



Centralized Change Management Platform for Melstacorp PLC

**A dissertation submitted for the Degree of
Master of Information Technology**

T. C. Kelaniyage

University of Colombo School of Computing

2019



Abstract

This document contains the details about a web application for change management and user management. Which also includes procedures that have been followed during the designing, implementation and testing process. This application specifically designed for Melstacorp PLC to centralize and acts as an all in one solution for different applications uses within the organization. With the new application all the change requests and user requests have to propagate within the workflow ensuring required approvals been obtained before migrate in to production environment. Process flow also ensure that each task will adhere to organizational policy and procedures. It helps management to overlook all the requests and assign responsibility for each request individually. The main purpose of this application is to integrate security, QoS, transparency, efficiency and accuracy into a change management and user management process. The application adopts global and organizational standards and best practices to enhance its efficiency. The application is designed to cater most of the organization requirements and streamline the process involved. The design is developed on the .net platform with the help of SQL server.

Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: T. C Kelaniyage

Registration Number: 2016/MIT/030

Index Number: 16550302

Signature:

Date:

This is to certify that this thesis is based on the work of

Mr./Ms.

under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name:

Signature:

Date:

Table of Contents

Abstract	i
Declaration	ii
List of Figures	v
List of Tables.....	vi
List of Abbreviations.....	vii
Acknowledgment	viii
Chapter 1 . Introduction	1
1.1 Motivation	1
1.2 Aims and Objectives	3
1.3 Scope	3
1.4 Structure of the Thesis.....	4
Chapter 2 . Background.....	5
2.1 Review of similar systems.....	5
2.2 ITIL Alignment	12
2.2.1 ITIL Access Management	13
2.2.2 ITIL Change Management	14
Chapter 3 .Analysis and Design	17
3.1 Requirements Gathering.....	17
3.2 System Requirement Analysis	18
3.3 Design Strategy	25
3.4 Design.....	26
3.4.1 Database Design.....	26
3.4.2 Application Design.....	28
3.4.3 User Interface Design.....	32
3.5 Testing Plan.....	39
Chapter 4 .Implementation	42
4.1 Database Backup	42
4.2 Basics of security	42
4.2.1 Hashing Passwords.....	42
4.2.2 Allow Only authorized users.....	43
4.2.3 Transaction Log.....	44

4.2.4	Input Validation.....	45
4.3	Code Module	46
Chapter 5	. Testing and Evaluation	48
5.1	Usability aspects of software.....	48
5.2	Information in the Program	50
5.3	Technical Aspects of software	50
5.4	Functional Aspect.....	51
5.5	Technical and Practical Difficulties.	53
Chapter 6	. Conclusion	54
6.1	Project Summary	54
6.2	Limitations of the Project.....	54
6.3	Future Work	55
Bibliography	56
Annexure A	58
Annexure B	62

List of Figures

Figure 2.1: Emergency Change Life cycle.....	7
Figure 2.2: Normal Change Life cycle.....	7
Figure 2.3: Routine Change Life cycle	8
Figure 2.4: Bugzilla Instance Lifecycle	10
Figure 2.5: ITIL Access Management process flow	13
Figure 2.6: ITIL Change process flow	15
<i>Figure 3.1:Change Management</i>	22
Figure 3.2:User Management.....	24
Figure 3.3:Table Structure.....	27
Figure 3.4:EER Diagram.....	28
Figure 3.5:State Diagram of Change Request.....	30
Figure 3.6:State Diagram of User Creation Request.....	31
Figure 3.7:Login Screen.....	33
Figure 3.8:Admin Screen	34
Figure 3.9:User Screen.....	35
Figure 3.10:Change Password.....	35
Figure 3.11:Responsive Interface.....	36
Figure 3.12:Color Theme 1	37
Figure 3.13:Color Theme 2	37
Figure 3.14:Popup	38
Figure 3.15:Warning	38
Figure 3.16 :Admin Dashboard.....	39
Figure 4.1:Database Backup Schedule.....	42
Figure 4.2:Hashed Passwords in SQL database	43
Figure 4.3:Function used to Hash the password	43
Figure 4.4:User Creation Interface.....	44
Figure 4.5: Password Complexity	45
Figure 4.6: Password Complexity Coding	46
Figure 4.7:Server Side Validation Example.....	46
Figure 4.8:Coding Structure	47

List of Tables

Table 3.1:User Stories	18
Table 3.2:Development Environment	25
Table 4.1:Change Management Roles	43
Table 4.2:User Management Roles	44
Table 5.1:Usability Feedback.....	49
Table 5.2:Spec Sheet	50
Table 5.3:User Level Wise Average Score	51
Table 5.4:Evaluation Criteria	52

List of Abbreviations

- CM – Change Management
- ITSM – Information Technology Service Management
- ITIL - Information Technology Infrastructure Library
- ERP - Enterprise resource planning
- IS – Information System
- BA – Business Analysis
- UAT – User Acceptance Testing
- UI – User Interface
- HR – Human Resources
- HTML - Hypertext Markup Language
- CAB - Change Advisory Board
- HRIS – Human Resources Information System
- Oracle EBS – Oracle E-Business Suit

Acknowledgment

I am using this opportunity to express my gratitude to everyone who supported me throughout the course of this project. I am thankful for their aspiring guidance, invaluable constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

I express my warm thanks to Dr. Prasad Wimalaratne for his support and guidance throughout the project.

I would also like to thank my project examiners and all the people who help me with useful and constructive recommendations for my project.

Thank you,

Chapter 1 . Introduction

1.1 Motivation

In today's context, most of the organizations use ERP (Enterprise Resource Planning) systems to increment the efficiency in their business. Sometimes organizations may use multiple fortifying systems to pair with ERP. Because of that managing, these systems became an intricate task to the management and system administrators. Prevalent tasks such as change management and User management, perform a critical task in system maintenance.

Most of the organizations use decade-old practices to manage the above tasks. Where they utilize manual forms or email communications to manage system changes and users. With the rapid transmutations in the business, these obsolete methods would engender more preponderant difficulties such as unable to track requests, document misplacement, unable to review statistics and many more. Since system administrators responsible for most of the activities in applications. Maintaining manual process could lead to inefficiency and erraticism.

Security administrators and programmers who endeavor to control, modify, or change applications programs without a well-orchestrated, meticulously administered system of change can encounter such difficulties as:

Lack of a precise program inventory. The IS department does not know precisely which programs have been approved for use or are authentically in accommodation, the location of these programs, and the programs that are scheduled for change.

An incomplete history of program changes. There is no consummate history of change requests, change sanctions, modified programs, or programmer change assignments.

Lack of program change integrity. Some systems experience inadvertent program code overlays. This quandary conventionally occurs when different programmers have uncontrolled access to engenderment source code with no log-out, log-in, or tracking mechanisms in place. For example, one programmer accesses a replica of a program to make enhancements. Concurrently, a second programmer, who has additionally accessed the same program, makes an expeditious bug fine-tune and updates the engenderment source code. The first programmer then consummates the enhancements, places the program into engendered, and expunges the bug fine-tune and the engenderment source code updates.

Duplicated program modules. Often modules that perform standard processes are duplicated, rather than being developed once and then reused, wasting both personnel and computing resources. Similarly, modules that are shared may be transmuted to accommodate one program, causing the disruption of another program's operation.

Unauthorized changes. For sundry reasons, users may make unauthorized changes to engenderment programs. Such changes may be irreprehensible, fraudulent, destructive, or the work of a midnight programmer (i.e., an employee who programs or makes changes to subsisting programs outside business hours).

Lack of documentation and testing. The pressure to fine-tune or modify a program may constrain efficacious program change documentation and may embolden the utilization of a program that has not been adequately tested.

Similarly, for Utilizer Access Management withal administrators face the following challenges.

Lack of Security: Manual document filing a high propensity of being misplaced or damaging documents. There are so many ways to be damaged, lost, or misplaced. A fire or natural disaster could mean the loss of all utilizer access request forms. But most importantly paper document filing can be less secure than electronic filing systems. Misplaced documents can facilely be placed in the erroneous hands. the organization expects its information to be secured and not fall into erroneous hands. A cabinet filled with forms is way more facile to access than a computer which requires a password and credentials to get into.

Takes Up a Lot of Space: Another downfall of manual document filing is the amount of space it can take up to store. With the magnification of the organization finding places to store documents will be a devastating quandary. It interrupts the productivity levels of the organization.

Access Time: Manual document filing is very time-consuming. Not only they require to organize and store the files, retrieving the information when it is required to take time. This can cause efficiency and productivity of the organizational functions.

1.2 Aims and Objectives

This project focus to implement a single platform to cater to system Change Management in an organization. Which allow the organization to manage all the request in a centralized platform by satisfying all above discrepancies. Which will help top management to overlook ongoing tasks and make decisions considering below aspects.

- Ongoing Tasks
- Progress of each task
- Who owns responsibility
- Priority of task
- Expected completion date

Not only the management but ground level users also have the opportunity to see the progress of each task.

Even though the introduction of this new system seems to increase the workload of the user. In reality, it will help to streamline change management and access management process by allowing the user to follow a common standard. By far different users in different departments adopt their own procedure or a process that is decades old.

Another main objective is the transparency of the activities. This system facilitates to capture all activities and the respective person who is responsible for each task. By having all these records management can inquire about any given activity at future date.

This system specifically developed to cater two main applications, Oracle EBS and HRIS. However, this system can be customizable to any other application.

1.3 Scope

The main focus of this project is to the design a centralized Change Management System to Melstacorp PLC. The system will be using to capture all changes initiate within the organization. Each change request will be routed through a sequence of steps before successfully migrate into the production environment. The most important aspect of this system is that management can monitor status of each request individually and to take appropriate actions. This application would be a stand-alone application with no direct integration to any other existing systems. At the initial phase user requirement analysis have been carried out to gather user requirements. Process flow has been design based on current practice obtain through requirement gathering while adhering to the ITSM and ITIL frameworks, a widely accepted models for the service management in the world. The system is a web-based application which allow users to access system with very limited resources.

1.4 Structure of the Thesis

Following the introduction above which contain reason behind initiating this project and it's expected objectives, this report consists of the following chapters as below,

Background: Content in this chapter describes the foundation for this project. Similar system study and identify the pros and cons in selected systems also discussed in this chapter. Furthermore, detail comparison of different design strategies to implement this system also discuss under this Background chapter.

Analysis and Design: This chapter mainly focuses on technical aspects of the system which includes design structure, database structure, and other supporting structures. Further, it describes the design aspects of the system in various angles.

Implementation: This chapter discussed implementation of the system. Coding structure, application security and database implementation were concerned in this chapter.

Evaluation: Evaluation chapter discusses the outcome of the project and how is achieve the expected result. The reaction of the user and the management towards the new system will be also evaluated in this section. Evaluation results are obtained through pre-defined evaluation criteria and will be analyzed root cause for such instances.

Conclusion: Finally, under conclusion, it discusses a summary of the project and its outcome to the organization. Possible future improvements for this application also discuss in this chapter.

Chapter 2 . Background

2.1 Review of similar systems

During analysis, there were projects that implement a partial solution by implementing either user management or change management. But most of them do not focus on the total solution. Further, most of them do not follow industry standards.

While conducting research on some similar systems such as iTOP, SARA, Omini Tracker, Smart Solve, Remedy ITSM and some in-house developed systems it was noted that some compulsory components are missing in those systems.

iTop

iTop stands for IT Operational Portal. iTop is an Open Source web application for the day to day operations of an IT environment. But there will be a cost if the organization has to use for business purposes. It has its own pre-defined authority structure where the user can assign access to individuals according to the requirement. Apart from that, it provides a structured overview of all the ongoing requests. Management can evaluate the progress of the requests at any given time. iTop is a commonly use platform in most of the organizations since it complies with ITSM (IT Service Management) and ITIL (Information Technology Infrastructure Library) standards. With the free version, the user gets very limited customization and if required more customization organization required to acquire paid version.

iTop facilitates the following tasks in an organization:

1. Document your IT infrastructure and all the relationships between the various pieces and stakeholders of the infrastructure (servers, applications, network devices, virtual machines, contacts, locations...)
2. Manage incidents, user requests, planned outages...
3. Document IT services and contracts with external providers including service level agreements
4. Export all the information in a manual or scripted manner
5. Import (manually and using scripts) or synchronize/federate any data from external systems

iTop also can be used by different types of persons which includes, Help Desk agents, Support engineers (1st level, 2nd level...), Service managers, IT managers, End-users: a simplified “portal” interface is available to let them submit their requests directly.

iTop is relying on Apache/IIS, MySQL, and PHP, so it can run on whatever operating system supporting those applications. It has been tested on Windows,

Linux (Debian, Ubuntu, and Redhat). The application also runs on Solaris and MacOS X. Since iTop is a web-based application organization don't need to deploy any client software on each user's PC. A simple web browser is sufficient (IE 8+, FF 3.5+, Chrome or Safari 5+).

iTop facilitate several types of changes which includes, Patch installations, System configuration changes, OS updates and Software installations

This way organization can track all the modifications made within IT. A lot of incidents are due to changes made to the IT environment. By documenting them, organisation can identify easily what changes had been made when an incident occurs and restore the service more quickly.

Moreover, this change management module allows to automatically analyze the impact of the changes on the infrastructure and the application solutions. IT engineers can then better control the unavailability of the critical services in the enterprise and improve user satisfaction.

In iTop the changes are managed by the people with the following profiles:

- Change the implementors plan and implement the changes
- Change supervisors follow up with the changes
- Change managers approve the changes

When installing iTop, you have a choice between two different modules for documenting changes. The module described here has been designed to implement ITIL V3 change management with three types of changes:

- Routine changes
- Normal changes
- Emergency changes

The differences between those three types of changes rely on their respective workflows.

According to iTop manual emergency, change is the highest priority change that can be defined in an organization. Emergency changes are defined as changes that need to be evaluated, assessed and either rejected or approved in a short timeframe. Simply defining a change as an emergency does not automatically entail the change should be implemented. The Emergency Change Advisory Board (ECAB) will assess the change and provide advice to the delegated person responsible for approving or rejecting emergency changes. Emergency Change objects in iTop application have the life cycle indicates on **Figure 2.1**:

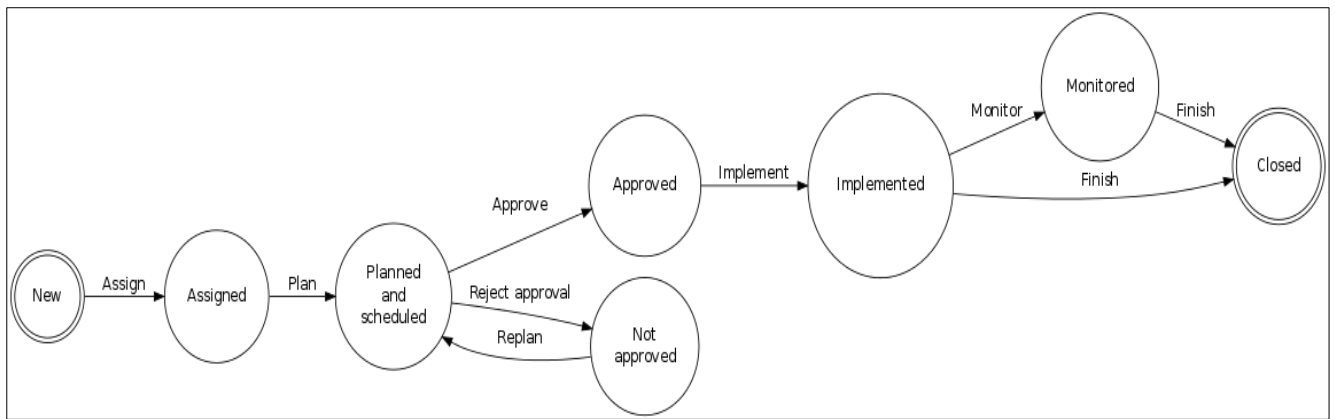


Figure 2.1: Emergency Change Life cycle

iTop normal change refers to changes that must follow the complete change management process. By definition, a normal change will proceed through all steps of the change management process and will eventually be reviewed by the Change Advisory Board (CAB). The CAB will provide advice regarding the change to the person who is deemed responsible to approve or reject normal changes. Normal Change objects have the life cycle indicates on **Figure 2.2:**

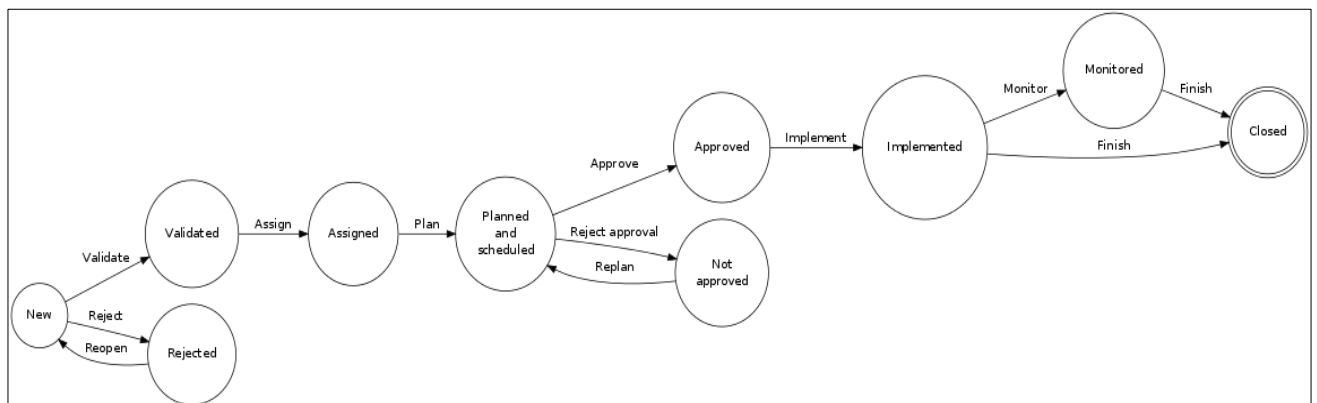


Figure 2.2: Normal Change Life cycle

Routine (standard) change quite simply refers to pre-approved changes. Pre-approved changes can be defined for a variety of tasks, but they will typically be low risk, low effort changes that have a low or known cost. Routine Change objects have the life cycle indicate on **Figure 2.3:**

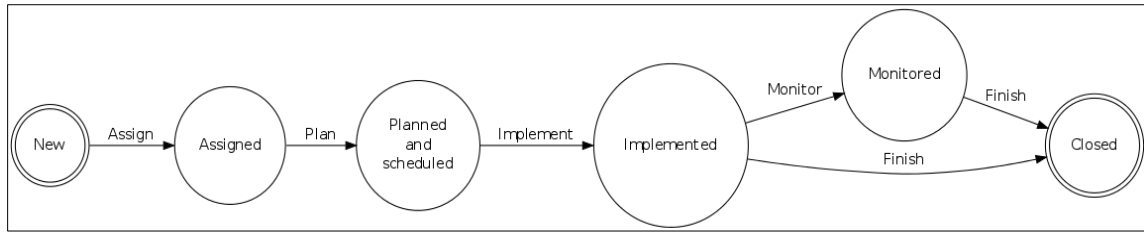


Figure 2.3: Routine Change Life cycle

The list of teams to which you can assign a change is defined by the delivery model of the corresponding organization. When creating a change, the agent has to select the customer organization, then the list of teams is strictly limited to the teams defined for this customer. If a team is missing, the delivery model of the customer must be updated to reflect this need. See More about the Delivery model for more information. [1]

A change ticket only has a private log to document all the activities and communications related to it. This one is not visible on the end user portal.

