Generic workflow engine

Software Requirements Specification

Version 2.0

Prepared by

Komal Chugh (2016csb1124)
Prerna Garg (2016csb1050)
Siddharth Nahar (2016csb1043)
Rishabh Singh (2016csb1054)

12 Feb 2020

Table of Contents

1. Introduction
- 1.1 Purpose
- 1.2 Intended Audience and Reading Suggestions
- 1.3 Product Scope
- 1.4 References

2. Overall Description
- 2.1 Product Perspective
- 2.2 Product Functions
- 2.3 User Classes and Characteristics
- 2.4 Operating Environment
- 2.5 Design and Implementation Constraints

3. External Interface Requirements
- 3.1 User Interfaces
- 3.2 Hardware Interfaces
- 3.3 Software Interfaces
- 3.4 Communications Interfaces

4. System Features (Use Cases)

5. Other Nonfunctional Requirements
- 5.1 Performance Requirements
- 5.2 Safety Requirements
- 5.3 Security Requirements
- 5.4 Software Quality Attributes

6. Design Architecture
- 6.1 Context Diagram
- 6.2 UML Modeling (Class diagram)
- 6.3 Activity diagrams for each use case
- 6.4 Dataflow Diagram
- 6.5 Deployment Diagram
- 6.6 Design issues and discussions

Revision History

Date	Reason For Changes	Version	Date of Approval	Assignment No. (Sections)
29-01-2020	Initial commit for SRS	1.0	<Date of Approval>	Assignment 1 (Sec 1-5)
12-02-2020	Added design architecture	2.0	<Date of Approval>	Assignment 2 (Sec 6)

1. Introduction

1.1 Purpose

The purpose of this document is to present a detailed description of the Automated Workflow system. It will explain the purpose and features of the system, the interfaces of the system, what the system will do and the constraints under which it must operate. This document is intended for both the stakeholders and the developers of the system.

1.2 Intended Audience and Reading Suggestions

ID	Stakeholder	Description
S-0	Administrator	There will be one admin who will handle the creation of new forms for an institute/industry/company. Other responsibilities and functionalities provided for the admin are listed in section 2.1
S-1	Customer	• Financial institutions (eg. budget approvals) • Educational institutions (eg. funding for events , sports equipment purchases) • Industries (eg. design approvals) • Government offices (eg. leave approvals)
S-2	Development team	Forming the accurate vision of the project, detailed functional and nonfunctional requirements. Making Test Plans and Test Cases, Planning Timeline of work. Relevant Sections are 2.1, 4.

1.3 Product Scope

This software is a generic and configurable system that can help in automating the business workflows. A business workflow is a series of processing steps that need to be applied to certain data forms by an organization or a team of people. These forms need to be approved by designated officers through a predefined route. This system will be designed to maximise productivity by providing tools to assist in this approval process which would otherwise have to be performed manually.

This system will allow easily defining and adding new forms, roles and various state transitions. Every workflow supported by the system will have a set of legal transitions

allowed between its states. At any point of time, the form is in one specific state. The form is forwarded to a designated officer who may forward it ahead in the chain, or may send back with comments. The users are assigned various roles, and the ability of any user to edit or view a workflow form in any given state will depend on the user's role.

2. Overall Description

2.1 Product Perspective

Generic workflow engines are designed to assist certain processes (e.g. approval) for different kinds of workflow forms used in various industries like educational institutions, industries, etc. Our aim is to develop a generic platform that handles all these processing steps like creating the form, apply through the form, approve, send back with comments etc. The detailed functionality has been described in the following section.

Product Functions

Following is the list of major functionalities of the system:

1. Creating new Generic Forms
    a. The admin will be able to add new forms. The system will allow him/her to add fields of varying types e.g. date, text, number etc. and also set certain constraints as per the fieldtype. For eg. date entered should be valid and meaningful as per the context, numbers can be constrained to be non-negative etc.
    b. The admin can set permissions on the forms to decide which forms can be filled by what roles. For example, for an educational institution, a student should not be able to raise a budget approval application for a professor's lab.
    c. Each form will go through a specific path based on the role of the applicant for approval. The admin will be required to set these paths for various roles.

2. Adding Roles relevant for the workflow
    a. The admin will add specific roles for particular institutions/industries. For eg. a software company will have software developers, product managers, business analyst etc. while an education institution will have department wise HOD, Dean Faculty, students etc

3. Manage logs for auditing
    a. The previous history of roles eg. who were previous HOD or Dean Faculty will be maintained
    b. The admin will be responsible for changing a specific person's designation after his/her tenure has begun/ended.

4. Authentication
    a. The system will allow the admin to enable domain checks to login/signup. For eg. if the system is being used for Microsoft, it will only allow users whose email-ID has microsoft.com as the domain.
    b. For specific roles like HOD, there will be a specific email handle given to the person who currently holds the position. For example, Dr. Somitra in IIT Ropar is a CSE faculty and the HOD of computer science. He will be given two handles, hodcse@iitrpr.ac.in and somitra@iitrpr.ac.in. In order to perform functionalities as a HOD, he will be required to login through the particular email.
    c. Users can change their password and also reset password in case they forget the old password.

5. User functionalities
    a. Users can initiate an application by filling a form from the available set of forms.
    b. Users can view the status of their current pending application/form. He will be able to check with whom the form is pending and since when. Also, he will be able to see the comments added by the authorities who have already signed.
    c. Users can view the entire history of their approved forms in the past.
    d. User can edit (eg. add extra information) the currently pending form but in that case rerouting will be done i.e. the form will follow the approval path from start.
    e. However, if an approval along the workflow path has asked the applicant to modify his/her application, then the edited form will be routed to the particular approval and continue forward. It will not be re-routed from the beginning.

6. Approver's profile
    a. Users can see pending requests of forms received for approval.
    b. Users have a choice to forward/reject/request to modify forms.
    c. Users will get notifications for pending forms.
    d. Users can see the complete history of the forms he received for approval.
    e. User has functionality to approve forms in bulk i.e. the user can filter all instances of a particular form and collectively approve them.

2.2 User Classes and Characteristics

ID	User classes	Description
U-1	Administrator	A signed up user who has completed the account activation. He owns expanded rights inside the portal. He has functionality to create new forms/roles/edit workflows.

U-2	Signed up user	A user here can fill a new form, check the status of previous filled forms in processing and check the history of all of his forms.
U-3	Signed up Approver*	A user here checks notification for pending requests of forms , has permission to forward it further or revert it.
U-4	Non-Signed Up User	A user having email of domain can signup and use the webapp.

