Automated Workflow System (AWS)

Software Requirements Specification

Version 1.0
Status: Draft

Prepared by Abhineet Pandey, Divesh Dodeja, Karan Sehgal & Vineet Madan
Group-10

CS-305, Spring 2020

January 29, 2020

Table of Contents

1. Introduction
- 1.1 Purpose
- 1.2 Document Conventions
- 1.3 Intended Audience and Reading Suggestions
- 1.4 Product Scope
- 1.5 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
- 2.6 User Documentation
- 2.7 Assumptions and Dependencies

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

4. Domain Model

5. System Features (Use Cases)
- 5.1 Use Case: Creation of Form and Workflow
- 5.2 Use Case: Creation of Roles
- 5.3 Use Case: Creation of Teams
- 5.4 Use Case: Assigning Roles/Teams
- 5.5 Use Case: Admin Password Change
- 5.6 Use Case: Update Role
- 5.7 Use Case: Accessing Logs (Admin)
- 5.8 Use Case: Filling the form - Initiation
- 5.9 Use Case: Filling the form (At Intermediate Node)
- 5.10 Use Case: View Status of Form Instance

6. Other Nonfunctional Requirements
- 6.1 Performance Requirements
- 6.2 Safety Requirements
- 6.3 Security Requirements
- 6.4 Software Quality Attributes

7. Other Requirements

Revision History

Name	Date	Reason For Changes	Version
Group-10	29 January 2020	Initial Version	1.0

1. Introduction

1.1 Purpose

This Software Requirements Specification (SRS) aims to provide a comprehensive description of the specifications and functions of the "Automated Workflow System", hereon referred to as 'AWS', which is a configurable system that helps in automating a business workflow. This document will illustrate the purpose and detailed overview of the development of the system.

1.2 Document Conventions

The document uses the following abbreviations.

DB	Database
ER	Entity Relationship
SRS	Software Requirements Specification
Admin	Administrator
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol *Secured*
SQL	Structured Query Language
QA Team	Question Answering Team
API	Application Programming Interface
JSON	JavaScript Object Notation
CSS	Cascading Style Sheets
HTML	HyperText Markup Language
JS	JavaScript
CSRF	Cross-Site Request Forgery

1.3 Intended Audience and Reading Suggestions

The intended audience includes the management of the firm using the software, the management, finance, and marketing teams of the organization responsible for selling the software, the users and the developers. The reading sequence suggested is the same as the sequence in Table of Contents.

ID	Stakeholder	Description
S-1	Customer	Checking if the requirements of the business align with the proposed functionality of the product
S-2	Development Team	To plan the further trajectory of the project implementation, using detailed functional and non-functional requirements
S-3	QA team	Designing test-cases to test the suitability of the software to a real scenario
S-4	Finance and Planning team	To estimate the financial resources required and the timeline for the project
S-6	User	The Privacy Policy and Terms of Services are created based on this document.
S-7	Marketing Team	To find relevant clients having requirements similar to ones served in AWS.

1.4 Product Scope

The main purpose of AWS is to help businesses create customizable workflows i.e. series of processing steps to apply on certain data form(s). This system would not only help in increasing the efficiency of basic business workflows but also increase the productivity of the employees of the corporation by creating an interactive and secure interface to categorize and keep track of multiple business workflows. For a business, the product would provide the authorized personnel to create custom data forms required for a specific workflow. The personnel would also be provided with a detailed view of workflows and the activities of users within the workflow, thereby creating a transparent system.

1.5 References

[1] Ullman, Garcia-Molina and Widom, *Database Systems, The Complete Book.* Pearson, 2009.

2. Overall Description

2.1 Product Perspective

The product is a replacement for traditional offline business workflow practices, where employees are required to process and forward data forms to respective business personnel in-person or through an intermediary. The product would provide a business with interactive web-based tools to automate the previously stated process accompanied by monitoring tools to make the process secure, transparent and convenient.

2.2 Product Functions

The product allows the employees of a corporation to log-in to a portal where they can search and fill out an application or a data form for a specific task. The said employee would also be able to keep track of the submitted form within its respective workflow.

The workflow and the blueprint of a form (i.e. the data fields of the form) can be created by the authorized personnel through the product. The creator would be given several features to customize workflow and the corresponding form. For a workflow, the features include creation and deletion of a node (i.e. a step in the series of steps) in the workflow, whereas, for a form, the features include creation and deletion of a data field, setting a type of the data field (such as text, blob, radio button, checkbox, etc) and managing the permissions of the form (which field can be viewed or filled-out by which individual).

In a specific workflow, an employee would be able to manage the applications forwarded by them, as well as, the applications forwarded to them, according to the respective permissions granted to the employee. The employee is allowed to send the application to preceding nodes with comments.

2.3 User Classes and Characteristics

ID	User Classes	Description
U-1	Administrator	A user who has rights to validate other users after they complete their sign up, and owns expanded rights in the portal. The administrator assigns the role to each user who is validated.
U-2	Employee	A user of the software who has completed the signup and has been validated, and assigned a role by the administrator.
U-3	Not signed up user	This user doesn't have any rights in the portal.

