",
    "name": "HR Manager",
    "hierarchy_level": 3,
    "reports_to": ["coo", "vp"],
    "responsibilities": [
        "Recruitment",
        "Employee relations",
        "Performance management",
        "Training & development",
        "Compliance"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "view_hr_tasks",
        "manage_users",
        "recruitment_approval",
        "employee_data_access"
    ],
    "screen_access": ["manager", "employee"],
    "default_roles": ["manager"],
    "can_approve": true,
    "approval_authority_level": 50
},

"finance_manager": {
    "id": "finance_manager",
    "name": "Finance Manager",
    "hierarchy_level": 3,
    "reports_to": ["coo"],

```

```

    "responsibilities": [
        "Financial planning",
        "Budget management",
        "Expense approval",
        "Financial reporting",
        "Cost control"
    ],
    "permissions": [
        "create_task",
        "assign_task",
        "view_finance_tasks",
        "expense_approval",
        "budget_management",
        "financial_reports"
    ],
    "screen_access": ["manager", "employee"],
    "default_roles": ["manager"],
    "can_approve": true,
    "approval_authority_level": 55,
    "approval_chain_position": "level_2_finance"
}
}
}

```

4. USER REGISTRATION FORMS

4.1 Initial Registration Form (Public)

```

{
    "formId": "user_registration_public",
    "name": "User Registration",
    "description": "Initial registration form for new TechVision employees",
    "version": 1,
    "visibility": "public",
    "fields": [
        {
            "id": "fullName",
            "type": "text",
            "label": "Full Name",
            "required": true,
            "placeholder": "e.g., Rajesh Kumar",
            "validation": "^[a-zA-Z\\s]{3,50}$",
            "hint": "Your full legal name",
            "minLength": 3,
            "maxLength": 50
        }
    ]
}

```

```

},
{
  "id": "email",
  "type": "email",
  "label": "Work Email",
  "required": true,
  "placeholder": "your.name@techvision.com",
  "validation": "^[^@]+@techvision\\.com$",
  "hint": "Use your company email address",
  "unique": true
},
{
  "id": "password",
  "type": "password",
  "label": "Password",
  "required": true,
  "placeholder": "Min 12 chars: uppercase, lowercase, number, special char",
  "validation": "^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[!@#$%^&*])[A-Za-z0-9!@#$%^&*]{12}",
  "hint": "Minimum 12 characters with mixed case, number, and special character",
  "minLength": 12
},
{
  "id": "confirmPassword",
  "type": "password",
  "label": "Confirm Password",
  "required": true,
  "placeholder": "Re-enter your password",
  "matchField": "password",
  "hint": "Must match the password above"
},
{
  "id": "mobileNumber",
  "type": "phone",
  "label": "Mobile Number",
  "required": true,
  "placeholder": "+91 XXXXX XXXXX",
  "validation": "^\\+91[6-9]\\d{9}$",
  "hint": "10-digit Indian mobile number with country code"
},
{
  "id": "designation",
  "type": "dropdown",
  "label": "Your Designation",
  "required": true,
  "hint": "Select your job position in TechVision",
  "dataSource": "firebase",

```

```

    "collection": "designations",
    "displayField": "name",
    "valueField": "id",
    "filter": { "status": "active" },
    "sortBy": "hierarchy_level"
},
{
    "id": "department",
    "type": "dropdown",
    "label": "Department",
    "required": true,
    "hint": "Your primary department",
    "dataSource": "firebase",
    "collection": "organizations",
    "displayField": "name",
    "filter": { "type": "team" },
    "sortBy": "name"
},
{
    "id": "reportingManager",
    "type": "userPicker",
    "label": "Reporting Manager",
    "required": true,
    "hint": "Who is your direct manager?",
    "dataSource": "firebase",
    "collection": "users",
    "filter": { "roles": ["manager", "admin"] },
    "searchFields": ["fullName", "email"]
},
{
    "id": "officeLocation",
    "type": "dropdown",
    "label": "Office Location",
    "required": true,
    "options": [
        { "label": "Bangalore HQ", "value": "bangalore" },
        { "label": "Delhi Office", "value": "delhi" }
    ]
},
{
    "id": "joiningDate",
    "type": "date",
    "label": "Joining Date",
    "required": true,
    "hint": "When are you joining TechVision?"
}

```

```

{
  "id": "terms",
  "type": "checkbox",
  "label": "I agree to the Terms of Service and Privacy Policy",
  "required": true
},
],
"workflow": {
  "onValidate": ["validateEmail", "validatePassword", "validateMobileNumber"],
  "onSubmit": "registerUser",
  "nextAction": "submitForApproval"
},
"validation_rules": {
  "validateEmail": {
    "type": "unique_check",
    "collection": "users",
    "field": "email",
    "message": "This email is already registered"
  },
  "validatePassword": {
    "type": "pattern_match",
    "pattern": "^(?=.*[A-Z])(?=.*[a-z])(?=.*[0-9])(?=.*[!@#$%^&*])",
    "message": "Password must contain uppercase, lowercase, number, and special character"
  },
  "validateMobileNumber": {
    "type": "unique_check",
    "collection": "users",
    "field": "mobileNumber",
    "message": "This phone number is already registered"
  }
}
}

```

4.2 Registration Approval Form (Manager)

```

{
  "formId": "registration_approval",
  "name": "Registration Approval Form",
  "description": "Manager approves new user registration",
  "version": 1,
  "visibility": "manager_only",
  "fields": [
    {
      "id": "applicantName",

```

```

    "type": "text",
    "label": "Applicant Name",
    "required": false,
    "readOnly": true,
    "bindsTo": "registration.fullName"
},
{
    "id": "applicantEmail",
    "type": "email",
    "label": "Applicant Email",
    "required": false,
    "readOnly": true,
    "bindsTo": "registration.email"
},
{
    "id": "proposedDesignation",
    "type": "text",
    "label": "Proposed Designation",
    "required": false,
    "readOnly": true,
    "bindsTo": "registration.designation"
},
{
    "id": "department",
    "type": "text",
    "label": "Department",
    "required": false,
    "readOnly": true,
    "bindsTo": "registration.department"
},
{
    "id": "joiningDate",
    "type": "date",
    "label": "Joining Date",
    "required": false,
    "readOnly": true,
    "bindsTo": "registration.joiningDate"
},
{
    "id": "approvalDecision",
    "type": "radio",
    "label": "Approval Decision",
    "required": true,
    "options": [
        { "label": "Approve Registration", "value": "approved" },
        { "label": "Request More Information", "value": "pending_info" },

```

```

        { "label": "Reject Registration", "value": "rejected" }
    ],
},
{
    "id": "comments",
    "type": "textarea",
    "label": "Comments / Feedback",
    "required": false,
    "placeholder": "Add any notes about this application",
    "showIf": {
        "field": "approvalDecision",
        "operator": "!=",
        "value": "approved"
    }
},
{
    "id": "actualDesignation",
    "type": "dropdown",
    "label": "Confirm Designation",
    "required": true,
    "showIf": { "field": "approvalDecision", "operator": "==", "value": "approved" },
    "dataSource": "firebase",
    "collection": "designations",
    "displayField": "name"
},
{
    "id": "costCenter",
    "type": "text",
    "label": "Cost Center Code",
    "required": true,
    "showIf": { "field": "approvalDecision", "operator": "==", "value": "approved" },
    "placeholder": "e.g., CC001-Engineering"
}
],
{
    "workflow": {
        "onSubmit": "processApproval",
        "onApprove": ["createUserAccount", "sendWelcomeEmail", "notifyHR"],
        "onReject": ["sendRejectionEmail", "archiveApplication"],
        "onPending": ["requestAdditionalInfo", "assignToHRTeam"]
    }
}

```

5. EMPLOYEE WORKFLOWS

5.1 Engineering Team Workflow

5.1.1 Task Creation Workflow

ENGINEERING TASK WORKFLOW

Step 1: TASK CREATION (Engineering Lead - Amit)

Amit logs in to Manager Screen
Clicks "Create New Task"
Fills engineering task form:
 Title: "Implement Payment Gateway Integration"
 Description: "Integrate Stripe payment gateway..."
 Project: "Mobile App v2.0"
 Story Points: 8
 Sprint: "Sprint 24 (Jan 1-14)"
 Priority: "High"
 Technical Stack: ["Node.js", "PostgreSQL"]
 Estimated Hours: 40
 QA Estimation: 8 hours
Submits form
Status: Task created (PENDING_ASSIGNMENT)

Step 2: TASK ASSIGNMENT (Amit)

Task shows in "Tasks to Assign" queue
Amit clicks task
Selects assignee: Sanjay (Senior Developer)
Confirms assignment
Task Status: ASSIGNED
Notification: Sanjay receives "New task assigned"
Task appears in Sanjay's EMPLOYEE SCREEN

Step 3: TASK START (Sanjay - Senior Developer)

Sanjay logs in to EMPLOYEE SCREEN
Sees "Implement Payment Gateway Integration" in My Tasks
Clicks task → View Details
Clicks "Start Work"
Task Status: IN_PROGRESS
Tracked: Started At: 2024-01-05 09:30 AM
Daily standup can include this task

Step 4: WORK IN PROGRESS

Sanjay works on implementation
Updates progress:
 Jan 6: "Completed Stripe API integration - 20 hours used"
 Jan 7: "Testing basic flow - 25 hours used"

Jan 8: "Ready for code review - 38 hours used"
Adds code review comment:
"Pull request #456 ready for review"
Task shows: Progress = 85%
Subtask: "Code Review" created automatically

Step 5: CODE REVIEW (QA Lead - Deepak)
Deepak (QA Lead) sees "Code Review" subtask
Reviews pull request
Deepak approves: "Code quality excellent, ready for QA"
Subtask Status: APPROVED
Notification: Sanjay receives approval
Main task waits for QA

Step 6: QA TESTING (QA Engineer - Priya Q)
Priya Q (QA Engineer) sees task in QA Queue
Starts testing implementation
Tests:
Test Case: "Valid payment flow" PASS
Test Case: "Invalid card handling" PASS
Test Case: "Network failure handling" PASS
Test Case: "Concurrent payments" PASS
Priya marks: "All tests passed"
Creates test report
Task Status: AWAITING_FINAL REVIEW

Step 7: FINAL APPROVAL (Amit - Engineering Lead)
Amit sees task in "Pending Approvals" (Manager Screen)
Clicks "Review"
Reviews:
Code Quality: Approved
Test Results: All passed
Performance: Meets targets
Documentation: Complete
Clicks "APPROVE"
Adds comment: "Excellent work, ready for production"
Task Status: COMPLETED
Notification: Sanjay sees "Task Approved"
Task removed from active list

5.1.2 Daily Standup Workflow