2.3 Operating Environment

This software will run on both windows and linux.

2.4 Design and Implementation Constraints

- On the initiation of this software for any particular institution the admin will be required to perform the following tasks:
    - Add users at each hierarchical level eg. faculty-HOD-Director or SDE1-Manager-Business Analyst etc.
    - Define approval paths for each role depending on the workflow
    - In order to define the hierarchy of an institution the admin needs to select a successor for each instance of a role at every level

3. External Interface Requirements

3.1 User Interfaces

The system will have the following interfaces:
Note: For every institution there will be a single admin who will be able to create new forms, add approval paths and define user roles through the UI.

1. User login/signup
2. The admin on logging in will see a list of existing forms and a button to create new forms.
3. On submitting the above form, the user will be redirected to a page where he/she will add the path that this form must follow based on the user role. He/she will also be able to define new roles as per the requirement.
4. On adding a new role, a drop down menu will allow the admin to select the employee/faculty who will currently hold this role.

5. For an end user, say a university professor, on logging in a list of existing forms will be displayed. The user can select a form to fill out to initiate the particular workflow.
6. The end user page will have a navigation bar with an option to view existing applications. This will include the entire user history i.e. approved applications, rejected applications, ongoing request etc. sorted by the initiation date.
7. The user may select any of the applications from the list to view the details or edit it. For some cases, editing the form will result in re-initialising the application through the beginning of the route e.g. date change in vacation request.

3.2 Hardware Interfaces

1. PC/Laptop
2. Since the application must run over the internet, hardware interface would include, Modem or WAN – LAN or Ethernet Cross-Cable

3.3 Software Interfaces

1. Operating System: Windows/Linux
2. Internet Connection
3. A browser that supports HTML, JavaScript

3.4 Communications Interfaces

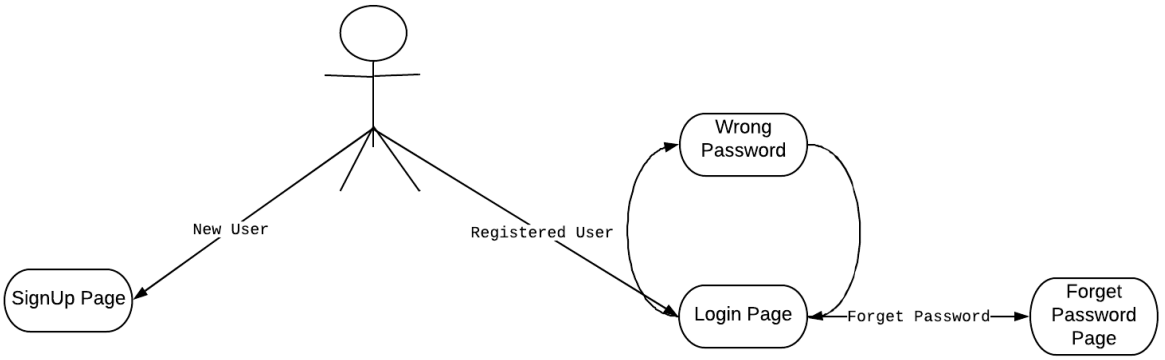
1. The end-user will be required to signup/login into the system using his email-ID.
2. The web service will have client-server architecture and use a stateless communication protocol (http).
3. A fixed set of four operations: PUT, GET, POST, and DELETE will be used to create, read, update, and delete resources, respectively.

4. Use Case Scenarios

4.1 Use Case Scenarios : Authentication

Purpose	User authentication
Actor	Admin, User and Approvers of webapp
User Input	1. Email Address 2. Password 3. Name 4. Department
Precondition	1. Email must be unique 2. Department must exist.

Postcondition	Successful Login/Signup
Basic Flow	Home Page, user clicks login/signup button does the transaction.



4.2 Use Case Scenarios : Creating Hierarchy

Purpose	Creating Department and Role Hierarchy
Actor	Admin of webapp
User Input	1. Name of department at each level. 2. Name of corresponding roles.
Precondition	1. The name of the departments should be unique. 2. Name of the roles should be unique. 3. Roles hierarchy should be subset of the department hierarchy.
Postcondition	Successful creation of both hierarchy and corresponding relational schema in database.
Basic Flow	Admin logins, Clicks the create hierarchy link and feeds the required input to create the hierarchy.

4.3 Use Case Scenarios : Creating Form Template

Purpose	Creating Form Template
Actor	Admin of webapp
User Input	1. Various input fields with their type ex. Text, numeric, date. 2. Permissions to view/edit forms.

	3. Workflow for the forms.
Precondition	1. There shouldn't be any existing form for similar usage. 2. For setting permissions and workflows, roles must be pre existing in database.
Postcondition	Successful template creation and users must be able to fill data.
Basic Flow	Admin logins, Clicks the create form link select type of input field required add it specify some constraints save it.

4.4 Use Case Scenarios : Editing Form Template

Purpose	Editing Form Template
Actor	Admin of webapp
User Input	1. Change input type for field. 2. Add additional data fields. 3. Change permissions for specific roles.
Precondition	1. There should be pre pre-existing form. 2. For setting permissions and workflows, roles must be pre existing in database.
Postcondition	Successful template update. Users must be able to fill out the form.
Basic Flow	Admin logins, Clicks the edit form link select form to be edited , does the task and saves it.

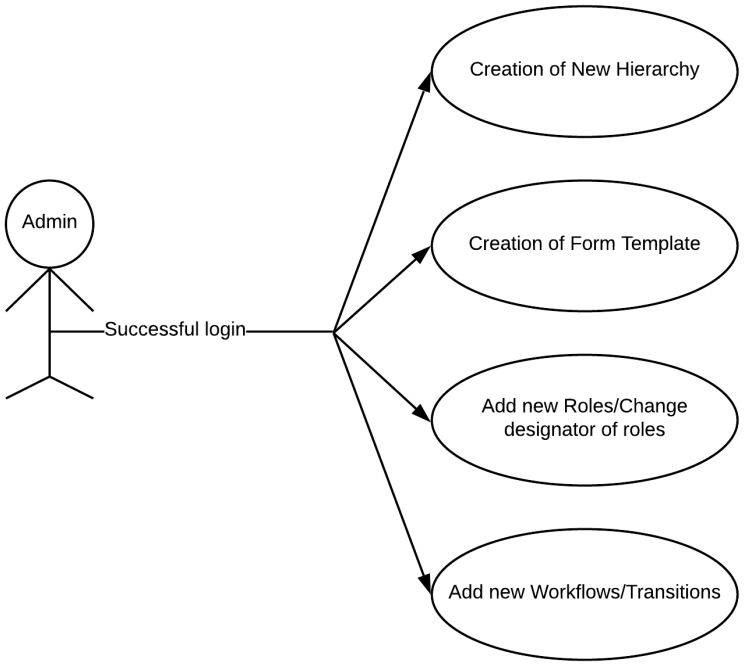
4.5 Use Case Scenarios : Adding/Editing New Roles

Purpose	Adding/Editing new Roles
Actor	Admin of webapp
User Input	1. Unique Role name to add. 2. User to fill in for a specific role.
Precondition	1. There shouldn't be pre pre-existing role name

	for adding a new role. 2. For users to fill in a role that role must be pre existing.
Postcondition	Successful role creation.
Basic Flow	Admin logins, Clicks the roles link select to add roles or edit user role and do the transaction.