U-4	Authorized Employee	A user who has certain permissions and privileges to perform create and update operations for specific cases.

2.4 Operating Environment

The initial version of the software should be deployed on a Linux machine with both SQL and NoSQL databases installed. A python and node development environment should also be available with permissions for installing open-source modules and internet access.

2.5 Design and Implementation Constraints

- The user-interface will be built using HTML, CSS and JavaScript and its libraries.
- The product will be developed using Django, a python based web framework.
- The standard data exchange format will be JSON.
- SQL and NoSQL will be used to manage the database.
- It uses a modular design where every feature is wrapped into a separate module and the modules depend on each other through well-written APIs.

2.6 User Documentation

A well-documented user manual will be delivered along with the software, which will best describe the steps to create a form. Its access will be given to only those who have access to create a new form.

2.7 Assumptions and Dependencies

The software doesn't handle the cases where the form is supposed to go from a higher position in the hierarchy to a lower position in the hierarchy, as it is reasonably assumed that in most cases the form flow is from lower positions to higher ones.
The software doesn't handle the cases where workflow explicitly has a loop, as it is also a rare and unforeseeable case. All the workflows must be linear in nature. However, the 'send back' feature is given, as explained in section 2.1.
The software handles the upload of small files only in the forms.
For the purpose of the workflow of a form, it is assumed that each employee is a part of only one team, which may or may not have a parent team or sub-teams.

