DESIGN DOCUMENT

WORKFLOW MANAGEMENT SYSTEM

Team Members :

Priya Jain : 2017CSB1099

Sapna Sharma : 2017CSB1105

Shivang Khare : 2017CSB1109

Shrikant Saxena : 2017CSB1111

TABLE OF CONTENT

1. INTRODUCTION

This System Design Document has been created to outline the proposed system design for Business Workflow Management System. The BWFMS is intended to replace the traditional way of managing workflows (manually sending emails, notifications and text documents used in organizations) by an automated WFMS. This is a web-based system for end-users, that can be easily accessed by them on browsers.

1.1 Purpose

The purpose of this Design Document is to provide a description of how the BWFMS will be constructed. The Design Document was created to ensure the BWFMS design meets the requirements specified in the BWFMS project requirements documentation. The System Design Document provides a description of the system architecture, software, hardware, database design, UI design and security. It is meant to give a detailed blueprint of the BWFMS to the coding team about the modules to be implemented and their coordinated working to fulfil the requirements specified in the SRS document.

We have considered various design options and pondered over their pros and cons. These design issues have been mentioned in detail in this document. Further, we have given specification about the testing procedures to be followed to ensure that the system is reliable, robust and efficient.

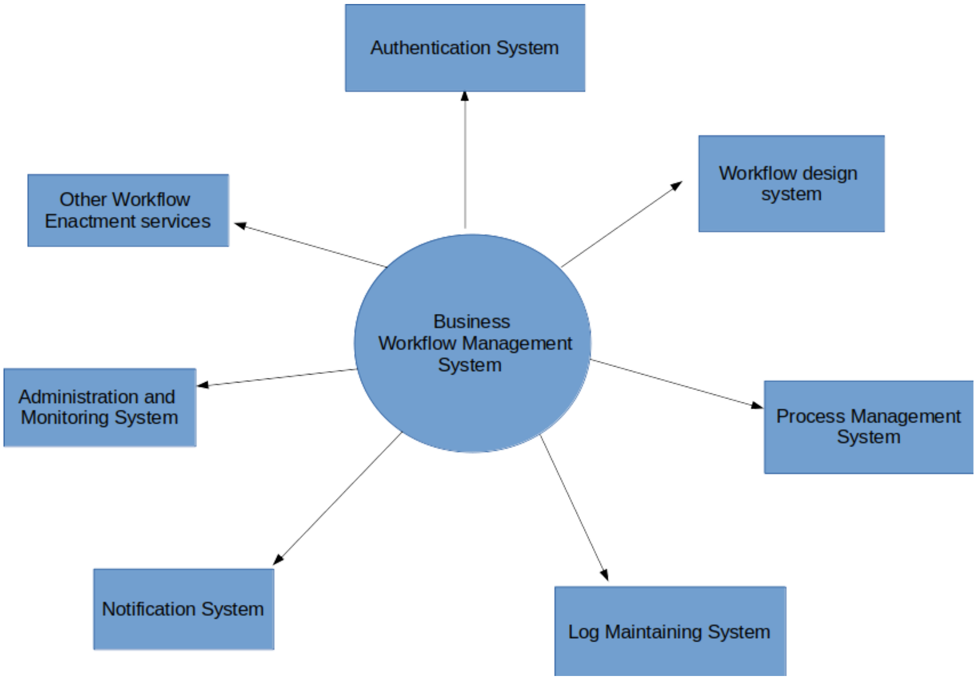
1.2 System Overview

A workflow management system (WfMS or WFMS) provides an infrastructure for the set-up, performance and monitoring of a defined sequence of tasks, arranged as a workflow application.

WfMS allow the user to define different workflows for different types of jobs or processes. For example, in a manufacturing setting, a design document might be automatically routed from a designer to a technical director to the production engineer. At each stage in the workflow, one individual or group is responsible for a specific task. Once the task is complete, WfMS ensures that the individuals responsible for the next task are notified and receive the data they need to execute their stage of the process. WfMS also automate redundant tasks and ensure that uncompleted tasks are followed up.

The system will have functionalities of designing and editing of workflows, initiating processes, responding to tasks, notification regarding pending tasks, creation of new roles, accounts and departments, distribution of privileges and duties and logging and fetching of the information regarding the activities taking place via the system.

This will be a distributed system consisting of servers which will be accessed by clients using traditional browsers on their system. The system will consist of authentication system to provide access to authorised users, workflow design system for designing of new workflows, process management system for initiation and progression of processes, log maintaining system for maintaining a record of activities, notification system to notify the users about the pending tasks, administration and monitoring system for broad level administering of the system by the system administrator and workflow enactment services for enforcing a process to follow the workflow design.


Authentication System
Workflow design system
Other Workflow Enactment services
Business Workflow Management System
Administration and Monitoring System
Process Management System
Notification System
Log Maintaining System

Context diagram of the system

1.3 Product Scope

The final product will be a Human task-centric workflow system which will enable the organization to manage their business processes in a hassle free manner by automating relatively simple and redundant processes. Admins will be able to create and deploy workflows for the processes that need to be automated. Users will be able to start the processes which are available to them and check the status of various processes initiated by them. The notification system will remind users of their pending tasks so that they do not miss the deadlines. The organization can get information about its staff such as how frequently an employee misses his/her deadlines and what action did he/she took on each process that is assigned to them through the logs available in the system. This will help the organization to take action against the employees who are not working efficiently. The department head and system administrator will be able to easily and efficiently administer the system based on the policies of the organisation.

In the initial releases, the system will not support complex workflows involving parallel tasks and redirection of the course of the workflow based on the conditions imposed on the value of variables at certain state. The workflow supported will be relatively simple composed of linear chains of tasks involving the designated users at each node.

The functionality for more complex workflows will be available in further releases, and their implementation will be governed by the availability and integrability of open source API's for the design of the workflow layout and configuration.

In the least, the final product must be able to support and enforce simple workflows, must be able to maintain the logs of the important information regarding the activities happening in the system and must support system administration and notification functionalities. On top of that, the system must fulfil various security criterias to safeguard against exploitation by external agents and information leak.

Note that the product is meant to achieve automation of form fill-up based processes and is not meant for more complex processes like the ones which need mutual discussions among people of various hierarchies for the approval of a task.

1.4 References

Software requirements specification is released one can get details from there and more information about the system will be made available in the future documentation such as how to install the software, user manual for how to use the system and also on the website where it will be hosted, whose link will also be provided in future documents.

1. Regarding form creation and deployment API : (See Form.io)
    a. https://www.form.io/how-it-works
    b. https://github.com/formio/formio
    c. Refer to Form_io_documentation document attached herewith

2. Regarding drag and drop based workflow design API: (See jsPlumb)
    a. https://jsplumbtoolkit.com/
    b. https://github.com/jsplumb/jsplumb
    c. Refer to jsPlumb_documentation document attached herewith

1.5 Key Stakeholders

Internal

Project team members: the group performing the work

Project management team: the members of the team directly involved in project management

In between

User/ Customer: System-Admin, Admin, users who are associated with the organization.

External

Professor and TA

1.6 Roles and their privileges

There will be different kind of roles with a different kind of roles/posts in the system, each having different privileges as follows:

Roles	Privileges
System Admin	Approve/reject new account creation requests. Manage(create/delete/edit) department,roles, and user accounts. Has access to all the data of the System.
Workflow Admin	Create, edit or delete a workflow.
Supervisor	Privilege to see all workflows, their processes and details of the employees.
Department Head	Can see the list of all employees. Reassign the task of one employee to another.
Normal Employ	Respond to pending tasks, depending upon the options given to him as per as the workflow designed.

1.7 Design Constraints and Limitations

Following are the design constraints binding the implementations of this software :

1. There is a hard-coded account of system admin into the software so that the system can be configured and set up after the installation. Through his account, further account creation requests can be handled, and the account could be used to create new roles and departments.

2. The software will support only simple workflows consisting of nodes joined in series. More complex workflows which involve parallel chains and conditional workflow routes will be attempted for implementation in the later releases, preferably towards the end. The implementation of this functionality will depend upon the availability of various API's which can be integrated in the system to support this functionality.
3. There will only be one user at a particular role so that role based tasks can be uniquely assigned to one particular user.
4. In case where a user misses a deadline to complete a certain task, a notification has to be sent to the department head. This constraint is necessary for supervision of the employees and proper functioning of the organisation. It also serves the purpose to notify the department head to re-assign the tasks of an employee in case the employee is not available to work at the moment.
5. A user cannot edit the comments or responses made by any other user. This constraint is necessary to prevent any employee fiddling with the information and disrupting the proper progress of a process.
6. A workflow cannot be deleted until all processes based on it are resolved. This constraint is necessary to ensure that users who have submitted an application before the deletion of the workflow do not feel bamboozled
7. In case a workflow is edited, the active process following that workflow will restart from the node at the end of the portion which is common in the original workflow and the new workflow. This way the new design can be brought into effect immediately.
8. A page has to be divided into different sections so that the most important portions can be loaded sooner. This will enable us to give an illusion of faster loading of the site, enhancing user experience.
9. Although the system admin will be given the privilege to see everything. But since one can't create tabs for displaying all kinds of data in all different types, we are giving system admin the privilege to run a SQL query, extract and check from the database.