Further, according to the feedback provided by users setting up the system is complex and required professional expertise and in order to obtain technical support, organization has to make an annual subscription fee.

Bugzilla

Bugzilla is another robust, featureful and mature defect-tracking system or bug-tracking system. Which also can be configured as change management system allowing teams of developers to keep track of outstanding bugs, problems, issues, enhancement and other change requests in their products effectively. Bugzilla originally developed and used by the Mozilla project, and licensed under the Mozilla Public License.

Simple defect-tracking capabilities are often built into integrated source code management environments such as Github or other web-based or locally-installed equivalents. organizations tend to turn into Bugzilla when they outgrow the capabilities of those systems - for example, because organizations need workflow management, or bug visibility control (security), or custom fields. [2]

Bugzilla is both free as in freedom and free as in price. Most commercial defect-tracking software vendors charge enormous licensing fees. Despite being free, Bugzilla has many features which are lacking in both its expensive and its free counterparts. Consequently, Bugzilla is used by hundreds or thousands of organizations across the globe including Sri Lanka.

The features include Advanced Search Capabilities, Email Notifications Controlled By User Preferences, Reports and Charts, Request System, High Security, Custom Workflows which allow organizations to change workflow according to their requirement. Bugzilla becomes popular all over the world due to following characteristics.

1. It is still under active development
2. Used in high-volume, high-complexity environments by Mozilla and others
3. Supported by a dedicated team
4. Packed with features that many expensive solutions lack
5. Trusted by world leaders In technology
6. Installable on many operating systems, including Windows, Mac, and Linux

Bugzilla's system requirements include:

- A compatible database management system
- A suitable release of Perl 5
- An assortment of Perl modules
- A compatible web server
- A suitable mail transfer agent, or any SMTP server

Currently, supported database systems are MySQL, PostgreSQL, Oracle, and SQLite. Bugzilla is usually installed on Linux using the Apache HTTP Server, but any web server that supports CGI such as Lighttpd, Hiawatha, Cherokee can be used. Bugzilla's installation process is command line driven and runs through a series of stages where system requirements and software capabilities are checked.

While the potential exists in the code to turn Bugzilla into a technical support ticket system, task management tool, or project management tool, Bugzilla's developers have chosen to focus on the task of designing a system to track software defects. Mandated design requirements include:

- The ability to run on freely available, open-source tools. While Bugzilla development includes work to support commercial databases, tools, and operating systems.
- The maintenance of speed and efficiency at all costs. One of Bugzilla's major attractions to developers is its lightweight implementation and speed, so calls into the database are minimized whenever possible, data fetching is kept as light as possible, and generation of heavy HTML is avoided.
- Tickets. For instance, Mozilla.org uses it to track feature requests as well. In this case items (called bugs) can be submitted by anybody, and will be assigned to a particular developer. Various status updates for each bug are allowed, together with user notes and bug examples.

In practice, most Bugzilla projects allowing the public to file bugs—such as the Bugzilla bug database for Bugzilla itself—assign all bugs to a gatekeeper, whose

job it is to assign responsibility and priority level. *Figure 2.4* illustrates lifecycle of a bug or change request.

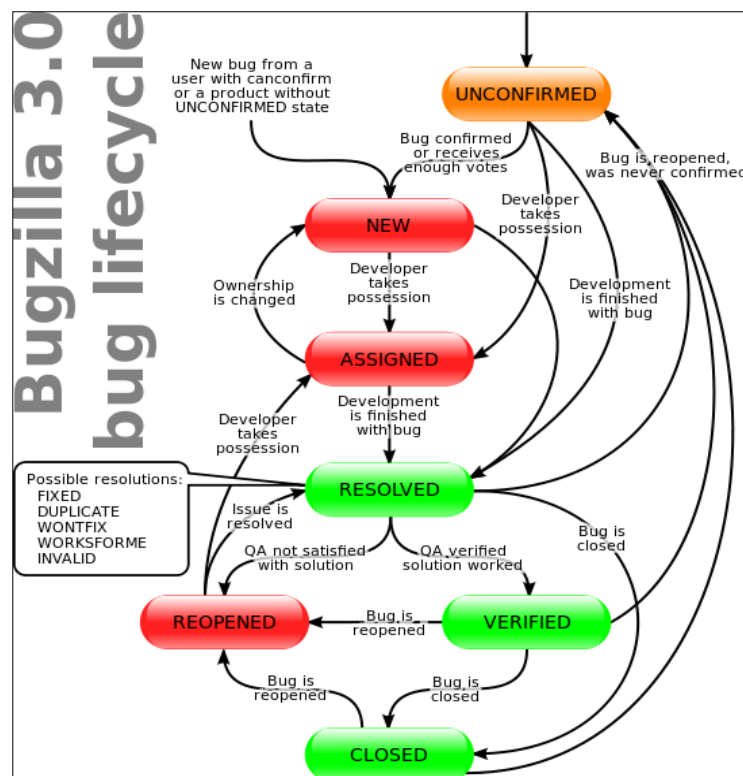


Figure 2.4: Bugzilla Instance Lifecycle

As a drawback Bugzilla does not cater all the aspects of change management process such as Business Analysis, User Acceptance Testing and so on. Additionally, it required paid support to add new modules and workflows to the existing design.

Remedy IT

Remedy IT system Management also another change management system developed by BMC Software. The Remedy ITSM Suites cover the main process of the ITSM aligned to ITIL best practices. All the Application are integrated as part of the ITSM Suite in order to manage the different scenarios that normally needs to be handled by the support staff.

The main usage scenario is incident management, work orders, and change requests. Remedy facilitates the flow of approvals through the organizational hierarchy. Remedy integrates the change management calendar and workforce load. KPI (key performance indicators) are easily exported from the Remedy system.

The system includes a notification system to issue email and SMS alerts based on

the level of criticality. We integrated Remedy with other IT monitoring system, where alerts are translated to work orders and assigned to the specific team. It also supports Integration with monitoring software like SolarWinds. where Remedy can transform notifications to work orders and assign to respective teams. With technical expertise Remedy, IT allows detailed Customization capability to fit the exact needs of the organization but the organization has to bear considerable cost over support service.

Change management calendar is another feature which allows integration of change orders from all organization departments and facilitates the scheduling process of project managers.

With Remedy, IT organizations do not need to worry about internal infrastructure such as Server Hardware, Databases, Network, server maintenance, database backup. and other aspects since it offered as a cloud-based service,

The license model of floating write licenses, lets the organizations share licenses pool. According to user reviews, there are several drawbacks affiliated with this application. Which includes customization can be a lengthy process due to the numerous features included in the software. New staff members require some sort of hands-on training to get to know all the use capabilities because of its complexity. Even though Remedy IT system equipped with cloud base solution it tends to be expensive compared to competitors. [3]

SARA

SARA also a paid application with limited customization. SARA is a web-based system where user can access from anywhere. It is mostly used as an Incident Tracking system rather than change management system. since the system is not capable of capturing the UAT process. But organizations which use this system for change management they have to neglect the UAT process and continue with the rest. SARA system also can be used as a User management system to create, modify and terminate users. More or less it functioning as a ticketing system to cater to organizational requirements. [4]

Omni Tracker

Omni Tracker also another change management application which is compatible with ITSM standards. This application has all-inclusive shared licenses which are comparatively cost effective. There are some very neat ways to customize this software, and some great little things like SMS integration, views can create, and a solid calendar make a unique experience to the user. It sends notifications automatically for the responsible people in case new actions are created; it also has the possibility of quick reporting for open actions / overdue actions/actions per responsible, etc. There are few drawbacks in the application as well such as limited customization in reports, the interface is cluttered and un-intuitive, and the performance slow. Omini Tracker comes with features including Approval

workflow, Change Planning, Release Planning, and Task Management capabilities. [4]

Smart Solve

Smart Solve is a new ITSM system introduced into the market by Pilgrim Quality Solutions. This application contains the following features. Approval Workflow, Audit Trail, Change Calendar, Change Planning, Compliance Management, Prioritization, Release Management, Task Management, and Training Management. Even though this application does have a number of features It was informed that a considerable amount of bugs associated with the application. Since this is a new application it seems to require time to well established. [5]

In-house developed applications which are tailor-made to the respective organization has its own features according to the organizational requirements. Some organizations use open-source applications such as Bugzilla and customize those applications according to their requirement. During the analysis, most of the organisations failed during the customization due to lack of technical and business knowledge. Further, in-built features that are not properly customized by the organization could lead to loopholes and backdoors with the application. Another reason for failing most of the in-house application is they do not follow industry standards. Bypassing of controls are quite common on those in-house applications since they do not concern on segregation of duties in the process flow. This lack of control could lead to misuse of information. Ultimately they do not provide the expected result to the users and management.

All the above pros and cons in the various system have been considered in designing process to minimize the loopholes and increase the control aspects on the proposed application.

2.2 ITIL Alignment

In the existing manual IT service system, the organization faced major issues due to the lack of proper processes, procedures, systems and policies, which necessitated a move toward implementing an automated ITIL solution encompassing IT service management (ITSM) best practices. The challenges that will face during the planning phase will be manual processes for access provisioning; manual/ad hoc change management processes; manual generation of reports and statistics and person-dependent, rather than process-dependent, IT services.

2.2.1 ITIL Access Management

The organization has locations throughout the country. The head office has 40 percent of the workforce; the other 60 percent is located in different physical locations. Some of the divisions are quite volatile, with high employee turnover, especially on the retail side, which gives rise to numerous instances of access provisioning and de-provisioning. Currently, the organization has manual process with a manual form for access provisioning and de-provisioning. In this respect, IT requests went to the line manager or department head, who, after approving the requests, scanned and sent them by email or faxed them. Most of the time, the process took a lot of time, and line managers were not sure where to send requisitions. On numerous occasions, the IT department, due to time constraints, gave provisional access pending the receipt of the signed request form, and there were instances of non-arrival of those documents.

The main objective of ITIL access management is to safeguard the data from being accessed by unauthorized users. This is extremely vital for an organization as critical data falling into the wrong hands could cause irreversible damage to the company. Following Figure 2.5 indicates Access Management process flow proposed by ITIL[6]

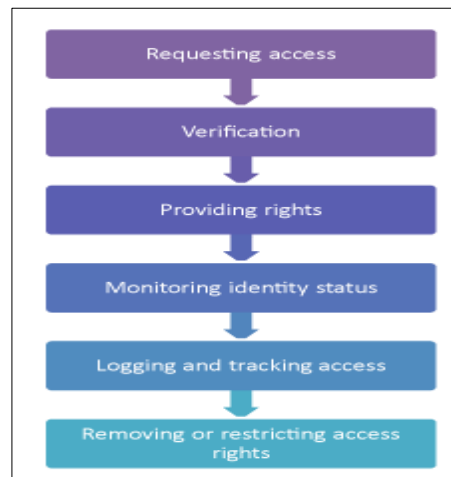


Figure 2.5: ITIL Access Management process flow

Requesting access

The first step is to request access to a particular IT service by sending one of the following types of requests.

1. A Standard request sent directly from the Human Resources department
2. A request for change sent from the change management process
3. A service request raised from the service desk
4. An auto-provisioning request where smaller requests are automatically handled.

Verification

It is the duty of access management to verify the identity of the user making the request and the legitimacy of the request being made.

Providing rights

After the identity of the user has been verified, the access management grants him the right to access a particular IT service as per the regulations defined during the ITIL Service Strategy. If undefined, the access management grants the user, rights to access the service after sending requests to the respective departments in question and receiving approval.

Monitoring identity status

In a large organization, there is a constant change of employees in a particular position or requesting IT service access. It is the duty of the access management to keep track of all the access rights granted to the various employees and update them once the relevant employees are no longer in that position.

Logging and tracking access

Access management also has to keep constant tabs on the IT service users who have been granted access. All the activities of Service Operation processes should be monitored to ensure that only the users with the relevant clearance and authorization are accessing a particular IT service. They should also define parameters which make it easy to detect intrusions into the system by unauthorized users, excess incorrect login attempts, and unusual activities.

Removing or restricting access rights

The access rights of a user have to be monitored so that when the role of the user changes over a period of time and he no longer needs access to a previously needed IT service, the authorization for access should be revoked. Depending on the present status of the user, their access needs to be restricted or terminated accordingly. [6]

2.2.2 ITIL Change Management

The second challenge was in the realm of change management. The company had a substantial number of IT applications, which required a lot of procedural effort to change, upgrade and modify. For example, if one user wanted to add new field to user interface, the IT support personnel looked at it purely as an upgrade, without looking at dependency, due to the lack of documentation of the application, thus proceeding with the update. There were instances of dependency with applications, where some applications that were working with the old version may not work with the new change. Hence, doing things in an unstructured way gave rise to reduced effectiveness of the

system, which, ultimately, led to a very unstable system environment prone to frequent outages.

The primary objective of ITIL Change management is to mitigate risk and impact. Change management does the authorization to approve any change to be deployed. It protects the production environment while executing a new change. Following are the objectives of ITIL Change management process.

1. Reduction of risk and impact
2. Maintenance of current working state
3. Communication and approval management
4. Effective change planning with optimized resources
5. Reduction in number of incidents due to change execution

ITIL Change management process includes different steps that capture every detail about a change request for future tracking. These processes ensure that the change is validated and tested before it moves to deployment. Release management process is responsible for successful deployment. Change Manager handles planning & evaluation whereas Release Manager takes care of the actual implementation of the change. The change management process flow looks similar to the diagram indicate in **Figure 2.6** :

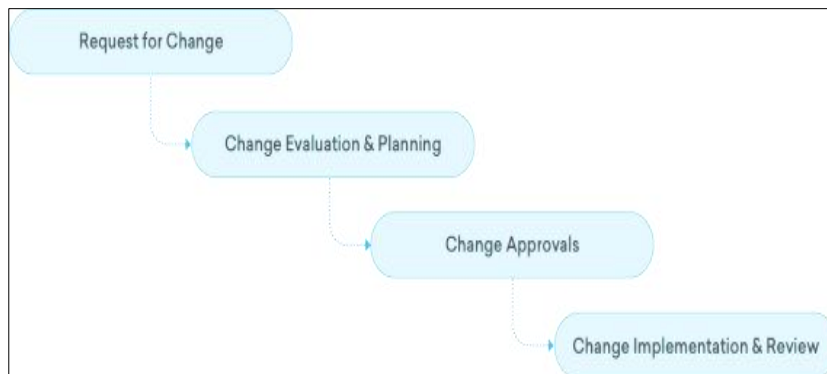


Figure 2.6: ITIL Change process flow

Request for Change

Request For Change (RFC) is submitted to the change management team for validation and approval. Change requests occur from one of the following sources.

- Incident that causes a change
- Existing problem that results in a change
- User requesting for a new change
- As a result of an ongoing maintenance

An RFC template is used to record the change

- **Reason for change** - Justifies why change is needed along with risk/benefit analysis.
- **Impact & Risk assessment** - Potential Impact and risk are calculated and documented including configuration items, CIs.
- **Cost benefit analysis** - Estimated cost versus potential benefits are compared.
- **Implementation planning** - Steps for implementing change that includes project members, timelines and methodology.

Change Assessment and Planning

This stage handles change evaluation and planning activities. It includes prioritization and planning activities to minimize risk and impact.

- **Prioritization** - Determine the type of change and prioritize the requests accordingly.
- **Scheduling** - Check the Release schedule to get an estimated time frame and fix the planned start date and end date
- **Roll out plan** - Plan out the implementation activities
- **Back out plan** - Back out plan in case of unexpected setbacks

Change approvals

Any change request that comes in, needs to be approved. The Change management team handles end to end communication and approvals from Change Advisory Board (CAB). Change approval is crucial to eliminate any execution failure and downtime. Approval process varies according to the change type. For example, a major change like ERP solution replacement requires approval from CAB as well as management whereas a standard change like patch deployment does not require any CAB approval as they are pre-approved. Change request is approved only if all the CAB members approve it. Upon rejection, reassessment review is done and submitted again for CAB approval.

Change implementation

Change implementation is handled by Release management team and Post Implementation Review (PIR) is taken care of, by the Change management team. Release management team follows their own processes that involve planning and testing. Change review happens once the implementation is completed to ensure everything has gone according to the plan The existing change management process is constantly reviewed and updated wherever necessary. [7]

Chapter 3 .Analysis and Design

3.1 Requirements Gathering

To gather the requirements effectively a series of face to face interviews were organized with the users. This is crucial to understand the requirements of the user and the practical issues they face while working. Interviews became the most efficient and effective method of conveying information to and within development and management of face-to-face conversations. But it noted that users were reluctant to convey their issues with the presence of top management. Therefore, the questioner has been distributed among the bottom level stakeholders to obtain their true opinion and their difficulties.

As shown in Table 3.1 the meetings were used to discuss what was needed in the application and gather feedback on progress. As this application is to be used by all level of employees in the organization it is essential to represent the feedback of all of them. The first meetings were very crucial to the project to get the most feedback possible about what was expected from the application and identify the drawbacks of the manual process. The challenge was that the conflicts of requirements among the users where a different level of users has different expectations from the system. Referring to the similar systems and Industry standards made sure everyone agreed to a common set of requirement that caters to the common objective of the organization. To help clarify what features will be needed for the application, user stories were used to illustrate. This technique was used throughout the project helped clarify exactly what features were going to be needed for the application in laymen terms. The most important stories were then organized into general deliverables for each iteration this is shown in *Table 3.1*. Creating clear objectives from the user stories helped plan out the project.