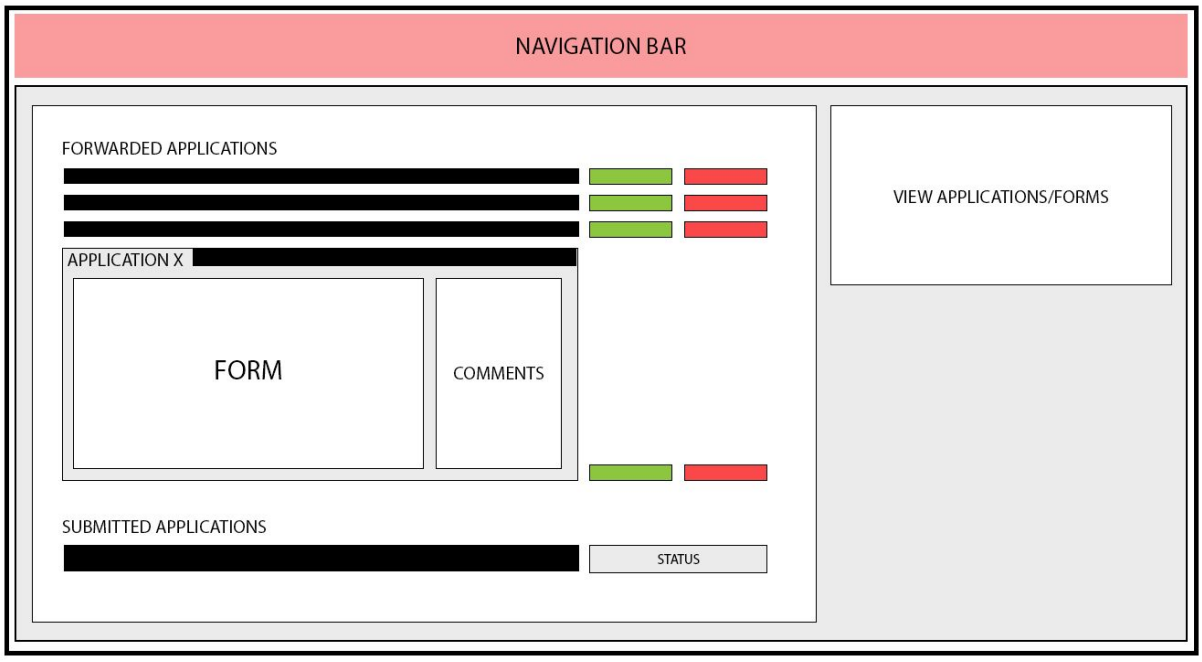
3. External Interface Requirements

3.1 User Interfaces

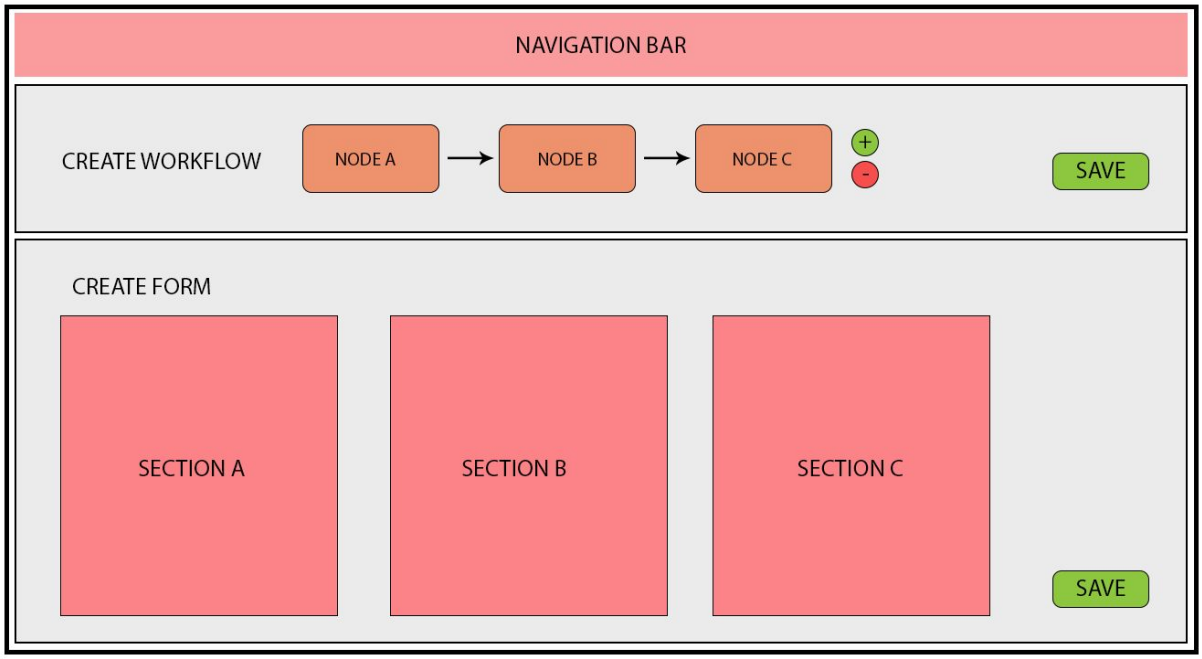
A first time user should see a log-in or sign-up page when they open the application. The first time the user is expected to sign-up before using the application. Since the application is focused on business or organization domain, the admin would be responsible for approving the sign-up requests for the system.

A signed-in employee should see the necessary information to help them navigate the application, this would be provided to them through a navigation bar. The navigation bar would contain several options such as dashboard, search for a form, logout, etc.

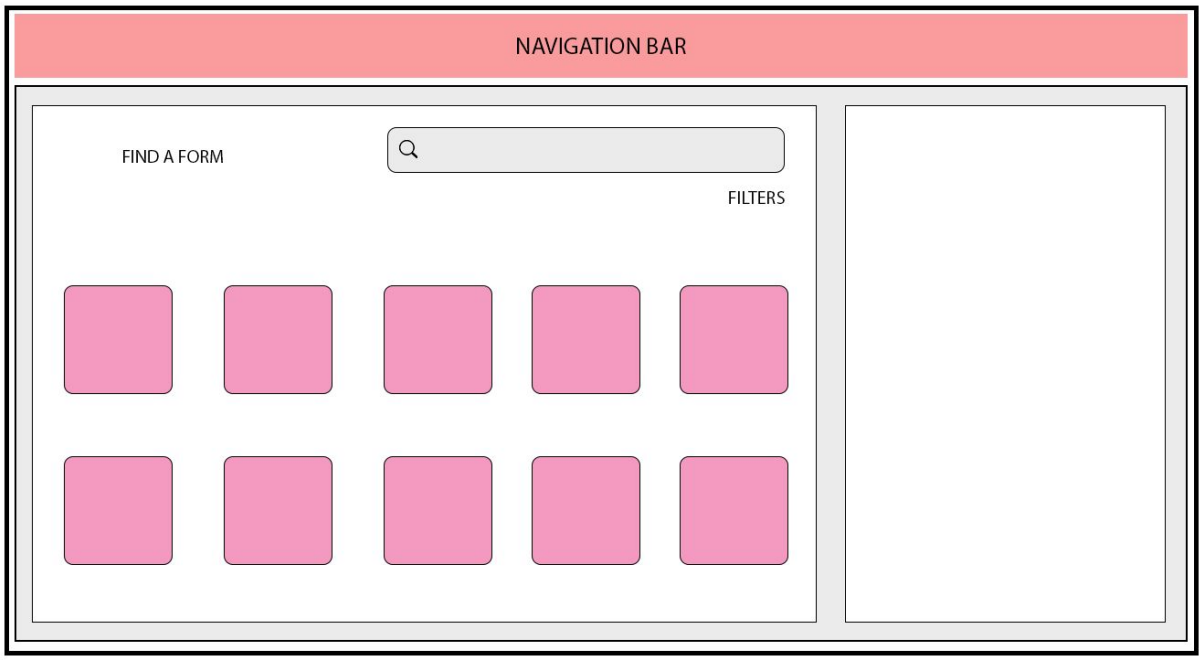
In the dashboard, a signed-in employee should see the necessary information regarding the applications/forms that they forwarded to the respective personnel, as well as, the applications/forms that were forwarded to them. The employee should have an option to approve, deny or comment on a given application, as well as view the status of the forwarded application.