```
{  
  "workflowId": "daily_standup_engineering",  
  "name": "Daily Engineering Standup",  
  "description": "Track daily progress on engineering tasks",
```

```

"triggers": [
  {
    "type": "scheduled",
    "time": "09:00 AM",
    "frequency": "daily",
    "days": ["Monday", "Tuesday", "Wednesday", "Thursday", "Friday"]
  }
], 

"standupFormId": "engineering_standup",

"participants": [
  {
    "role": "engineering_lead",
    "responsibility": "Lead standup, assign blockers"
  },
  {
    "role": "senior_engineer",
    "responsibility": "Report on progress, help unblock juniors"
  },
  {
    "role": "junior_engineer",
    "responsibility": "Report progress, raise blockers"
  },
  {
    "role": "qa_engineer",
    "responsibility": "Report testing status"
  }
], 

"fields_per_person": [
  {
    "id": "yesterday_completed",
    "type": "textarea",
    "label": "What did you complete yesterday?",
    "required": true
  },
  {
    "id": "today_plan",
    "type": "textarea",
    "label": "What will you do today?",
    "required": true
  },
  {
    "id": "blockers",
    "type": "text"
  }
]

```

```

    "type": "textarea",
    "label": "Any blockers or issues?",
    "required": false
},
{
  "id": "time_used_yesterday",
  "type": "number",
  "label": "Hours worked yesterday",
  "required": true,
  "min": 0,
  "max": 24
}
],
}

"post_standup_workflow": {
  "aggregate_report": "Send to Engineering Lead",
  "highlight_blockers": "Notify relevant team leads",
  "update_progress": "Update all task statuses"
}
}
}

```

5.2 Sales Team Workflow

5.2.1 Lead Management Workflow

SALES LEAD MANAGEMENT WORKFLOW

Step 1: LEAD CREATION (Sales Rep - Neha)

Neha logs in to EMPLOYEE SCREEN
 Clicks "New Lead"
 Fills Lead Form:
 Company Name: "ABC Enterprises Pvt Ltd"
 Contact Person: "Mr. Rajesh Sharma"
 Email: "rajesh@abc-enterprises.com"
 Phone: "+91-9876543210"
 Industry: "E-commerce"
 Company Size: "1000+ employees"
 Current Solution: "Custom built system"
 Budget Range: "50-100 lakhs"
 Requirements: "Mobile app + Web platform"
 Creates Lead
 Status: LEAD_CREATED

Step 2: LEAD ASSIGNMENT (Sales Manager - Surbhi)

Surbhi sees new lead in "Unassigned Leads" queue
 Reviews lead details

Clicks "Assign to Sales Rep"
Selects: Neha (appropriate rep for industry)
Sends message: "High-value lead, prioritize follow-up"
Lead Status: ASSIGNED
Notification: Neha receives assignment

Step 3: LEAD QUALIFICATION (Neha)
Neha creates task: "Qualify ABC Enterprises Lead"
Calls Mr. Rajesh Sharma
Qualification checklist:
 Budget confirmed: 75 lakhs
 Timeline: Want solution in 3 months
 Decision maker: Mr. Rajesh is the CTO
 Technical Requirements: Clear and documented
 Competition: No active RFP yet
Marks task: COMPLETED
Lead Status: QUALIFIED
Notification: Surbhi sees lead is qualified

Step 4: PROPOSAL CREATION (Neha)
Neha creates task: "Create Proposal for ABC Enterprises"
Uses proposal template
Customizes:
 Solution architecture
 Timeline & milestones
 Pricing: 75 lakhs
 Payment terms: 30-30-40
 Team composition
Saves draft
Lead Status: PROPOSAL_DRAFT
Task waits for sales manager approval

Step 5: PROPOSAL APPROVAL (Surbhi)
Surbhi sees "Proposal Pending Approval" task
Reviews proposal:
 Scope: Matches requirements
 Pricing: Competitive
 Timeline: Feasible
 Team: Appropriate resources
Approves: "Looks good, send to client"
Task Status: APPROVED
Notification: Neha gets approval
Lead Status: PROPOSAL_SENT

Step 6: CLIENT FOLLOW-UP (Neha)
Neha sends proposal to Mr. Rajesh

Creates task: "ABC Enterprises - Follow-up (Day 3)"
Creates reminder: "Follow up if no response"
Lead Status: AWAITING_FEEDBACK
Days 1-3: No response
Day 4: Neha calls → Client says "Reviewing internally"
Task updates: FOLLOW_UP_SCHEDULED

Step 7: NEGOTIATIONS (Neha & Surbhi)
Client responds: "We can do 70L, not 75L"
Neha creates task: "Negotiate pricing with ABC"
Surbhi reviews: "Can we do 72L + 20% discount first year?"
Neha negotiates → Client agrees to 72L + 15% first-year discount
Surbhi approves: "Good deal, proceed to contract"
Lead Status: DEAL AGREED

Step 8: CONTRACT APPROVAL (Sales Manager - Surbhi)
Surbhi reviews final contract
Checks: Payment terms, deliverables, SLAs
Approves: "Send to Legal for final review"
Contract sent to Legal Team (task created for them)
Lead Status: CONTRACT_PENDING_LEGAL

Step 9: DEAL CLOSURE (Surbhi)
Legal approves contract
Client signs and returns
Surbhi marks: "DEAL WON"
Updates lead with:
Contract value: 72 lakhs
First payment: 24 lakhs (received)
Timeline: 90 days
Primary contacts: Rajesh, CFO (Finance contact)
Creates task: "Handoff to Project Manager"
Notifies Engineering: "New project - ABC Enterprises"
Lead Status: CLOSED WON

5.3 HR Team Workflow

5.3.1 Recruitment Workflow

HR RECRUITMENT WORKFLOW

Step 1: JOB OPENING REQUEST (Amit - Engineering Lead)
Amit logs in to MANAGER SCREEN
Clicks "Request New Hire"
Fills Job Opening Form:
Position: "Senior Backend Engineer"

Department: "Engineering"
Reporting to: "Amit (Team Lead)"
Urgency: "High"
Budget: "8-12 LPA (approx)"
Required Skills: ["Node.js", "PostgreSQL", "System Design"]
Preferred Skills: ["AWS", "Docker", "Kubernetes"]
Experience: "4-6 years"
Notice Period: "30 days max"
Job Description: [Detailed description]
Submits form
Status: PENDING_HR_APPROVAL

Step 2: HR REVIEW & APPROVAL (HR Manager - Vikram)
Vikram sees "New Job Opening Request" in HR queue
Reviews request:
Role matches org structure:
Budget approved by Finance:
Reporting structure valid:
Skills realistic for market:
Approves: "Approved for posting"
Task Status: APPROVED
Notification: Amit sees approval
Job Opening Status: POSTING_IN_PROGRESS

Step 3: JOB POSTING
HR publishes to:
TechVision careers page
LinkedIn
Indeed
Internal network
Referral program
Creates task: "Job Application Review"
Opens application collection
Job Opening Status: ACTIVE (RECEIVING_APPLICATIONS)

Step 4: CANDIDATE SCREENING (HR Staff - Priya S.)
Candidates start applying (automatically captured in Firestore)
Priya receives applications daily
Screens each application:
Resume review: Matches required skills?
Experience: 4-6 years?
Notice period: 30 days?
Current location: Bangalore/Delhi or willing to relocate?
Salary expectation: Within budget?
Marks candidates:
Interview_Ready: Schedule for next round

Maybe: Interesting but missing something
Reject: Not matching criteria
Task: "Send Interview Invites"

Step 5: TECHNICAL INTERVIEW (Amit - Engineering Lead)
Scheduled candidates: 5 candidates
Interview 1: Candidate A - Codility test + Q&A
Interview 2: Candidate B - System design discussion
Interview 3: Candidate C - Live coding exercise
Candidate A: Excellent (Rating 4.5/5)
Candidate B: Good (Rating 4/5)
Candidate C: Average (Rating 2.5/5)
Amit submits: "Recommend A and B for next round"
Task: "Send to HR for Offer Round"

Step 6: HR ROUNDS (Vikram - HR Manager)
Interviews Candidates A & B:
Culture fit assessment
Salary negotiation
Joining date flexibility
Benefits discussion
Candidate A: Excellent fit, asks 10 LPA
Candidate B: Good fit, asks 9.5 LPA
Vikram: Recommends offering to both
Task: "Create Offers"

Step 7: OFFER CREATION (Vikram)
Creates Offer to Candidate A:
Position: Senior Backend Engineer
CTC: 10 LPA
Joining Date: Feb 1, 2024
Reporting to: Amit
Benefits: Standard package
Department: Engineering
Sends offer letter via email
Task: "Await Candidate Response"
Offer Status: SENT

Step 8: OFFER ACCEPTANCE (Candidate A)
Candidate accepts offer
Submits signed offer letter
Provides:
Background check approval
PAN number
Bank account details
Emergency contact

Task: "Onboarding Preparation"
 Offer Status: ACCEPTED

 Step 9: ONBOARDING WORKFLOW (Priya S. - HR Staff)
 Priya creates comprehensive onboarding checklist
 Pre-joining tasks:
 System access: Request IT
 Equipment: Laptop, peripherals (Order)
 Workspace: Desk assignment (Facility)
 Orientation: Schedule with HR
 Team intro: Schedule with Amit
 Amit prepares:
 Team introduction slides
 Project overview
 Mentor assignment: Senior Engineer (Rajesh)
 Candidate joins on Feb 1
 First week: Orientation + Setup
 New Employee Status: ONBOARDED

6. DEPARTMENT-SPECIFIC WORKFLOWS

6.1 Marketing Team Task Workflow