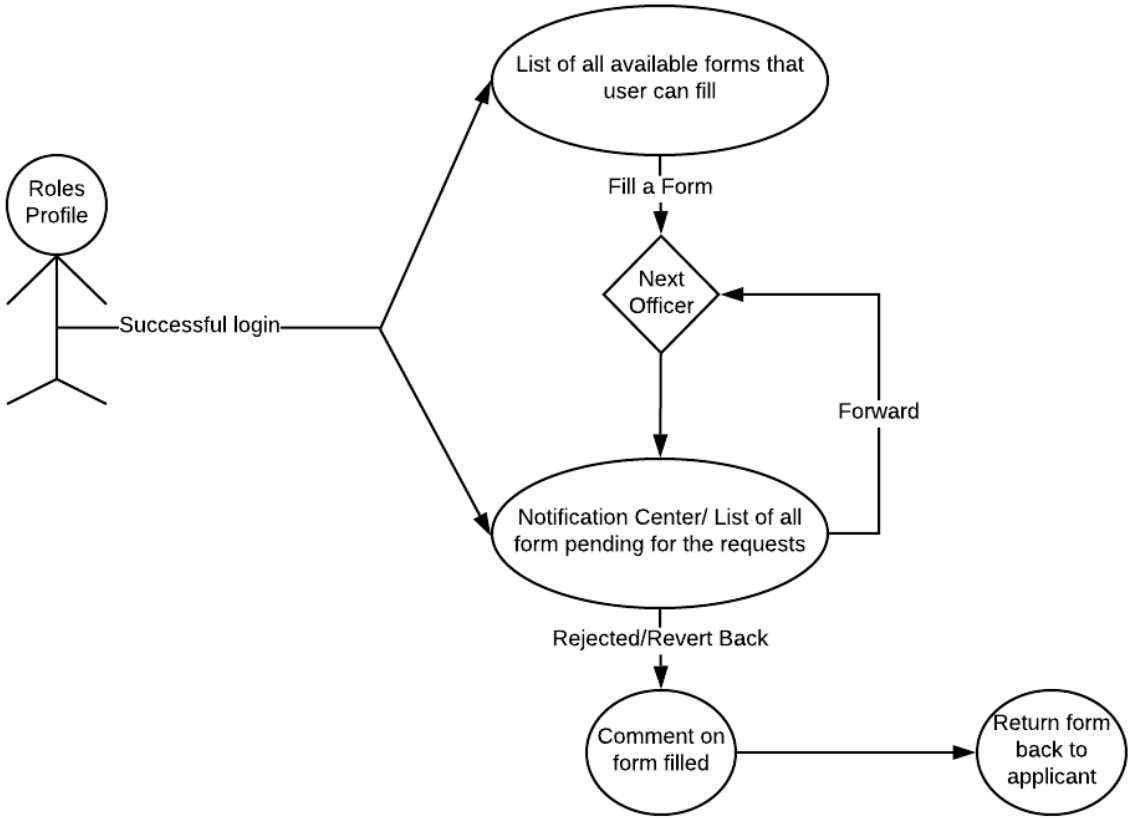
4.6 Use Case Scenarios : Add/Edit Workflow

Purpose	Add/Edit Workflow
Actor	Admin of webapp
User Input	1. Sequence of roles for specific form showing the workflow. 2. Remove/Add roles for some workflow.
Precondition	1. There should be pre pre-existing form for workflow. 2. Roles must be pre existing in the database.
Postcondition	Successful workflow update for form.
Basic Flow	Admin logins, Clicks the Workflow link Add new workflow or edit pre existing one and save the transaction.



4.7 Use Case Scenarios : Forward/Revert/Comment Form

Purpose	Forward/Revert/Comment the form
Actor	Designated Officers of webapp
User Input	Users fill out the comment section in the form.
Precondition	There should be pending forms to respond. Users must have permission to edit the form.
Postcondition	1. In case of Forwarding form will be forwarded to the next designated officer or if it is the last person then it gets approved. 2. In case of Revert it is rolled back to the applicant.
Basic Flow	User logins, Click on the pending requests to get details and then Click on the Forward/Revert after giving Comments.

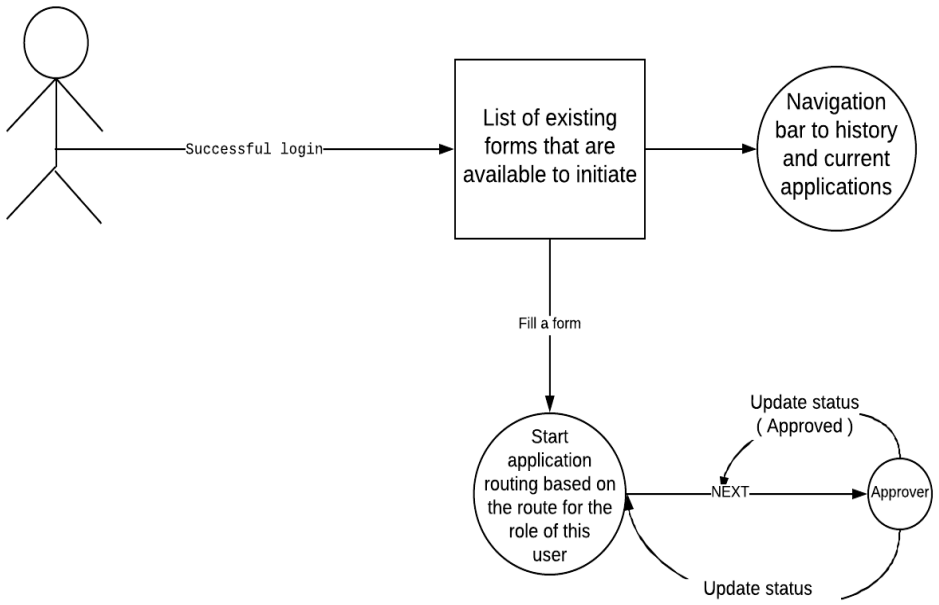


4.8 Use Case Scenarios : Fill The Form

Purpose	Fill the form
Actor	User of the webapp
User Input	1. User fill out all data fields in form 2. Forwards it to the designated officer.
Precondition	1. There should be pre pre-existing form for similar usage. 2. Users must meet all constraints specified by admin.
Postcondition	Successful form saving.
Basic Flow	User logins, Click on type of form to be filled and click submit.