Authorized business personnel should be able to create a business workflow that would act as a route for a new application/form, whose data fields can be created only after the workflow is fixed. This is because certain fields would only be visible or could only be filled by certain individuals who are a part of the workflow.



An employee should be able to search for a form that they are required to fill. The employee would have the flexibility to filter through multiple forms by the name and the category of the form (i.e. accounting, human resources, etc). After selecting the form, the employee would fill in the required details and forward it to the respective personnel.



3.2 Hardware Interfaces

Since the web portal does not have any designated hardware, it does not have any direct hardware interfaces, except the hardware the web portal would be accessed through, like a computer or a mobile device.

3.3 Software Interfaces

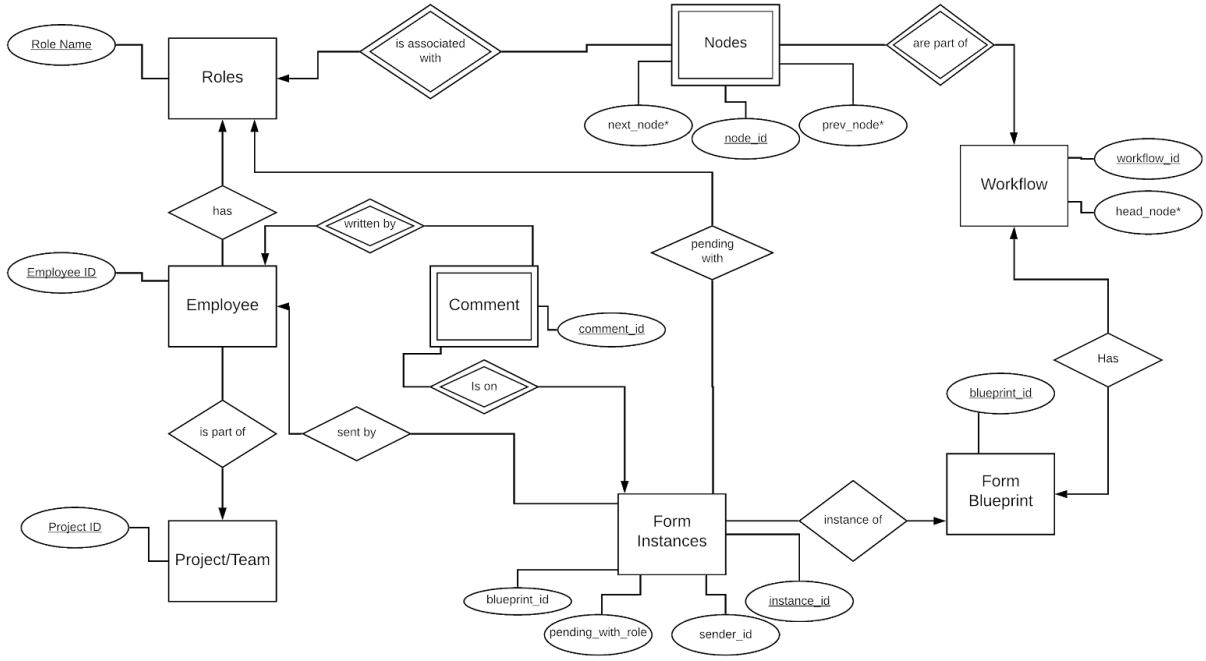
The application communicates with both the SQL and NoSQL based databases in order to add, retrieve, update and delete relevant information.

3.4 Communications Interfaces

Communication between different parts of the system is through HTTP or HTTPS protocols.

4. Domain Model

Following is the Entity-Relationship diagram of the software. The convention followed is derived from reference [1].



5. System Features (Use Cases)

5.1 **Use Case:** Creation of Form and Workflow

Brief Description:	Authorized personnel will be able to create a new form. For example, a person in the accounts team will be able to create a form which is to be used for reimbursement. While creating the form, the person will define its workflow(route) and add fields. This person will also divide the form into sections. A certain section in the form can be edited only by the designated person in the workflow path.
Business Trigger:	The need for introduction of a new form (workflow)
Preconditions:	The person creating the form should be authorised to do so by the admin

Basic Flow: An authorised person (referred to as the user from now on) opens the page for creating a form. The first task in this use case is to set the workflow(route) for this form. The number of nodes in the workflow is dynamic. At each node, the user selects the post pertaining to that node.

A form has two general attributes. A string attribute which is the title of the form. Secondly, a boolean attribute which marks if the form is active or not.

Once the flow is saved, the user starts adding the fields of the form. Each field has three main attributes i.e label, type (int, string, file, etc.) and the section this field belongs to. While creating the form, the user will first add all fields of the first section, then the second and so on.

Once all the fields have been created, the form is saved and becomes public if it is marked as active.