```
{
  "workflowId": "marketing_campaign_workflow",
  "name": "Marketing Campaign Workflow",

  "stages": [
    {
      "stage": 1,
      "name": "Campaign Planning",
      "owner": "marketing_manager",
      "tasks": [
        {
          "taskType": "strategy",
          "description": "Define campaign objectives & KPIs",
          "duration_days": 3,
          "required_approvals": 1
        },
        {
          "taskType": "audience_research",
          "description": "Research target audience",
          "duration_days": 5,
          "required_approvals": 1
        },
        {
          "taskType": "content_plan",
          "description": "Create initial content plan",
          "duration_days": 2,
          "required_approvals": 1
        }
      ]
    }
  ]
}
```

```

    {
      "taskType": "budget_allocation",
      "description": "Allocate budget across channels",
      "duration_days": 2,
      "required_approvals": ["finance_manager", "manager"]
    }
  ],
},
{
  "stage": 2,
  "name": "Content Creation",
  "owner": "content_team",
  "tasks": [
    {
      "taskType": "copywriting",
      "description": "Write campaign copy",
      "duration_days": 5,
      "required_approvals": 1
    },
    {
      "taskType": "design",
      "description": "Create visual assets",
      "duration_days": 7,
      "required_approvals": ["designer", "marketing_manager"]
    },
    {
      "taskType": "video_production",
      "description": "Produce video content (if needed)",
      "duration_days": 14,
      "required_approvals": ["marketing_manager"]
    }
  ]
},
{
  "stage": 3,
  "name": "Review & Approval",
  "owner": "marketing_manager",
  "tasks": [
    {
      "taskType": "content_review",
      "description": "Review all content",
      "duration_days": 2,
      "required_approvals": ["marketing_manager", "vp_sales_marketing"]
    },
    {
      "taskType": "legal_compliance",
      "description": "Check legal compliance of all content"
    }
  ]
}

```

```

        "description": "Legal review for compliance",
        "duration_days": 2,
        "required_approvals": ["legal_team"]
    }
]
},
{
    "stage": 4,
    "name": "Campaign Launch",
    "owner": "marketing_manager",
    "tasks": [
        {
            "taskType": "schedule_posts",
            "description": "Schedule social media posts",
            "duration_days": 1
        },
        {
            "taskType": "email_campaign",
            "description": "Setup and launch email campaign",
            "duration_days": 1
        },
        {
            "taskType": "ad_setup",
            "description": "Setup paid ads on platforms",
            "duration_days": 2
        }
    ]
},
{
    "stage": 5,
    "name": "Monitoring & Optimization",
    "owner": "marketing_manager",
    "tasks": [
        {
            "taskType": "daily_monitoring",
            "description": "Monitor campaign performance",
            "duration_days": 30,
            "frequency": "daily"
        },
        {
            "taskType": "optimization",
            "description": "Optimize underperforming channels",
            "duration_days": 30,
            "frequency": "weekly"
        }
    ]
}

```

```

    },
    {
      "stage": 6,
      "name": "Results & Reporting",
      "owner": "marketing_manager",
      "tasks": [
        {
          "taskType": "data_analysis",
          "description": "Analyze campaign results",
          "duration_days": 3
        },
        {
          "taskType": "report_creation",
          "description": "Create comprehensive report",
          "duration_days": 2
        },
        {
          "taskType": "executive_presentation",
          "description": "Present results to leadership",
          "duration_days": 1,
          "required_approvals": ["vp_sales_marketing", "ceo"]
        }
      ]
    }
  ]
}

```

6.2 Finance Team Expense Approval Workflow

```

{
  "workflowId": "expense_approval_workflow",
  "name": "Expense Approval Workflow",

  "triggers": [
    {
      "type": "on_expense_submission",
      "description": "When employee submits expense report"
    }
  ],

  "approval_levels": [
    {
      "level": 1,
      "name": "Department Manager Review",
      "approvers": ["immediate_manager"],
      "requirements": {

```

```

        "review_receipts": true,
        "verify_business_purpose": true,
        "check_policy_compliance": true
    },
    "sla_hours": 24,
    "can_approve_up_to": "10000",
    "actions": {
        "approve": ["mark_approved_level_1", "notify_finance"],
        "reject": ["request_revision", "notify_employee"],
        "escalate": ["send_to_director", "add_notes"]
    }
},
{
    "level": 2,
    "name": "Finance Manager Review",
    "approvers": ["finance_manager"],
    "requirements": {
        "budget_check": true,
        "cost_center_validation": true,
        "policy_audit": true
    },
    "sla_hours": 48,
    "can_approve_up_to": "50000",
    "actions": {
        "approve": ["process_payment", "update_ledger", "notify_employee"],
        "reject": ["request_revision", "notify_manager"],
        "escalate": ["send_to_coo", "flag_for_review"]
    }
},
{
    "level": 3,
    "name": "Director/VP Approval",
    "approvers": ["director", "vp"],
    "requirements": {
        "amount_threshold": "> 50000",
        "policy_exception": true
    },
    "sla_hours": 72,
    "can_approve_up_to": "500000",
    "actions": {
        "approve": ["process_payment", "notify_all", "archive"],
        "reject": ["deny_permanently", "notify_employee"]
    }
}
],

```

```

"escalation_rules": [
  {
    "trigger": "not_approved_after_hours",
    "after_hours": 24,
    "escalate_to": "department_head"
  },
  {
    "trigger": "not_approved_after_hours",
    "after_hours": 48,
    "escalate_to": "director"
  },
  {
    "trigger": "policy_violation",
    "escalate_to": "finance_manager"
  }
]
}

```

7. FORMS & DATA MODELS

7.1 Engineering Task Form

```

{
  "formId": "engineering_task_form",
  "name": "Create Engineering Task",
  "description": "Create a new engineering task/story",
  "version": 2,
  "fields": [
    {
      "id": "title",
      "type": "text",
      "label": "Task Title",
      "required": true,
      "placeholder": "e.g., Implement Payment Gateway Integration",
      "maxLength": 100
    },
    {
      "id": "description",
      "type": "richtext",
      "label": "Task Description",
      "required": true,
      "placeholder": "Detailed description of what needs to be done"
    },
    {
      "id": "acceptance_criteria",
      "type": "richtext",
      "label": "Acceptance Criteria"
    }
  ]
}

```