User Story	Features
As a user, I should be able to initiate a change request	Raise change request
As a user should be able to check the status of my request	View status of the requests
As a user once change been completed I should be able to confirm	Acknowledge the solution
As Head of the department I should be able to authorize or reject requests	Request Authorisation

User Story	Features
As a Head of Department, I should be able to add comments to the requests	Add comments
As a HOD I should able to change the priority of the request	Set Request priority
As HOD I should be able to check the ongoing requests from my department	Follow up requests
As Business Analysis I should be able to review the request and made remarks on it	Add remarks
As BA I should be able to approve or reject the request	Approve/Reject Request
As a Developer, I should be able to add remarks to the request	Add Remarks
As the developer, I should be able to confirm the development	Confirm development
As a QA I should be able to make comment on the development	Add comments for the solution
As a QA I should be able to approve, reject or further develop the solution	Approve/Reject/Further Develop the solution
As Head of IT, I should be able to approve or reject for live deployment	Approve/Reject live deployments
As System Administrator, I should able to confirm the live deployment	Confirm Deployment

Table 3.1: User Stories

3.2 System Requirement Analysis

The foundation of this system has been laid base on the requirement gathers from ground level users and the application owners. Apart from that all the functionalities and requirement has to align with the organizational policies and procedures since they set up the direction of the organization.

In order to gather end-user requirement, interviews have been used. Because it seems to be the most effective way of gathering end user requirements with the busy schedule of them.

The current manual procedures organization has followed has plenty of loopholes due to lack of controls, therefore, a combination of current practice and the industry best practice has to merge to identify most critical functionalities need to address via the proposed system.

After fine-tuning identified critical functionalities swim lane diagram of **Figure 3.1** indicates process flow of system with responsibility of each user.

Main Functional Requirements as follows:

Change Management

1. Change Request-

The end user can request a change through the system where he needs to enter following details to the system,

- Type of the change/modification – Bug Fix, New Requirement, Report change, Parameter change
- Description of the change and justification
- Priority – Major/Minor/Emergency*
- The module of the system that modification needs to be done
- Supporting Documents – Any document or screenshot that helps to illustrate the requirement or the problem.

Apart from that requested date and requester details will be captured from the system.

2. Authorize change –

Every change request needs to be authorized by the relevant department head of the business owner. He will be able to put a comment pertaining to the requested change and alter the priority state if required. After completing the required amendments department head has to authorize the change request. Also, he has the option to reject the request if it not required. Even it is authorized or rejected, rejected user and the date will be captured from the system.

3. BA Team Approval-

Upon authorization of the request Business Analysis(BA), the team can analyze the change request and comment on their opinion on that. They also have the option to change the priority of the request for one last time since they are in a position to evaluate the criticality of the request with the business impact. After setting their approval they have to assign that task to selected development team under a team lead.

**BA team also have another option if the local support/development team does not have expertise in engaging on particular task BA team can consult the vendor on that. On such instance, BA team is responsible for follow up the request and lodging on the support portal of the vendor.

4. Development:

If the requested change forwarded to develop locally. The development team lead has the responsibility to complete the task he has to provide the following details to the system

- Team members who engage in a given task.
- Expected completion date
- Description of tasks perform and tools that been used

After complete, the development Team lead has to sign off the development task.

5. QA Test :

Upon completion of the development Quality Assurance(QA) team has to conduct the technical evaluation of the new development which includes whether the development team adopts best practices and standards. Whether its derive expected outcome. Will there be any unexpected outcome upon deployment and etc. The system facilitates with a check-list to be filled by QA person.

If the QA team does not satisfy with the development they also can revert to the previous stage with appropriate justification. If the Quality Assurance person senses that there is problem with requirement understanding that person can re-direct the request to business analysis.

6. User Acceptance Testing (UAT)

This stage needs to be completed by change requester or the end user. Where he has to confirm whether new change provides the expected result. If not, he also has the option to reject the provided solution. Whatever the situation user has to attach test cases and results he used to conclude his decision.

If the solution has been rejected by the user, the task has to bounce back to the BA team to re-analyze the request.

7. Approval for deployment

Before deploying the change to live environment IT department head needs to give approval.

8. Change Deployment:

Deployment of new change to the live environment has to be done by the system administrator and he has to lodge completion date and time and any special incidents occurred during the deployment.

9. Confirmation of deployment:

The final confirmation has to be done of IT head or a technical person assigned by IT manager upon inspection of the live system.

Note:

* Depend on the criticality of the system changes has been divided into 3 categories

1. Major Changes – Most of the time these type of changes take considerable time period to implement. Change in the process flow, adding new functions, Implementing new module and etc.
2. Minor changes (Low Priority) – Minor changes take a limited time period and can be easily fixed. Their type of changes can be bug fixes, operational changes and etc.
3. Emergency changes – These are highly critical changes and need to implement as soon as possible unless the organization will end up in losses and brand image.

**Proposed system will not directly integrate with Oracle Support Portal or HRIS Portal. The system will only cater to inter-organization functions.

Depend on the user requires access will be granted to the system. As organization management need to decide the access level to each employee.

High-level change management process flow depicts in the below flow chart diagram.

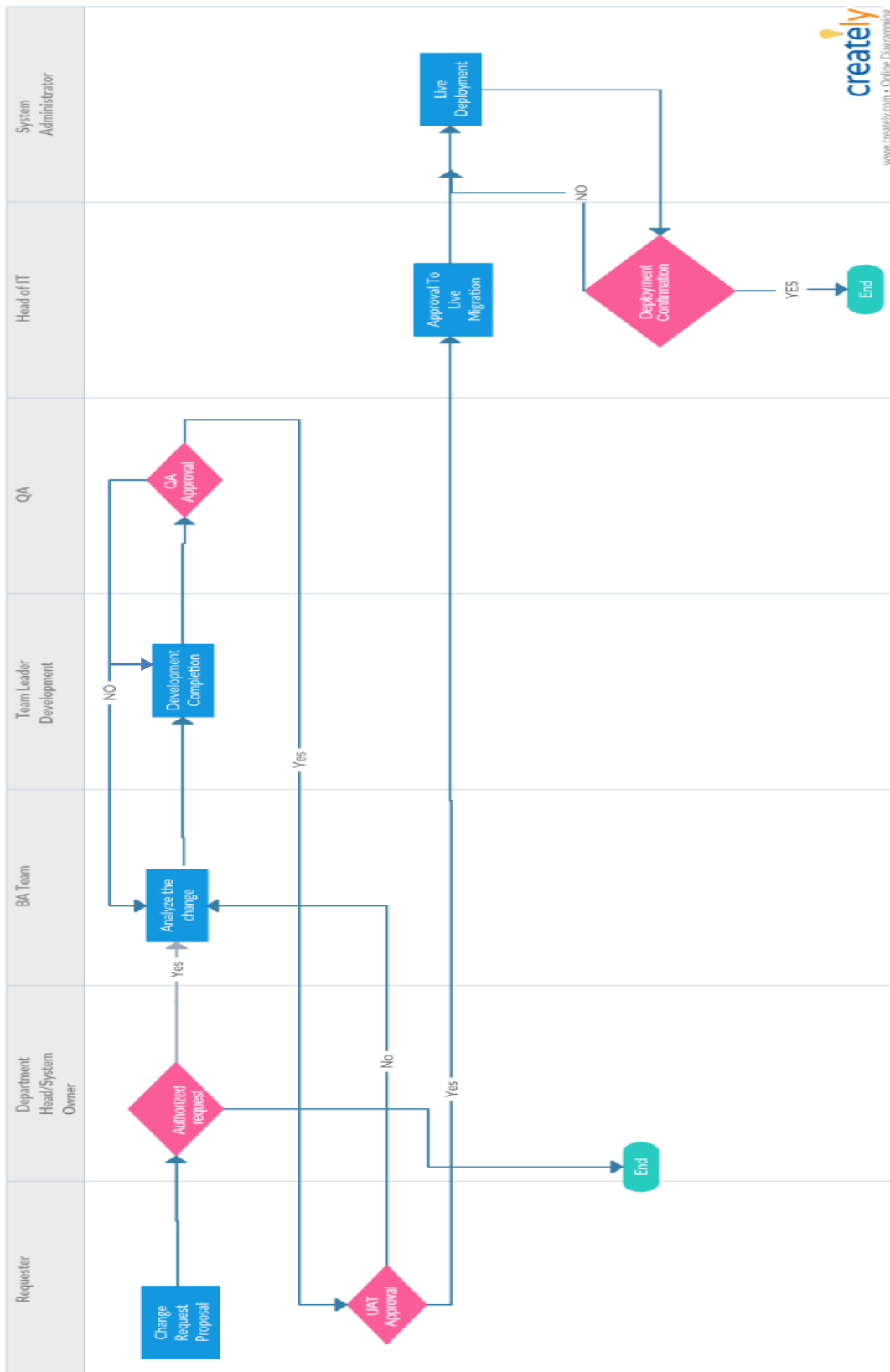


Figure 3.1: Change Management

User Management

1. User Request -

The end user has to request access to an ERP or HRIS system through this application where he needs to enter following details,

- To whom access request –
- Type of request – New Request/Access Modification
- Applications that need access – ORACLE ERP, HRIS
- Designation
- Access Role

Apart from that requested date and requester details will be captured from the system.

If it is a user termination request, it can be only initiated through HR department any other users do not have the authority to initiate such request. User termination has an additional option where the requester can set a future date to terminate.

2. Authorize Request –

Every request needs to be authorized by the relevant department head or supervisor. He has to approve the access requested by user if required department head/supervisor should be able to revise the access level.

If it is a user termination request particular request need to approve by the HR department head.

3. Create/Modify/Terminate user

Upon authorization system, the support team has to create, modify or terminate user from the system. The support team member has to set the date and time of the action he performs.

4. Confirmation of creation/termination/modification

IT department head has to confirm the action performed by a support team member. Once the confirmation has been done task should be shift to complete state.

Note :

Proposed application not directly interact with main applications Oracle and HRIS system. By means once make a user creation request that will not create a user in Oracle or in HRIS. This system only coordinates with inter-organization processes to automate the traditional paper-based process. Each user in the organization should have a separate profile in the system.

The access level for the functions has to define base on their job requirement.

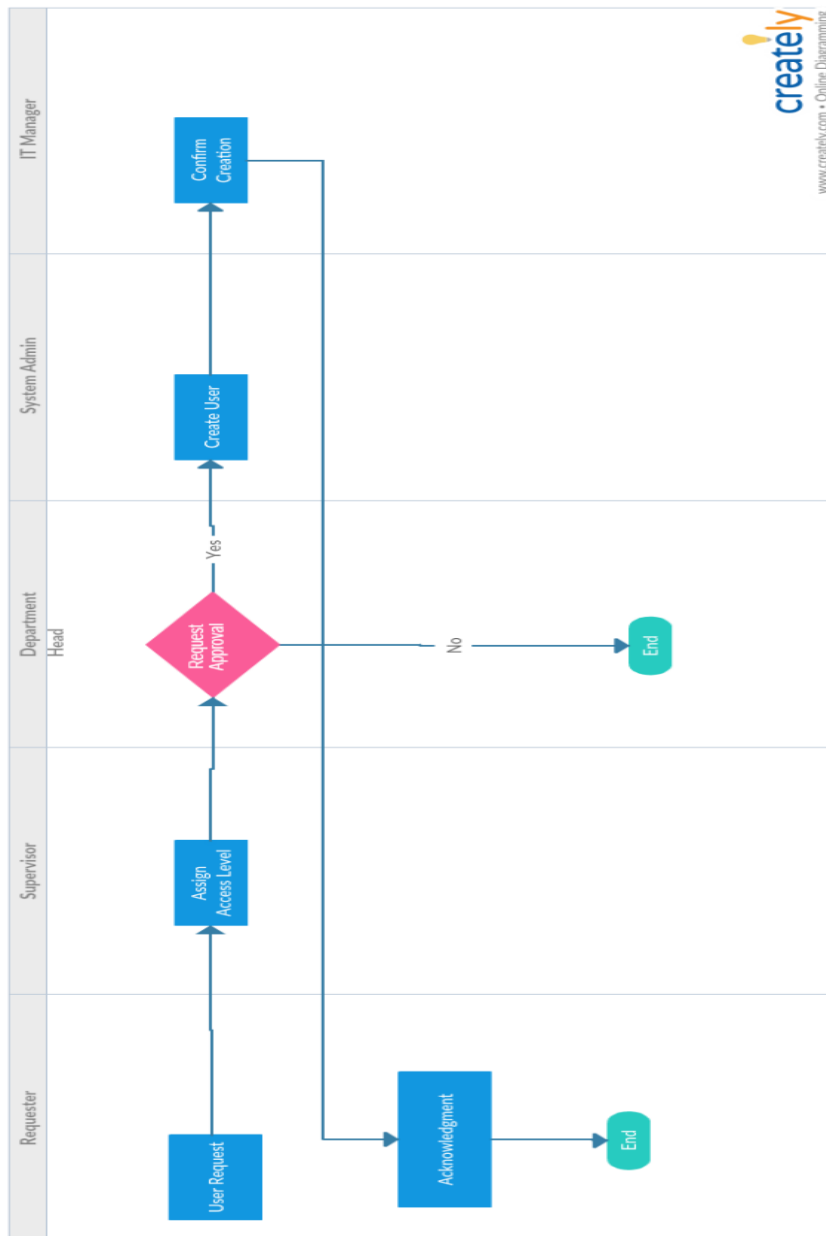


Figure 3.2: User Management

Non- Functional Requirements:

1. System should capable of taking frequent backups.
2. It is better to have colorful themes to give more attractive user experience.
3. System should have high availability since there are frequent requests all the time
4. Detailed audit trail needs to be available since these requests tend to audit frequently

3.3 Design Strategy

After evaluating different design strategies in software development such as structured, function oriented and object orient; Structured Design strategy has been selected to develop this proposed system because of its unique features. Structured design is a conceptualization of problem into several well-organized elements of the solution. It is basically concerned with the solution design. Similarly, this application has been developed as a solution to problems that been identified with regards to change and user management within the organization. As stated in the introduction chapter there were several malpractices and discrepancies along with the manual process followed. The main focus was to rectify the problem and provide the most suitable solution for those discrepancies. But when its come to function-oriented design it divides the whole system into smaller functions, which provides means of abstraction by concealing the information and their operation. Which simply provide more focus on functions of the system and focus less on problems.

Further, the proposed system is not a core application in the organization where functionalities really matter. If this proposed system had to focus more on to the feature there is a tendency where core problems will not address.

This application has developed on a personal computer with limited resources. The technical development environment as indicates in *Table 3.2* below.

Operating System	Windows 8.1 and mac os 12.1
Development tools	Vision Studio Community Edition, Cristal Reports, Adobe flash
Database	MSSQL Community version

Table 3.2:Development Environment

3.4 Design

This section discusses how the plan for the application was constructed. This includes user interface design, the structure of the application and server and how the requirements were gathered.

3.4.1 Database Design

The database design was conducted much later in the project schedule than expected. This occurs due to the problems of understanding what information needs to be captured in the application and stored in the database. The problem was discussed with management and finalized the information that is critical and needs to stores in the database. The main challenge of creating a plan for the database was solving the problem of how the data could be organized in tables and the relationships between them. The first problem was to decide what tables should exist. Therefore, a set of rules were followed to ensure the database is as efficient as possible. These rules are called "normal forms" which are part of a process called database normalization. To create the tables, Microsoft SQL 2012 was used. As displayed in *Figure 3.3* the tool gives the ability to create an enhanced table structure to display the tables in the database and relationships between the tables.

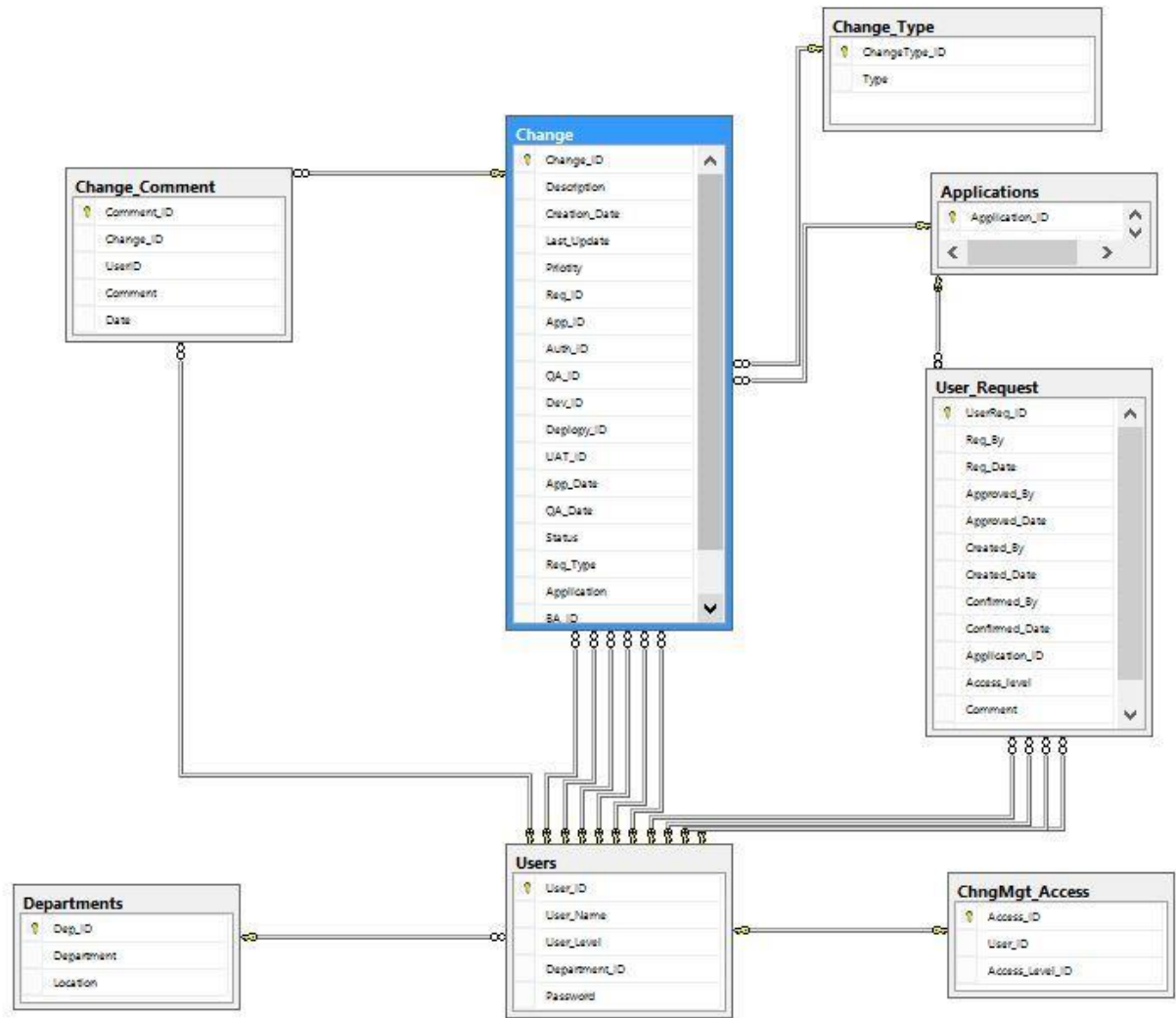


Figure 3.3: Table Structure

The relationships are also shown in the EER diagram in **Figure 3.4** to display how the tables interact with each other.