Assumptions: The workflow for a form cannot be altered once saved.

Line	System Actor Action	System Response
1	The college administration asks the user to create a new form for some workflow. For example, a staffer in accounts department is asked to create a new form for reimbursement.	The system asks the user for authorization.
2	The user uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the user can take including the ‘create form’ action.
3	The user clicks on the ‘create form’ link/button.	The system redirects the user to the ‘create form’ page.

4	The sets the workflow i.e adds nodes and sets the posts and saves this workflow.	The system prompts the user for title of the form.
5	The user sets the title and saves the progress so far.	The system prompts the user to add a section to the form.
6	The user sets the section title and selects the role of the person who can fill the section and saves the section.	The system prompts the user to add certain data fields in the section
7	The user sets the attributes (label, type and section) of the field and saves the current field added.	The system prompts the user to add another field or section.
8	The user keeps on adding fields or sections until done and saves.	The system prompts the user for the 'active' attribute.
9	The user selects whether to make the form visible or not and saves.	The system redirects the user to the home page.
Post Condition:	1. The form object is created in the database. 2. Transaction creating the form is logged in the database.	

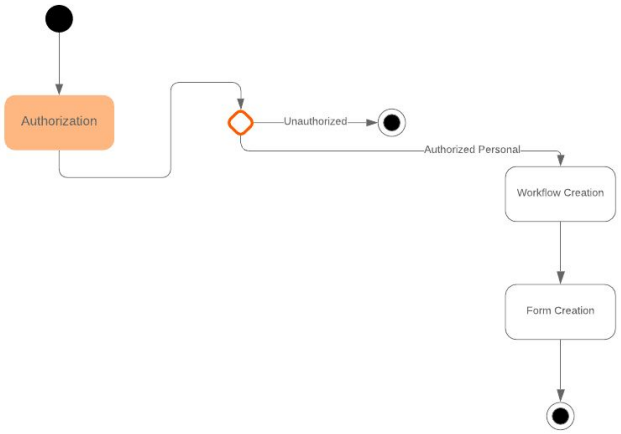
Business Rules:

1 The workflow should not contain a loop, hence after submission of a workflow, the application would detect the possibility of a loop, and modify the database only when the workflow is acyclic.

Data Requirements:

1. Only those who are authorised to create a form should be allowed to do so. The data of the authorized personnel is required.

Activity Diagram:



5.2 **Use Case:** Creation of Roles

Brief Description:	The admin (superuser) will be able to create new roles in the organisation.
Business Trigger:	A new role is introduced in the organisation.
Preconditions:	Only the admin will be able to create a new role. Admin authorisation is a pre condition.

Basic Flow: The admin opens the page for creating a role. The role object has a name attribute. The admin sets the name attribute for the role and saves the role to the database.		
Assumptions: 1. A role once created cannot be deleted. 2. The names of roles must be unique.		
Line	System Actor Action	System Response
1	The admin gets a request to create a new role in the organisation from competent authority.	The system asks the user for admin authorization.
2	The admin uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the ‘create role’ action.
3	The admin goes to the ‘create role’ link.	The system serves the ‘create role’ form on a page. The form has ‘name’ field.
4	The admin sets the ‘name’ field and saves the role.	The system saves the role object to the database if there is no exception.
Post Condition:	1. The role object is saved to the database. 2. The transaction creating the role is logged in the database.	

Non Functional Requirements:
1. The roles created must be defined as such within the organisation and not an arbitrary choice by the admin.

Data Requirements:
1. The data of the roles that exist within the organisation must be provided to the admin.

Screen Entry Exception Table:	
Exception	**Response**

1. Name of new role being created is the same as an already existing one.	Message: “A role with the same name exists. Role creation was not successful.”

5.3 **Use Case:** Creation of Teams

Brief Description:	The admin (superuser) will be able to create new teams in the organisation.
Business Trigger:	A new team is introduced in the organisation.
Preconditions:	Only the admin will be able to create a new team. Admin authorisation is a pre condition.

Basic Flow: The admin opens the page for creating a team. The team object has a parent team id attribute, name attribute, an id attribute (auto primary key) and a current status boolean attribute which is true if team is currently active . The admin sets the name attribute for the team and the parent team id attribute and saves the team object to the database.

Assumptions:
1. A team once created cannot be deleted. The team can only be marked as active or inactive.
2. A team’s name must be unique.
3. There exists a tree hierarchy kind of structure within the organisation. Every team has a parent team except the ‘root’ team. For example, the board of a company can be a root.

Line	System Actor Action	System Response
1	The admin gets a request to create a new team in the organisation from competent authority.	The system asks the user for admin authorization.
2	The admin uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the ‘create team’ action.
3	The admin goes to the ‘create team’ link.	The system serves the ‘create team’ form on a page. The form has ‘parent team id’ field, ‘name’ field and ‘current status’ field.
4	The admin sets the ‘parent_team_id’, ‘name’, ‘current status’ and saves the role.	The system saves the team object to the database if there is no exception.