```

    "type": "textarea",
    "label": "Acceptance Criteria",
    "required": true,
    "placeholder": "List of acceptance criteria separated by newlines",
    "rows": 5
},
{
    "id": "project",
    "type": "dropdown",
    "label": "Project",
    "required": true,
    "dataSource": "firebase",
    "collection": "projects"
},
{
    "id": "sprint",
    "type": "dropdown",
    "label": "Sprint",
    "required": true,
    "dataSource": "firebase",
    "collection": "sprints",
    "filter": { "status": "active" }
},
{
    "id": "story_points",
    "type": "dropdown",
    "label": "Story Points",
    "required": true,
    "options": [1, 2, 3, 5, 8, 13, 21, 34]
},
{
    "id": "priority",
    "type": "dropdown",
    "label": "Priority",
    "required": true,
    "options": [
        { "label": "Low", "value": "low", "color": "green" },
        { "label": "Medium", "value": "medium", "color": "yellow" },
        { "label": "High", "value": "high", "color": "orange" },
        { "label": "Critical", "value": "critical", "color": "red" }
    ]
},
{
    "id": "assignee",
    "type": "userPicker",
    "label": "Assign To",

```

```

    "required": false,
    "dataSource": "firestore",
    "collection": "users",
    "filter": { "team": "engineering", "status": "active" }
},
{
  "id": "technical_stack",
  "type": "multiselect",
  "label": "Technical Stack Required",
  "required": true,
  "options": [
    "Node.js", "Python", "Java", "Go",
    "PostgreSQL", "MongoDB", "Redis",
    "React", "Vue", "Angular",
    "Docker", "Kubernetes", "AWS"
  ]
},
{
  "id": "estimated_hours",
  "type": "number",
  "label": "Estimated Hours",
  "required": true,
  "min": 1,
  "max": 160
},
{
  "id": "qa_estimation",
  "type": "number",
  "label": "QA Estimated Hours",
  "required": true,
  "min": 1,
  "max": 80
},
{
  "id": "dependencies",
  "type": "textarea",
  "label": "Dependencies",
  "required": false,
  "placeholder": "List any blockers or dependent tasks"
},
{
  "id": "attachments",
  "type": "fileupload",
  "label": "Attachments",
  "required": false,
  "maxFiles": 5
}

```

```

        "maxSize": "10MB",
        "allowedTypes": [".pdf", ".doc", ".docx", ".jpg", ".png"]
    }
]
}

```

7.2 Sales Lead Form

```

{
  "formId": "sales_lead_form",
  "name": "Create New Sales Lead",
  "description": "Create and track a new sales lead",
  "version": 1,
  "fields": [
    {
      "id": "company_name",
      "type": "text",
      "label": "Company Name",
      "required": true,
      "placeholder": "ABC Enterprises Pvt Ltd"
    },
    {
      "id": "contact_person",
      "type": "text",
      "label": "Contact Person Name",
      "required": true,
      "placeholder": "Mr. Rajesh Sharma"
    },
    {
      "id": "contact_email",
      "type": "email",
      "label": "Contact Email",
      "required": true
    },
    {
      "id": "contact_phone",
      "type": "phone",
      "label": "Contact Phone",
      "required": true
    },
    {
      "id": "company_website",
      "type": "url",
      "label": "Company Website",
      "required": false
    },
  ],
}

```

```
{
  "id": "industry",
  "type": "dropdown",
  "label": "Industry",
  "required": true,
  "options": [
    "E-commerce", "Finance", "Healthcare", "Education",
    "Manufacturing", "Retail", "Travel", "SaaS", "Other"
  ]
},
{
  "id": "company_size",
  "type": "dropdown",
  "label": "Company Size",
  "required": true,
  "options": [
    "1-50", "51-200", "201-500", "501-1000",
    "1001-5000", "5000+"
  ]
},
{
  "id": "current_solution",
  "type": "textarea",
  "label": "Current Solution",
  "required": false,
  "placeholder": "What are they currently using?"
},
{
  "id": "budget_range",
  "type": "dropdown",
  "label": "Budget Range",
  "required": true,
  "options": [
    "< 10L", "10-25L", "25-50L", "50-100L",
    "100-200L", "200L+"
  ]
},
{
  "id": "timeline",
  "type": "dropdown",
  "label": "Implementation Timeline",
  "required": true,
  "options": [
    "Urgent (1 month)", "Quick (1-3 months)", "Medium (3-6 months)",
    "Long-term (6+ months)", "Exploring options"
  ]
}
```

```

},
{
  "id": "pain_points",
  "type": "textarea",
  "label": "Identified Pain Points",
  "required": true,
  "placeholder": "What are their main challenges?"
},
{
  "id": "requirements",
  "type": "textarea",
  "label": "Initial Requirements",
  "required": false,
  "placeholder": "Mobile app, Web platform, Custom features, etc."
},
{
  "id": "decision_maker_name",
  "type": "text",
  "label": "Primary Decision Maker",
  "required": true
},
{
  "id": "decision_maker_designation",
  "type": "text",
  "label": "Decision Maker Designation",
  "required": true
},
{
  "id": "source",
  "type": "dropdown",
  "label": "Lead Source",
  "required": true,
  "options": [
    "Referral", "LinkedIn", "Website", "Event",
    "Cold Call", "Inbound Request", "Partner", "Other"
  ]
},
{
  "id": "notes",
  "type": "textarea",
  "label": "Additional Notes",
  "required": false
}
]
}

```

8. TASK WORKFLOWS

8.1 Complete Task Lifecycle

TASK LIFECYCLE IN TECHVISION

STATE 1: CREATED

Status: CREATED
Created By: Manager/Lead
Assigned To: Nobody yet
Created At: 2024-01-05 10:30 AM
Data: Title, Description, Story Points, etc.
Action: Manager assigns to team member

STATE 2: ASSIGNED

Status: ASSIGNED
Assigned To: Junior Dev (Ram)
Assigned At: 2024-01-05 10:45 AM
Assigned By: Engineering Lead (Amit)
Due Date: 2024-01-12
Priority: High
Notification: Assignee notified
Action: Assignee starts working or communicates blockers

STATE 3: IN_PROGRESS

Status: IN_PROGRESS
Started At: 2024-01-05 11:00 AM
Progress: 0%
Time Logged: 0 hours
Comments: Daily standup updates
Action: Continuous work and updates

STATE 3.5: BLOCKED (Optional)

Status: BLOCKED
Blocked By: Waiting for DB design from Sanjay
Blocked At: 2024-01-07 3:00 PM
Reason: "Cannot proceed without database schema"
Notify: Team Lead (to resolve blocker)
Action: Blocker resolution

STATE 4: AWAITING REVIEW

Status: AWAITING REVIEW

Completed By: Ram (Junior Dev)
Completed At: 2024-01-10 5:30 PM
Total Hours: 35 hours
Code Review Link: PR #456
QA Assignment: Sent to Deepak (QA Lead)
Action: QA reviews and tests

STATE 5: PENDING_APPROVAL
Status: PENDING_APPROVAL
Approvals Needed: 3
 Level 1: QA Lead (Deepak) - PENDING
 Level 2: Tech Lead (Sanjay) - PENDING
 Level 3: Engineering Manager (Amit) - PENDING
Approval Created At: 2024-01-11 10:00 AM
Approval Expires: 2024-01-13 10:00 AM (48 hours)
Action: Each approver reviews their level

APPROVAL FLOW:

QA Lead reviews (Level 1)
 Tests all acceptance criteria: PASS
 Performance testing: PASS
 Approves: "Quality excellent"
 Status: APPROVED

Tech Lead reviews (Level 2)
 Code quality: Good
 Architecture: Aligned
 Approves: "Ready for production"
 Status: APPROVED

Engineering Manager reviews (Level 3)
 Business value: Excellent
 Timeline: On schedule
 Team impact: Positive
 Approves: "Approved for production"
 Status: APPROVED

STATE 6: COMPLETED
Status: COMPLETED
Completed At: 2024-01-11 4:00 PM
Final Approver: Amit (Engineering Manager)
All Approvals: APPROVED
Time Logged: 35 hours
Story Points: 8
Velocity Contribution: +8 points
Release: Scheduled for Jan 15 release

Archived: Auto-archived after 30 days of completion

9. APPROVAL CHAINS

9.1 Engineering Task Approval Chain

```
{  
    "workflowId": "engineering_task_approval",  
    "name": "Engineering Task Approval Workflow",  
    "description": "Multi-level approval for engineering tasks",  
  
    "approval_triggers": [  
        {  
            "trigger": "task_completed",  
            "condition": "status == AWAITING REVIEW",  
            "action": "initiate_approval_chain"  
        }  
    ],  
  
