Travel Management System for NGO

Fatema Tuj Jahra
Student Id: 012131023

A Project

in

The Department

of

Computer Science and Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Master of Science in Computer Science and Engineering

United International University

Dhaka, Bangladesh

February 2018

Approval Certificate

This project titled "**A Project on Travel Management System for NGO**" submitted by **Fatema Tuj Jahra**, Student ID: **012131023**, has been accepted as Satisfactory in fulfillment of the requirement for the degree of Master of Science in Computer Science and Engineering on 28-02-2018.

Board of Examiners

1. _____ Supervisor
Suman Ahmmed
Assistant Professor, Director CDIP

2. _____ Examiner
Mohammad Mamun Elahi
Assistant Professor

3. _____ Ex-Officio
Mohammad Nurul Huda
Professor & Coordinator - MSCSE

Declaration

This is to certify that the work entitled "**Travel Management System for NGO** " is the outcome of the research carried out by me under the supervision of Suman Ahmmed, Assistant Professor & Director CDIP

_____

Fatema Tuj Jahra
ID: 012131023
Dept. of CSE

In my capacity as supervisor of the candidate's project, I certify that the above statements are true to the best of my knowledge.

_____

Suman Ahmmed
Assistant Professor & Director CDIP

Abstract

Travel Management System (TMS) is a web application that covers all the areas required for a VISA & Protocol Department of a Company. Its main modules are Application User Management, VISA Services, Approval Management, Ticket management and reports. TMS is a web based application. The main purpose of this system is to provide a convenient way for an employee to visit a country and keep updated records of every visit. The objective of this project is to develop a system that automates the processes and activities of VISA & Protocol Department (VPD). In this project, I will make an easier task of taking approvals and keep records of all employee who has gone abroad. In the present system, an employee has to approach VPD to know the initial steps and all requirements of visiting a specific country. VPD works like a Travel Agency and assist them to get VISA. When Management asks for a specific Country visit details report then this requires a lot of time and effort to prepare. The project TMS is developed to replace the currently existing manual system, which helps in keeping records of the employee details of destination, VISA duration, Ticket Fund details. This application is developed to provide best travelling services to the employees and management. This system also provide a better way to connect with various events. It also gives tours related information like which places are tourist attractions, cities, provinces and culture. Employees can also get the Map temperature, weather and basic information. This system also keeps a history of visited places of its users.

Acknowledgement

First, all praises and gratitude to the almighty Allah for giving me the patients, strength and audacity to complete this project. I would like to convey my whole-hearted gratitude to my supervisor Mr. Suman Ahmmed Sir for his valuable guidance, motivation and friendly approach. I am thankful to him for being my Supervisor. I would like to say thanks to my Coordinator Nurul Huda Sir and the Staffs of United International University. My thanks goes to the NGO staffs- without their assistance it was not possible for me to know their manual system. I also like to express my deepest gratitude and love to my respective family especially my parents, my husband. Last but not the least; I am very much grateful to United International University authority to give me this opportunity.

Table of Contents

Chapter 1

1 Introduction

There are many Travel Management Applications that manages tours, travels and booking of hotel and transport. But no customizable application is available for tracking the approval process and tour details. It's not that much expensive to develop a system like this. Also this kind of system can be used in Tours, Travels or VISA support department of a company.

It takes long time for Decision makers, Administrative Staffs and Managers to prepare a wide variety of reports on demand and in more detail in a manual system. VISA & protocol Department is not able to respond to queries quickly and precisely. They spend all the time on making larger reports on tedious tasks. Year or Project Wise Visit Summary, Monthly Departure History, Tracking No wise details reports extraction from manual system is very time consuming. So an automated system is needed here.

Objectives of the project are-

- Automated Approval taking system
- Automated mail generated in every required step
- Maintain a life cycle of every ticket for each employee
- Less time waste in manual report preparation
- Construct an appropriate shell for this system.