4.9 Use Case Scenarios : Check the Status and History

Purpose	Check Status and History
Actor	User and Designated Officer
User Input	No Input
Precondition	There should be pre pre-existing form that the user/officer has touched.
Postcondition	All histories viewed successfully.
Basic Flow	User logins, Click on history tab, click on specific form to see its history close it.



5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. The system shall support multiple clients efficiently.
2. The number of forms being processed simultaneously is proportional to the response time.

5.2 Safety Requirements

1. The system shall use secure sockets in all transactions that include any confidential customer information.
2. The system shall automatically log out all customers after a period of inactivity.

3. The system shall confirm all transactions with the customer's web browser by displaying appropriate messages.

5.3 Security Requirements

1. The system shall not leave any cookies on the customer's computer containing the user's confidential information.
2. The user password shall be encrypted and hidden while typing

5.4 Software Quality Attributes

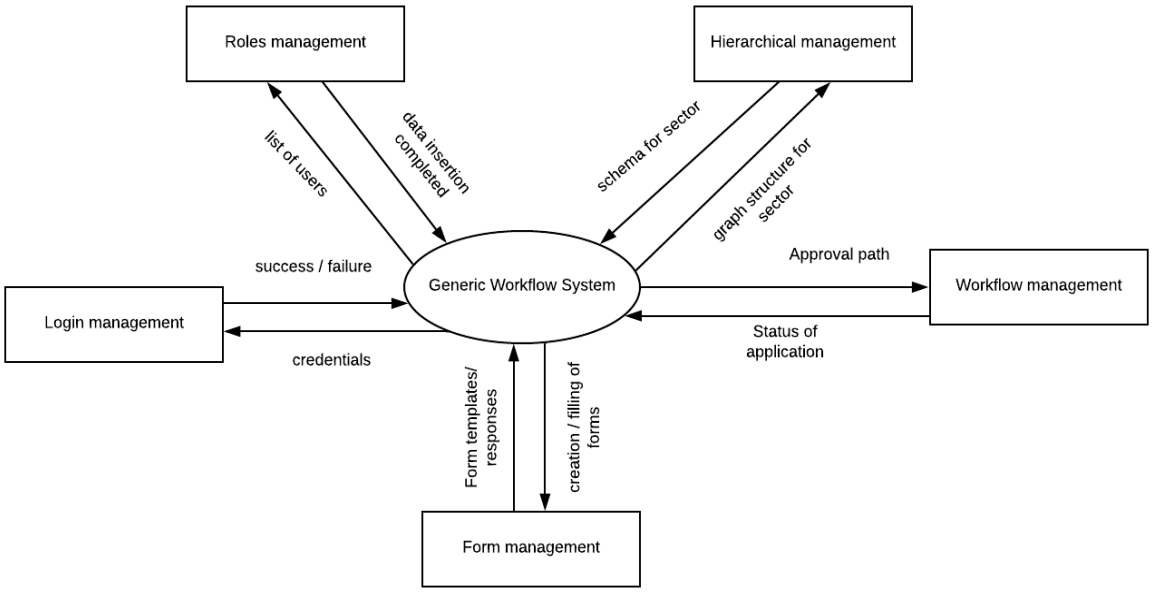
1. The system will be maintained through version control systems like Github.
2. The source code shall be modularised to maximise reusability.

6. Design Architecture

6.1 Context diagram

Various modules of our system are :

1. **Login Management :** Based on credentials inputted by user, Login management module will either Login or give a failure message.
2. **Form Management :** This module allows creation of templates of forms via Admin, allows to initiate an application via user, allows to edit templates, set permission who can initiate this form.
3. **Workflow Management** : This module allows admin to set the path for a particular form at each stage set the permission of users who can approve/reject/revert the form. This module will return the current status of form.
4. **Hierarchical Management** : This module basically takes graph structure of Departmental Hierarchy/Roles Hierarchy as shown in fig and creates Schema for database which will be relevant to the sector.
5. **Roles Management :** This module basically does the task to assign roles like “HOD, STUDENT, DEAN” to list of users provided and fills the data accordingly in schema previously created.

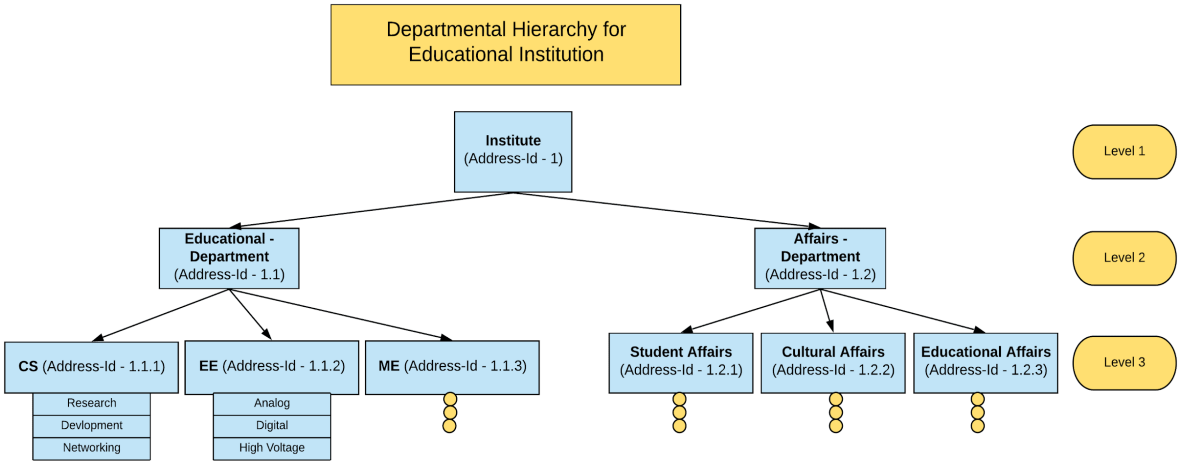


6.2 UML Modeling (Class diagram for static behaviour)

Generally every sector has a hierarchical structure for responsibilities and each position in that hierarchy is guided by a person. Considering this, we label the hierarchy of teams/departments as Departmental Hierarchy and hierarchy of persons in-charge of those departments is labelled as Roles Hierarchy. Suppose an educational institute wants to use our application, First task for the Administrator is to input the graph structure of the hierarchy of Departments and Roles. This hierarchy will help us to develop Relational schema for our application.