    "approval_levels": [  
        {  
            "level": 1,  
            "name": "QA Lead Review",  
            "approver_designation": "qa_lead",  
            "approver_id_source": "task.qa_lead_id",  
            "responsibilities": [  
                "Test case verification",  
                "Functional testing",  
                "Performance testing",  
                "Quality assurance sign-off"  
            ],  
            "required_fields": [  
                "test_results",  
                "test_coverage_percentage",  
                "performance_metrics"  
            ],  
            "sla_hours": 24,  
            "can_approve": true,  
            "can_reject": true,  
            "rejection_action": "send_back_to_developer",  
            "approval_comment_required": false,  
            "notification": {  
                "channels": ["in_app", "email"],  
                "message_template": "engineering_qa_approval_needed"  
            }  
        }  
    ]  
}
```

```

},
{
  "level": 2,
  "name": "Technical Lead Review",
  "approver_designation": "engineering_lead",
  "approver_id_source": "task.tech_lead_id",
  "wait_for": "level_1_approval",
  "responsibilities": [
    "Code quality review",
    "Architecture compliance",
    "Best practices check",
    "Technical decision making"
  ],
  "required_fields": [
    "code_review_link",
    "architecture_compliance",
    "tech_debt_assessment"
  ],
  "sla_hours": 24,
  "can_approve": true,
  "can_reject": true,
  "rejection_action": "send_back_to_developer",
  "approval_comment_required": false,
  "notification": {
    "channels": ["in_app", "email"],
    "message_template": "engineering_tech_approval_needed"
  }
},
{
  "level": 3,
  "name": "Engineering Manager Final Approval",
  "approver_designation": "engineering_lead",
  "approver_id_source": "organization_node.manager_id",
  "wait_for": "level_2_approval",
  "responsibilities": [
    "Business value verification",
    "Delivery timeline confirmation",
    "Team capacity check",
    "Final production approval"
  ],
  "required_fields": [
    "estimated_hours_vs_actual",
    "business_impact",
    "production_readiness"
  ],
  "sla_hours": 24,

```

```

        "can_approve": true,
        "can_reject": true,
        "rejection_action": "schedule_discussion",
        "approval_comment_required": false,
        "notification": {
            "channels": ["in_app", "email"],
            "message_template": "engineering_manager_approval_needed"
        }
    }
],
"escalation_rules": [
{
    "trigger": "level_not_approved_after_hours",
    "level": 1,
    "after_hours": 24,
    "escalate_to": "engineering_lead",
    "notification": "urgent"
},
{
    "trigger": "level_not_approved_after_hours",
    "level": 2,
    "after_hours": 24,
    "escalate_to": "cto",
    "notification": "urgent"
},
{
    "trigger": "level_not_approved_after_hours",
    "level": 3,
    "after_hours": 36,
    "escalate_to": "ceo",
    "notification": "critical"
}
],
"post_approval_actions": [
{
    "trigger": "all_approvals_complete",
    "actions": [
        "update_task_status_to_completed",
        "create_release_note",
        "notify_product_management",
        "update_sprint_metrics",
        "archive_approval_records"
    ]
}
]

```

```
    ]  
}
```

10. REAL-TIME SCENARIOS

10.1 Scenario 1: New Junior Developer First Day

TIMESTAMP: 2024-01-15, 09:00 AM

CHARACTER: Ashish - New Junior Engineer

MORNING:

- Ashish joins TechVision office
- Receives laptop, welcome package
- Logs into Wall-D desktop app (Flutter Web)
- Screens available: Employee screen only
- Sees notification: "Welcome to TechVision!"
- Dashboard shows:
 - Onboarding checklist (HR created)
 - "Pending Training: Git Basics" (Task)
 - "Pending Training: Codebase Introduction" (Task)
 - "Meet Senior Engineer - Rajesh" (Task)
- Ashish clicks: "View Onboarding Tasks"

10:00 AM - FIRST TRAINING TASK:

- Task: "Git Basics Training"
- Created by: HR Manager Vikram
- Assigned to: Ashish
- Task form has:
 - Training video links
 - Documentation links
 - Practical exercises
 - Completion checklist
- Ashish watches videos (45 minutes)
- Completes exercises (30 minutes)
- Clicks: "Mark Training Complete"
- Status: Awaiting Trainer Review

TRAINER REVIEW (Senior Engineer - Rajesh):

- Gets notification: "Ashish completed Git training"
- Reviews Ashish's exercise submissions
- Checks: All 5 exercises completed correctly
- Approves: "Great job, ready to move on"
- Task Status: APPROVED
- Notification: Ashish receives "Training Approved"
- Next task auto-created: "Codebase Overview Training"

2:00 PM - FIRST CODE ASSIGNMENT:

Engineering Lead (Amit) assigns first task

Task: "Fix typos in API documentation"

Task Details:

Title: "Fix Typos in API Documentation"

Description: "Correct spelling/grammar issues in API docs"

Project: "API Gateway v2.0"

Story Points: 1

Priority: Low

Assigned to: Ashish

Due Date: 2024-01-15 EOD

Estimated Hours: 1

Notification: Ashish sees new task

Ashish clicks: "Start Work"

Task Status: IN_PROGRESS

TASK COMPLETION:

Ashish edits documentation (30 minutes)

Finds and fixes 12 typos

Creates pull request #789

Adds comment: "Fixed all identified typos in API.md"

Marks task: "Ready for Review"

Task Status: AWAITING REVIEW

CODE REVIEW (Rajesh - Senior Engineer):

Gets notification: "Code review needed"

Reviews PR #789:

Changes are minimal

No breaking changes

Documentation quality

Formatting correct

Approves with comment: "Excellent first task! Keep it up."

Task Status: PENDING_APPROVAL

Automatically escalates to Amit (Tech Lead)

TECH LEAD REVIEW (Amit):

Gets notification: "Task pending final approval"

Sees Rajesh already approved

Reviews: Documentation quality

Approves: "Great start, Ashish!"

Task Status: COMPLETED

Notification: Ashish sees "Task Completed"

Achievement: "First task completed!"

Next task auto-assigned: Another documentation task

END OF DAY:

- Ashish completes 2 tasks
- Gains 2 story points
- Completes all onboarding checklist items
- Dashboard shows: "Great start! 2 tasks completed"
- Email from Amit: "Welcome to the team, Ashish!"
- Tomorrow: Real feature development task assigned

10.2 Scenario 2: Emergency Bug Fix Escalation

TIMESTAMP: 2024-01-20, 02:00 PM
INCIDENT: Production bug in payment system

DETECTION:

- Customer reports: "Payments failing for large transactions"
- Support ticket created (external)
- Escalated to Engineering Lead (Amit)
- Time: 2:00 PM

INCIDENT TASK CREATION (Amit - 2:05 PM):
Amit creates URGENT task:

- Title: "CRITICAL: Payment processing failure - Production"
- Description: "Customers unable to complete transactions > 10,000"
- Priority: CRITICAL
- Story Points: -1 (urgent, no estimation)
- Assigned to: Sanjay (Senior Dev, Payment Expert)
- Due Date: ASAP (2-hour SLA)

Notification: URGENT alert to Sanjay
Dashboard: CRITICAL flag shown in red
Escalates to: Amit + CTO automatically

INCIDENT ACKNOWLEDGMENT (Sanjay - 2:07 PM):
Sanjay clicks: "Start Work"
Task Status: IN_PROGRESS
Dashboard shows: "Investigating..."
Time tracking: Started at 2:07 PM
Workspace: Dedicated incident window
Integration: Links to production logs

INVESTIGATION (Sanjay - 2:07-2:25 PM):
Sanjay reviews:

- Recent code changes (last 3 days)
- Production logs for errors
- Payment gateway API responses
- Database transaction logs

Finds: Database constraint error

Amount field truncated for values > 9999.99
Root cause: Data type migration bug (Jan 18)
Affects: All transactions > 10,000
Sanjay adds comment: "Found root cause! Quickfix in progress"
Notification: Amit sees update

HOTFIX DEVELOPMENT (Sanjay - 2:25-2:45 PM):
Creates emergency branch: "hotfix/payment-critical"
Modifies: Payment amount validation
Quick fix: Add decimal precision handling
Tests locally: All transactions pass
Creates PR #1001: "HOTFIX: Payment amount precision"
Adds comment: "URGENT FIX - Ready for immediate review"
Task Status: AWAITING REVIEW

CODE REVIEW (Tech Lead - Sanjay - 2:46 PM):
SKIPPED for critical incidents
Direct approval by Sanjay himself
Justification: Only 3-line change, extensively tested
Comments: "This is safe, merge immediately"
Merge: PR #1001 merged to master

DEPLOYMENT (DevOps - Ravi - 2:50 PM):
Deployment task: "Deploy hotfix to production"
Ravi takes action immediately
CI/CD pipeline runs (5 minutes)
All tests pass
Deployment to production (3 minutes)
Verification: Test payment transactions
\$10,000 transaction: SUCCESS
\$50,000 transaction: SUCCESS
\$100,000 transaction: SUCCESS
Notification: "Deployment successful"

INCIDENT RESOLUTION (Amit - 2:58 PM):
Amit verifies fix: All systems operational
Closes task: "COMPLETED"
Time to fix: 56 minutes (excellent!)
Marks in system: "Critical Incident Resolved"
Creates follow-up task: "Post-incident review meeting"
Notification: "Incident resolved, customers notified"

POST-INCIDENT (Jan 20, 3:30 PM):
Incident review meeting scheduled
Sanjay presents: Root cause analysis
Discussion: "How did this slip through QA?"

Action items:

- Add data type validation tests
- Improve database constraint testing
- Create payment edge case test suite
- Implement better monitoring for amount fields

Assigned to: Sanjay + QA team

Due date: Jan 22 (2 days)

ANALYTICS:

- Incident duration: 56 minutes
- Customers affected: ~500 transactions
- Revenue impact: Minimal (most retried successfully)
- Team response: Excellent (all hands on deck)
- Communication: Clear and timely
- Resolution: Technical excellence
- Prevention: Implemented (tests added)

10.3 Scenario 3: End-of-Sprint Review & Retrospective

TIMESTAMP: 2024-01-19, End of Sprint 24 (2-week sprint)

SPRINT METRICS DASHBOARD (Amit viewing):

- Sprint Duration: Jan 5 - Jan 19 (14 days)
- Team: Backend team (8 engineers)
- Planned Capacity: 40 story points
- Completed: 38 story points (95% - Excellent!)
- Incomplete: 2 story points (1 task blocked)
- Velocity Trend: ↑ Up from 35 last sprint
- Defects Found: 3 (2 fixed during sprint, 1 in production)
- Deployment Success Rate: 100%
- Team Satisfaction: 4.2/5.0

SPRINT COMPLETION CEREMONY (Amit - 4:00 PM):

Sprint Planning Task created

Task Type: "Sprint Review & Retrospective"

Participants assigned:

- Sanjay (Tech Lead)
- Rajesh (Senior Dev 1)
- Vikram (Senior Dev 2)
- Ashish (Junior Dev)
- Ram (Junior Dev)
- Priya Q (QA Lead)
- Deepak (QA Engineer)
- Amit (Team Lead)

Meeting scheduled: 4:30 PM - 5:30 PM

Agenda:

Sprint summary & metrics review
Completed vs planned discussion
Team retrospective
Planning next sprint
Celebration of achievements
Notification: Meeting invite sent to all

SPRINT REVIEW MEETING (4:30-5:00 PM):

Amit presents sprint metrics
"Great job team, 95% completion rate!"
"5 customer-facing features delivered"
"Zero critical incidents, excellent stability"

Sanjay reviews: Key achievements
"Payment gateway integration complete"
"API performance improved by 30%"
"Database optimizations deployed"

Priya Q reviews: QA metrics
"Test coverage increased to 75%"
"3 defects found, all fixed"
"Zero production issues from our team"

Team discussions:
Rajesh: "Ashish's onboarding went great!"
Vikram: "Documentation improvements were key"
Priya Q: "New test framework saved us 20% QA time"
Outcomes: 2 blocked items resolved, blockers documented

RETROSPECTIVE (5:00-5:30 PM):

Amit leads retrospective discussion
"What went well?"
Team communication: "Standups are really effective"
New tools: "The new test framework is amazing"
Onboarding: "Ashish integration was smooth"
Collaboration: "QA and Dev partnership excellent"

"What didn't go well?"
Testing setup: "Initial environment setup took time"
Requirements clarity: "Payment spec had ambiguity"
Blockers: "Waiting for DB team slowed one task"

"Action items for next sprint?"
Sanjay takes: "Create better requirements template"
Priya takes: "Document test environment setup"
Amit takes: "Improve cross-team coordination"
All: "Continue with current standup format"
Assigned deadlines: Before next sprint starts

Team satisfaction survey: Posted in-app
Team morale: 4.2/5.0 (\uparrow from 3.8 last sprint)
Workload balance: 4.0/5.0 (good)

Communication: 4.4/5.0 (excellent)
Tools & support: 4.1/5.0 (good)
Celebration: "Awesome sprint, team! "

NEXT SPRINT PLANNING (Implicit in system):

Sprint 25 starts: Jan 22
Backlog items for sprint:
Customer onboarding flow (8 points)
Payment reconciliation (5 points)
API documentation (3 points)
Performance optimization (5 points)
Tech debt: Database queries (3 points)
Total planned: 24 story points (conservative estimate)
Capacity available: 40 points
Reserve: 16 points (for unplanned work, escalations)
Sprint planning meeting: Jan 22, 10:00 AM

POST-SPRINT SYSTEM UPDATES:

All sprint tasks marked: SPRINT_COMPLETED
Metrics saved to: Sprint 24 record
Reports generated:
Executive summary
Detailed metrics
Team performance review
Notification: Management sees sprint summary
Archival: Sprint 24 data archived (with retention policy)
Notification: Team sees "Sprint 25 starting soon"

11. FIREBASE STRUCTURE

11.1 Complete Firestore Collection Structure