2. REQUIREMENTS

Necessary requirements to build the system, requirements for installing and configuring the system by system administrator and requirements for using the system by end-users are enlisted here.

2.1 Software Requirements

Language Requirements

Front end: Ajax(Javascript and XML) with jQuery framework

Why use Ajax?

Ajax can selectively modify a part of a page displayed by the browser, and update it without the need to reload the whole document as we need this functionality in our system to reload the comments posted by the user on any process without reloading the whole page. It will save a lot of time when many users are concurrently accessing the system.

Ajax is a set of technologies, supported by a web browser, including these elements:

- HTML for the interface.
- CSS for the look and feel.
- JavaScript (ECMAScript) for local processing and DOM (Document Object Model) to access data inside the page or to access elements of XML file read on the server.
- The XMLHttpRequest object is used to read or send data on the server asynchronously.
- Supports python language used on the server.

Backend

Python with Flask framework

Database

Postgresql (Relational database)

For the scope of this project, we are using a relational database. A relational database can be considered for any information need in which data points relate to each other and must be managed in a secure, rules-based, consistent way. We need these properties of relational databases to maintain, access and store data securely. It is the best at maintaining data consistency across applications and database copies (called instances).

We will be using stored procedures so the code can be used over and over again; database locking and concurrency as many users are accessing the system at the time so Locking prevents other users and applications from accessing data while it is being updated and Concurrency manages the activity when multiple users invoke queries at the same time on the same database; indexing for enhancing the performance of the system; roles to assign privileges to database objects they own, enabling access and actions to those objects. Roles have the ability to grant membership to another role. Attributes provide customization options for permitted client authentication.

Operating System

It is a web-based application that will be available to users on all mobile and desktop operating systems.

2.2 HARDWARE REQUIREMENTS

a) For the development of this software, the developers will need a standard pc in use. The pc should have at least 4GB RAM, with a processor clock of speed 1GHz. A server computer will be needed to host the website, during the testing and deploying phase. Considering the scope of this course project, it is expected that the developer’s laptop will be used as a server.

b) For using this software, the system administrator should install this software on a system having enough secondary memory to download the software file, at least 4GB RAM (physical memory), with processor clock speed 1GHz. For real-time use of the software, additional space (several GB) should be allowed for growth in the Web Server database and RAM requirements can increase too. Also, make sure that it is available all the time for anyone trying to access it.

c) End-users can access the system through web, and they will need a standard pc or mobile device with at least 100 MB of free RAM to run any standard browser like google chrome.

3. System Architecture

3.1 Overall System Architecture

The overall system can be divided into major components that are shown in the diagram below. Description of each component:

1). Web user interface: It is the topmost layer in our system that provides the users with a graphical interface with which they can interact. System admin, Admin and end-users have some differences in their interfaces as this will depend on the functions they can perform in the organization. This layer takes the data from the users and sends it to the lower layers and it fetches the data from the lower layer and presents them to the users. This layer is implemented using the front end languages mentioned later in this document. For more details of user interface refer to section 5.

2). Workflow Editor: Only admin has the right to create the workflow. So, this functionality only appears to him/he. We provide the GUI to create, edit and delete the workflows.

Create Workflow: User firstly creates a form for a workflow by selecting the attributes and by mentioning their data types he/she wants in that form. After designing the form, this form will be stored in the source code as an HTML file and viewed to all users who initiate a process. After that admin will create the nodes of the workflow in series. Admin will assign the role on every node, duration in which task should be completed, default settings of notification. Following are the further design decisions pertaining to this module:

A) While the workflow is being designed, its incomplete components will be stored in a temporary table. Once the design is complete, this will be transferred to the permanent database.

Reason for this design: This design decision is taken so that the main database needs not be accessed again and again. This will save the round trips needed and improve the efficiency of the system. Further in case the system is interrupted in between, the user can continue the task further from the last saved checkpoint in the temporary database

Edit Workflow: All the admins who have the right to edit the workflow can see this functionality. Admin can select the nodes to be deleted and insert any new node between two nodes. Admin can change the role assigned to a particular node and other factors like duration or notification. Admin can change the form and create a new form. After editing the workflow, this get stored into the system and from now onward new workflow will be displayed to all users and the pending applications of the edited workflow start from the node which is common in previous and new workflow.

Delete Workflow: In case that a workflow is no longer required due to change in internal working or change in policies of the organisation, the workflow can be deleted by the admin who created the workflow.

Following are the further design decisions pertaining to this functionality:

A) The privilege to delete a workflow is provided only to the admin who has created that workflow. Further a workflow can only be deleted if there are no active processes running on that workflow. In case an admin tries to delete such a workflow, an error message will be displayed to him / her.

Reason for this design : First of all the privilege of deleting a workflow is given only to the admin who has created the workflow to prevent accidental or deliberate tampering with the workflows stored in the database. Further a workflow can only be deleted when there are no active processes running on it to ensure that such processes don’ t jam during their execution phase.

3) Activity transaction component: This component manages the course and progress of the processes running based on workflow design stored in the database. Once a certain task is completed,the component logs the responses made by the designated users of that task into the database. The component fetches the details regarding the next task in the workflow from the system database. These details include the users involved in the next task, the preconditions needed to activate the next task and other configuration options of the next task. The users associated with that task are notified via the notification system about the task pending at their part. The users can respond based on their convenience or based on the task deadlines as dictated by the task configuration. The responses given by the users in form of comments or fill-up of the form fields are then logged into the database. Only after the present task is completed and the preconditions of the next task is satisfied (like the application must be forwarded for it to be judged by the next higher authority) the next task is activated.

Note that the workflow can chose one of the multiple branches available depending upon certain conditions being satisfied

Following are the further design decisions pertaining to this module:

A) In the initial releases, we are implementing only three types of final responses on a task: Forward, Reject, Send Back. In case of Forward, the form is forwarded to the next node in the workflow. In case of Reject, no next task is activated and the process terminates. In case of Send Back, the the form is forwarded to one of the previous nodes in the workflow.

   Reason for this decision : In the initial release we try to keep the user interface as simple as possible so that we can have a robust foundation. Implementing multi branching workflow will need an extensive UI for their design, which will be a hindrance in the timely deliverance of the initial system release

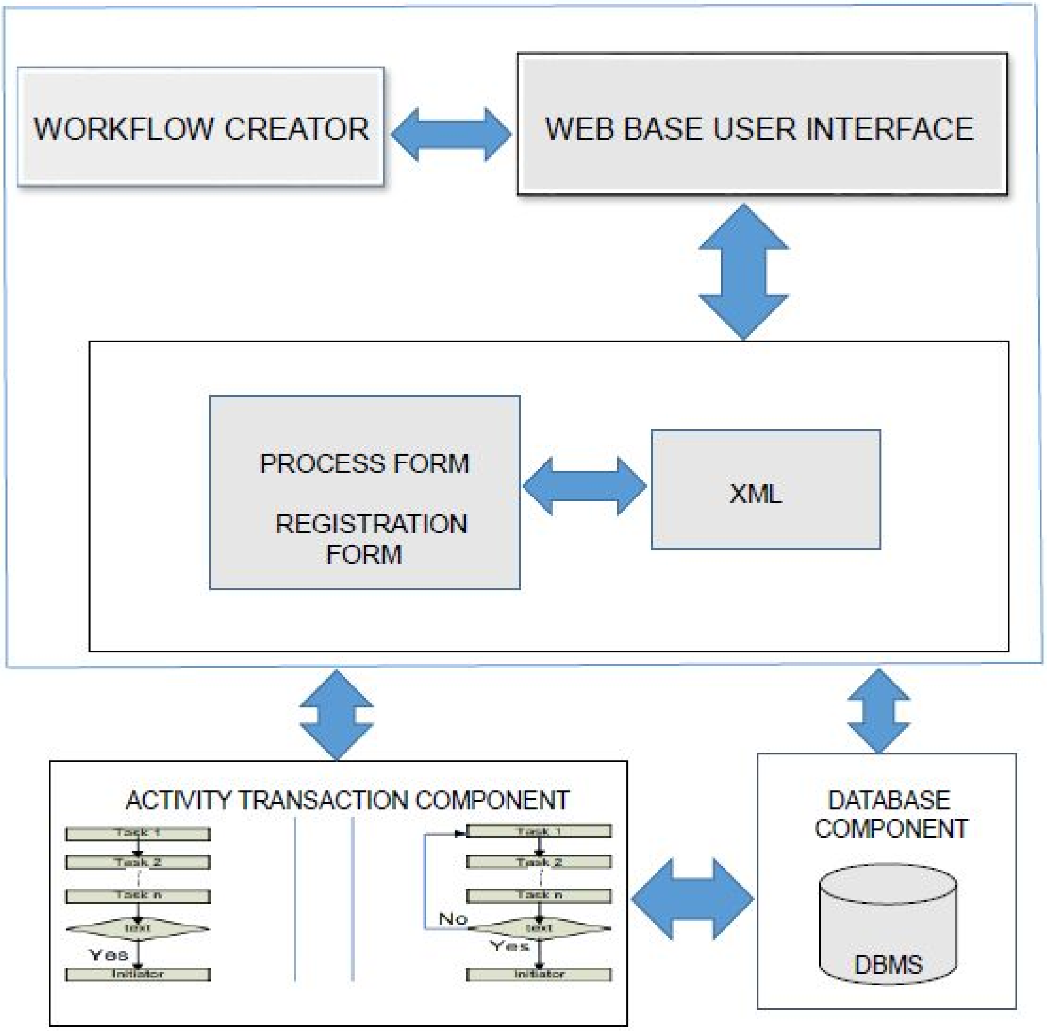
4) Log Maintaining System: Each and every activity which ever took place in the system will be logged into the system database. Following is a detailed description of the objects to be stored by the logging system