Post Condition:	1. The team object is saved to the database. 2. The transaction creating the team is logged in the database.

Data Requirements:
1. The names of the already existing teams should be visible to the admin.

Screen Entry Exception Table:	
Exception	**Response**
1. Name of new team being created is the same as an already existing one.	Message: "A team with the same name exists. Team creation was not successful."
2. If the parent team id is invalid.	Message: "A team with the parent team id does not exist. Team creation was not successful."

5.4 **Use Case:** Assigning Roles/Teams

Brief Description:	The admin (superuser) will be able to assign roles and teams to an employee.
Business Trigger:	1. A new employee joins the organisation 2. An existing employee is assigned a new role or a team within the organisation.
Preconditions:	Only the admin will be able to change the team or role of an employee. Admin authorisation is a pre condition.

Basic Flow: The admin opens the profile page of the concerned employee and goes to the edit profile page for that employee. The admin sets the new team and role for the employee and saves.		
Assumptions: 1. An employee can be assigned only one team at a time. 2. Team must not be null. 3. An employee can be assigned only one role at a time. 4. Role must not be null.		
Line	System Actor Action	System Response
1	The admin gets a request to assign a new role or team in the organisation to an employee from competent authority.	The system asks the user for admin authorization.
2	The admin uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the 'employees list' action.

3	The admin goes to the ‘employees list’ link.	The system serves the employees list on a page. Each item in the list is a link to the corresponding employee’s profile page.
4	The admin goes to the link of the concerned employee who is to be assigned a new role or team.	The system serves employee ‘profile’ page having an ‘edit profile’ link.
5	The admin goes to the ‘edit profile’ link.	The system serves the ‘edit profile’ form which has ‘role’ and ‘team’ fields.
6	The admin sets the new ‘role’ or ‘team’ or both from a dropdown menu.	The system saves the employee-team relationship object to the database if there is no exception.
Post Condition:	1. The employee-team relationship object is saved to the database. 2. The transaction assigning the role or team is logged in the database.	

Non Functional Requirements:
1. The roles created must be defined as such within the organisation and not an arbitrary choice by the admin.

Data Requirements:
1. The data of the roles that exist within the organisation must be provided to the admin.

Screen Entry Exception Table:	
Exception	**Response**
1. Name of new role being created is the same as an already existing one.	Message: “A role with the same name exists. Role creation was not successful.”

5.5 **Use Case:** Admin Password Change

Brief Description:	The admin will be able to change the password for the admin account.
Business Trigger:	1. The admin feels the need to change the password for the admin account.
Preconditions:	Only the admin will be able to change the password for the admin. Admin authorisation is a pre condition.

Basic Flow: The admin opens the configuration page of the system and goes to the change password page. The admin sets the new password after giving the existing password once more and answering a security question(s).		
Assumptions:		
Line	System Actor Action	System Response
1	The admin feels the need to change the password for the admin account due to a specific security concern or as a precaution.	The system asks the user for admin authorization.
2	The admin uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the 'change configuration' action.
3	The admin goes to the 'change configuration' link.	The system serves the configuration on a page.
4	The admin goes to the 'change password' link.	The system serves the role 'change password' page.
5	The admin goes answers the security questions.	The system saves the new credentials to the database.
Post Condition:	1. The new credentials of the admin are saved to the database. 2. The transaction changing the admin password is logged to the database	

Data Requirements:
1. The data of the security questions set while initializing the system is required.

Screen Entry Exception Table:	
Exception	**Response**
1. The security question(s) isn't answered correctly.	Message: "Incorrect answers to the security questions."

5.6 **Use Case:** Update Role

Brief Description:	The admin (superuser) will be able to edit the attributes of an existing role.
Business Trigger:	1. The business undergoes restructuring and the names of roles are changed 2. A specific role's name is changed.
Preconditions:	Only the admin will be able to edit the attribute role of an employee. Admin authorisation is a pre condition.

Basic Flow: The admin opens the status page of the concerned role and goes to the edit role page for that employee. The admin sets the new name for the role and saves.

Assumptions:
1. A role's name is unique.

Line	System Actor Action	System Response
1	The admin gets a request to change the name from competent authority.	The system asks the user for admin authorization.
2	The admin uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the 'roles list' action.
3	The admin goes to the 'roles list' link.	The system serves the roles list on a page. Each item in the list is a link to the corresponding role's detail page.
4	The admin goes to the link of the concerned role whose attribute is to be changed.	The system serves the role 'detail' page having an 'edit role' link.
5	The admin goes to the 'edit role' link.	The system serves the 'edit role' form which has 'name' field.
6	The admin sets the new 'name'.	The system saves the role object to the database if there is no exception.

Post Condition:	1.The role object is saved to the database. 2. The transaction changing the role is logged in the database.