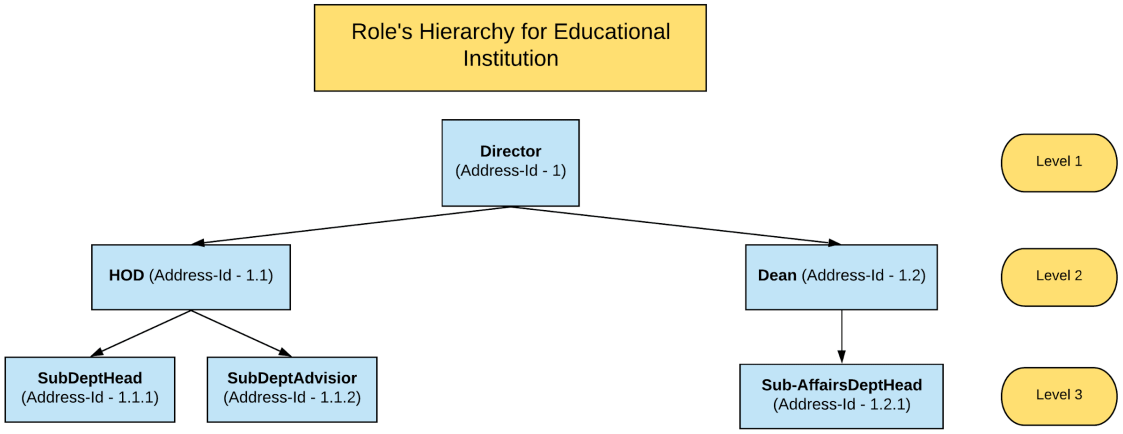
1. **Departmental Hierarchy**

   We can see in the below figure how a hierarchy can look for an educational institution. There are multiple Levels of hierarchy as shown in figure. We have a unique address for each node in the graph. Institute address is 1, Educational Department is 1.1, Affairs Department is 1.2 . Address : "Parent-id concat Child-id" child-id will be according to BFS traversal of graph.



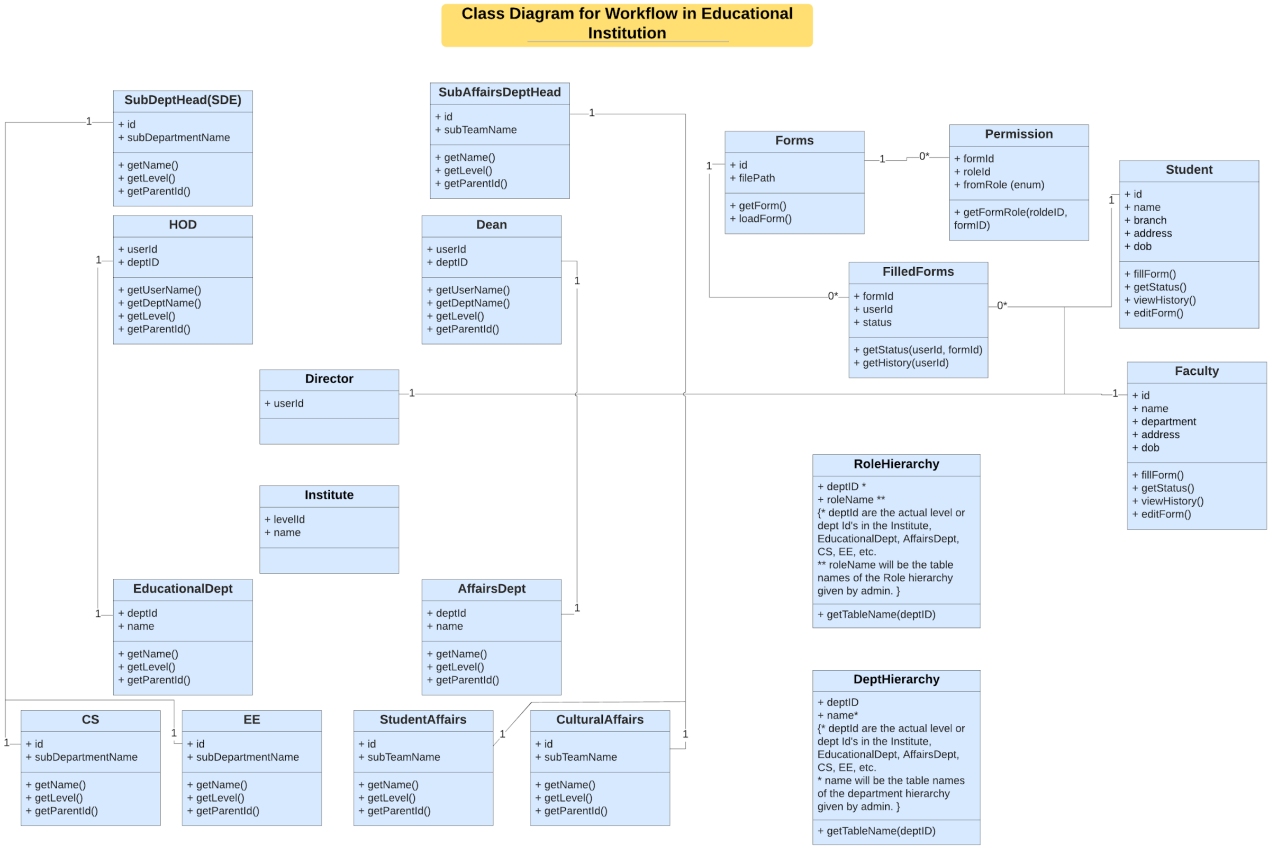
2. **Role Hierarchy**

   Each node in the Department Hierarchy has a certain number of values. In our notation, Values and Childs of nodes are two different things for ex. Research, Development, Networking are values of node CS but CS, EE and ME are children of the Educational Department. So we now know each node has multiple persons leading in roles. So that role will be the same for each of the values of node for ex. Institute Node in the Department hierarchy is led by Director, Similarly Each department value is led by HOD and the Affairs department is led by Dean. For each CS node value we will have SubDepartment head and Sub-Department Advisor. So we can see we have a parallel relation between two hierarchies.