```
firestore/  
  tenants/  
    techvision_inc/  
      metadata/  
        designations (document)  
          {alldesignations.json}  
        formSchemas (collection)  
          user_registration  
          engineering_task_form  
          sales_lead_form  
          expense_approval_form  
          ...  
        workflowDefinitions (collection)
```

```

engineering_task_approval
expense_approval_workflow
sales_lead_workflow
...
escalationRules (document)
notificationTemplates (collection)
    task_assigned
    approval_needed
    task_completed
...
settings (document)
    autoLogoutMinutes: 30
    dateFormat: "DD/MM/YYYY"
    currency: "INR"
    language: "English"
    timezone: "Asia/Kolkata"
    theme: "light"
customFields (collection)

users (collection)
    user_ceo_priya
        fullName: "Priya Sharma"
        email: "priya.sharma@techvision.com"
        designation_id: "ceo"
        department_id: "executive_team"
        status: "active"
        roles: ["admin", "manager", "developer"]
        permissions: [...all permissions...]
        createdAt: timestamp
        lastLogin: timestamp
        profile_picture_url: "..."
        phone: "+91-..."
        office_location: "bangalore"
        joining_date: "2015-06-01"
        cost_center: "CC001-Executive"
        preferences/
            notification_preferences
            theme: "light"
            language: "en"
            doNotDisturbHours

    user_junior_dev_ram
        fullName: "Ram Kumar"
        email: "ram.kumar@techvision.com"
        designation_id: "junior_engineer"
        department_id: "backend_team"

```

```

status: "active"
roles: ["employee"]
permissions: ["view_tasks", "complete_task", ...]
manager_id: "user_senior_dev_sanjay"
created_at: "2024-01-15"
onboarding_status: "completed"
sprint_velocity: 8

[...more users...]

organizations (collection)
root_node
  id: "root_node"
  name: "TechVision Inc."
  type: "company_root"
  parent_id: null
  manager_id: "user_ceo_priya"
  level: 0
  children: ["executive_team", "engineering_div", ...]
  metadata: {...}

engineering_div
  id: "engineering_div"
  name: "Engineering Division"
  parent_id: "root_node"
  manager_id: "user_cto_rajesh"
  level: 1
  children: ["backend_team", "frontend_team", ...]
  members: [...engineer IDs...]

backend_team
  id: "backend_team"
  name: "Backend Engineering Team"
  parent_id: "engineering_div"
  manager_id: "user_lead_amit"
  level: 2
  members: [...]
  metadata: {...}

[...more nodes...]

tasks (collection)
task_001
  id: "task_001"
  title: "Implement Payment Gateway Integration"
  description: "Integrate Stripe..."
```

```

        created_by: "user_lead_amit"
        created_at: timestamp
        assigned_to: "user_senior_dev_sanjay"
        assigned_at: timestamp
        due_date: "2024-01-12"
        status: "COMPLETED"
        priority: "high"
        project_id: "payment_system_v2"
        sprint_id: "sprint_24"
        story_points: 8
        estimated_hours: 40
        actual_hours: 38
        completed_at: timestamp
        custom_fields: {...}
        approvals (sub-collection)
            approval_001
                level: 1
                approver_id: "user_lead_deepak"
                status: "APPROVED"
                created_at: timestamp
                approved_at: timestamp
                comment: "Quality excellent"

            approval_002
                [level 2 approval...]

            approval_003
                [level 3 approval...]