- **Employee data:** Following details pertaining to the employee will be logged into the system
  - Personal details of each and every employee which ever worked with the organisation

    - All the processes initiated by the employee, including the date of initiation and the details filled by the employee into the form.
    - All the responses given by the employee on the tasks assigned to him / her along with the timestamp of assignment and completion of those tasks.
    - All the deadlines missed by the employee, the process to which the missed task belonged to, the temporal details of the task missed.
    - All the posts handled by the employee along with their tenures will be logged into the system.
- **Workflow data**: Following data pertaining to the workflow will be logged into the system
    - The name and id of each and every workflows ever designed and deployed in the system.
    - When the workflow was designed
    - The designer of the workflow
    - The workflow layout
    - The configuration option applicable to each and every node of the workflow (like the cardinality of the approval from the applicable users etc.)
    - If a workflow was deleted, when was it deleted.
    - The number of processes currently supported by the workflow will be logged into the system.

- **Process data:** Following details regarding the active or completed processes pertaining to the system will be maintained:
    - The process id of each and every process ever initiated
    - The user id of the initiator of the process
    - The current status of the process
    - The users involved at each and every member task of the process
    - The responses and comments given by the users at each and every task involved in the process
    - The timestamp of assignment and completion of a task of the process by a user.

FIGURE 3.1.1: LOGICAL ARCHITECTURE OF SYSTEM



5) Notification System Component: This component will ensure that the users are always notified whenever it's necessary so that the user doesn't check for things again and again themselves. The following are the situations in which users will be notified:

- Whenever a new task is assigned to a user or whenever he gets associated with a process.
- Whenever the roles of a user is updated or approved by the system admin.
- Notification to system admin whenever a new account is requested.
- Whenever a user gets a new privilege. For example, the privilege to start new process of a workflow, the privilege to edit workflow.
- Notification to HOD in case a user is not able to complete the task upon a time.
- Notification to process initiator in whenever a process completes successfully.
- Notification to the system admin in case of any errors or issues like sudden system failure, or if any component gets corrupted.

3.2 System Integration

The system components work in integration in the following manner:

- **Authentication system** :
  - Before interacting with the system the user needs to access an account. In case the user is not pre-registered into the system, he/she needs to submit a request through the authentication system for a new account.
  - Once the request is approved, the user can access the account using the credentials provided to him / her. The user must pass through the authentication system layer to gain access to the account. The user enters his / her login credentials. The authentication system then passes these credentials to the **Database System.** Through the execution of certain SQL functions and queries, the authentication system gets the info if the user credentials are logged into the Database System or not. The authentication system then redirects or rejects the account access request accordingly.

- **Workflow design system:** Once the user has accessed his / her accounts, he/she gets the functionalities applicable to the account. In case a user has administrator privileges, he can design or edit workflows. This functionality is handled by the workflow design system:
  - In case the user is designing a new workflow, he has to go through the following modules :
    - The layout design modules enable the user to design the layout and connectivity of the workflow. This module works in connection with the **Database System.** Initially, during the design phase, the workflow

        design is stored in a temporary table. Once the design is complete, it is transferred to the permanent table in the database system.
        - The form design module enables the user to design the form layout to be associated with a workflow. This module consists of a form designing API. This API works with the **Database System** to store the forms in the form of XML documents.

    - In case a user is editing an existing workflow, the user can select the desired workflow and further select the layout or the form option. Depending upon the selection, the respective module gets initiated with the original design recreated in the module. The module will work in coordination with **the database system** and make changes at the appropriate places in the database tables.

- Process management system: This system works together with the **database system** and the **notification system.** It extracts the workflow details from the database system for the execution of successive tasks. Further, it interacts with the notification system and passes it the signals so that the notification system can notify users associated with a certain task when the task becomes active.
- Log management system: This system works in association with the **Database System and Process management system.** The process management system sends it the data regarding the activities taking place in the system. It then records that data into the **database system** and fetches the data whenever there is an auditing need.
- Notification system: This system works in association with the **database system** and the **process management system**. It receives signals by process management system and retrieves the contact details from the **database system** and sends the notification to the respective user.

3.3 Database Architecture

We chose a relational database Postgresql to store the data of the system. All process forms and registration forms of the users are stored in the xml form in the table only.

Overview of how data is stored and retrieved from the database:

- Tables storing employee data

1. Current_employees table: It contains those employees which do not hold any special position. It contains their userid, email_id and password.
2. por_table: This stores roles in the system. It contains the userid of the person holding this role, their userid, email-id of the role, password, email_id of the employee and date of joining of department. This table contains records of those people who are employees as well as holding special positions in the system.
3. Employees_info table: It contains information about all the employees involved in the system. It contains their first name, last name and XML form containing extra information filled during them while registration
4. Sys_management_users table: This table contains records of those people who are not employees of the organisation but work to manage the system and business processes of the organisation. It stores their email-id, password, post name and user id.

    - Functionality:

1. When a user logins into the system, the credentials of the user are matched against the Current_employees table, por_table and sys_management_user table. In case a user is found in the current_employees table, he is given access to the normal employee account. In case his record is found in the por table, he is given access to the special role account that he is handling .

- Tables for storage of workflows:

    1) Workflow table: It contains the workflow id and the description of the workflow.
    2) state table: The individual states of the workflows are stored in the state table. It stores the workflow id, state

id, email_id of the role handling that state, deadline duration of that state

3) transition table : the transitions of the workflows after completion of certain tasks are stored in the transition table. It stores the workflow id, the current state id and the next state id
4) Workflow_editor table: Contains the record of the users who are allowed to edit a certain workflow.
5) Workflow_visibility table: Records which posts can start a process based on a certain workflow

- Tables for storage of application:
    1) Application table: It stores the information about the applications like the application id, workflow id, date, timestamp, email-id of the initiator, current state id where the application is currently held, redirect-id in case the application is redirected to some arbitrary previous node.
    2) Application_action table: It stores the details regarding the responses taken by the applicable users on the application. It contains application id, post email-id of the responder, personal-email id of the responder, name, comments, final action, attachments and the timestamp of task activation and action taken.