3. Class Diagram

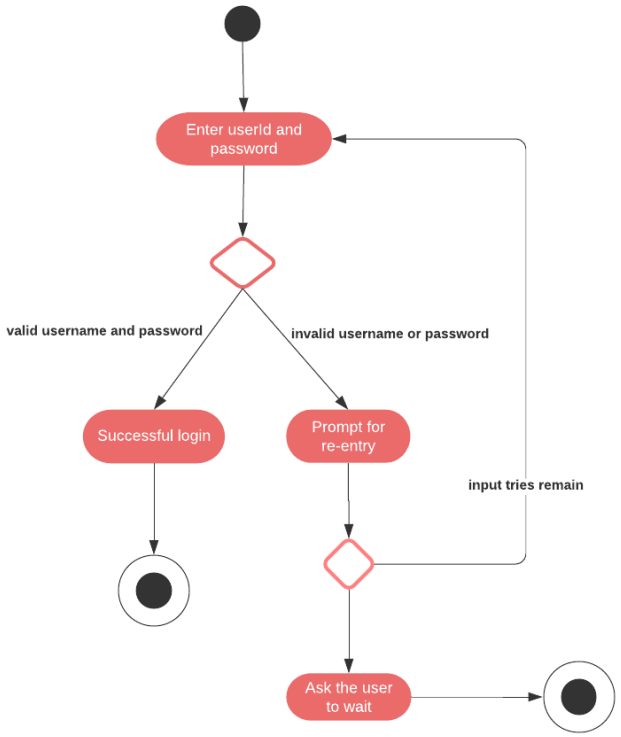
We have to organize the above details in a relational Schema. Each node will represent a different table in Schema. Each role table will have a foreign key to the corresponding department. We can see a class diagram below corresponding to the above two hierarchies.



Forms will be saved as html file on hard disk with its meta-info in database. For setting permission for each form will have permission table saving form-id and role-id and type of permission of form.Both Hierarchy will be stored as separate Tables with its address this will act as starting block for each query.

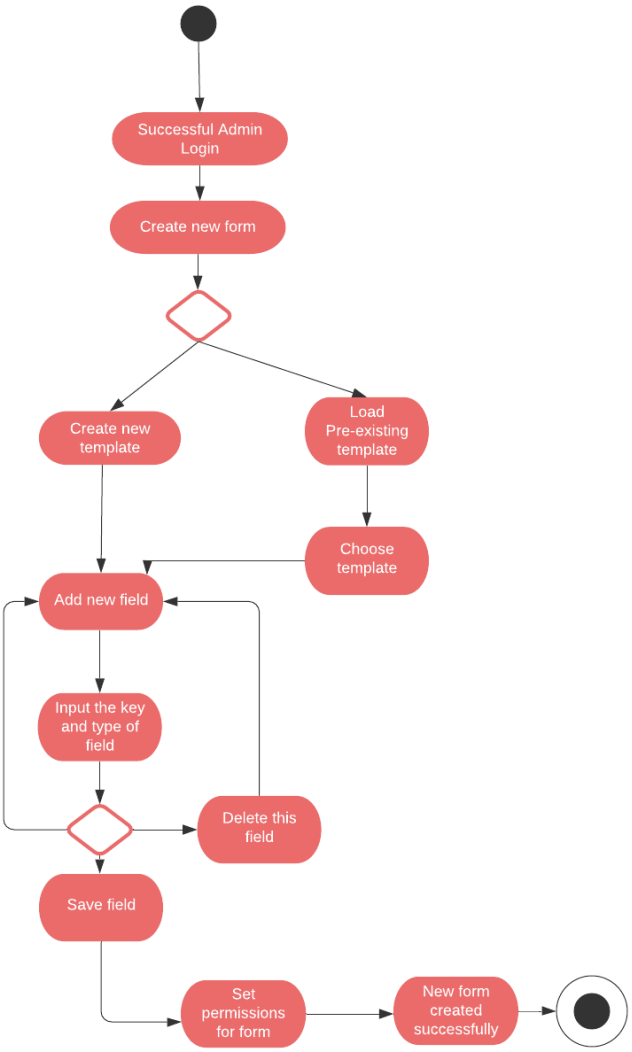
6.3 Activity diagrams for each use case

1. **User Authentication:** The following activity diagram explains the working of authentication use case. The system prompts the user to enter loginId and password and leads to successful login if the credentials are correct. Otherwise, it checks if there are input tries remaining or not. If yes, it prompts the user for re-entry. Else the system asks the user to wait for sometime and try again.

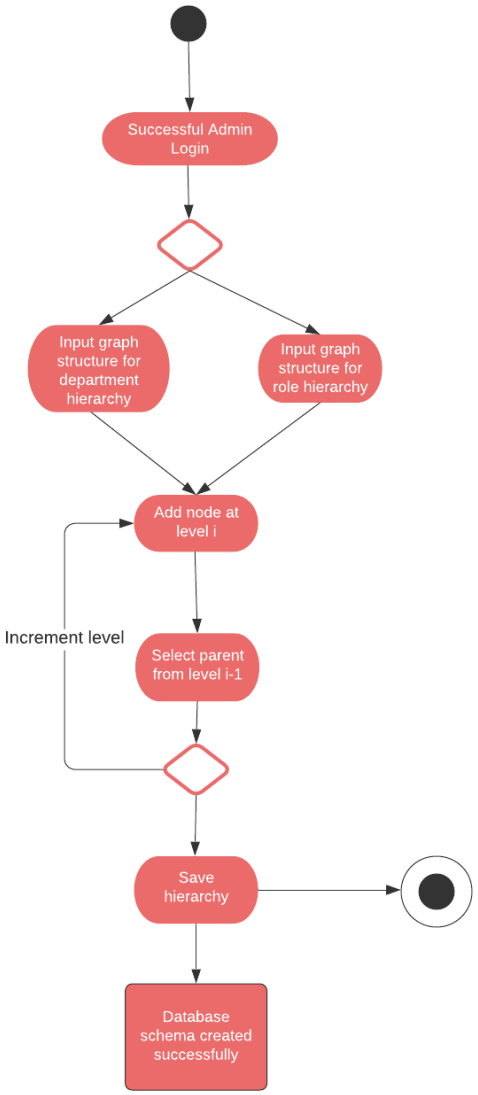


2. **Create / Edit form template:** The administrator may wish to edit a previously created form template or create a new form template. The following activity diagram explains the working of creating form template and editing form template use case. After the administrator has logged in successfully, he can choose the form template he wants to edit or can click on the 'Create new template' button.