Non Functional Requirements:

1. The roles created must be defined as such within the organisation and not an arbitrary choice by the admin.

Data Requirements:

1. The data of the roles that exist within the organisation must be provided to the admin.

Screen Entry Exception Table:

Exception	Response
1. The new name of the role being updated is the same as an already existing one.	Message: “A role with the same name exists. Role update was not successful.”

5.7 **Use Case:** Accessing Logs (Admin)

Brief Description:	The admin (superuser) will be able to access the logs or transaction history.
Business Trigger:	1. The organisation wants to look at the transactions involved in an instance of a workflow. 2. There is a security breach or violation of security norms by an employee.
Preconditions:	Only the admin will be able to view all the logs and transactions. Admin authorisation is a pre condition.

Basic Flow: The admin opens the ‘view logs’ page and filters the logs according to parameters like ‘role’, ‘form’, ‘form_instance_id’, etc.		
Line	System Actor Action	System Response
1	The admin gets a request to retrieve the logs of particular employee, role, workflow instance, etc. from competent authority.	The system asks the user for admin authorization.
2	The admin uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the ‘view logs’ action.
3	The admin goes to the ‘view logs’ link.	The system serves the logs list on a page. Each item in the list is a link to the corresponding type of logs like ‘role update’ logs, ‘password change’ logs, etc.
4	The admin goes to the link of the concerned logs type.	The system serves the logs of the selected type on a page in ‘tabular’ form.
5	The admin filters the logs according to the requested parameter.	The system serves the filtered logs in tabular form as a page.

Non Functional Requirements:
1. The filtering must give as many options as possible and make it easy to get to a particular log.

2. A policy which states the period for which logs must be saved. 3. Organisation must comply with legal requirements.

Data Requirements:
1. The logs of the requested period must be there in the database.

Screen Entry Exception Table:	
Exception	**Response**
1. The filtering parameter is invalid.	Message: "Filtering parameter is invalid. Please change."

5.8 **Use Case:** Filling the form - Initiation

Brief Description:	An employee will be able to create an instance of a form and fill the section which he/she has to fill and forward to the appropriate authority for approval.
Business Trigger:	An employee wants to create a workflow instance for accomplishing a task and get the required approval.
Preconditions:	Employee authorisation is a pre condition.

Basic Flow: The employee chooses a form and creates an instance of it. He/she then fills the form section which he/she is supposed to fill and forwards for approval.		
Assumptions: 1. Only those form types which can be instantiated by an employee with that role will be visible to the employee for creation.		
Line	System Actor Action	System Response
1	The employee wants to create an instance of a workflow.	The system asks the user for employee authorization.
2	The employee uses his/her credentials to identify himself/herself to the system and gain authorization.	The system displays a screen with a list of links to the possible actions the admin can take including the 'create form instance' action.
3	The employee goes to the 'create form instance' link.	The system serves the list with types of forms on a page. Each item in the list is a

		lik to the corresponding form instance's creation page.
4	The employee goes to the link of the concerned form.	The system serves the 'form instance creation' page having the field pertaining to that form. Only those fields belonging to the section meant for filling by that employee are visible to that employee.
5	The employee sets the values of the fields in the section meant for him and saves.	The system saves the form instance object to the database and forwards to the approving authority.
Post Condition:	1.The form instance object is saved to the database. 2. The transaction recording that the form is created is logged in the database.	

Business Rules:
1. If the team of the receiver (person to which the form is forwarded to) is not mentioned explicitly, the software forwards the form to the person having the given role, belonging to the same team as the sender.

Data Requirements:
1. The data of the roles that exist within the organisation must be provided to the admin.

Screen Entry Exception Table:	
Exception	**Response**
1. The employee leaves a required field blank.	Message: "A required field is empty. Please re-check the form"

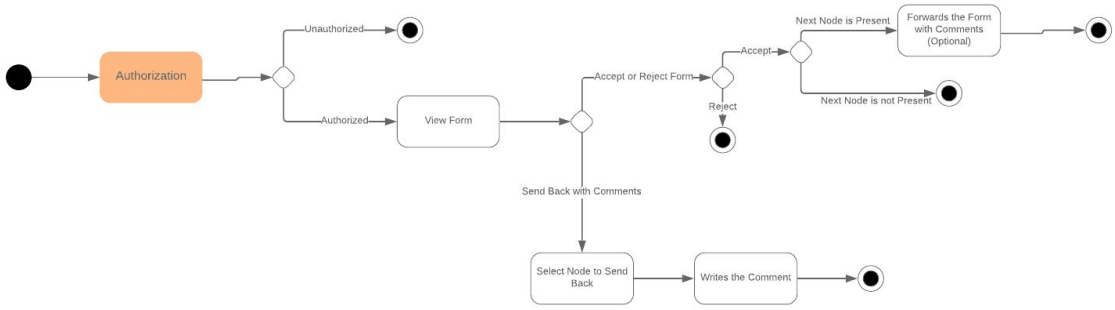
5.9 **Use Case:** Filling the form (At Intermediate Node)

Brief Description:	An employee will be able to fill the form, approve, reject or send back to the previous node.
Business Trigger:	An employee is part of the one of the intermediate nodes of a workflow and gets a form instance for approval from someone else in the organisation.
Preconditions:	Employee authorisation is a pre condition. Employee is part of the workflow for that form instance.