            [...more tasks...]

        approvals (collection)
            approval_001
                id: "approval_001"
                task_id: "task_001"
                approver_id: "user_lead_deepak"
                level: 1
                status: "APPROVED"
                created_at: timestamp
                expires_at: timestamp
                approved_at: timestamp
                comment: "Testing completed"

            [...more approvals...]

    forms (collection)

```

```

[All form schema documents...]

workflows (collection)
[All workflow definition documents...]

notifications (collection)
notif_001
  user_id: "user_junior_dev_ram"
  type: "task_assigned"
  title: "New task assigned"
  message: "Fix typos in API documentation"
  action_link: "task/task_001"
  created_at: timestamp
  read_at: null
  channels: ["in_app", "email"]

[...more notifications...]

comments (collection)
comment_001
  task_id: "task_001"
  author_id: "user_senior_dev_sanjay"
  comment: "Ready for QA, PR #456 submitted"
  created_at: timestamp
  attachments: [...]

[...more comments...]

auditLogs (collection)
log_001
  action: "task_created"
  actor_id: "user_lead_amit"
  resource: "task_001"
  timestamp: timestamp
  ip_address: "192.168.x.x"
  before_state: {}
  after_state: {...created task...}

[...more logs...]

```

12. FLUTTER IMPLEMENTATION GUIDE

12.1 Project Structure

wall_d_flutter/

```

lib/
    main.dart (entry point)
    config/
        firebase_config.dart
        routes.dart
        theme.dart

    models/
        user_model.dart
        task_model.dart
        organization_model.dart
        designation_model.dart
        form_schema_model.dart
        approval_model.dart
        notification_model.dart

    services/
        auth_service.dart (Firebase Auth)
        firestore_service.dart (Firestore operations)
        realtime_service.dart (Realtime DB subscriptions)
        notification_service.dart
        storage_service.dart (Firebase Storage)
        sync_service.dart (offline sync)

    providers/ (State Management - GetX/Riverpod)
        auth_provider.dart
        task_provider.dart
        organization_provider.dart
        form_provider.dart
        notification_provider.dart
        ui_provider.dart

    screens/
        authentication/
            login_screen.dart
            registration_screen.dart
            password_reset_screen.dart
            mfa_screen.dart

        common/
            splash_screen.dart
            main_layout.dart
            error_screen.dart

    employee_screen/
        employee_dashboard.dart

```

```
my_tasks.dart
task_detail.dart
create_task.dart
profile_screen.dart

manager_screen/
    manager_dashboard.dart
    team_tasks.dart
    approvals_queue.dart
    team_analytics.dart
    create_task_advanced.dart

admin_screen/
    admin_dashboard.dart
    organization_management.dart
    form_builder.dart
    workflow_designer.dart
    user_management.dart
    settings.dart

developer_screen/
    developer_dashboard.dart
    firestore_explorer.dart
    tenant_management.dart
    system_logs.dart
    analytics.dart

widgets/
common/
    app_bar.dart
    navigation_drawer.dart
    custom_button.dart
    custom_text_field.dart
    loading_widget.dart
    error_widget.dart

forms/
    dynamic_form_renderer.dart
    form_field_factory.dart
    text_field_widget.dart
    dropdown_widget.dart
    date_picker_widget.dart
    user_picker_widget.dart
    file_upload_widget.dart

tasks/
```

```

    task_card.dart
    task_status_badge.dart
    priority_badge.dart
    approval_widget.dart
    timeline_widget.dart

    notifications/
        notification_toast.dart
        notification_badge.dart
        notification_center.dart

    org_hierarchy/
        org_tree_widget.dart
        org_node_card.dart
        hierarchy_builder.dart

    utils/
        validators.dart
        formatters.dart
        constants.dart
        extensions.dart
        logger.dart
        date_time_utils.dart

    database/
        local_db.dart (SQLite for offline)
        sync_manager.dart

    pubspec.yaml
    firebase.json
    analysis_options.yaml

```

12.2 Core Services Implementation Pattern

```

// lib/services/firestore_service.dart

import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:firebase_auth/firebase_auth.dart';

class FirestoreService {
    final FirebaseFirestore _firestore = FirebaseFirestore.instance;
    final String _tenantId = 'techvision_inc'; // From auth/config

    // Get user by ID
    Future<UserModel> getUser(String userId) async {
        try {

```

```

    final doc = await _firestore
        .collection('tenants/${_tenantId}/users')
        .doc(userId)
        .get();

    if (!doc.exists) throw Exception('User not found');
    return UserModel.fromJson(doc.data()!);
} catch (e) {
    logger.error('Error fetching user: $e');
    rethrow;
}
}

// Get user's tasks (REAL-TIME STREAM)
Stream<List<TaskModel>> getUserTasks(String userId) {
    return _firestore
        .collection('tenants/${_tenantId}/tasks')
        .where('assigned_to', isEqualTo: userId)
        .where('status', isNotEqualTo: 'COMPLETED')
        .orderBy('due_date')
        .snapshots()
        .map((snapshot) => snapshot.docs
            .map((doc) => TaskModel.fromJson(doc.data()))
            .toList());
}

// Get manager's team tasks
Stream<List<TaskModel>> getTeamTasks(String managerId) async* {
    try {
        // Get manager's organization nodes
        final orgNodes = await _firestore
            .collection('tenants/${_tenantId}/organizations')
            .where('manager_id', isEqualTo: managerId)
            .get();

        final nodeIds = orgNodes.docs.map((doc) => doc.id).toList();

        if (nodeIds.isEmpty) {
            yield [];
            return;
        }

        // Get all tasks assigned to team members in these nodes
        yield* _firestore
            .collection('tenants/${_tenantId}/tasks')
            .where('organization_node_id', whereIn: nodeIds)
    }
}

```

```

        .orderBy('created_at', descending: true)
        .snapshots()
        .map((snapshot) => snapshot.docs
            .map((doc) => TaskModel.fromJson(doc.data()))
            .toList());
    } catch (e) {
        logger.error('Error fetching team tasks: $e');
        rethrow;
    }
}

// Create new task
Future<String> createTask(TaskModel task) async {
    try {
        final docRef = await _firestore
            .collection('tenants/${_tenantId}/tasks')
            .add(task.toJson());

        // Log action for audit
        await _logAuditEvent('task_created', 'task', docRef.id, {}, task.toJson());

        return docRef.id;
    } catch (e) {
        logger.error('Error creating task: $e');
        rethrow;
    }
}

// Update task status
Future<void> updateTaskStatus(String taskId, String newStatus) async {
    try {
        final taskDoc = await _firestore
            .collection('tenants/${_tenantId}/tasks')
            .doc(taskId)
            .get();

        final beforeState = taskDoc.data();

        await _firestore
            .collection('tenants/${_tenantId}/tasks')
            .doc(taskId)
            .update({
                'status': newStatus,
                'updated_at': FieldValue.serverTimestamp(),
            });
    }
}

```

```

    // Handle workflow based on new status
    if (newStatus == 'AWAITING REVIEW') {
        await _initializeApprovalChain(taskId);
    }

    // Log action
    await _logAuditEvent('task_status_updated', 'task', taskId,
        beforeState ?? {}, {'status': newStatus});

} catch (e) {
    logger.error('Error updating task: $e');
    rethrow;
}
}

// Get approval chain for task
Future<List<ApprovalModel>> getApprovalChain(String taskId) async {
try {
    final approvals = await _firestore
        .collection('tenants/${_tenantId}/tasks/$taskId/approvals')
        .orderBy('level')
        .get();

    return approvals.docs
        .map((doc) => ApprovalModel.fromJson(doc.data()))
        .toList();
} catch (e) {
    logger.error('Error fetching approval chain: $e');
    rethrow;
}
}

// Initialize approval chain (called when task completed)
Future<void> _initializeApprovalChain(String taskId) async {
try {
    // Get task details
    final taskDoc = await _firestore
        .collection('tenants/${_tenantId}/tasks')
        .doc(taskId)
        .get();

    final task = TaskModel.fromJson(taskDoc.data()!);

    // Get workflow definition
    final workflowDoc = await _firestore
        .collection('tenants/${_tenantId}/metadata/workflowDefinitions')

```

```

.doc('engineering_task_approval')
.get();

final approvalLevels = workflowDoc.data()?'approval_levels' as List?;

if (approvalLevels == null || approvalLevels.isEmpty) return;

// Create approval records for each level
for (int i = 0; i < approvalLevels.length; i++) {
  final level = approvalLevels[i];
  final approverId = await _findApproverInHierarchy(
    task.assignedTo,
    level['approver_designation']);

  if (approverId != null) {
    await _firestore
      .collection('tenants/${_tenantId}/tasks/$taskId/approvals')
      .add({
        'level': i + 1,
        'approver_id': approverId,
        'status': 'PENDING',
        'created_at': FieldValue.serverTimestamp(),
        'expires_at': Timestamp.fromDate(
          DateTime.now().add(Duration(hours: level['sla_hours']))),
      });
  }