4. **Application Domains**

4.1 DATA FLOW/CONTROL

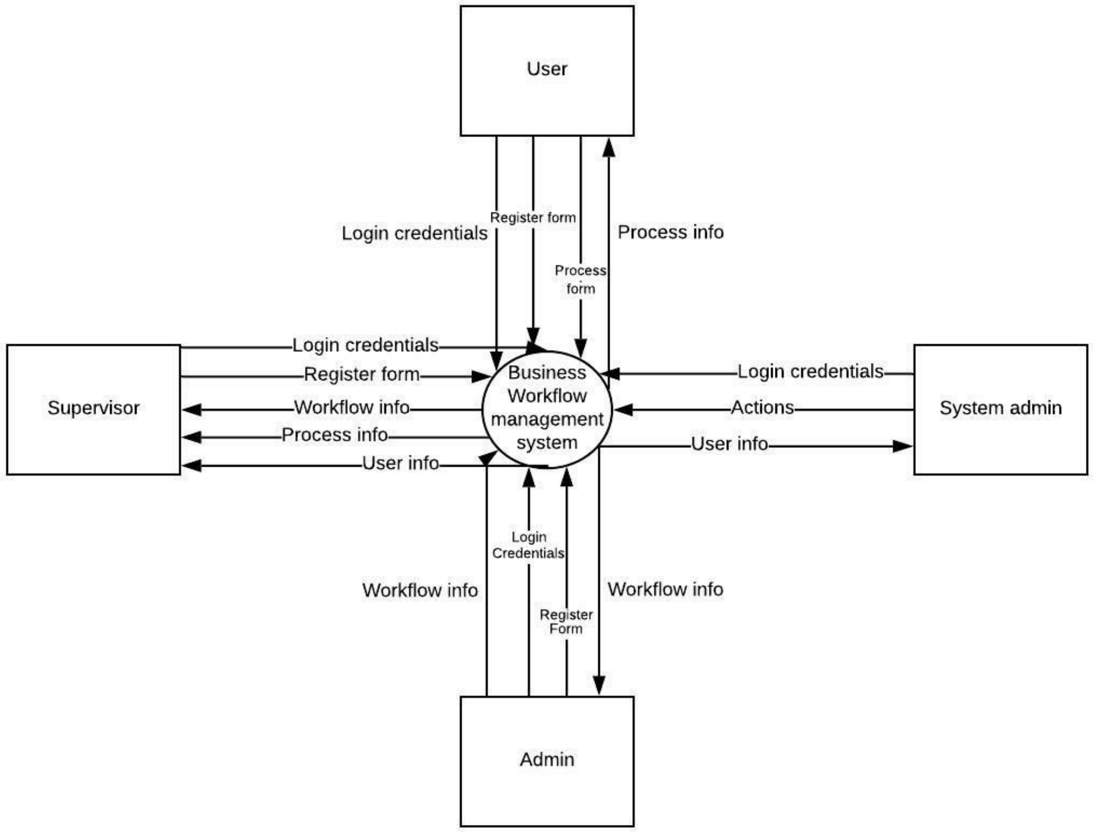


Figure 4.2.1 : LEVEL 0 CONTEXT LEVEL DFD FOR BFMS

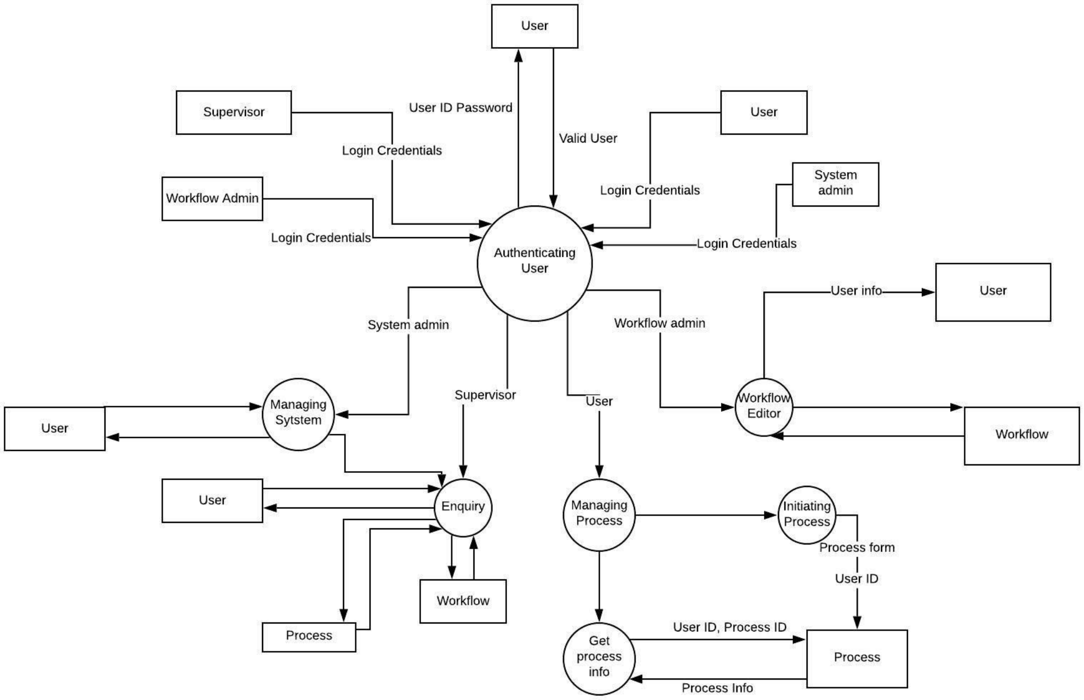


Figure 4.2.2 : LEVEL 1 DFD FOR BFMS

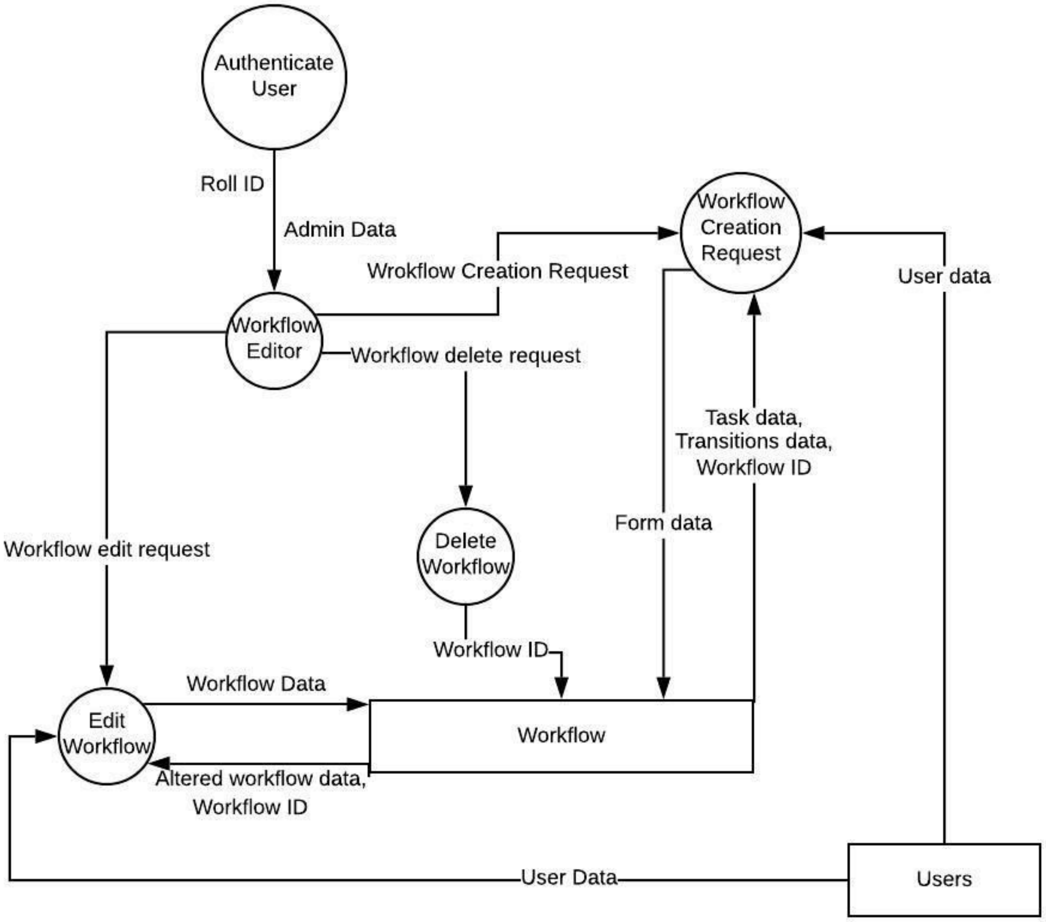


Figure 4.2.3: LEVEL 2 DFD THAT REFINES THE WORKFLOW EDITOR PROCESS

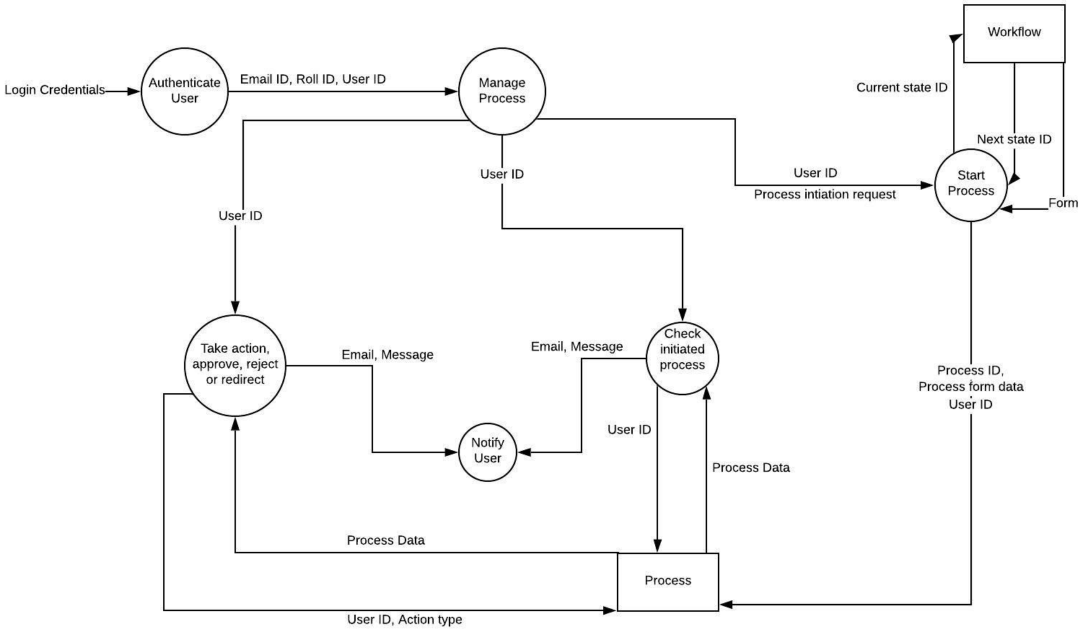


Figure 4.2.4: LEVEL 2 DFD THAT REFINES PROCESS MANAGER

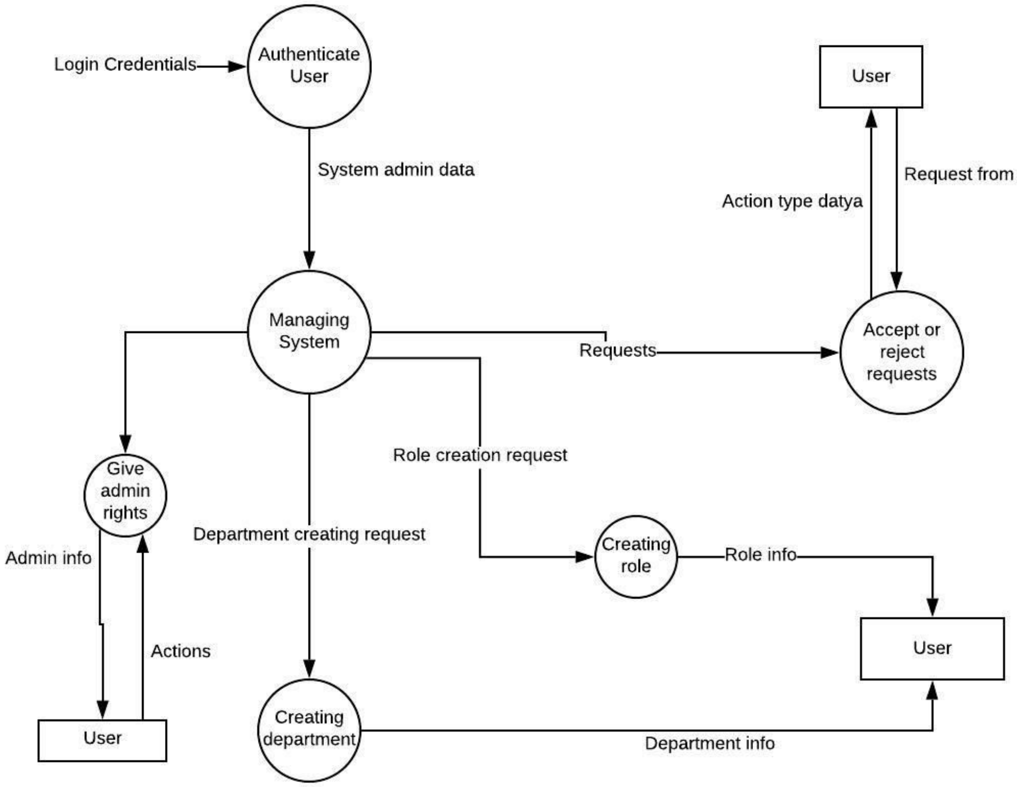


Figure : LEVEL 2 DFD THAT REFINES THE SYSTEM MANAGEMENT PROCESS

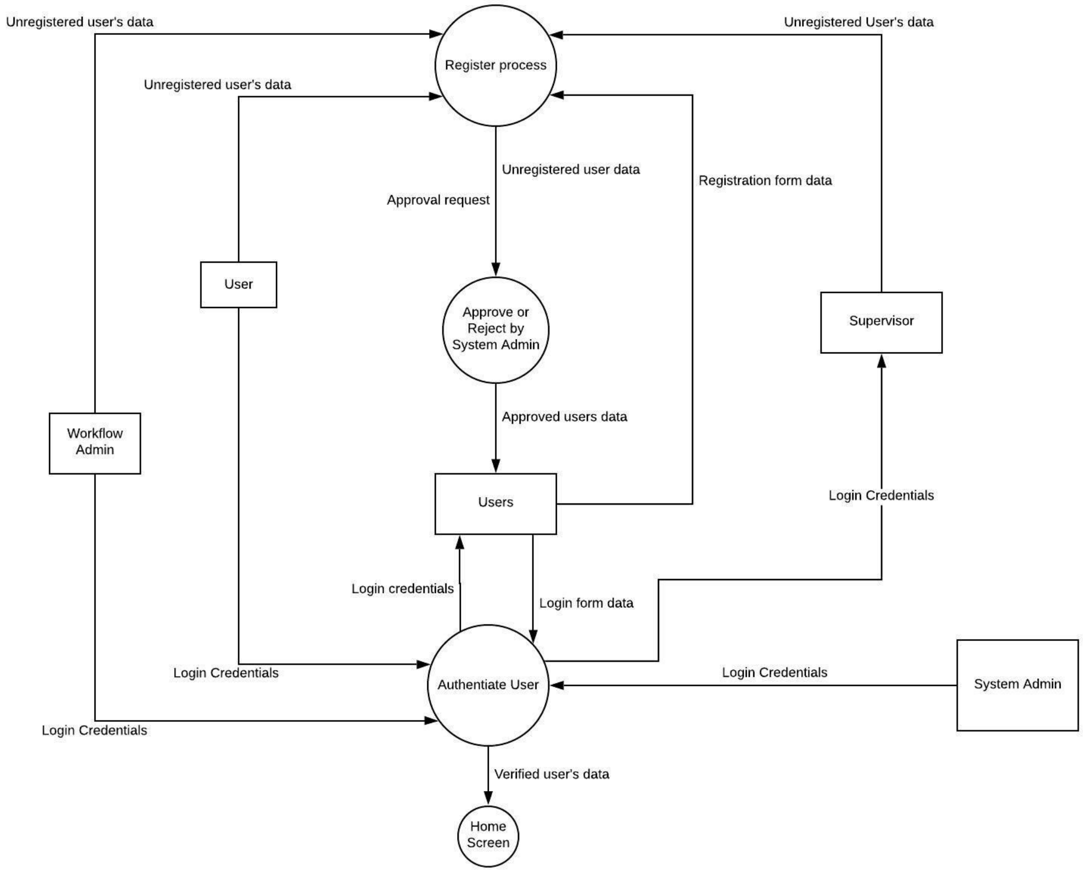


Figure: LEVEL 2 DFD THAT REFINES THE AUTH. AND REGISTRATION PROCESS

5. Data Design

5.1 DATA DESCRIPTION

Major entities in the system and description of the data related to them are as follows:-

Employee: Employees data include personal information of employers in the firm like Employers ID, email ID, password role ID, department name. This data is generated when the employers get a successful registration by filling the registration form and by getting it approved by the admin. This data is queried to verify the employer's login credential, during his authentication.

Workflow: It is the major entity in our system, it includes the data which defines the workflow. Workflow data includes data of every newly created/under creation workflows, their states, transitions between states and role at each state, conditions at each state. Each workflow is uniquely determined by its Workflow ID. This data is generated by the admin, only admin has the right to alter the data of workflow and create/edit/delete the workflow.

Processes: A process is basically an instance of the workflow. One or more processes can be started using a workflow. The attributes of the process entity will majorly include process ID, Workflow ID, process initiator details(email id, employer's id, etc.), application form data filled by process initiator, status of the application, every process has a set of states or nodes, the roles at each state and transitions in the process will be defined by the workflow. The action taken at each node/state will be defined by another entity called as process details. The process details will include the actions(approve, reject, redirect) taken at each state, attachment and comments attached by the role at that state, the deadline for action to be taken and the time when the action is taken.

Actions: It includes the actions like approve, reject, redirects that can be taken on an application by an authorised user.

Department: This data is defined by the admin, admin can add and delete a department.