Chapter 2

2 Background and Literature Review

In a year, more than three thousand employee of this NGO go abroad for official purpose. They take approval from multiple authorized person and do lots of paper work. After VISA apply, employee become detached with VISA & Protocol Department (VPD). In most cases, VPD remains uninformed about individual employee's VISA detail status. Also VPD does not get to know whether the employee returned or not. When management wants to see report like how many staff went to Uganda this year or who have Afghanistan's VISA right now- VPD can't answer properly. To prepare this kind of report VPD had to do lots of tedious and time consuming work. The manual system requires the employees to contact the VPD staffs, go to their office and wait for their turn just to get information about travel approval procedures and required documents name.

There is risk of not following the procedure which was mentioned in that respective countries VISA requirement. This can cause a huge problem. The Employee will not get VISA. If a management asks for total cost in this year in buying Tickets then VPD can't answer. If management wants to know the total fund of Other Source apart from BRAC in buying tickets- VPD takes few days to prepare the report!

There were some limitations in this project. Initially it was difficult to obtain exact process information from the VPD which would be needed for the project.

Chapter 3

3 System Concept, Design & Implementation

To accomplish the objective, an efficient process was followed. The study process begins with the identification of the investigation topic (Travel Management System) where studies was carried out to obtain enough information on the topic. In addition to this, frequent interviews with BRAC VISA & Protocol Department employees were conducted to gather more information on their daily operations and on the methods and features required in the travel management system web application. After gathering feedback from employees, an analysis was led to draw out important information for the development of the portal. In this section the requirement analysis for Travel Management system is described. The abstract system block diagram is included. And a detail selection criterion of each component is discussed.

3.1 Requirement Analysis and Features Outline

Primary role of this app is to track travel approval and make country specific visa related information available for BRAC staffs for smooth and hassle free operation. User will use this application for applying for any travel approval and through approval workflow his/her supervisor will approve the travel. There will be information section in this app where all the country specific information related to VISA will be available. Along with that the list of steps to be followed to get the service from BRAC Travel & Protocol Management Department will be there. The information will be linked with external information sources like VFS global, IVAC etc. User will request the supervisor for travel approval and once the travel is approved by the supervisor, user can request for the VISA services to Travel & Protocol Department. User can upload required documents in system during placing the requisition. Once the VISA is accepted/rejected user/ representative from Travel & Protocol Management Department will update the status in the system. Unless the visa decision is updated in system, no ticket can be issues. Reports Required: Countrywide/Region wise / Program wise travel reports. Note: Ticketing is being considered as a future scope

Integration: HR system, ticket system

Features are:

- User Management System with SSO
- Country Wise Information Setup: Requirement, Steps to be followed to get service from VPD, Link With VFS Global and IVAC
- Project & Country Setup
- Department wise Director Assign.
- Approval Request To Director
- Approve/Reject/Call For Discuss by Director
- Country Wise VISA Information
- Steps to follow & FAQ
- Employee can Cancel Approval Request before director's Approval
- VISA Request to VISA & Protocol Department
- VISA Status Update
- Un processed ticket close notification by mail
- Various Reports from each level.
- All Issued Ticket details

Stakeholders:

- VPD Admin

- Director
- Staff
- System Admin

3.2 Software Design

Proposed Web Application Architecture

The Web relies sturdily in the client-server model. It uses markup languages such as XML, HTML to transfer and signpost data. Under it there are many programming and scripting languages that can vigorously process, generate and modify data, or give an UI. The

development of Web applications can be placed under the protection of software engineering in this approach but need to be prolonged. Web applications are multidisciplinary (software engineering, effective interface design, database demonstrating techniques and network computing). They are built in a interminable changing environment where requirements are unsteady and the user community is broader than before. Web applications handle information from numerous sources (text, video, audio, graphics) dealing with shaping, processing, loading and presenting this material.

In this kind of model, the outermost level contracts with the presentation of the content and interaction with the user. It can be called presentation, UI, view. The application shows to the user what is needed to be seen and gives the tools for interaction in this layer. The particular kind of interaction depends on the application; one can create a web app that only shows information to the user without any kind of interaction, not even hyperlinks to be clicked, in such a case it does not require an advanced architecture. In maximum cases the user will produce some input, send it for processing and then receive a feedback that can be the ultimate result or a step for further actions.