  // Send notification to approver
  await _sendApprovalNotification(approverId, taskId, i + 1);
}

} catch (e) {
  logger.error('Error initializing approval chain: $e');
  rethrow;
}
}

// Log audit event
Future<void> _logAuditEvent(String action, String resource,
  String resourceId, Map beforeState, Map afterState) async {
try {
  await _firestore
    .collection('tenants/${_tenantId}/auditLogs')
    .add({
      'action': action,
      'resource': resource,
      'resource_id': resourceId,
    });
}

```

```

        'actor_id': FirebaseAuth.instance.currentUser?.uid,
        'before_state': beforeState,
        'after_state': afterState,
        'timestamp': FieldValue.serverTimestamp(),
        'ip_address': '', // Would be captured from device info
    });
} catch (e) {
    logger.error('Error logging audit event: $e');
}
}

// Helper: Find approver in hierarchy
Future<String?> _findApproverInHierarchy(
    String userId, String approverDesignation) async {
try {
    // Get user's organization node
    final userDoc = await _firestore
        .collection('tenants/${_tenantId}/users')
        .doc(userId)
        .get();

    final currentOrgNodeId = userDoc.data()?['organization_node_id'];

    // Traverse hierarchy upward to find approver
    String? currentNodeId = currentOrgNodeId;
    while (currentNodeId != null) {
        final nodeDoc = await _firestore
            .collection('tenants/${_tenantId}/organizations')
            .doc(currentNodeId)
            .get();

        final managerId = nodeDoc.data()?['manager_id'];

        if (managerId != null) {
            // Check manager's designation
            final managerDoc = await _firestore
                .collection('tenants/${_tenantId}/users')
                .doc(managerId)
                .get();

            if (managerDoc.data()?['designation_id'] == approverDesignation) {
                return managerId;
            }
        }
    }

    currentNodeId = nodeDoc.data()?['parent_id'];
}

```

```
        }
    }

    return null;
} catch (e) {
    logger.error('Error finding approver: $e');
    return null;
}
}

// Send approval notification
Future<void> _sendApprovalNotification(
    String approverId, String taskId, int level) async {
try {
    await _firestore
        .collection('tenants/${_tenantId}/notifications')
        .add({
            'user_id': approverId,
            'type': 'approval_needed',
            'task_id': taskId,
            'approval_level': level,
            'title': 'Task Approval Required',
            'message': 'A new task requires your approval',
            'action_link': 'task/$taskId/approve',
            'created_at': FieldValue.serverTimestamp(),
            'read_at': null,
            'channels': ['in_app', 'email'],
        });
} catch (e) {
    logger.error('Error sending notification: $e');
}
}
```

12.3 Dynamic Form Rendering

```
// lib/widgets/forms/dynamic_form_renderer.dart

import 'package:flutter/material.dart';
import 'package:cloud_firestore/cloud_firestore.dart';

class DynamicFormRenderer extends StatefulWidget {
    final String formId;
    final String tenantId;
    final Function(Map<String, dynamic>) onSubmit;

    const DynamicFormRenderer({
```

```

        required this.formId,
        required this.tenantId,
        required this.onSubmit,
    });

    @override
    State<DynamicFormRenderer> createState() => _DynamicFormRendererState();
}

class _DynamicFormRendererState extends State<DynamicFormRenderer> {
    late FirebaseFirestore _firebase;
    final GlobalKey<FormState> _formKey = GlobalKey<FormState>();
    final Map<String, dynamic> _formData = {};
    late Future<Map<String, dynamic>> _formSchemeFuture;

    @override
    void initState() {
        super.initState();
        _firebase = FirebaseFirestore.instance;
        _formSchemeFuture = _loadFormSchema();
    }

    Future<Map<String, dynamic>> _loadFormSchema() async {
        final doc = await _firebase
            .collection('tenants/${widget.tenantId}/metadata/formSchemas')
            .doc(widget.formId)
            .get();

        return doc.data() ?? {};
    }

    Widget _renderField(Map<String, dynamic> field) {
        final fieldType = field['type'] as String;
        final fieldId = field['id'] as String;
        final label = field['label'] as String;
        final required = field['required'] as bool? ?? false;

        switch (fieldType) {
            case 'text':
                return TextFieldWidget(
                    label: label,
                    required: required,
                    onChanged: (value) => _formData[fieldId] = value,
                    validator: (value) => _validateField(field, value),
                );
        }
    }
}

```

```

    case 'email':
        return TextFieldWidget(
            label: label,
            required: required,
            keyboardType: TextInputType.emailAddress,
            onChanged: (value) => _formData[fieldId] = value,
            validator: (value) => _validateEmail(value),
        );

    case 'dropdown':
        return DropdownFieldWidget(
            label: label,
            required: required,
            field: field,
            tenantId: widget.tenantId,
            onChanged: (value) => _formData[fieldId] = value,
        );

    case 'date':
        return DatepickerWidget(
            label: label,
            required: required,
            onDateSelected: (date) => _formData[fieldId] = date,
        );

    case 'userPicker':
        return UserPickerWidget(
            label: label,
            required: required,
            field: field,
            tenantId: widget.tenantId,
            onUserSelected: (userId) => _formData[fieldId] = userId,
        );

    case 'textarea':
        return TextAreaWidget(
            label: label,
            required: required,
            onChanged: (value) => _formData[fieldId] = value,
            validator: (value) => _validateField(field, value),
        );

    case 'checkbox':
        return CheckboxWidget(
            label: label,
            required: required,

```

```

        onChanged: (value) => _formData[fieldId] = value,
    );
}

case 'multiselect':
    return MultiSelectWidget(
        label: label,
        required: required,
        options: field['options'] as List? ?? [],
        onSelectionChanged: (values) => _formData[fieldId] = values,
    );
}

case 'fileupload':
    return FileUploadWidget(
        label: label,
        required: required,
        maxFiles: field['maxFiles'] as int? ?? 1,
        onFilesSelected: (files) => _formData[fieldId] = files,
    );
}

default:
    return SizedBox(
        child: Text('Unknown field type: $fieldType'),
    );
}
}

String? _validateField(Map<String, dynamic> field, String? value) {
    if (field['required'] == true && (value == null || value.isEmpty)) {
        return '${field['label']} is required';
    }

    final validation = field['validation'] as String?;
    if (validation != null && value != null && value.isNotEmpty) {
        final regex = RegExp(validation);
        if (!regex.hasMatch(value)) {
            return 'Invalid ${field['label']} format';
        }
    }
}

return null;
}

String? _validateEmail(String? value) {
    if (value == null || value.isEmpty) {
        return 'Email is required';
    }
}

```

```

final emailRegex = RegExp(r'^[^@]+@[^@]+\.[^@]+$');
if (!emailRegex.hasMatch(value)) {
    return 'Enter valid email address';
}

return null;
}

@Override
Widget build(BuildContext context) {
    return FutureBuilder<Map<String, dynamic>>(
        future: _formSchemeFuture,
        builder: (context, snapshot) {
            if (snapshot.connectionState == ConnectionState.waiting) {
                return const Center(child: CircularProgressIndicator());
            }

            if (snapshot.hasError) {
                return Center(child: Text('Error loading form: ${snapshot.error}'));
            }

            final formSchema = snapshot.data ?? {};
            final fields = formSchema['fields'] as List? ?? [];

            return SingleChildScrollView(
                child: Form(
                    key: _formKey,
                    child: Padding(
                        padding: const EdgeInsets.all(16),
                        child: Column(
                            children: [
                                Text(
                                    formSchema['name'] ?? 'Form',
                                    style: Theme.of(context).textTheme.headlineSmall,
                                ),
                                const SizedBox(height: 24),
                                ...fields.map((field) => Padding(
                                    padding: const EdgeInsets.only(bottom: 16),
                                    child: _renderField(field as Map<String, dynamic>),
                                )),
                                ElevatedButton(
                                    onPressed: () {
                                        if (_formKey.currentState!.validate()) {
                                            widget.onSubmit(_formData);
                                        }
                                    }
                                )
                            ],
                        )
                    )
                )
            );
        }
    );
}

```

```
        ],
        child: const Text('Submit'),
      ),
    ],
  ],
),
),
),
),
),
);
},
);
}
}
```

CONCLUSION

This comprehensive demo company setup demonstrates:

1. **Realistic Hierarchy:** Multi-level org structure with real departments
2. **Complete Workflows:** Engineering, Sales, HR, Finance workflows documented
3. **Forms & Data:** Task forms, lead forms, expense forms all defined
4. **Approval Chains:** Multi-level approvals with escalation rules
5. **Real Scenarios:** Actual workflow scenarios showing system in action
6. **Firebase Structure:** Complete Firestore collection layout
7. **Flutter Implementation:** Code examples for core services and dynamic forms

This serves as the **complete template** for implementing Wall-D with Flutter + Firebase for any enterprise organization.