Roles: This entity includes data of the role associated with the user, his/her email ID, role ID, the start and end date when he/she is associated with that role. This data is modified by the system administrator.

5.2 Static / Persistent Data

These data have static nature and have a tendency of not modifying frequently.

User login credentials and his personal information(department, user id, date of joining etc.) will be static. The password may change if the user forgets it and requests the system administrator to change.

Attributes of the registration form will be static once defined with the capability to be changed by system administrator in exceptional cases.

Workflow once defined by the admin will have static data. The roles at each state, transitions and conditions for each transition, duration of staying at a state and the attributes of the application form are unlikely to be modified frequently(though the system will provide the functionality of editing and deleting workflow) and can be considered as static data.

Data of type of action is persistent, only three actions will be provided to the user, approve, reject and redirect.

Data of type of role is persistent. For example a system administrator in a firm may define roles as, employer, department manager, vice president, president, HR. These roles are predefined in a firm and are unlikely to change in a firm. Though our system will have capability to add and delete roles under certain constraints.

Data of type of department is also static once defined by the system administrator. Though system administrator will have the right to modify in exceptional cases

Final status of the application (approved or rejected)

5.3 Dynamic Data

Following data have dynamic nature, these data may change periodically and frequently by external stimuli by any user or by the system administrator.

Status of the application of the user with respect to each role. It changes from(pending -> rejected/approved)

The user associated with the role may change periodically and hence user's data under each role will be dynamic.

5. User interface

The user interface of the software is designed keeping in mind the internal hierarchical structure of the organisations. We have also kept in mind that the interface must be made so that a user can operate this software even with negligible knowledge of the underlying functioning components.

5.1 UI overview

The front-end design of this software consists of the following components :

- A login/signup page, through which a user can get access to his account. Once he logs in, he is redirected to the landing page.