   He has the option to add new fields where the field refers to the key and type. For example, for the name field, the key can be 'First Name' and type would be varchar. He will also have the option to delete this field and add new fields. After adding all the fields, he can click on the Save button and after this, the user will have to set permissions for this form for the initiators. Only the initiator roles defined by the Admin will be able to initiate the form.

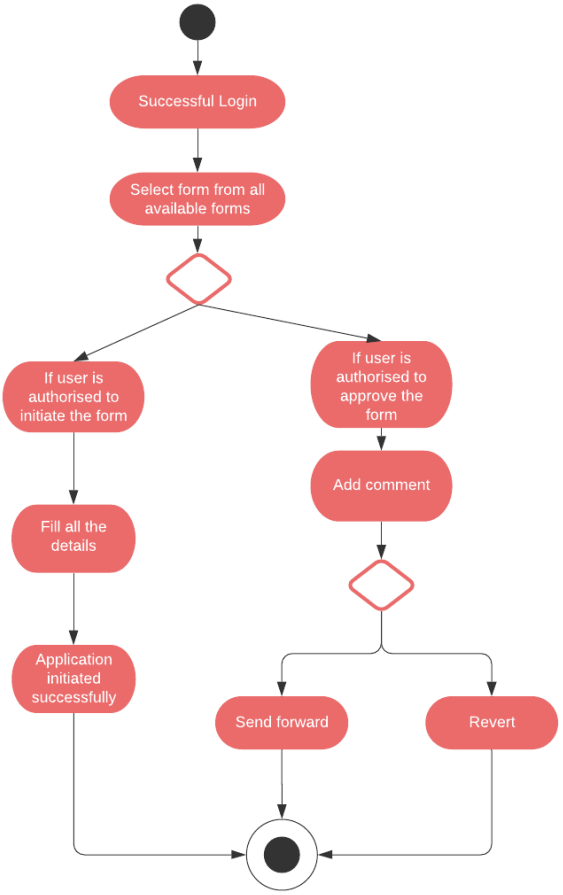


3. **Create hierarchy:** The initial setup of our application requires the Administrator to input Departmental hierarchy and Roles hierarchy as explained in UML Model. The following activity diagram explains the working of creating hierarchy use cases. After successful login, the Admin will have the option to add graph structure for Departmental hierarchy and Roles hierarchy. For each structure, the Admin has the option to add nodes to level i and for each node, he will have to specify the parent node. For example, for 'CS' node, 'Educational Departments' node will be the parent node. Finally after he has finished adding all the nodes, he can save the hierarchy and this will lead to creation of relational schema in our database.



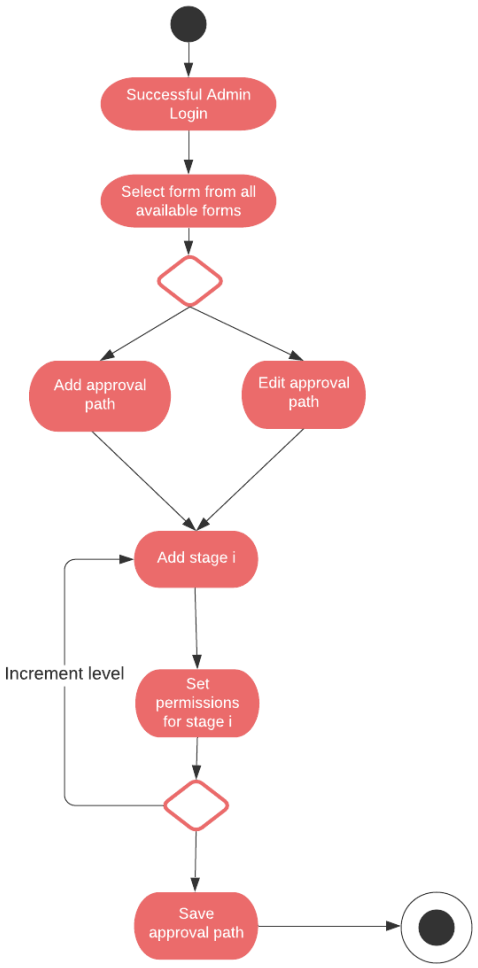
4. **Initiate / Approve a form:** The following activity diagram explains the working of filling the form and add comments to form use cases. After a user has logged in successfully, he has the option to initiate a request by filling a form or can add comments to pending applications/forms if the user is authorised to do so. For initiation, the user will have to fill all the necessary fields and then click on the save button. This will automatically forward the application to the immediate approver set by the Admin.

   If the user is authorised to approve an application, he can add comments to that application and then forward it to the next approver or can revert it.