Main Tables:

- **Change_Req:** This table contains details relevant to a new change request
- **Change_Comment:** This table contains comment details pertaining to each change request
- **Users:** This table contains details pertaining to the users who utilize the application
- **User_Request:** This table contains details about new user creation request

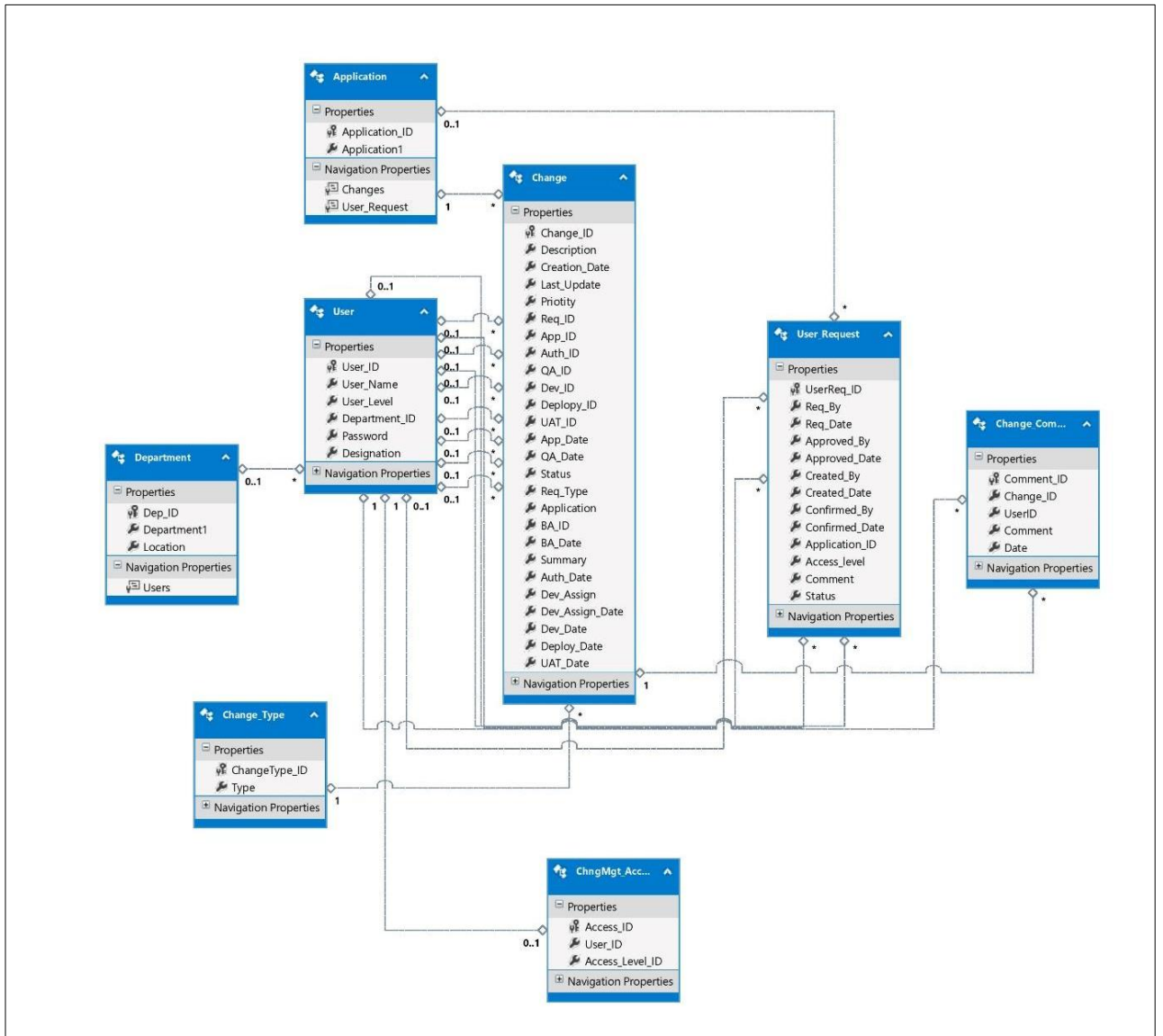


Figure 3.4: EER Diagram

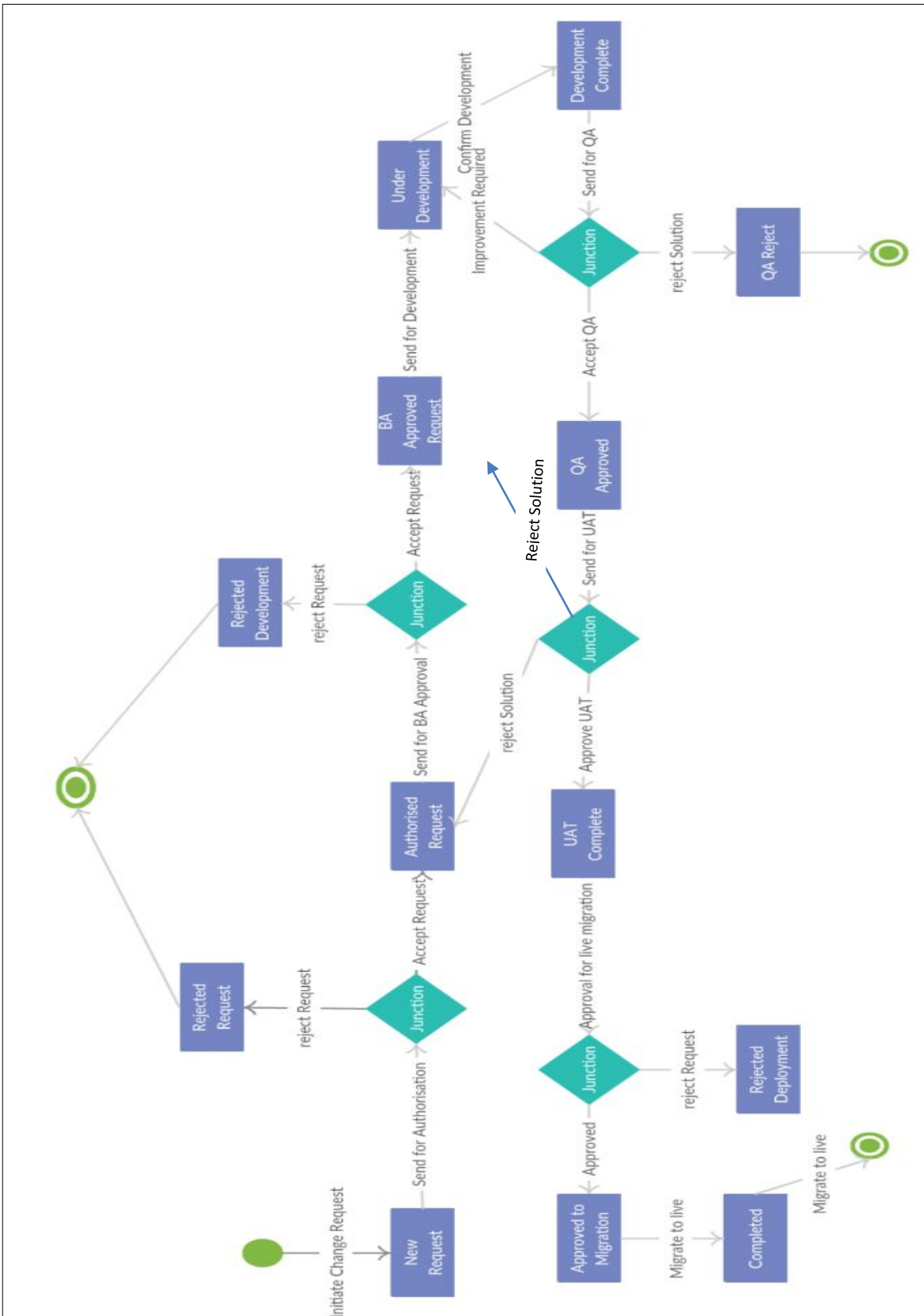
3.4.2 Application Design

As mentioned before this is a web-based application developed using C# language and uses Microsoft Visual Studio 2015 for development. Since this application is a web-based application client-server architecture is used in development. Client-side representation is developed using HTML, Bootstrap, JQuery and Javascript while server-side developed using the .net framework.

Mainly this application caters two types of user requests Change Management and User Management Requests. Each request passes several stages in the process flow. Following State diagrams indicate stages that each request passes along the process flow.

In change management workflow as indicated on *Figure 3.5* each request has to go through different stages. At the point of initialing, the request will be in “New Request” state. Once the Head of Department or Line manager authorized the request it will change into “Authorized” state. After change been authorized business analysis can conduct their analysis and convert in to “BA Approved” state by assigning particular request to a developer.

Once the developer finish development request status can be change into “developed” state. Respectively the request change then can be change in to “QA Approved” and “UAT complete” states once the QA and UAT have been completed by QA team and the end user. Upon completion of UAT head of IT can approve request for live migration and the request state change into “Approved for Live Migration”. As the final stage system admin who responsible of migrating change into production environment can switch request change into “Complete”. Apart from above states if the request has been rejected at any stage it will move into “Rejected” state.



F

Figure 3.5: State Diagram of Change Request

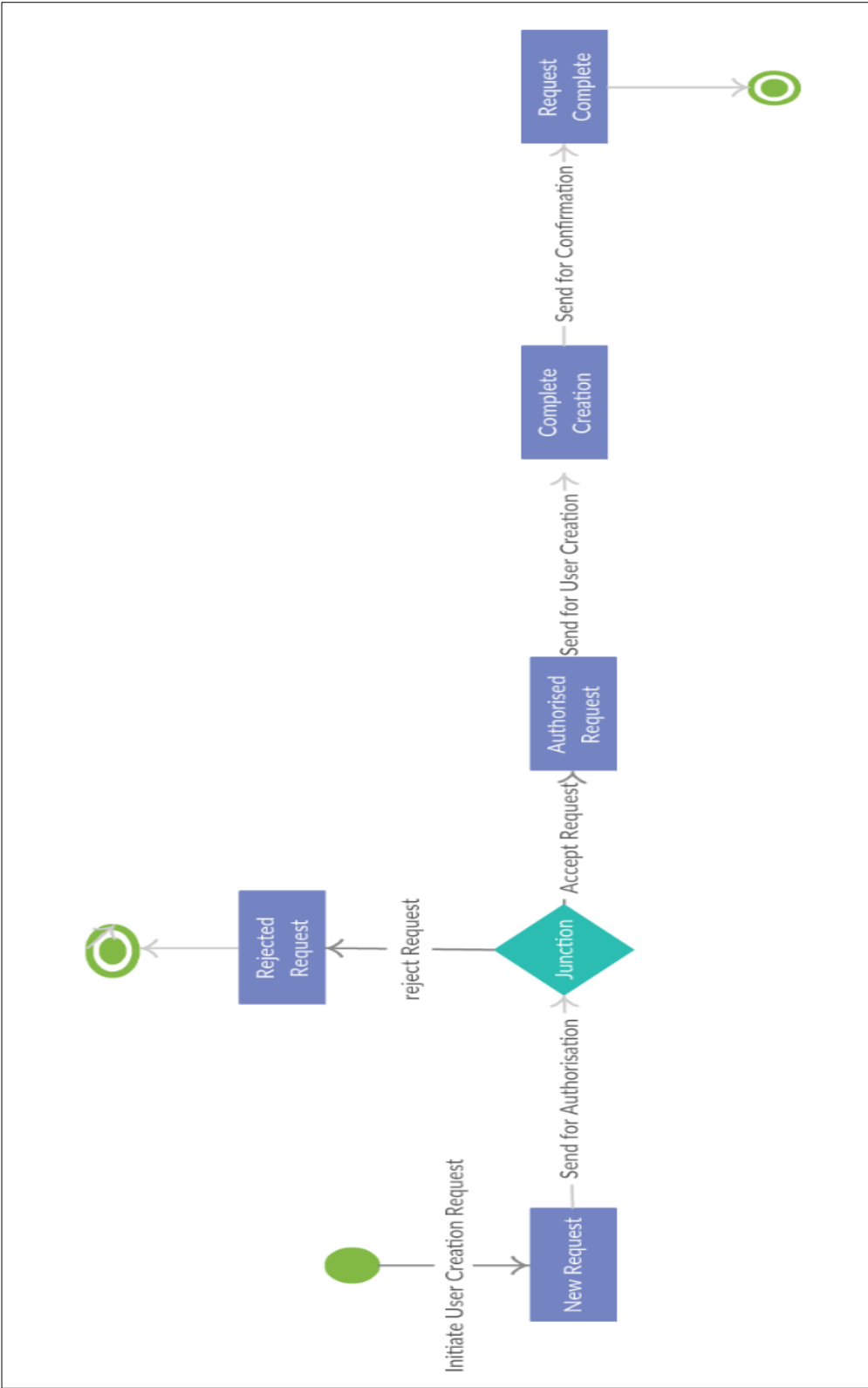


Figure 3.6: State Diagram of User Creation Request

3.4.3 User Interface Design

User Interface Design focuses on anticipating what users might need to do and ensuring that the interface has elements that are easy to access, understand, and use to facilitate those actions. Users have become familiar with interface elements acting in a certain way, so the design of interfaces focus to be consistent and predictable in choices and their layout. Which would help with task completion, efficiency, and satisfaction. [8]

In the application mainly following interface elements have been used but not limited to.

- **Input Controls:** buttons, text fields, checkboxes, radio buttons, dropdown lists, list boxes, toggles, date field
- **Navigational Components:** breadcrumb, pagination, tree view, icons
- **Informational Components:** tooltips, icons, message boxes,
- **Containers:** Panels

User stories help to understand their goals, skills, preferences, and tendencies. Which allow the application to cater following best practices in user interface design.

- **Keep the interface simple.** The best interfaces are almost invisible to the user. They avoid unnecessary elements and are clear in the language they use on labels and in messaging. Since this application uses by various level of employees the appearance kept simple and understandable.
- **Create consistency and use common UI elements.** By using common elements in user interfaces, expect users to feel more comfortable and are able to get things done more quickly. Further application users' common patterns in language, layout, and design throughout the application to help facilitate efficiency. In order to achieve the above task "Master Page" concept has been used throughout the application to ensure a common layout. When designing the layout of the page placement of each item carefully considered.
- **The system communicates what's happening.** The application always indicate the current location to users with the help of "Site map path" which facilitate user to navigate the exact location and where user can head. Further, the application is configured to acknowledge user action by prompting dialog

boxes. For example, once a user raises a change request system pop up a dialog box indicating acceptance of the request and request ID. Which help user to communicate the status of their action and reduce frustration.

- **Think about the defaults.** In order to reduce the burden on the user and reduce mistakes such as spelling, application provide dropdowns, checklist and radio buttons to select options instead of typing. This becomes particularly important when it comes to dealing with a different type of user with different literacy levels.

Login

The very clean interface has been provided to log in to the system where the user has to enter his/her Username and password to the system. If the user login to the system for the 1st time system is prompt to change the initial password given by application admin at the time of creating the user. The system does not allow to enter without changing the initial password.

Further, relevant warning messages are incorporated into the login screen to make it more convenient to the user. **Figure 3.7** shows the login screen of the application.



Please login with your Username and Password.

User Name

Password

Remember me

Login

Figure 3.7:Login Screen

Admin Panel

Admin Panel provides superuser access to the user. This account is a privileged account and should be grant access to designated personnel only. Because providing access to the unauthorized user could harm the operation. Admin has access to all functionalities including user management option of the application. But the system is configured to capture all admin actions and admin is not allowed to delete any of the entry in the system. **Figure 3.8** shows the Admin screen of the application.