- The landing page displays a navigation bar and a drop down menu containing a list of tabs. The tabs which are shown depend on the role of the user. The navigation bar shows the path that the user has taken to reach the current page. Other than that, the landing page also displays the list of pending tasks and the status of processes initiated by the user.

- The tabs available to a user depends upon his role :

  - System administrator :

    - Account creation requests: Opens the page containing a new request for account creation.

- Manage :
  - Department : Create new departments and handle other department related options
  - Role: Create a new role, edit privileges of existing roles
  - Accounts: Reset account profile and password or delete an account.

- Profile :
  - Edit profile: Change Name, Password and other details.
  - Notifications : Edit the preferences regarding notifications like notification mediums, details regarding notification mediums etc.
  - Logout: logout from the system.

➢ Administrator :

- Tasks :
  - Pending tasks : Check the pending tasks at your level and details about them like deadlines etc.
  - History : Check the history of the responses given by the user on different tasks.

- Workflow :
  - Design workflow : Redirects to the workflow design screen which contains drag and drop interface for designing workflows. It also

contains other options to configure individual tasks of workflows. It further contains a form designer to create form for the workflow.

        - Edit workflow : Edit the design, task configuration or form of existing workflows.

- ■ Processes :

    - Initiate process : Opens the process initiate page where you can select a workflow, fill a form and start a process based on that workflow.

    - History : See the history of the processes initiated by you, their outcomes, your responses in form, their current status etc.

- ■ Profile :

    - Edit profile : Change Name, Password and other details.

    - Notifications : Edit the preferences regarding notifications like notification mediums, details regarding notification mediums etc.

    - Logout : logout from the system.

➢ Department head :

- ■ Tasks :

    - Pending tasks : Check the pending tasks at your level and details about them like deadlines etc.

    - History : Check the history of the responses given by the user on different tasks.

    - Process :
        - Initiate process : Opens the process initiate page where you can select a workflow, fill a form and start a process based on that workflow.
        - History : See the history of the processes initiated by you, their outcomes, your responses in form, their current status etc.
    - Department employees : See the employees in your department and redirect the processes in which they are involved to other employees in case they go on leave etc.
    - Profile :
        - Edit profile : Change Name, Password and other details.
        - Notifications : Edit the preferences regarding notifications like notification mediums, details regarding notification mediums etc.
        - Logout : logout from the system.
- Supervisor : For supervisor, the landing screen would consist of widgets pertaining to each department. Upon selecting the required department, the supervisor will get the following tabs :
    - Workflow : Inspect the logs created regarding the workflows existing in the system
    - Processes : Inspect the logs created regarding the actions taken during the course of each process, whether running or completed.
    - Employees : Inspect the logs regarding the actions taken and other details pertaining to the employees having an account in the system.
- Normal employee :
    - Tasks :
        - Pending tasks : Check the pending tasks at your level and details about them like deadlines etc.

        - History : Check the history of the responses given by the user on different tasks.

    - Profile :

        - Edit profile : Change Name, Password and other details.

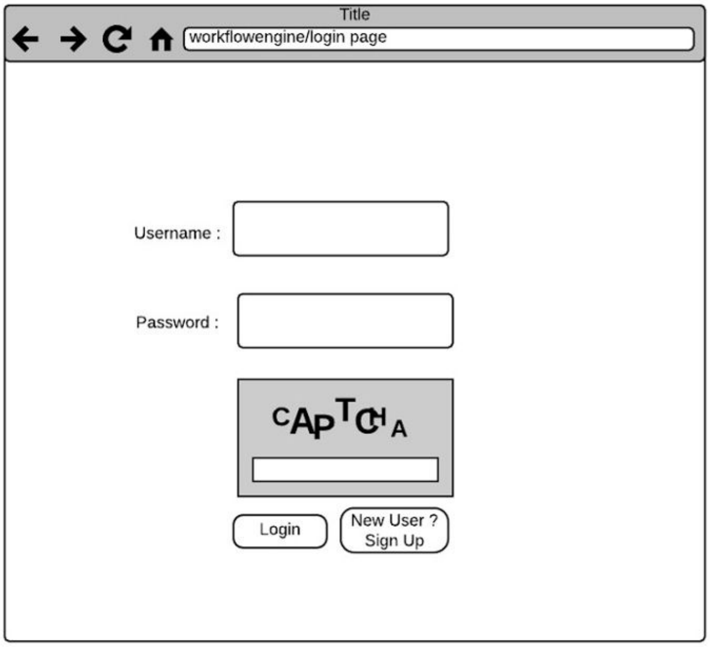
        - Notifications : Edit the preferences regarding notifications like notification mediums, details regarding notification mediums etc.

        - Logout : logout from the system.

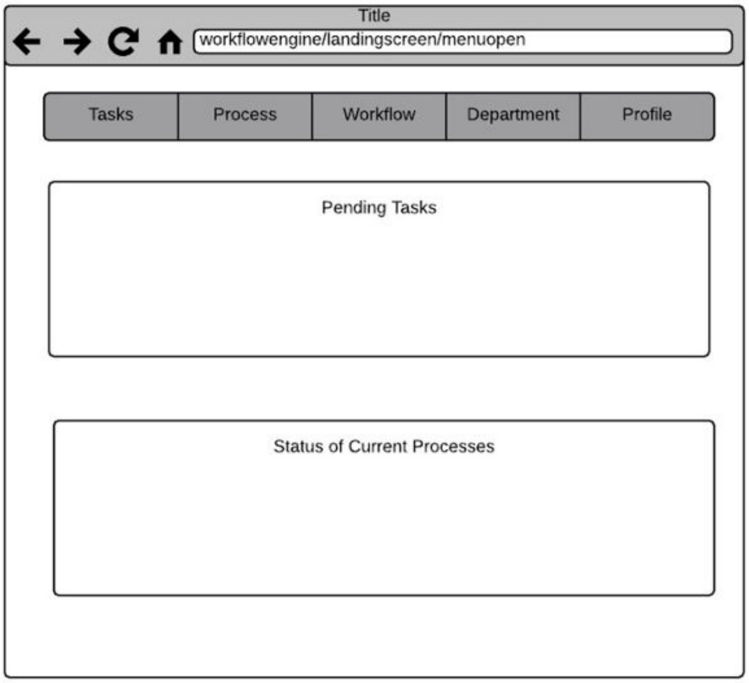
6.2 Navigational Flow

6.3 Function / Screen Usage

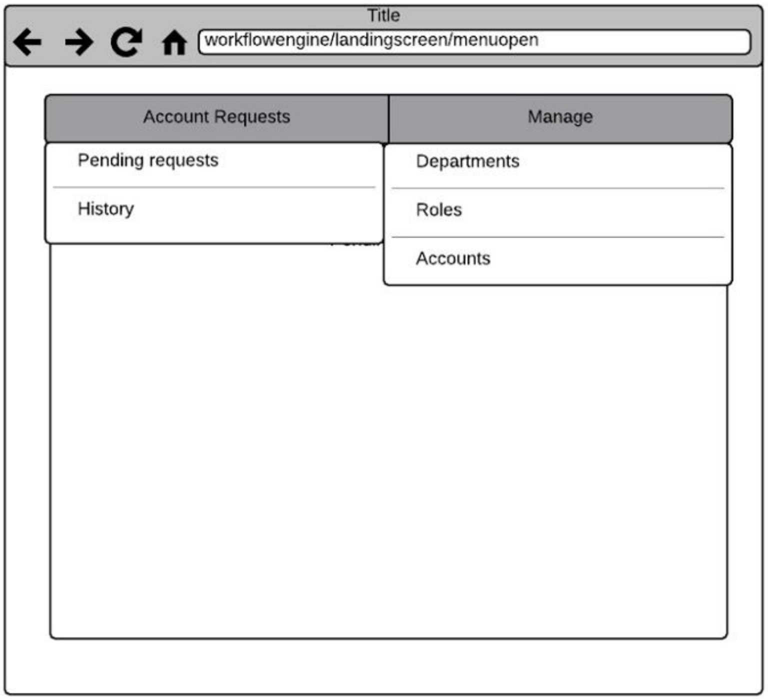
The following diagrams illustrate screen usage more clearly :



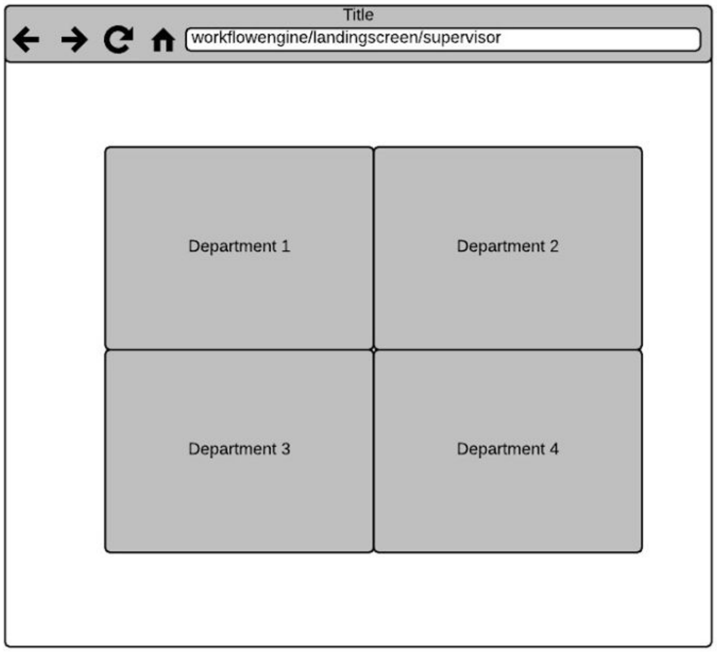
Login Screen


Title
workflowengine/landingscreen/menuopen
Tasks
Process
Workflow
Department
Profile
Pending Tasks
Status of Current Processes