This layer interconnects with the business logic layer beneath it, pass the information from the user and direct it, then give back any response it produces, not leaving any conclusions of the application's reasoning to be firm by the UI. This information passing is usually done through forms, alike a user log-in in a system by giving username and password, but there are additional ways. AJAX is an asynchronous way to pass information to the server and get replies. The cited a-synchronicity arises from the fact that in a form the content demands to be passed and then the response will come after a page refresh, but with AJAX the requested information, that is the result of the user's deed will come in the actual page. It saves time and gives user the impression that the application is really interacting with him/her.

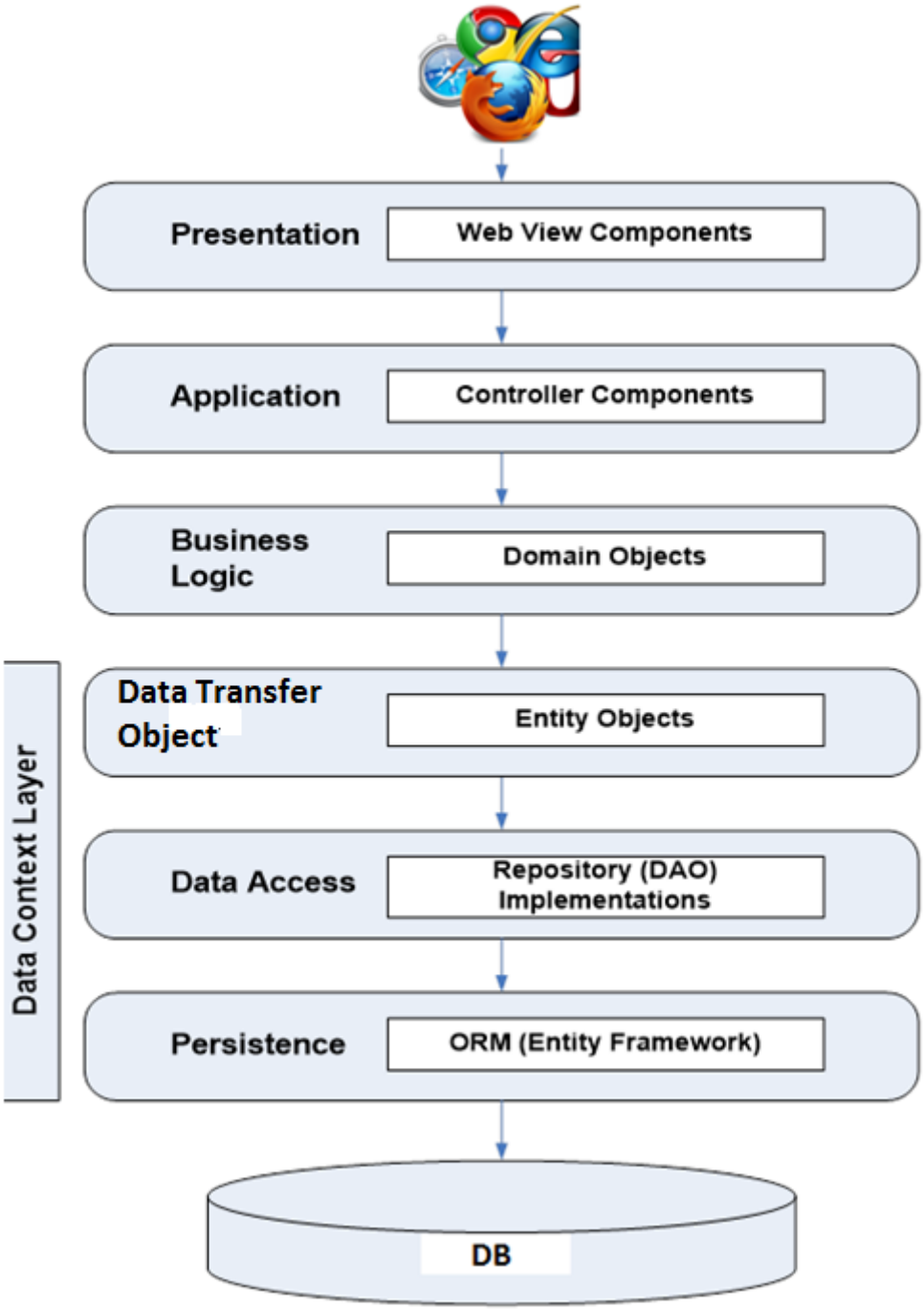

Figure 3.2.1 Web Application Architecture

The fundamental layer of this model deals with the logic of the program. It receives data from the upper level and converts it, using in the inner application logics. It also recovers data from the inmost data level and uses it to the logics. And, by integrating these two procedures, it can do modifications in both levels also.