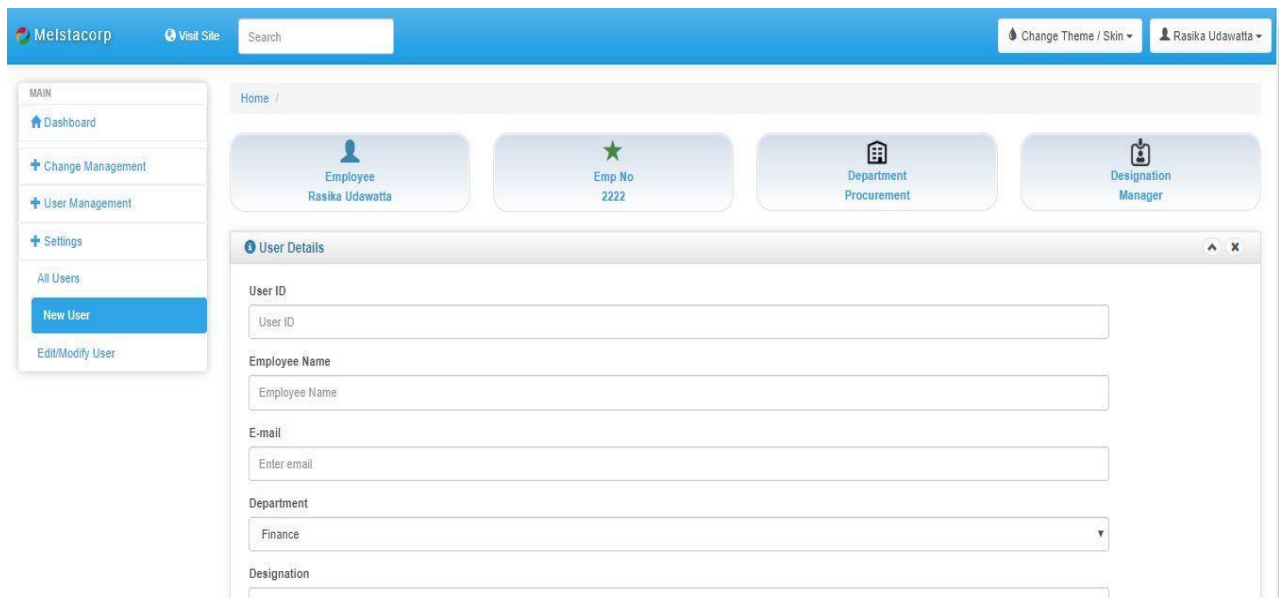


Figure 3.8:Admin Screen

User Panel

Similar to other user groups the basic user is allowed to function with within his/her job role. In this particular case, the user is allowed only to raise a change request, check their personal request and approve UAT. **Figure 3.9** shows the user screen of the application.

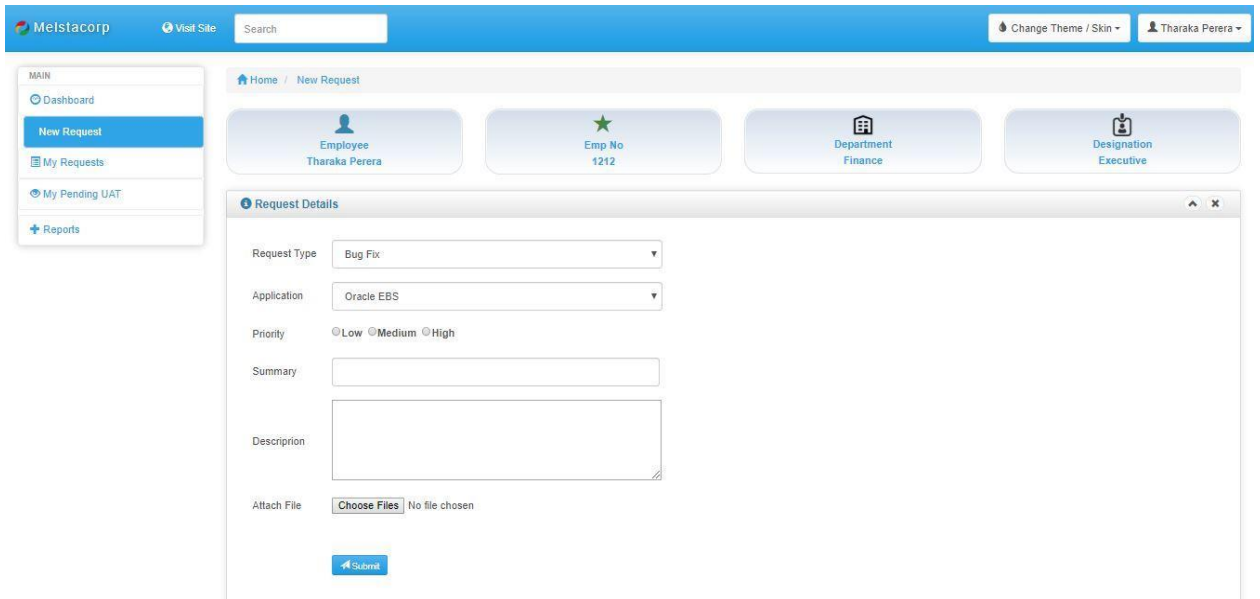


Figure 3.9: User Screen

Change Password

At any given time user is allowed to change their password if they suspect their password has been compromised. **Figure 3.10** shows the password changing screen of the application.

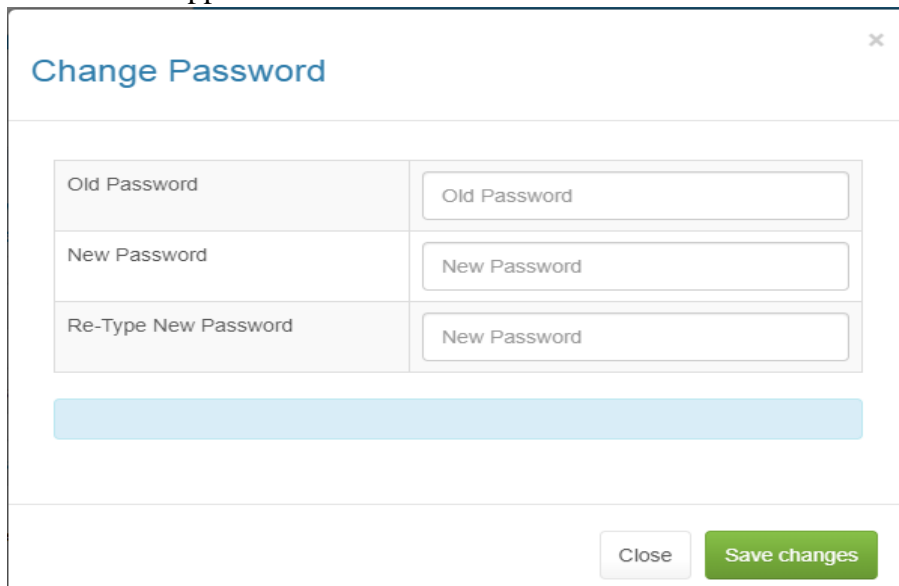


Figure 3.10: Change Password

Responsive Interfaces

Since the user may have to access the web application from various devices it's essential to facilitate responsive interfaces to the application. This application provides the capability to use the web application such as mobile phones, tablets and etc, *Figure 3.11* shows the sample responsive user interface of the application.

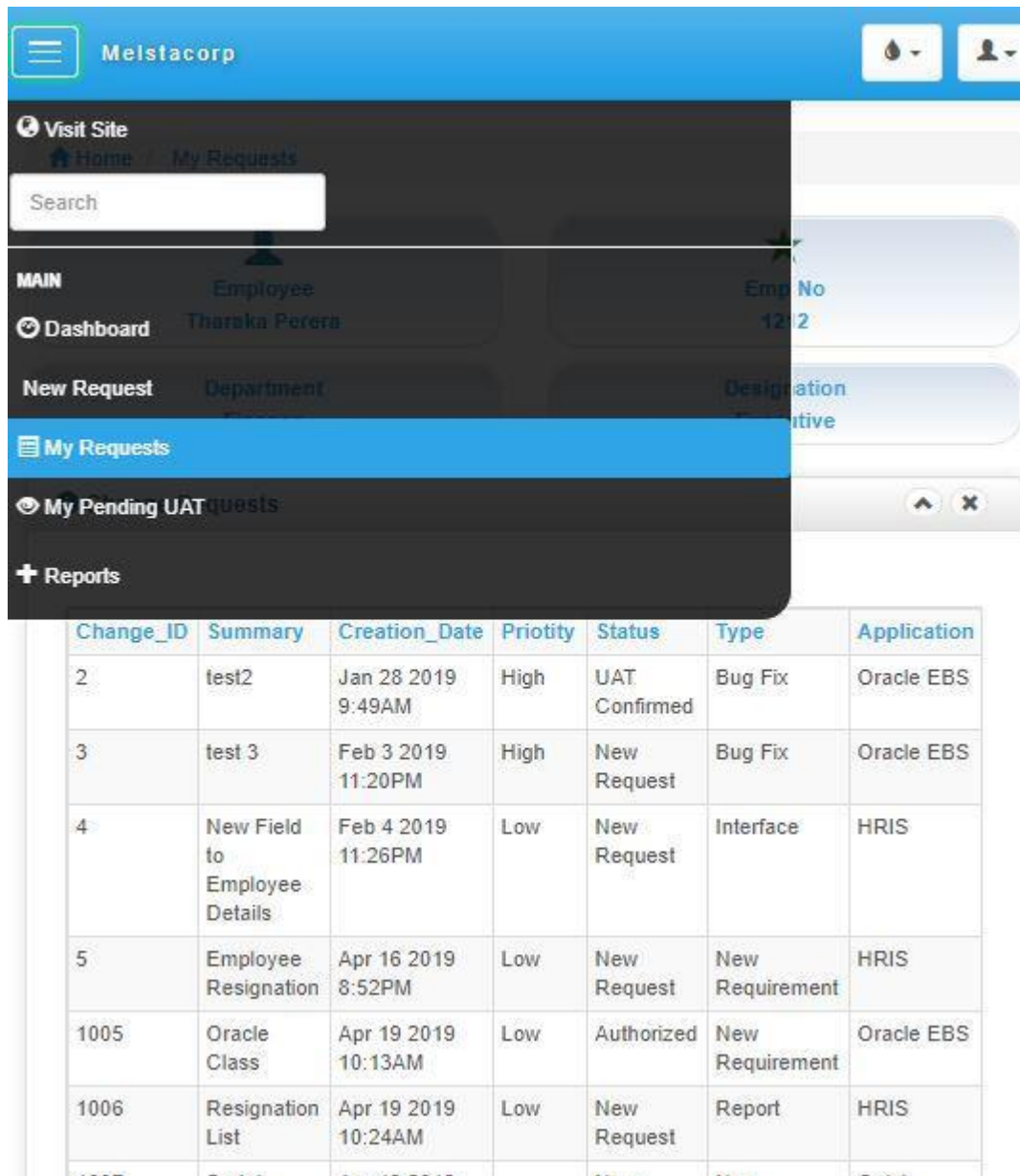


Figure 3.11: Responsive Interface

Color themes

It is common that different users prefer different themes to work on. For example color blind people or users with weak eyesight prefer to work a different theme which is comfortable for them. During our requirement gathering, it was noted that the system has to cater to the different type of users which includes users with low eyesight and the user who prefer to change the appearance of application frequently. *Figure 3.12* and *Figure 3.13* show the sample user interface themes of the application.

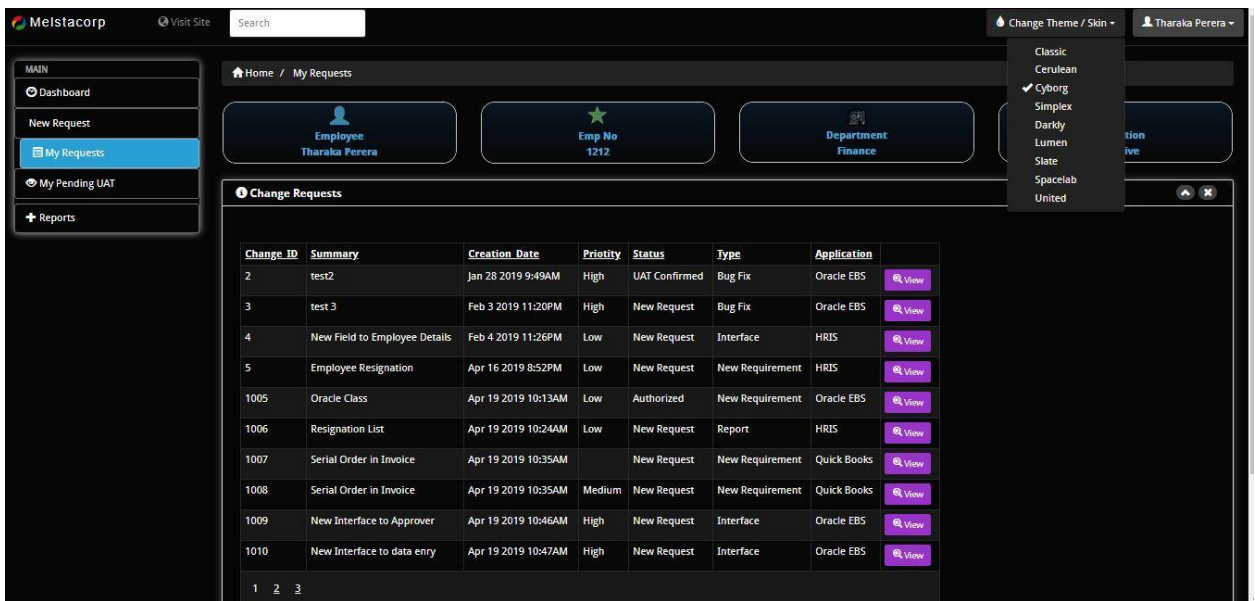


Figure 3.12:Color Theme 1

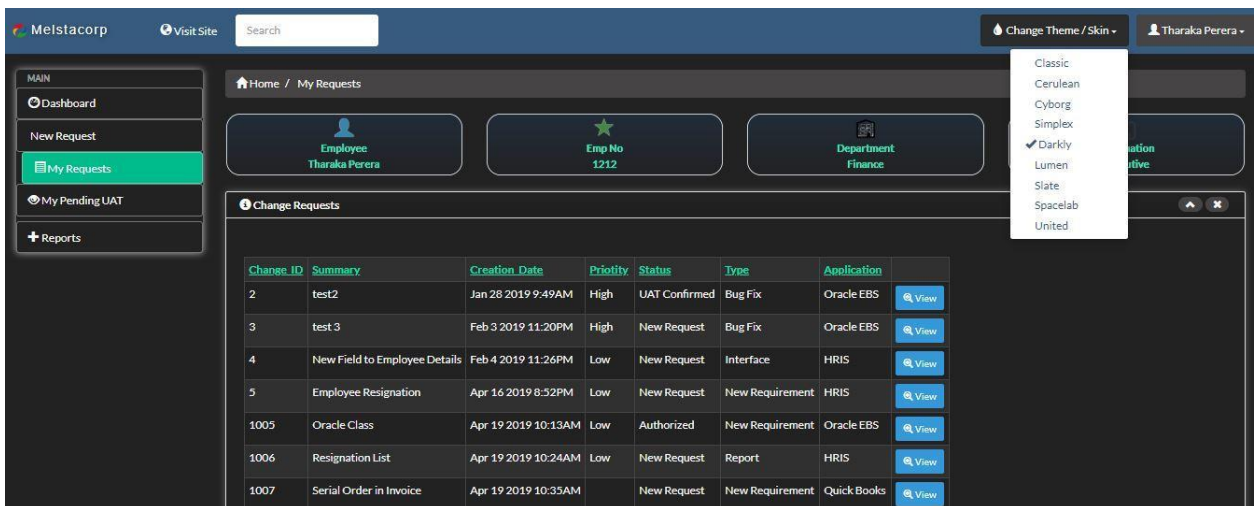


Figure 3.13:Color Theme 2

Popups

Popups have been used to immediately catch up user's attention where necessary. For an example to view additional information, warn users pertaining to the action they perform, to provide feedback about the action and etc, Further, it is the most efficient and effective way convey details to the user.

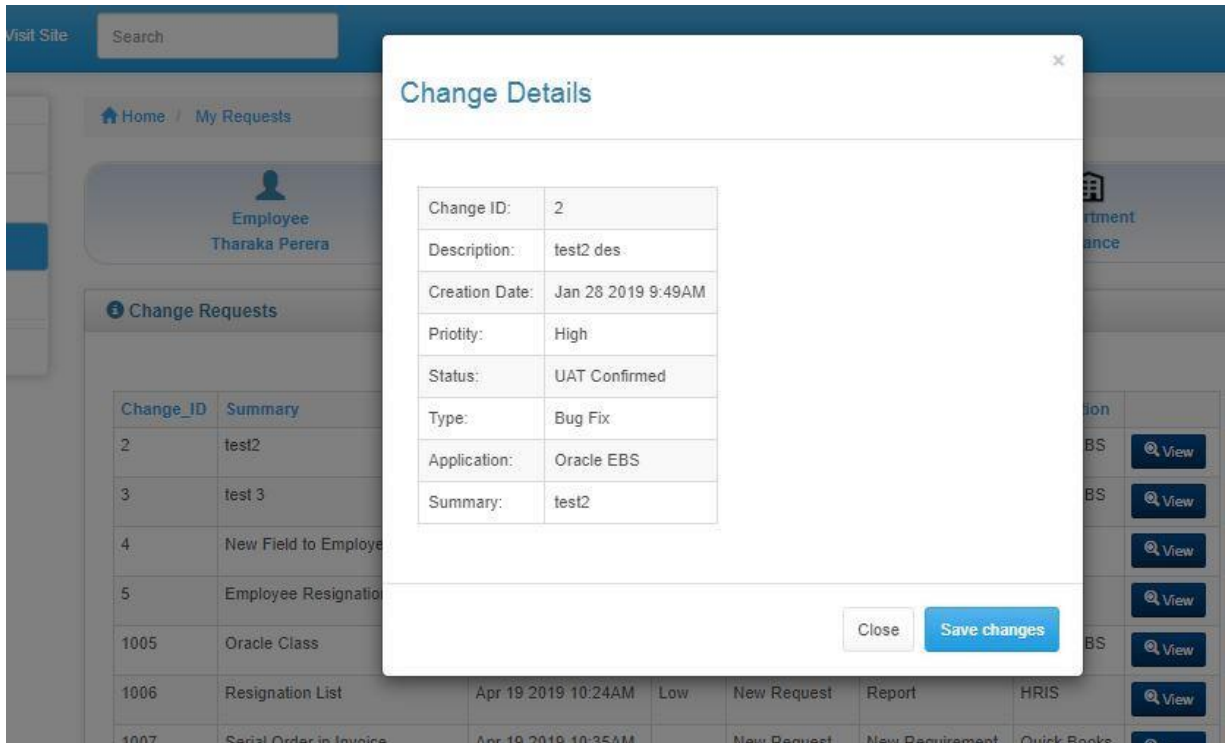


Figure 3.14: Popup

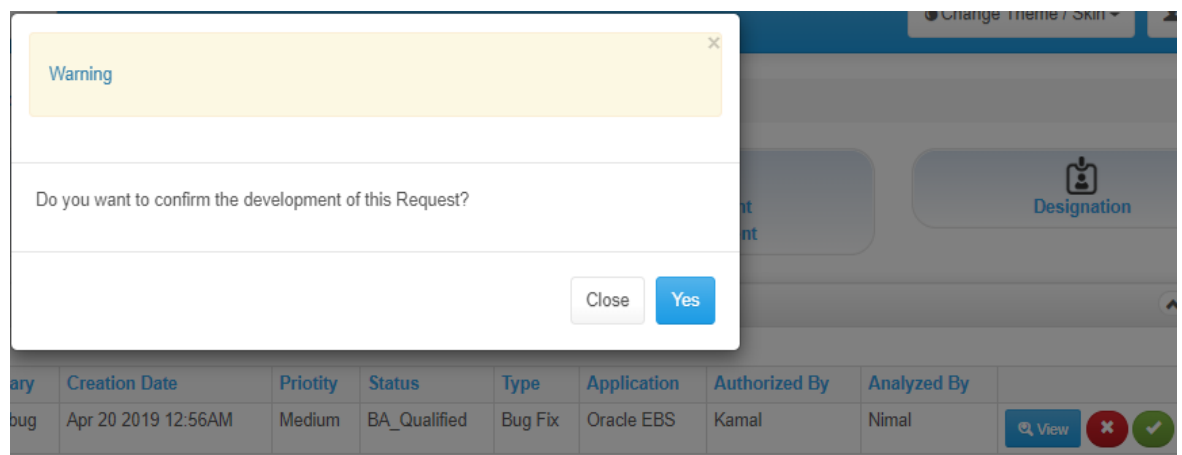


Figure 3.15: Warning

Dashboard

Graphical dashboards have been provided to each user category to give a snapshot of the existing information in the application. Which will help the user to analyze and make decisions immediately. Most importantly administrator provided with a more robust dashboard which facilitate up-to-date statistics such as progress of requests, user status, timelines and many more.