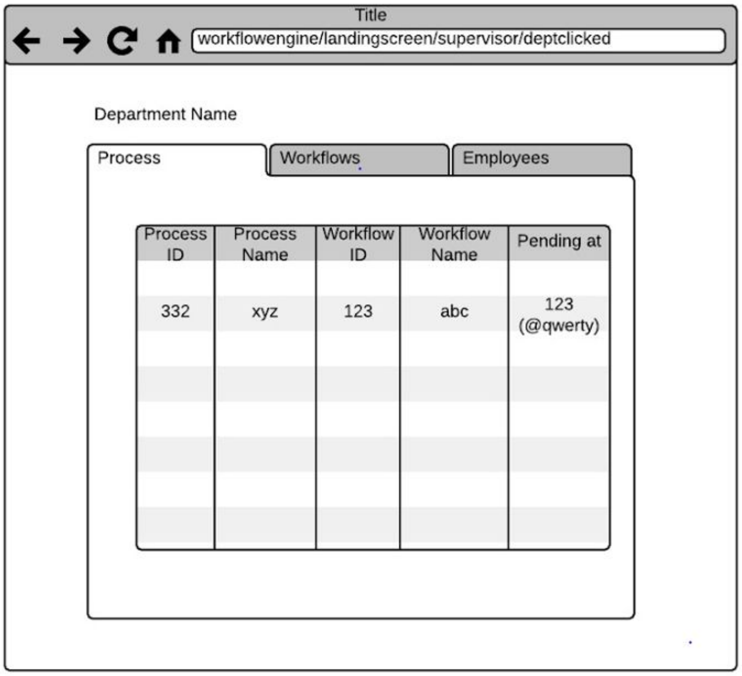
Landing Screen



Landing screen for system administrator

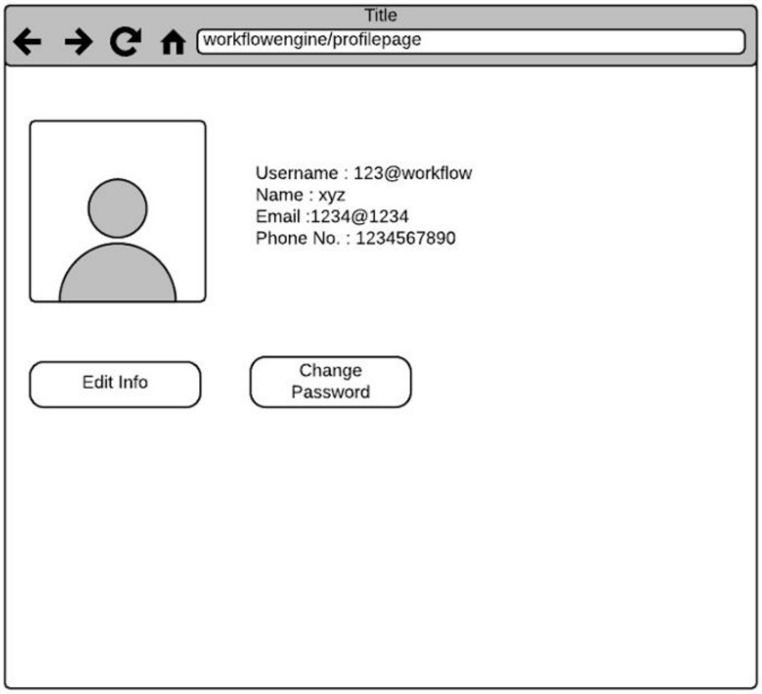


Landing screen for supervisor



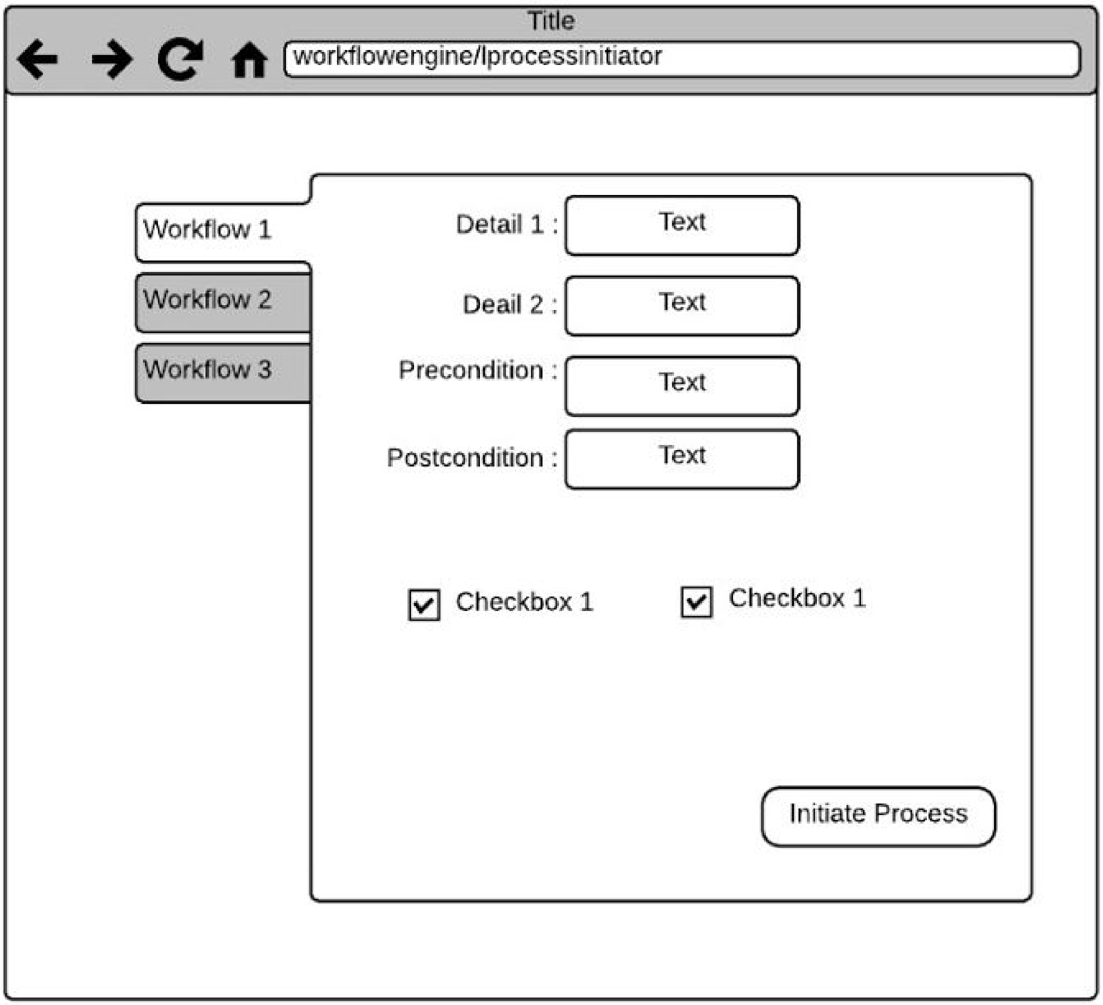
Process ID	Process Name	Workflow ID	Workflow Name	Pending at
332	xyz	123	abc	123 (@qwerty)