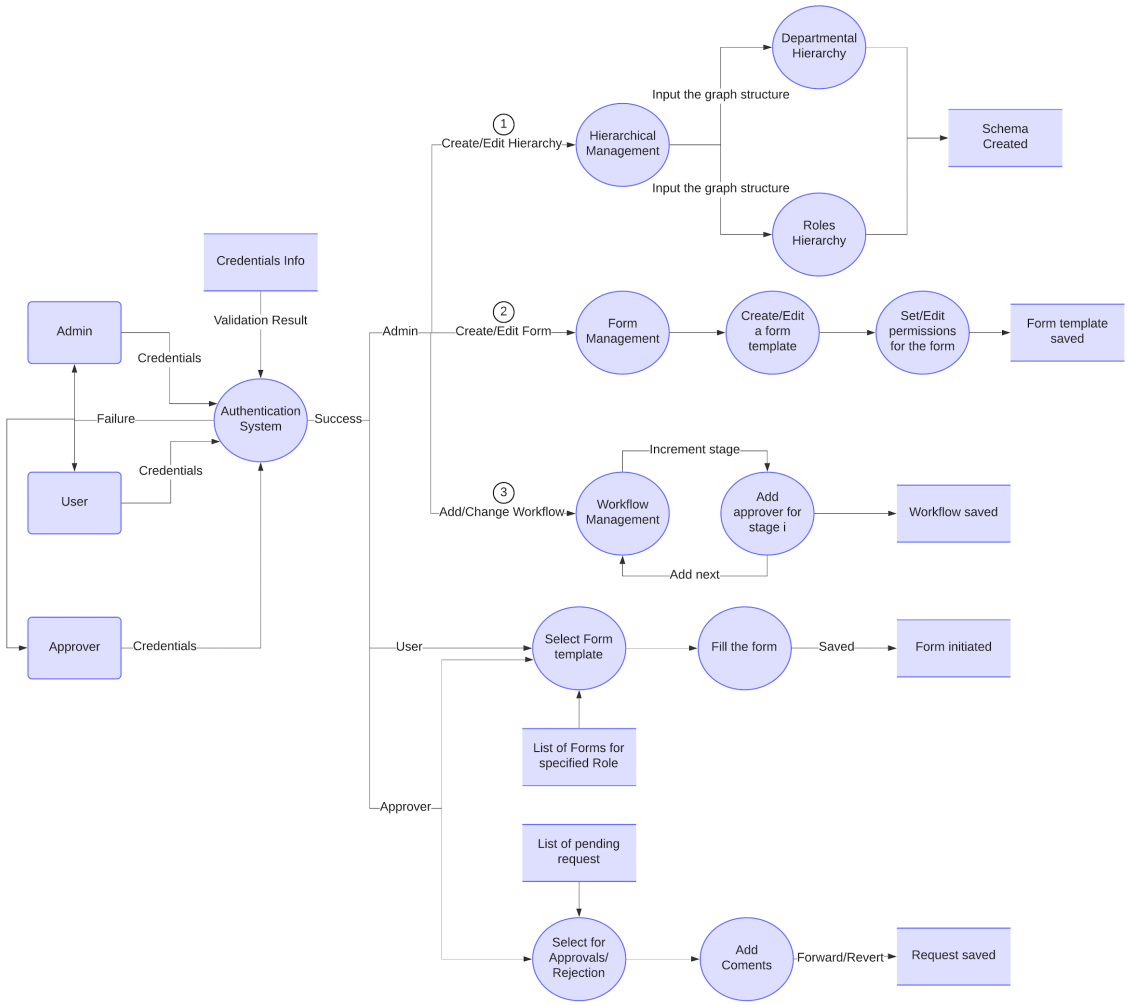


5. **Add / Edit workflow (approval path):** After the Admin has created a form template, he will have to add a workflow path for that form. Or the Admin may wish to edit workflow path for pre-existing forms. The following activity diagram explains the working of Add/Edit Workflow use case.

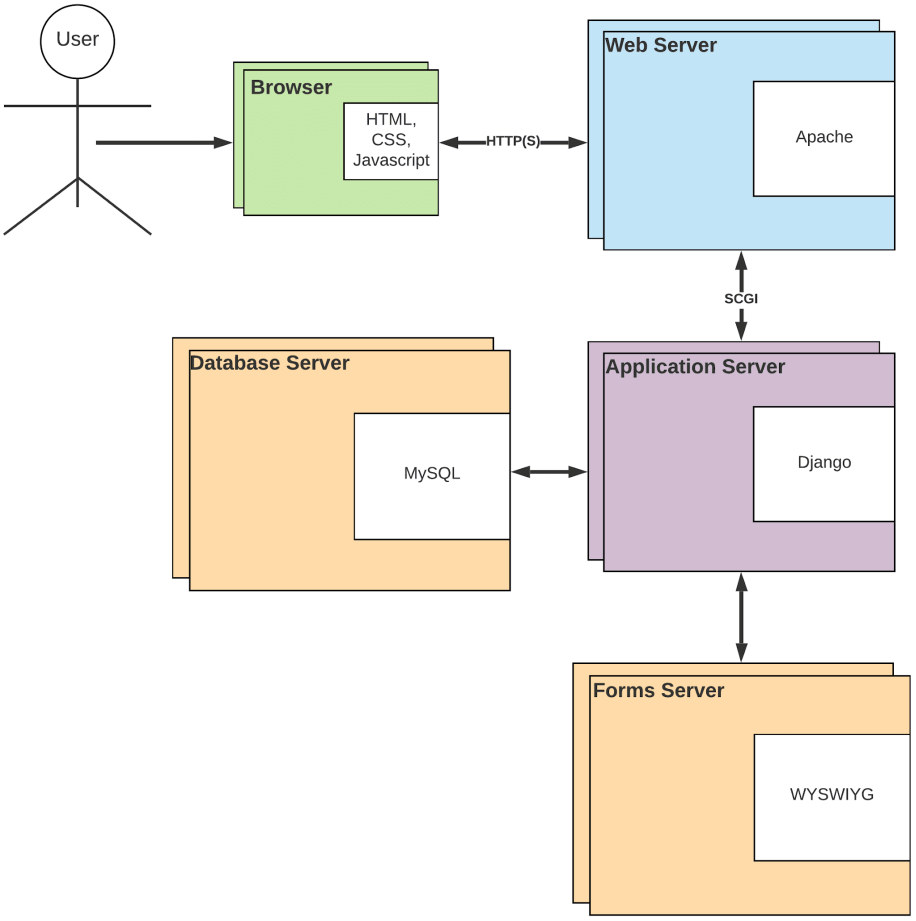
   After selecting the form for which he wants to add/edit workflow, the Admin will have to choose a role at every stage. He can add as many stages as he wants. But for every stage, there should be a unique role selected. For eg, for an educational institute, the Admin creates faculty leave application form, then he can add workflow like SubDeptHead → HOD → Director. If the Admin edits the workflow for pre-existing forms, then the forms which are currently under processing will follow the old approval path and forms initiated after the change will follow the new approval path.


Successful Admin Login
Select form from all available forms
Add approval path
Edit approval path
Add stage i
Increment level
Set permissions for stage i
Save approval path

6.4 Dataflow Diagram



6.5 Deployment diagram



6.6 Design issues and discussions

1. **Hierarchical management module:** It is mandatory for Admin to input Departmental hierarchy and Roles hierarchy. An alternative to this design choice was to take roles alone (eg HOD, Dean, Director, etc.) and not consider the hierarchical structure. The hierarchy offers an advantage over the other design choice. For instance, consider a case where Admin wants to create a form which can be initiated by anyone belonging to the CS Department.

   In case of hierarchical design choice, the Admin can simply specify CS department node and it will automatically allow people from Networking, AI, ML, etc. to initiate that form. Whereas for the other design choice (considering only roles), the Admin would have to specify each and every node of the subtree under ‘CS’ node.

2. **Bulk approval:** An Approver can accept or revert forms in bulk instead of doing it one by one. A choice will be given to the Approver for bulk processing. It can be made efficient by using Batching techniques. This design choice would be useful in cases where the priority of form is less and mostly the requests are accepted. Consider the case of Add/Drop requests sent to Faculty Advisor during course registration. There are very few chances of reverting the request in this case.

3. **Template saving:** After the Admin has created the form template, our system will save the HTML file to hard disk so that when a user wants to initiate a form, the system can directly load the file and display the contents. The system will store the file path in the database instead of storing the file as BLOB object in database directly. It is because there is no requirement or use case leading to querying in the file. Hence it is better to retrieve the file directly from disk.