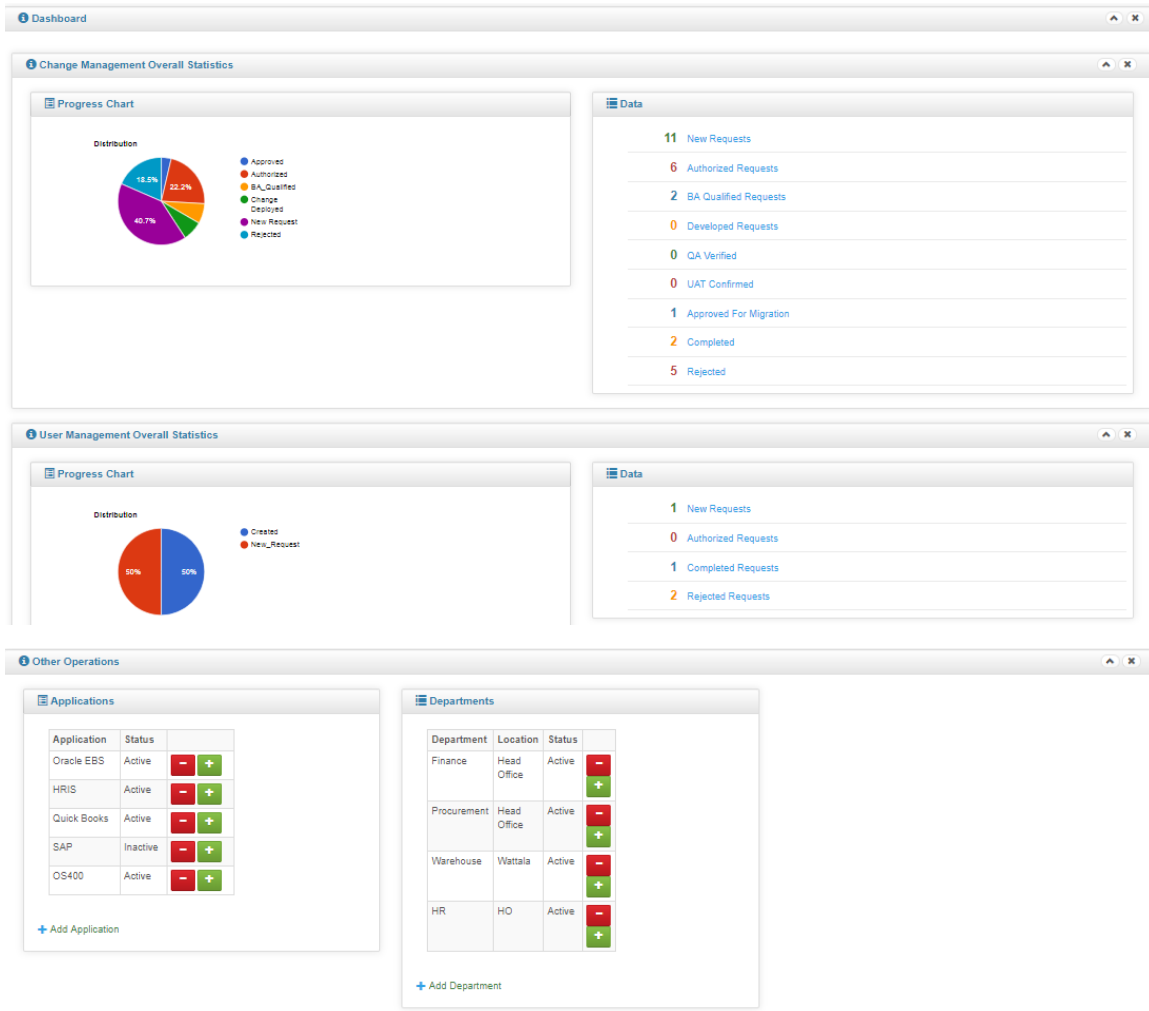


Figure 3.16 :Admin Dashboard

3.5 Testing Plan

Test plan describes the testing approach and overall framework that will drive the testing of the application.

Test Objectives

- Testing will be focused on meeting business objectives, cost efficiency, and quality.
- There will be common, consistent procedures for all teams supporting testing activities.
- Testing processes will be well defined, yet flexible, with the ability to change as needed.
- Testing activities will build upon previous stages to avoid redundancy or duplication of effort.
- Testing environment and data will emulate a production environment as much as possible.
- Testing will be a repeatable, quantifiable, and measurable activity.
- Testing will be divided into distinct phases, each with clearly defined objectives and goals.
- There will be an entrance and exit criteria.

Scope and Levels of Testing

Exploratory

PURPOSE: the purpose of this test is to make sure critical defects are removed before the next levels of testing can start.

SCOPE: First level navigation, User and admin modules

TESTERS: Testing team.

METHOD: this exploratory testing is carried out in the application without any test scripts and documentation

TIMING: at the beginning of each cycle.

Functional Test

PURPOSE: Functional testing will be performed to check the functions of the application. The functional testing is carried out by feeding the input and validates the output from the application.

Scope: Functional Requirements that been agreed with the management will be tested here

TESTERS: Testing Team.

METHOD: The test will be performed according to Functional requirements.

TIMING: after Exploratory test is completed.

User Acceptance Test (UAT)

PURPOSE: this test focuses on validating business logic. It allows the end users to complete one final review of the system prior to deployment.

TESTERS: the UAT is performed by the end users.

METHOD: Since the business users are the most indicated to provide input around business needs and how the system adapts to them, it may happen that the users do some validation according to the test cases developed.

TIMING: After all other levels of testing (Exploratory and Functional) are done. Only after this test is completed the product can be released to production.

Chapter 4 .Implementation

4.1 Database Backup

MS SQL Server configured to perform a daily full backup on the database. The task is scheduled to perform daily at 12.00 a.m since there is less traffic at that time of the day. With every backup, a backup log is generating to evaluate the status of the backup. Following *Figure 4.1* indicates a backup schedule of the SQL database.

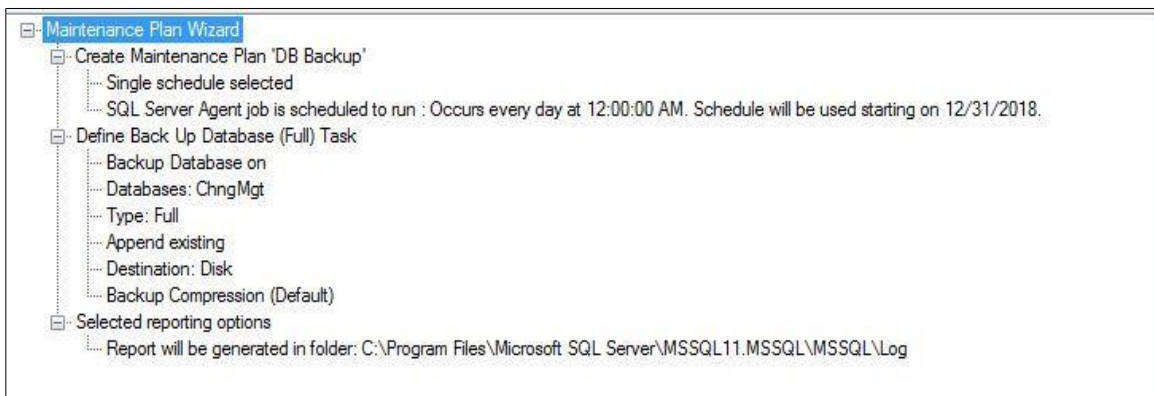


Figure 4.1:Database Backup Schedule

4.2 Basics of security

There are some common security requirements and goals in these areas that involve the protection of data. These concepts will be named and explained in the following sections.

4.2.1 Hashing Passwords

All the passwords in the application have been encrypted using the SHA256 hashing algorithm instead of storing as plain text in the database. Which ensure no one can steal the passwords. Further, since application store hash value of the password intruders can't decrypt the password to its plain text value.

	User_ID	User_Name	User_Level	Department_ID	Password
1	1234	Amal Silva	2	1	mDSHbc+wXLFnpcJJU+uljErImxfV/KPL50JrxB+6PA=
2	1111	Kamla Deva	1	2	jGI25bVBBBW96Qi9Te4V37Fnqchz/Eu4qB9vKrRlqRg=

Figure 4.2: Hashed Passwords in SQL database

```

z.references
public String GenerateSHA256Hash(String input)
{
    byte[] bytes = System.Text.Encoding.UTF8.GetBytes(input);
    System.Security.Cryptography.SHA256Managed sha256hashstring = new System.Security.Cryptography.SHA256Managed();
    byte[] hash = sha256hashstring.ComputeHash(bytes);
    return Convert.ToBase64String(hash);
}

```

Figure 4.3: Function used to Hash the password

4.2.2 Allow Only authorized users

To protect the confidentiality and integrity of data access is grant only to authorized users and access levels have been maintained to ensure information access base on the need to know and need to do basis. Following **Table 4.1** and **Table 4.2** indicate basic access levels implement in the application and the functionalities incorporate with each access level.

Role	Functions
Admin	All the functions including application settings
Authorizer	Authorize Change request
BA Approver	Approve/Reject Request for Development
QA Approver	Accept
Developer	Complete development
Deployment Approval	Approval for live migration
Deployer	Migrate change to the live environment
User	Raise change request, Accept UAT

Table 4.1: Change Management Roles

Role	Functions
Admin	All the functions including application settings
Authorizer	Authorize User Creation request
User Creator	Confirm of user creation
Confirmation	Acknowledgment of creation
User	Raise user creation request

Table 4.2: User Management Roles

The administrator of the application facilitate to provide access to other users who uses the application. Depending on their job role and responsibilities administrator can allocate access. Since Graphical User Interface has been provided to select required access it is easy for the administrator to provide access without any confusions.

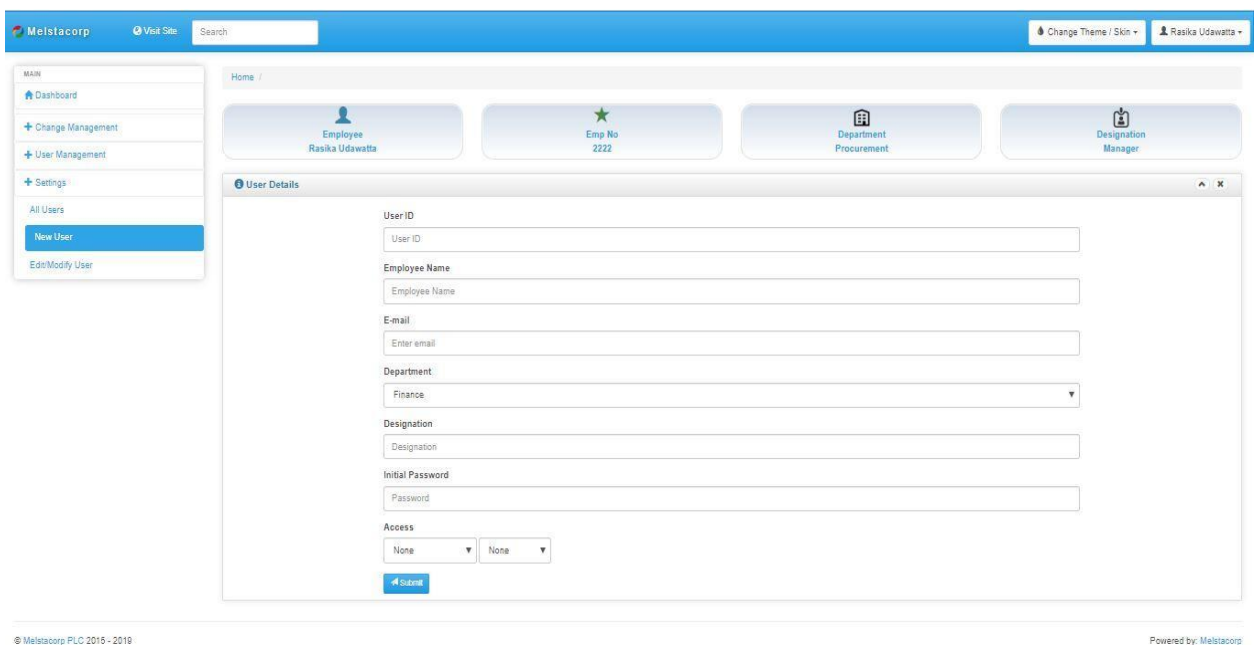


Figure 4.4: User Creation Interface

4.2.3 Transaction Log

Transaction logs/ Audit Logs highly important to held accountability of user actions and investigates any misuse of information. Also, transaction logs/audit logs provide transparency among the process. This application caters this transparency by capturing every action conduct by the user with the user ID and the time stamp. Once the request is made application does not allow to delete the request during the process flow.

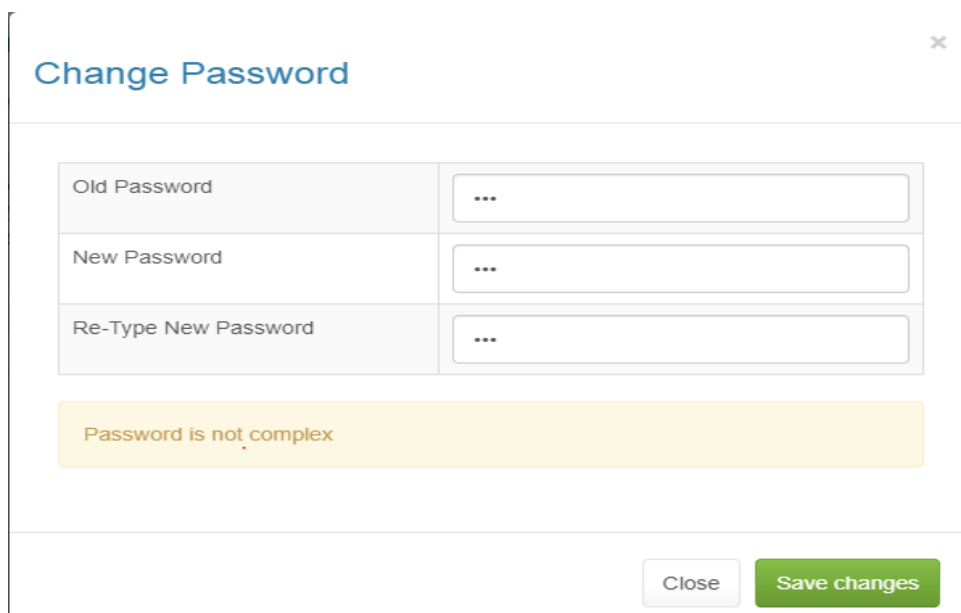
When comes database Microsoft SQL inbuilt C2 Audit log function has been enabled to ensure capturing all modification conduct through the backend.

4.2.4 Input Validation

The application complies with various input validation methods before passing data to the storage. This helps to prevent mistakes as well as intentional intruder attacks such as SQL injections and etc. There are two different types of form validations used in this application, Client-side validation, and Server-side validation. Most of the occasions client-side validation has been used rather than server-side validation since it is more user-friendly than server-side validation as it gives an instant response.

Validations such as Password validation to ensure a minimum number of characters and complexity have been maintained when creating a password, Employee ID validation to ensure only numeric values been entered have been incorporated into the application.

Server-side validations which occur on the server after the data has been submitted. For an example user login uses server-side code to validate the login data before it allows access to the application.



The screenshot shows a 'Change Password' dialog box. It contains three input fields: 'Old Password', 'New Password', and 'Re-Type New Password'. Each field is currently empty and has a masked input type (indicated by three dots). Below the input fields, there is a yellow error message that reads 'Password is not complex'. At the bottom right of the dialog, there are two buttons: 'Close' and 'Save changes'.

Figure 4.5: Password Complexity

```

8 references
public bool CheckPassword(string password)
{
    string MatchPassPattern = "(?=^.{6,}$)(?=.*d)(?=.*[a-zA-Z])";

    if (password != null) return Regex.IsMatch(password, MatchPassPattern);
    else return false;
}

```

Figure 4.6: Password Complexity Coding

```

0 references
protected void BtnLogin_Click(object sender, EventArgs e)
{
    try
    {
        Int64 uid = System.Convert.ToInt64(TxtUserName.Text);
        string pass = enc.GenerateSHA256Hash(TxtPWD.Text);

        con.Open();
        string qry = "select * from Users where User_Id=@User_ID and Password=@Password";
        SqlCommand cmd = new SqlCommand(qry, con);
        cmd.Parameters.AddWithValue("@User_ID", uid);
        cmd.Parameters.AddWithValue("@Password", pass);
        SqlDataReader sdr = cmd.ExecuteReader();
        if (sdr.Read())
        {
            LoginStatus.Text = "Login Success.....!!";
            Response.Redirect("~/Home.aspx");
        }
        else
        {
            LoginStatus.Text = "UserId & Password Is not correct Try again..!!";
        }
        con.Close();
    }
    catch (Exception ex)
    {
        Response.Write(ex.Message);
    }
}

```

Figure 4.7: Server Side Validation Example

4.3 Code Module

The system design used in the application is a layered model, which is the client server three-tier system architecture of a browser front-end, a web-server in the middle tier and a databases server at the backend shown in **Figure 4.8**.