Auditing screen



Username : 123@workflow
Name : xyz
Email :1234@1234
Phone No. : 1234567890

Profile screen



Detail 1 : Text

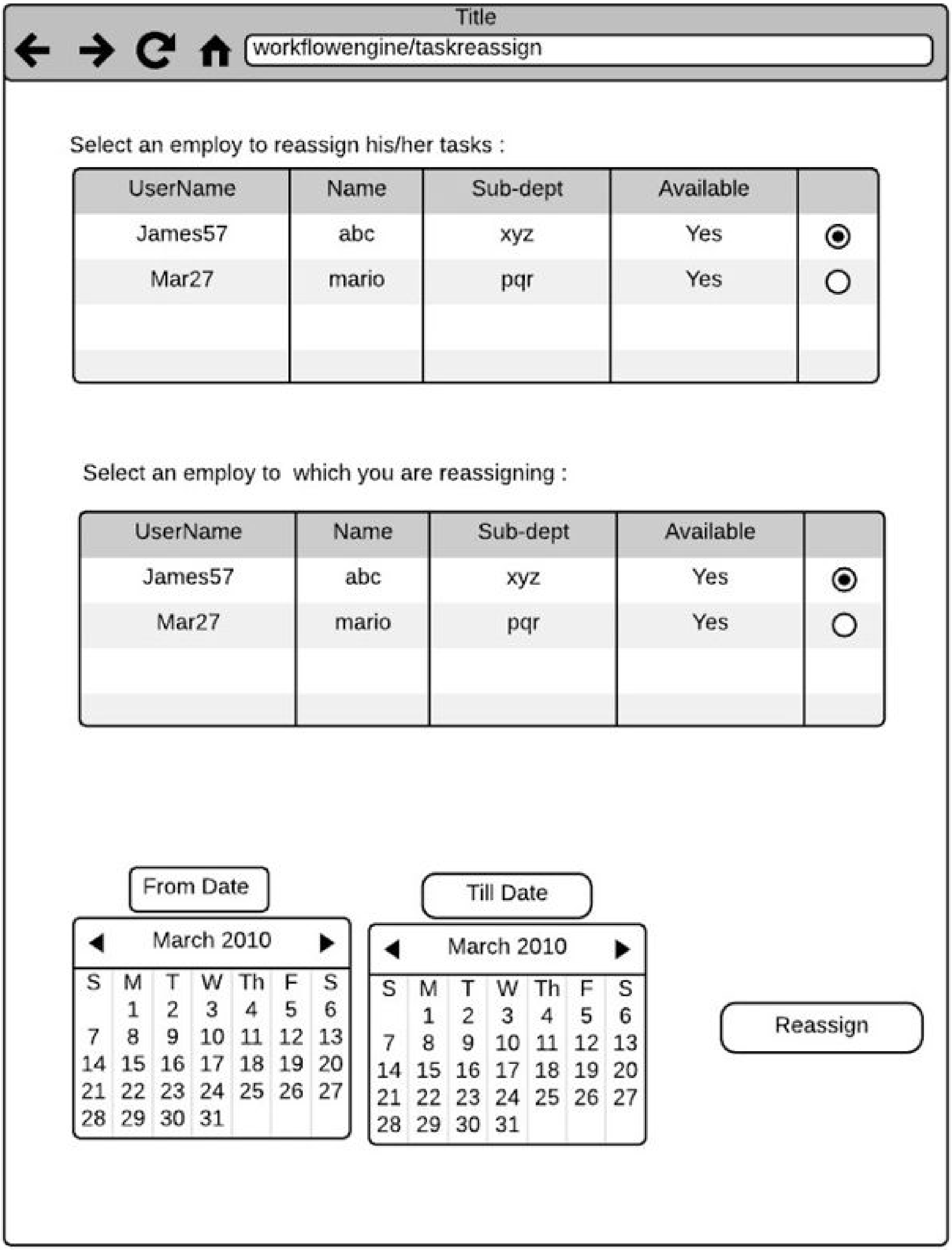
Deail 2 : Text

Precondition : Text

Postcondition : Text

☑ Checkbox 1 ☑ Checkbox 1

New process initiation screen



Select an employ to reassign his/her tasks :

UserName	Name	Sub-dept	Available	
James57	abc	xyz	Yes	●
Mar27	mario	pqr	Yes	○

Select an employ to which you are reassigning :

UserName	Name	Sub-dept	Available	
James57	abc	xyz	Yes	●
Mar27	mario	pqr	Yes	○

◀ March 2010 ▶

S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

◀ March 2010 ▶

S	M	T	W	Th	F	S
	1	2	3	4	5	6
7	8	9	10	11	12	13
14	15	16	17	18	19	20
21	22	23	24	25	26	27
28	29	30	31			

Redirect task screen

5.0 Design decisions with their advantages and disadvantages

Following are the design decisions that we made along with their foresighted advantages and disadvantages:

User Interface

Login / Signup page :

- ➢ We decided to directly show the login form when the user accesses a fresh session. We give a link to signup page separately in the login page itself instead of asking the user to choose from login page or signup page.
    - Advantages : Since most of the time user will be logging into his / her account and sign up requests are very less in number as compared to login requests, we decided to direct to the login page as soon as a session from the server is activated
    - Disadvantages : None so far
- ➢ We decided to add a captcha based turing test for the login process.
    - Advantages : Prevents automated scripts to create login / signup requests.
    - Disadvantages : Adds additional steps in the login process.

Landing page :

- ➢ We decided to use drop down menus in the form of tabs containing different sub options available to the user. The contents of the tabs have been carefully designed so that the maximum depth of the menu is 2.

    - Advantages : Since the tabs are not much in number (maximum 4 for administrator role), there is no need to arrange the tabs in a bar. The length of the menu is short enough so as not to deform the view. The depth of menu is kept limited so that there is no problem in navigation. Also in future if system is extended, more options can be easily included into the interface

    - Disadvantages : This adds a bar to the top of the screen. Some of the space may be wasted and in worst case

- ➢ We decided to display pending tasks and status of the running processes on the landing screen

    - Advantages : These are the two most important entities for which the user will be using the system and they will be visible to the user as soon as he logs into his / her account.

Auditing page :

- ➢ We decided to display different departments section wise at the landing screen of the auditing page. Audited information regarding the matters of that department is displayed after clicking on the link of that department

    - Advantages : Displaying information department wise will help user to easily find the record that he / she is searching for. Displaying all the information together will clutter the space and make searching difficult.

    - Disadvantages : To search a record, the user will have to go by the department. This may become a bit difficult in case user does not has idea about the department of the entity that he / she has been looking for.

Workflow designer :

- ➢ We decided to incorporate a drag and drop based workflow designer and form designer so as to facilitate the users who have no coding background and make the experience seamless. The nodes of the workflow will be interactive so that the user can configure individual nodes easily.
    - Advantages : The user can easily design whatever workflow he needs
    - Disadvantages : This adds up a whole new process designer component into the system.

Database :

- ➢ We decided to use PostgreSQL and MongoDB as the database client. PostgreSQL is a relational database while MongoDB is based on NoSQL.
    - Advantages : Indexing, auditing and authentication will be facilitated by PostgreSQL while other extra optional information that the system may need to store will be facilitated by MongoDB

Miscellaneous

- ➢ We decided that there needs to be a criteria of approval of new account request by the system administrator instead of the usual user driven direct account creation through sign up.
    - Advantages : This prevents creation of unnecessary and arbitrary accounts by people.
    - Disadvantages : A user will have to wait for his / her request to be approved by system administrator, and system administrator will have to approve or reject request after manually reviewing them.

➢ We decided to add a classification attribute ‘department’ to the entities in our system.

  - Advantages : Since in most corporate or educational institutions people as well as processes are associated with certain department (like mess tender, IT equipment acquisitions), we decided to introduce the department based classifications. This makes it easier to query the information

  - Disadvantages : Every entity needs to be associated with a department, so there may arise a need to create a miscellaneous department for general entities.

➢ We decided to limit the auditing power to supervisors only

  - Advantages : Only certain authorised users will be able to access critical information. This way the possibility of information leakage is reduced

  - Disadvantages : It makes it necessary to approach a supervisor in case some information needs to be extracted.

➢ We decided to introduce a role of department head into the system and vested into it the privilege to reassign tasks for certain period of time. Moreover if a deadline is missed the department head is sent a notice.

  - Advantages : In case an employee goes on leave or is unable to tend to a task, the tasks due on him can be reassigned to other employee and the department head can supervise his / her employees.

  - Disadvantages : This role needs to be handled by some person for this functionality to work.

➢ We decided to make a notification system component in the system.

- Advantages : The employees are automatically send the notifications, therefore they do not require to click the buttons again and again to check the things out.
- Disadvantages : This role needs to be handled by some person for this functionality to work.

6. Software Testing:

6.1 Types of tests to be conducted

The following types of tests need to be performed on the system:

- Unit testing : To verify if the authentication subsystem, process and form designer module and notification module are working as expected.

- Integration testing : To take unit tested components and interweave them to produce the system structure as dictated by the proposed design and verify the functioning of the workflow engine as a whole.

- Regression testing : To make sure that as new components are added with successive releases, the whole system keeps working as expected.

- Performance testing : This would be done to verify that the system meets the designated time constraints and effectiveness criteria.

- Stress testing : During this phase we need to verify if the system is robust enough to handle large workload and pressure of a large number of requests.

- Alpha testing : Before the final release of the software, the system needs to be validated.

7.2 Expected software response

Following are the expectations from the software:

- Access control systems must be able to reliably verify user credentials within a reasonable time frame. The implementation of captcha based turing test must be tested to produce captchas within 1s of being accessed

- The user interface must display the required information and the system must process requests within reasonable time limits. The round trips for data fetching from the database component must be minimised.

- The drag and drop based workflow designer must be organised so that it is user-friendly with all tools categorised properly and within the sight of the user

- The notification system must auto – generate and deliver notifications within 1 min of assignment of tasks.

- The process manager must activate a task only if its preconditions are satisfied.

- All the data must also be backed up simultaneously so that there is no loss in case of power failure or so.