Basic Flow: The employee logs onto the portal and forms pending for his/her approval are shown on his/her dashboard. The employee goes to a particular form instance's detailed page, fills/edits section of the form which is supposed to be filled by him/her, optionally comments, approves, rejects or send back.		
Assumptions: 1. A form cannot be sent back without comments. 2. A form can be sent back multiple steps in the hierarchy. 3. The workflow is linear.		
Line	System Actor Action	System Response
1	The employee logs into the system and sees forms waiting for his/her approval.	The system serves links for going to the detailed pages of the forms waiting for approval.
2	The employee goes to the detailed page of a particular form.	The system serves the detailed view of the form.
3	The employee is able to view the information filled in the previous sections (nodes) and comments from previous nodes. The employee approves i.e sends forward, rejects with comments, sends back with comments or optionally comments and saves.	The system updates the form instance object in the database and other relevant tables.
Post Condition:	1.The form instance object is updated in the database. 2. The transaction recording that the form is updated is logged in the database.	

Business Rules:
1. If the team of the receiver (person to which the form is forwarded to) is not mentioned explicitly, the software forwards the form to the person having the given role, belonging to the same team as the sender.

Screen Entry Exception Table:	
Exception	**Response**
1. The employee leaves a required field blank.	Message: "A required field is empty. Please re-check the form"
2. The employee sends back or rejects the form without comments, etc.	Message: "Commenting while sending back or rejecting is a must."

Activity Diagram


5.10 **Use Case:** View Status of Form Instance

Brief Description:	An employee will be able to view the status and transaction history of a form instance if he/she is a node in the workflow of the form instance. Admin can view any of the form status/history without restriction.
Business Trigger:	An employee is part one of the intermediate nodes of a workflow and wants to view the current status of the form instance.
Preconditions:	Employee authorisation is a pre condition. Employee is part of the workflow for that form instance.

Basic Flow: The employee logs onto the portal and forms whose workflow goes through this employee are visible to the employee. The employee goes to a particular form instance's detailed page and is able to view the transactions/history and current status of that particular instance.		
Assumptions: 1. A form instance's detail can only be viewed by the user only if he/she is admin or a node in the workflow.		
Line	System Actor Action	System Response
1	The employee logs into the system and sees form instances in which he/she is a node.	The system serves links for going to the detailed pages of the corresponding forms.
2	The employee goes to the detailed page of a particular form.	The system serves the detailed view of the form.
3	The employee is able to view the information filled in the previous sections (nodes), comments from	

	previous nodes and the current status of the form.	

6. Other Nonfunctional Requirements

6.1 Performance Requirements

Concurrently many users should be able to use the software. The intricate metrics of performance (say speed), however, may depend on the hardware used by the user.

6.2 Safety Requirements

It is the responsibility of the customer to ensure archival storage is provided for backup, to avoid any substantial loss in the case of catastrophic failures, say a disk crash. The backend servers' access should be restricted and only authenticated personnel shall have access to them.

6.3 Security Requirements

- The customer is expected to choose their database partner prudently, as the security of the database is the most significant aspect of the security of the system. The users shall be provided with the terms of service and privacy policy before they consent to use the software. Any sign-up request on the portal shall be redirected to the admin, and the admin can take action as required to validate the authentication credentials of the user.
- Passwords should not be displayed in plain text while typing. The passwords should be stored in an encrypted manner.
- Each POST request shall be accompanied by a CSRF token as a request parameter, ensuring that a valid request can't be generated by an attacker.

6.4 Software Quality Attributes

- **Availability:** As the portal shall be hosted on customers' servers, it shall be available all the time.
- **Correctness:** The system's activity logs should contain all the information pertinent to the workflows.
- **Maintainability:** The code should be modular, each service wherever possible should be a different module. The code should be refactored after every major change in order to increase readability for different developers.
- **Usability:** The system should provide user-friendly interfaces and should be easy to get familiar with.

7. Other Requirements

End user should have information about actors who have access to their data. The user should be provided with the terms and conditions.

Appendix A: Glossary

Term	Description
Project/Team	A group of people in the customer's organization that represents a unit from the perspective of the software and form workflow, e.g., each 'team' may have a manager.
Blueprint	A blueprint of form represents a type of form, say 'Leave form'. Each blueprint has an associated workflow, whose interpretation is described below. Each form in the portal shall be an instance of some form blueprint.
Workflow	Workflow is a route, associated with each form blueprint, which is followed as the form moves in the hierarchy for approval/comments.
Nodes	Nodes in a workflow represent the roles through which the form has to move through. Each node in a workflow is associated with a role.
Admin or Superuser	This refers to an administrator.
The document	This SRS