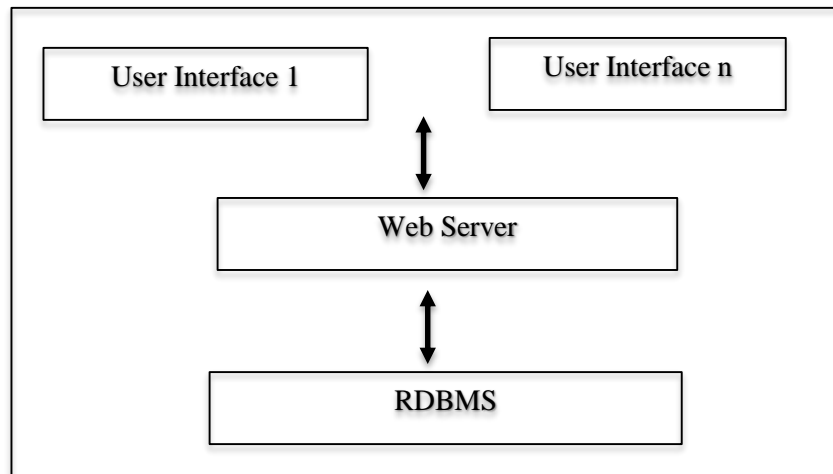


Figure 4.8: Coding Structure

In the three-tier client server, architecture the user, using a web browser located at the client end requests web pages from device. The web server takes these requests and passes them on to the database as SQL queries. The SQL server processes the queries and sends the results set back to the web server. This data is processed as specified in the script and the results served back to the user as HTML pages for display depending on the client's web browser in the device.

The ASPX script is executed in the web server is called server side scripting and is not visible to the client. This protects the script from tampering by users and protects the processing functionality of the application. Sometimes it is not advisable to perform all data validations on the server as they tie up the web server system resources leading to a drop in system performance. To overcome this problem there should be a client side script, which is executed by the browser-scripting engine. The application makes use of server side script for its processing functionality.

In view of the interactivity that is required in a web-enabled application to modularize the code and distribute it among several interconnected web pages. This results in having to address the problem of maintaining state between web pages. This is done by the use of session variables (like global variables) and local variables, which are populated by transferring data required.

Handling sessions is very important part in any form of a web application. It has to follow two steps, define the session variable and assign a value to it when submit the request at the web form.

When the user clicks the Submit button and if it is a valid request, and if all the information is correctly entered, request is processed by the system. If any error occurred while submitting, it is captured using a try-catch block and the error information passes to the user.

Chapter 5 . Testing and Evaluation

The assessment involves checking whether the software and the project that develops it, conforms to various characteristics or exhibits various qualities that are expected of a sustainable product.

The more characteristics and requirements that are satisfied, the more sustainable the software. In performing the evaluation consider how different user classes affect the importance of the criteria. For example, for Usability-Understandability, a small set of well-defined, accurate, task-oriented user documentation may be comprehensive for Users but inadequate for Developers. Assessments specific to user classes allow the requirements of these specific user classes to be factored in and so, for example, show that a project rates highly for Users but poorly for Developers, or vice versa. Scoring can also be affected by the computer literacy of users themselves

Evaluation is an ongoing process where most of the time developers and Users including top management require time to adapt to the new system and their functionalities. Performing a post-implementation evaluation provides limited feedback from stakeholders. But this will help to identify whether the system has been managing to achieve the expected outcome.

The project evaluation of this application mainly focuses on the following four criteria

5.1 Usability aspects of software

Usability Evaluation focuses on how well users can learn and use an application to achieve their goals. It also refers to how satisfied users are with the process. By conducting usability evaluation, product owners will be able to identify the user's experience when interacting with applications.

It is important to realize that usability is not a single, one-dimensional property of a product, system, or user interface. 'Usability' is a combination of factors including:

Intuitive design: a nearly effortless understanding of the architecture and navigation of the site

Ease of learning: how fast a user who has never seen the user interface before can accomplish basic tasks

The efficiency of use: How fast an experienced user can accomplish tasks

Memorability: after visiting the site once, the user should be able to remember enough to use it effectively in future visits

Error frequency and severity: how often users make errors while using the system, how serious the errors are, and how users recover from the errors

Subjective satisfaction: If the user likes using the system

In order to have a clear understanding about the usability of the application, separate section has been incorporated into the questioner (Annexure_A). The questions are consist of various sub criteria where users were expected to give their honest opinion about the application. According to the feedback received over roll response as follows.

User Level	Sample Size	Strongly Agree	Agree	Disagree	Strongly Disagree	N/A
End User	15	9	5	1	-	-
Authorizer (Head of Department)	8	5	3	-	-	-
Business Analysis	5	2	3	-	-	-
Developer	5	2	3	-	-	-
Quality Checker	3	1	2	-	-	-
IT Managers	2	2		-	-	-
System Admins	3	1	2	-	-	-
Total	41	22	18	1	-	-

Table 5.1: Usability Feedback

Feedback denotes that, as an average most of the users comfortable with the usability features provided in the application. This indicates that there will be less resistance towards new application.

5.2 Information in the Program

This application contains a vast amount of information but it's a matter of how well an application can present that information to the relevant people in clear format. For example, passwords are confidential information in an application and nobody should have access to the retrieve that information. It is paramount important to control access to the information.

Enabling restrictions is not enough to facilitate a more robust user experience. Presentation of information also plays an important role in information accessibility. Presentation of information should be clear and understandable base on the audience. This application tries to provide maximum user satisfaction while ensuring information security to minimize user resistance towards new application. Few steps took to achieve information control in the application as follows,

- Facilitate Access Control
- Use of simple English and use common symbols where possible
- Using color codes
- Use of generic template throughout the application
- Presentation of information in tables

5.3 Technical Aspects of software

This application is not designed to function as a high-end application such as ERP systems. This application provides supporting function to the organization which is also critical to streamlining organization processes. Therefore, it is not acceptable to use high tech equipment and software to provide a platform. This application is design to run with basic technical specifications. Below **Table 5.2** indicates the required specifications.

Operating System	Min Windows 7 (Windows Server 2008 or higher recommended)
Database	SQL Server 2015 or higher
Graphics	Standard
Internet Connection	ADSL for higher bandwidth
Capacity	8 GB or higher
Backup	Recommended daily backup

Table 5.2:Spec Sheet

In technical evaluation also focus on the physical and logical security of the application even though this application can be run on primitive hardware and software it is recommended to run on reliable resources. For an example this application can use windows server 2003 and below also can use SQL Server 2003 and below. But it is not recommended to use those Servers since Microsoft has stopped supporting those versions.

Harding of the application also considers when developing the application to minimize resource usage and improve performance of the application.

5.4 Functional Aspect

Function aspects evaluate whether application manages to satisfy a higher number of functional requirements collected at the requirement gathering phase. It is essential to cater to all critical functionalities unless application may not be able to achieve the expected outcome. It is acceptable to skip minor requirements if there are any technical or practical difficulties but those requirements also satisfy where possible to avoid user frustration and resistance. There are some requirements application unable to cater due to technical and practical difficulties those will be discussed in the latter part of this section.

The evaluation Feedback form has been used to obtain a response from the deferent level of stakeholders which includes end users, department heads, IT personnel and top management. base on their feedback average marks has been given to each evaluation criteria. Please Refer *Annexure_A* for Feedback format.

Below *Table 5.3* indicate User level wise total average score.

Designation	Sample Size	Average Score
User	15	8.2
Authorizer (Head of Department)	8	7.6
Business Analysis	5	7.8
Developer	5	7.5
Quality Checker	3	8.0
IT Managers	2	8.2
System Admins	3	8.9

Table 5.3:User Level Wise Average Score

Below *Table 5.4* indicate Criteria wise total average score obtained by the user.

Criteria	Score (Out of 10)
Information in the Program	
The information is clear, concise, attractive and informative to the intended audience	7.9
The language is nondiscriminatory. Content is free from race, ethnic, gender, age, and other stereotypes	8.8
The content is free from spelling and grammatical errors	9.1
Information is accessible only base on the privileged level	8.2
Updated information is distributed promptly	8.5
Information is validated before the store	8.0
Technical Aspects of the Software	
The system uses standard equipment that is reliable, widely available, and applicable to a variety of uses	8.3
Computer capabilities such as graphics, color, or sound are used for appropriate instructional reasons	8.5
The program is reliable in normal use. Software is bug-free	8.0
If the program creates a permanent record for a user, that record is secure and confidential	8.6
The system is implemented to utilize the resources for the maximum capacity	8.6
Functional Aspect	
All the functions operational as per the acceptable level	8.9
The pre-defined workflow is maintained for each function.	8.0
Functions are error-free for all test data	8.2

Table 5.4: Evaluation Criteria

5.5 Technical and Practical Difficulties.

There are a few tasks that unable to complete due to technical difficulties such as there was a requirement to integrate this application with the organization's ERP system to create users. Which facilitate once the user creation request has been made it will automatically create a user in the ERP system upon approval. But integration with ERP system is highly technical and specialized expertise required to cater the integration. Further, organization denied to provide administration access to ERP applications to perform implementation and testing.

As depicts on the evaluation criteria score is less under "The information is clear, concise, attractive and informative to the intended audience" criteria. Upon investigation, it was identified that the main reason for that is even though interfaces provide adequate information to the users, they mainly focus on the attractiveness of interfaces. Further, end-user expects to view all details pertaining to their change request. Including, suggested patch by analytical team, parameters set by developers to fix the bug and communications had with vendors. But due to security reasons and technical complexity the application does not display all the information but the relevant and useful information to a particular user group.

Chapter 6 . Conclusion

6.1 Project Summary

The aim of this project is to provide an innovative and convenient approach to business process reengineering and the development of centralized Change Management system. Application development has a close linkage with business process reengineering within the organization.

Not only is one tenet to streamline IT support over application changes and also to provide transparency among business processes along with the workflow.

As concluded in chapter 1, information systems are part of the business environment and their design and use cannot be specified solely in just technical terms. The lack of this consideration has contributed to the high rates of failure for application management. Also, it was found in this project that information systems management in conventional paradigms were not capable enough of fulfilling frequent change requirement of an organization.

Another common problem faced by both change management and user management is the role of human factors because human is the weakest link in information security, productivity, and cost.

Automating most of the processes could help to mitigate the above deficiencies. As a solution, this application helps to streamline internal processes and integrate better controls into the process flow.

6.2 Limitations of the Project

During the project, some difficulties and several technical limitations have been arising. Such limitations may partly result from the scope of this project, and partly from limited time and resources available.

The aim of this project has been to introduce a new change management application for different groups of employees to be involved in the process of system change management. However, there will be some social issues, such as user satisfaction and worker resistance, which cannot be fully addressed in the initial stage. Such social issues can only be highlighted through other methodologies such as training, awareness sessions and technical support. Addition to those non-technical approaches, application itself provides with user-friendly interfaces, navigation, and guidance.

Even though this application implemented for the specific organization they did not allow access to existing ERP applications such as Oracle EBS, HRIS, and Quickbooks to make the integration. Therefore, this application functions as a standalone application.

6.3 Future Work

There is clearly future work to be done on exploring the scalability of the application. The initial approach would be integrating change management application with existing ERP applications. By doing that organization gain competence to directly interfere with change management and user management activities. For example, if the application coupled with the existing Oracle ERP system, once the development has been completed and approved deployer (System Admin) could be able to migrate change to production environment through change management application itself, which provides more transparency over change management process. Similarly, user management also could be integrated with existing applications.

Another enhancement would be the addition of an incident management module. Which could help the organization to cater to all the incidents in a centralized platform. Which include virus attack, device malfunctions, repairs and etc.

Even though this application supports various devices through responsive interfaces developing mobile application could help users to offline access and more efficient interaction with the system.

Bibliography

- [1] iTop, "iTop Hub," iTop, 19 12 2018. [Online]. Available: https://www.itophub.io/wiki/page?id=2_0_2%3Astart.
- [2] bugzilla org, "bugzilla," bugzilla org, 2019. [Online]. Available: <https://www.bugzilla.org/about/>.
- [3] BMC Software, Inc., "IT Service Management," BMC Software, Inc., 2019. [Online]. Available: <https://www.bmc.com/it-solutions/remedy-itsm.html>.
- [4] Capterra Inc., "Change Management Software," Capterra Inc., 2019. [Online]. Available: <https://www.capterra.com/change-management-software/>.
- [5] Pilgrim Quality Solutions, "SmartSolve Change Management," Pilgrim Quality Solutions, 2019. [Online]. Available: <https://www.pilgrimquality.com/solutions/process-management/change-management-software>.
- [6] INVENSIS LEARNING, "What is ITIL Access Management?," Invensis Learning Pvt Ltd., 2019. [Online]. Available: <https://www.invensislearning.com/resources/itil/what-is-itil-access-management>.
- [7] Freshworks, "ITIL Change Management Process," Freshworks, [Online]. Available: <https://freshservice.com/change-management-software/itil-change-management>.
- [8] J. J. Garrett, P. Morville and L. Rosenfeld, "User Interface Design Basics," Usability.gov, [Online]. Available: <https://www.usability.gov/what-and-why/user-interface-design.html>.
- [9] Refsnes Data., "w3schools," Refsnes Data., 2019. [Online]. Available: <https://www.w3schools.com/>.
- [10] MIT, "Bootstrap," MIT, 2019. [Online]. Available: <https://getbootstrap.com/>.
- [11] U. Contribution, "Stack Over Flow," Stack Exchange Inc., 2019. [Online]. Available: <https://stackoverflow.com/>.
- [12] The Bugzilla Team, "The Bugzilla Guide - 4.2.16+ Release," 22 12 2015. [Online]. Available: <http://www.bugzilla.org/docs/>.
- [13] B. B. D. Kevin P. Duffy, "Using ITIL to Improve IT Services," *Association for Information Systems*, no. 2008, 2008.

- [14] S. K. R. M. Mathew Nicho, "Challenges and Lessons Learned Implementing ITIL," *ISACA JOURNAL*, vol. 4, 2017.
- [15] D. Bandyopadhyay, "10 Key Rules for Using the ITIL Framework Effectively," *ISACA Journal* , vol. 4, 2012.

Annexure A

System Evaluation Form

Part A: Personal Details

Name:.....

(Not Mandatory)

Gender: a. Male b. Female

Working Experience:.....

Age:.....

Designation:.....

Department:.....

Applications Uses :

- a. Oracle EBS b. HRIS c. Quickbooks
d. Other (specify)

Part B: Views on ICT based learning

	Very high	High	Moderate	Low	Not at all
a. Computer availability	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b. Availability computer with internet facility	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c. Usage of a computer for day-to-day activities	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d. Support from supervisor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
e. The responsiveness from IT support service	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Part C : Application Usability Evaluation Details

	Strongly Agree	Agree	Disagree	Strongly Disagree	N/A
a) Overall, I am satisfied with how easy it is to use this application.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
b) It was simple to use this application.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
c) I could effectively complete the tasks and scenarios using this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
d) I was able to complete the tasks and scenarios quickly using this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
e) I was able to efficiently complete the tasks and scenarios using this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
f) The system gave error messages that clearly told me how to fix problems.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

g) Whenever I made a mistake using the system, I could recover easily and quickly.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
h) The information (such as online support, on-screen messages, and other documentation) provided with this system was clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
i) It was easy to find the information I needed.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
j) The information provided for the system was easy to understand.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
k) The organization of information on the system screens was clear.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
l) The interface of this system was pleasant.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
m) I liked using the interface of this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
n) This system has all the functions and capabilities I expect it to have.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
o) The information was effective in helping me complete the tasks and scenarios.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
p) Overall, I am satisfied with this system.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Part D : Application Technical Evaluation Details

Criteria	Score (Out of 10)
Information in the Program	
The information is clear, concise, attractive and informative to the intended audience	
The language is nondiscriminatory. Content is free from race, ethnic, gender, age, and other stereotypes	
The content is free from spelling and grammatical errors	
Information is accessible only base on the privileged level	

Updated information is distributed promptly	
Information is validated before the store	
Technical Aspects of the Software	
The system uses standard equipment that is reliable, widely available, and applicable to a variety of uses	
Computer capabilities such as graphics, color, or sound are used for appropriate instructional reasons	
The program is reliable in normal use. Software is bug-free	
If the program creates a permanent record for a user, that record is secure and confidential	
The system is implemented to utilize the resources for the maximum capacity	
Functional Aspect	
All the functions operational as per the acceptable level	
The pre-defined workflow is maintained for each function.	
Functions are error-free for all test data	

Annexure B

Test case ID:	0001		
Test priority (Low/Medium/High):	High	Test Designed By:	Thilina Kelaniyage
Module Name:	Change Management	Test Designed Date:	28/01/2019
Test Executed By:		Test Execution Date:	28/01/2019

Test Title:	Change Management												
Test Summary/Description:	Request new change												
Test Steps:	<p><u>Step 1:</u> Log into the system via user login Login ID:1212</p> <p><u>Step 2:</u> Navigate to the "New Request" from the side menu</p> <p><u>Step 3:</u> Feed request details which include change type, Summary, Description and etc,</p> <table border="1" data-bbox="537 953 1344 1163"> <tr> <td>Change Type</td> <td>Bug Fix</td> </tr> <tr> <td>Change Summary</td> <td>Oracle invoices cannot approve</td> </tr> <tr> <td>Priority</td> <td>High</td> </tr> <tr> <td>Description</td> <td>Oracle invoices cannot approve from system</td> </tr> <tr> <td>Application</td> <td>Oracle EBS</td> </tr> </table> <p>Upload supporting files if required</p> <p><u>Step 4</u> Submit the request</p>			Change Type	Bug Fix	Change Summary	Oracle invoices cannot approve	Priority	High	Description	Oracle invoices cannot approve from system	Application	Oracle EBS
Change Type	Bug Fix												
Change Summary	Oracle invoices cannot approve												
Priority	High												
Description	Oracle invoices cannot approve from system												
Application	Oracle EBS												
Expected Result:	Reflect the initial request on the "New Request" queue. Automatically generate request Id, Requester ID, request date												
Actual result:	<table border="1" data-bbox="537 1434 1390 1575"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>Requested By</td> <td>1212</td> </tr> <tr> <td>Requested Date</td> <td>28/01/2019</td> </tr> <tr> <td>Status</td> <td>New Request</td> </tr> </table>			Change ID	2	Requested By	1212	Requested Date	28/01/2019	Status	New Request		
Change ID	2												
Requested By	1212												
Requested Date	28/01/2019												
Status	New Request												

Status (Pass/Fail):	Pass
Comments:	

Test case ID:	0002		
Test priority (Low/Medium/High):	High	Test Designed By:	Thilina Kelaniyage
Module Name:	Change Management	Test Designed Date:	28/01/2019
Test Executed By:		Test Execution Date:	28/01/2019

Test Title:	Change Management										
Test Summary/Description:	Authorize change										
Test Steps:	<p><u>Step 1:</u> Log into the system via Authorizer login (same department) Login ID: 1213</p> <p><u>Step 2:</u> Navigate to the "Pending Requests" from the side menu</p> <p><u>Step 3:</u> From pending requests authorizer will know request pertaining to his/her department. View for additional details</p> <p><u>Step 4:</u> After checking for validity authorized the request</p>										
Expected Result:	Reflect the initial request on Authorized queue. Automatically generate Authorizer ID, Authorized date										
Actual result:	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>Authorized By</td> <td>1213</td> </tr> <tr> <td>Authorized Date</td> <td>28/01/2019</td> </tr> <tr> <td>Status</td> <td>Authorized</td> </tr> </table>			Change ID	2	Authorized By	1213	Authorized Date	28/01/2019	Status	Authorized
Change ID	2										
Authorized By	1213										
Authorized Date	28/01/2019										
Status	Authorized										
Control Test:	<p><u>Step 1:</u> Log into the system via Authorizer login from another department Login ID: 1213</p> <p><u>Step 2:</u> Navigate to the "Pending Requests" from the side menu</p>										
Control Test Expected Result:	Initial request should not populate under control login										
Control Test Actual result:	The initial request did not populate under control login										

Status (Pass/Fail):	Pass
Comments:	

Test case ID:	0003		
Test priority (Low/Medium/High):	High	Test Designed By:	Thilina Kelaniyage
Module Name:	Change Management	Test Designed Date:	28/01/2019
Test Executed By:		Test Execution Date:	28/01/2019

Test Title:	Change Management										
Test Summary/Description:	Conduct business analysis (BA) for the request										
Test Steps:	<p><u>Step 1:</u> Log into the system via BA login</p> <p><u>Step 2:</u> Navigate to the "Pending Analysis" from the side menu</p> <p><u>Step 3:</u> Check the request details. Perform analysis and upload relevant supporting documents Make additional comments Assign request to a developing team member</p> <p><u>Step 4:</u> Confirm BA Analysis</p>										
Expected Result:	Reflect the BA confirmed request on "BA Confirmed" queue. Automatically generate BA Id, BA Confirmed date Assign request to the development team lead										
Actual result:	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>BA Confirmed By</td> <td>1214</td> </tr> <tr> <td>Confirmed Date</td> <td>28/01/2019</td> </tr> <tr> <td>Status</td> <td>BA Confirmed</td> </tr> </table>			Change ID	2	BA Confirmed By	1214	Confirmed Date	28/01/2019	Status	BA Confirmed
Change ID	2										
BA Confirmed By	1214										
Confirmed Date	28/01/2019										
Status	BA Confirmed										
Alternative Test Steps:	<p><u>Step 1:</u> Log into the system via BA login</p> <p><u>Step 2:</u> Navigate to the "Pending Analysis" from the side menu</p> <p><u>Step 3:</u> Check the request details. Reject the request</p>										
Expected Result For Alternative Test	Reflect the request on "Rejected" queue. Automatically update rejected date and rejecter ID										

Actual result for Alternative test:	Reflect the request on “Rejected” queue. Automatically update rejected date and rejecter ID
--	--

Status (Pass/Fail):	Pass
Comments:	

Test case ID:	0004	Test Designed By:	Thilina Kelaniyage
Test priority (Low/Medium/High):	High	Test Designed Date:	28/01/2019
Module Name:	Change Management	Test Executed By:	
		Test Execution Date:	30/01/2019

Test Title:	Change Management								
Test Summary/Description:	Confirm the development by Developer								
Test Steps:	<p><u>Step 1:</u> Log into the system via Developer login</p> <p><u>Step 2:</u> Navigate to the "Pending Developments" from the side menu</p> <p><u>Step 3:</u> Check the request details., BA comments and other supporting documents. After making the development add related comments such as test instance location, modules got changed, Users Involved and etc.</p> <p><u>Step 4</u> Confirm Development.</p>								
Expected Result:	Reflect the Development confirmed request on “Development Complete” queue. Automatically generate Developer Id, Developer Confirmed date.								
Actual result:	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>Developed By</td> <td>1215</td> </tr> <tr> <td>Confirmed Date</td> <td>30/01/2019</td> </tr> <tr> <td>Status</td> <td>Development Complete</td> </tr> </table>	Change ID	2	Developed By	1215	Confirmed Date	30/01/2019	Status	Development Complete
Change ID	2								
Developed By	1215								
Confirmed Date	30/01/2019								
Status	Development Complete								

<u>Control Test Steps:</u>	<p><u>Step 1:</u> Log into the system via developer login. (From another developer login)</p> <p><u>Step 2:</u> Navigate to the “Pending Developments” from side menu</p>
<u>Expected Result For Control Test</u>	<ul style="list-style-type: none"> • Previous change request should not visible to developer
<u>Actual result for Control test:</u>	Previous change request did not visible to developer

<u>Status (Pass/Fail):</u>	Pass
<u>Comments:</u>	

<u>Test case ID:</u>	0005	<u>Test Designed By:</u>	Thilina Kelaniyage
<u>Test priority (Low/Medium/High):</u>	High	<u>Test Designed Date:</u>	28/01/2019
<u>Module Name:</u>	Change Management	<u>Test Execution Date:</u>	30/01/2019
<u>Test Executed By:</u>			

<u>Test Title:</u>	Change Management
<u>Test Summary/Description:</u>	Confirm the Quality Assurance Testing
<u>Test Steps:</u>	<p><u>Step 1:</u> Log into the system via QA login</p> <p><u>Step 2:</u> Navigate to the “Pending Quality Assurance” from side menu</p> <p><u>Step 3:</u></p>

	<p>Check the request details., BA comments, Developer comments and other supporting documents. Make sure requirements are satisfied by the development <u>Step 4</u> Confirm Quality check.</p>								
<u>Expected Result:</u>	<p>Reflect the Development confirmed request on “QA Confirmed” queue. Automatically generate QA Id, QA Confirmed date.</p>								
<u>Actual result:</u>	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>QA Confirmed By</td> <td>1216</td> </tr> <tr> <td>Confirmed Date</td> <td>30/01/2019</td> </tr> <tr> <td>Status</td> <td>QA Confirmed</td> </tr> </table>	Change ID	2	QA Confirmed By	1216	Confirmed Date	30/01/2019	Status	QA Confirmed
Change ID	2								
QA Confirmed By	1216								
Confirmed Date	30/01/2019								
Status	QA Confirmed								
<u>Alternative Test Steps:</u>	<p><u>Step 1:</u> Log into the system via QA login <u>Step 2:</u> Navigate to the “Pending Quality Assurance” from side menu <u>Step 3:</u> Check the request details., BA comments, Developer comments and other supporting documents. Make sure requirements are satisfied by the development <u>Step 4</u> Reject Quality check and reassign task to Developer or business analysis.</p>								
<u>Expected Result For Alternative Test</u>	<ul style="list-style-type: none"> • QA Reject flag should be set and visible under QA rejection queue of Business Analysis. • Rejection ID and date should automatically update 								
<u>Actual result for Alternative test:</u>	<ul style="list-style-type: none"> • QA Reject flag should be set and visible under QA rejection queue of Business Analysis/Developer. • Rejection ID and date should automatically update 								

<u>Status (Pass/Fail):</u>	Pass
<u>Comments:</u>	

<u>Test case ID:</u>	0006	<u>Test Designed By:</u>	Thilina Kelaniyage
<u>Test priority (Low/Medium/High):</u>	High	<u>Test Designed Date:</u>	28/01/2019
<u>Module Name:</u>	Change Management		

Test Executed By:		Test Execution Date:	30/01/2019
--------------------------	--	-----------------------------	------------

Test Title:	Change Management								
Test Summary/Description:	Confirm the User Acceptance Testing								
Test Steps:	<p><u>Step 1:</u> Log into the system via User login</p> <p><u>Step 2:</u> Navigate to the “Pending UAT” from side menu</p> <p><u>Step 3:</u> Check whether solution satisfy the requirements</p> <p><u>Step 4</u> Confirm UAT.</p>								
Expected Result:	Reflect the Development confirmed request on “UAT Confirmed” queue. Automatically generate UAT Id, UAT Confirmed date.								
Actual result:	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>UAT Confirmed By</td> <td>1212</td> </tr> <tr> <td>Confirmed Date</td> <td>30/01/2019</td> </tr> <tr> <td>Status</td> <td>UAT Confirmed</td> </tr> </table>	Change ID	2	UAT Confirmed By	1212	Confirmed Date	30/01/2019	Status	UAT Confirmed
Change ID	2								
UAT Confirmed By	1212								
Confirmed Date	30/01/2019								
Status	UAT Confirmed								
Alternative Test Steps:	<p><u>Step 1:</u> Log into the system via User login</p> <p><u>Step 2:</u> Navigate to the “Pending UAT” from side menu</p> <p><u>Step 3:</u> Check whether solution satisfy the requirements. If not,</p> <p><u>Step 4</u> Reject UAT Confirmation.</p>								
Expected Result For Alternative Test	<ul style="list-style-type: none"> • UAT Reject flag should be set and visible under the UAT rejection queue of Business Analysis. • Rejection ID and date should automatically update 								
Actual result for Alternative test:	<ul style="list-style-type: none"> • UAT Reject flag did set and visible under the UAT rejection queue of Business Analysis. • Rejection ID and date should automatically update 								

Status (Pass/Fail):	Pass
Comments:	

Test case ID:	0007		
Test priority (Low/Medium/High):	High	Test Designed By:	Thilina Kelaniyage
Module Name:	Change Management	Test Designed Date:	28/01/2019
Test Executed By:		Test Execution Date:	30/01/2019

Test Title:	Change Management										
Test Summary/Description:	Approval for live deployment										
Test Steps:	<p><u>Step 1:</u> Log into the system via Approver login</p> <p><u>Step 2:</u> Navigate to the “Pending Approval” from side menu</p> <p><u>Step 3:</u> Check whether solution obtain adequate clearance from QA testing and UAT</p> <p><u>Step 4:</u> Confirm Approval.</p>										
Expected Result:	Reflect the approved request on “Approved for Migration” queue. Automatically generate Approver Id, approved date.										
Actual result:	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>Approved By</td> <td>1217</td> </tr> <tr> <td>Approved Date</td> <td>30/01/2018</td> </tr> <tr> <td>Status</td> <td>Approved for Migration</td> </tr> </table>			Change ID	2	Approved By	1217	Approved Date	30/01/2018	Status	Approved for Migration
Change ID	2										
Approved By	1217										
Approved Date	30/01/2018										
Status	Approved for Migration										
Alternative Test Steps:	<p><u>Step 1:</u> Log into the system via Approver login</p> <p><u>Step 2:</u> Navigate to the “Pending Approval” from side menu</p> <p><u>Step 3:</u> Check whether solution obtain adequate clearance from QA testing and UAT</p> <p><u>Step 4:</u> Reject Migration.</p>										
Expected Result For Alternative Test	<ul style="list-style-type: none"> Request should be listed under rejected queue. Rejection ID and date should automatically update 										

<u>Actual result for Alternative test:</u>	<ul style="list-style-type: none"> Request listed under rejected queue. Rejection ID and date automatically updated
---	---

<u>Status (Pass/Fail):</u>	Pass
<u>Comments:</u>	

<u>Test case ID:</u>	0008	<u>Test Designed By:</u>	Thilina Kelaniyage
<u>Test priority (Low/Medium/High):</u>	High	<u>Test Designed Date:</u>	28/01/2019
<u>Module Name:</u>	Change Management	<u>Test Execution Date:</u>	28/01/2019
<u>Test Executed By:</u>			

<u>Test Title:</u>	Change Management								
<u>Test Summary/Description:</u>	Change deployment								
<u>Test Steps:</u>	<p><u>Step 1:</u> Log into the system via deployer login</p> <p><u>Step 2:</u> Navigate to the “Pending deployments” from side menu</p> <p><u>Step 3:</u> Check necessary comments</p> <p><u>Step 4:</u> Confirm Deployment.</p>								
<u>Expected Result:</u>	Reflect the approved request on “Completed” queue. Automatically generate Deployer Id, Deploy date.								
<u>Actual result:</u>	<table border="1"> <tr> <td>Change ID</td> <td>2</td> </tr> <tr> <td>Deployed By</td> <td>1218</td> </tr> <tr> <td>Deployed Date</td> <td>30/01/2019</td> </tr> <tr> <td>Status</td> <td>Completed</td> </tr> </table>	Change ID	2	Deployed By	1218	Deployed Date	30/01/2019	Status	Completed
Change ID	2								
Deployed By	1218								
Deployed Date	30/01/2019								
Status	Completed								

<u>Status (Pass/Fail):</u>	Pass
<u>Comments:</u>	

<u>Test case ID:</u>	0009	<u>Test Designed By:</u>	Thilina Kelaniyage
<u>Test priority (Low/Medium/High):</u>	Medium	<u>Test Designed Date:</u>	28/01/2019
<u>Module Name:</u>	User Management	<u>Test Execution Date:</u>	01/02/2019
<u>Test Executed By:</u>			

<u>Test Title:</u>	User Management												
<u>Test Summary/Description:</u>	Request new user creation request												
<u>Test Steps:</u>	<p><u>Step 1:</u> Log into the system via user login Login ID:1212</p> <p><u>Step 2:</u> Navigate to the "New Request" from side menu</p> <p><u>Step 3:</u> Feed request details which include change type, Summary, Description and etc,</p> <table border="1" data-bbox="565 1360 1372 1606"> <tr> <td>Name</td> <td>Lakmal Perera</td> </tr> <tr> <td>Designation</td> <td>Manager</td> </tr> <tr> <td>Department</td> <td>Finance</td> </tr> <tr> <td>Emp No.</td> <td>3231</td> </tr> <tr> <td>Applications</td> <td>Oracle EBS</td> </tr> <tr> <td>Access Level</td> <td>Order Management View, Journal Entry Approval</td> </tr> </table> <p><u>Step 4:</u> Submit the request</p>	Name	Lakmal Perera	Designation	Manager	Department	Finance	Emp No.	3231	Applications	Oracle EBS	Access Level	Order Management View, Journal Entry Approval
Name	Lakmal Perera												
Designation	Manager												
Department	Finance												
Emp No.	3231												
Applications	Oracle EBS												
Access Level	Order Management View, Journal Entry Approval												
<u>Expected Result:</u>	Reflect the initial request on the "New Request" queue. Automatically generate request Id, Requester ID, request date												
<u>Actual result:</u>	<table border="1" data-bbox="565 1810 1416 1877"> <tr> <td>Request ID</td> <td>1001</td> </tr> <tr> <td>Requested By</td> <td>3231</td> </tr> </table>	Request ID	1001	Requested By	3231								
Request ID	1001												
Requested By	3231												

	Requested Date	
	Status	New Request

Status (Pass/Fail):	Pass
Comments:	

Test case ID:	0010		
Test priority (Low/Medium/High):	Medium	Test Designed By:	Thilina Kelaniyage
Module Name:	User Management	Test Designed Date:	28/01/2019
Test Executed By:		Test Execution Date:	01/02/2019

Test Title:	User Management								
Test Summary/Description:	Authorize User Creation Request								
Test Steps:	<p><u>Step 1:</u> Log into the system via Authorizer login (same department) Login ID: 1213</p> <p><u>Step 2:</u> Navigate to the “Pending Requests” from side menu</p> <p><u>Step 3:</u> From pending requests authorizer will know request pertaining to his/her department. View for additional details</p> <p><u>Step 4:</u> After checking for validity authorized the user creation request</p>								
Expected Result:	Reflect the initial request on Authorized queue. Automatically generate Authorizer ID, Authorized date								
Actual result:	<table border="1"> <tr> <td>Request ID</td> <td>1001</td> </tr> <tr> <td>Authorized By</td> <td>1213</td> </tr> <tr> <td>Authorization Date</td> <td>01/02/2019</td> </tr> <tr> <td>Status</td> <td>Authorized</td> </tr> </table>	Request ID	1001	Authorized By	1213	Authorization Date	01/02/2019	Status	Authorized
Request ID	1001								
Authorized By	1213								
Authorization Date	01/02/2019								
Status	Authorized								

--	--

Status (Pass/Fail):	Pass
Comments:	

Test case ID:	0011	Test Designed By:	Thilina Kelaniyage
Test priority (Low/Medium/High):	Medium	Test Designed Date:	28/01/2019
Module Name:	User Management	Test Execution Date:	01/02/2019
Test Executed By:			

Test Title:	User Management								
Test Summary/Description:	Complete User Creation								
Test Steps:	<p><u>Step 1:</u> Log into the system via User Creator login</p> <p><u>Step 2:</u> Navigate to the "Pending User Creations" from the side menu</p> <p><u>Step 3:</u> Review requested Access Create a user on the application</p> <p><u>Step 4:</u> Confirm Creation.</p>								
Expected Result:	Reflect the approved request on "Completed" queue. Automatically generate Creator Id, creation date.								
Actual result:	<table border="1"> <tr> <td>Request ID</td> <td>1001</td> </tr> <tr> <td>Created By</td> <td>3215</td> </tr> <tr> <td>Creation Date</td> <td>01/02/2019</td> </tr> <tr> <td>Status</td> <td>Completed</td> </tr> </table>	Request ID	1001	Created By	3215	Creation Date	01/02/2019	Status	Completed
Request ID	1001								
Created By	3215								
Creation Date	01/02/2019								
Status	Completed								