The Business Logic Layer (BLL) holds the determinant part of the application logic. It embraces-

- Performing all prerequisite calculations and validations
- Managing workflow

To provide application services in a consistent way managing all data access for the presentation layer.

The Business Logic Layer is generally implemented inside an Application server (like Microsoft Transaction Server, Oracle Application Server). The Application server generally automates a number of services such as transactions, security, messaging, connection pooling, persistence and name services.

The user that desires to see the weather in a specific place, when the information is given by the user the application recovers it and process. If the user wants to check the weather forecast for two days. After receiving request from UI the application the data is directed to the server. A script catches it and then make the calls for the subordinate level to get the needed data. When a reply arrives, being it the desired information or a failed request, it is dealt and then prepared to be sent once more to the upper level.

The tools used in this level are usually server-side scripts Groovy. node.js is a solution that uses server-side Javascript. These technologies, following the information serving that comes from the upper level, can do any computational handling that is required.

The Data Layer is the deepest level in the layered architecture, it deals with data retrieval from its sources. It is an abstraction to get the basic data, which can be in a wide diversity of forms. Once again, it plays a huge role on the reusability and altercation of technologies: if one data source is changed to another, but the proper data is still the same, a good layered design can help by providing the same data to the upper level with the same interfaces, changing only its internal logic.

The requirement by the user for the next day's forecast will come to this level as a request for the forecasts that it might have. Then a search will be made in the data using the given parameters, and then the will be sent again to the upper level.

The technologies used in this layer are database management systems like PostgreSQL for relational database. For the management systems generally an API will be used for building queries and retrieving data, and for the plain text, a script will do the needed operations. Inside it there can be any level of complexity anticipated by the application designer, so there can be integrity checks, stored procedures, and virtually anything needed to maintain the data in the desired state.

Inside the Data Layer, as it was outlined, many different technologies can be used. Most of the web applications currently use relational databases.
For communication with other layers. In this multiple layer application, a data access layer and a business layer, determine a strategy for communication between presentation layer and other layers. If I have a separate business layer, my presentation layer will

communicate with the business layer. If I don't have a business layer, my presentation layer will communicate directly with the data access layer.

I have used following techniques to access other layers-
Direct method calls. If the layer with which I am communicating is on the same physical tier as the presentation layer, I made direct method calls.

Web services. I use a Web service interface if I want to share the data access or business logic with other applications, or when I get data from other applications. If the business layer or data access layer are deployed on a separate tier from presentation layer, or if decoupling is important. Consider Service-oriented architecture. It is an evolution of distributed computing based on the request/reply scheme model for synchronous and asynchronous applications. For example, a service can be implemented either in .Net or J2EE, and the application using the service can be on a different platform or language.

Hibernate Architecture:

ORM is a piece of software/product for the demonstration and transformation of data between the database and the object-oriented programming language. **Hibernate** is such an ORM solution and it is an open-source project.

Though **Hibernate Framework** is not the only persistence solution, it has become very familiar over the recent past because of its enormous variety of features when compared with its contenders. It takes considerable of the database related boiler-plate code from the developers, thereby asking the designers to ponder on the core business logic of the application and not with the error-prone SQL syntax.

Most of the **Hibernate** features look very analogous that are found in *Java Persistence API (JPA)* specification. Also **Hibernate** offers some add-on functionalities that are not bring up in the *JPA* specification. ***Hibernate*** doesn't replace *JDBC*. ***Hibernate*** is sitting on top of JDBC to connect to the database. Internally **Hibernate** is using these *JDBC* requests to connect to the database. ***Hibernate*** can well mix with all kinds any of *J2EE/ J2SE* application and with any sort of frameworks (like *Grails, Spring* etc).

The architecture of Hibernate, consists of the following:

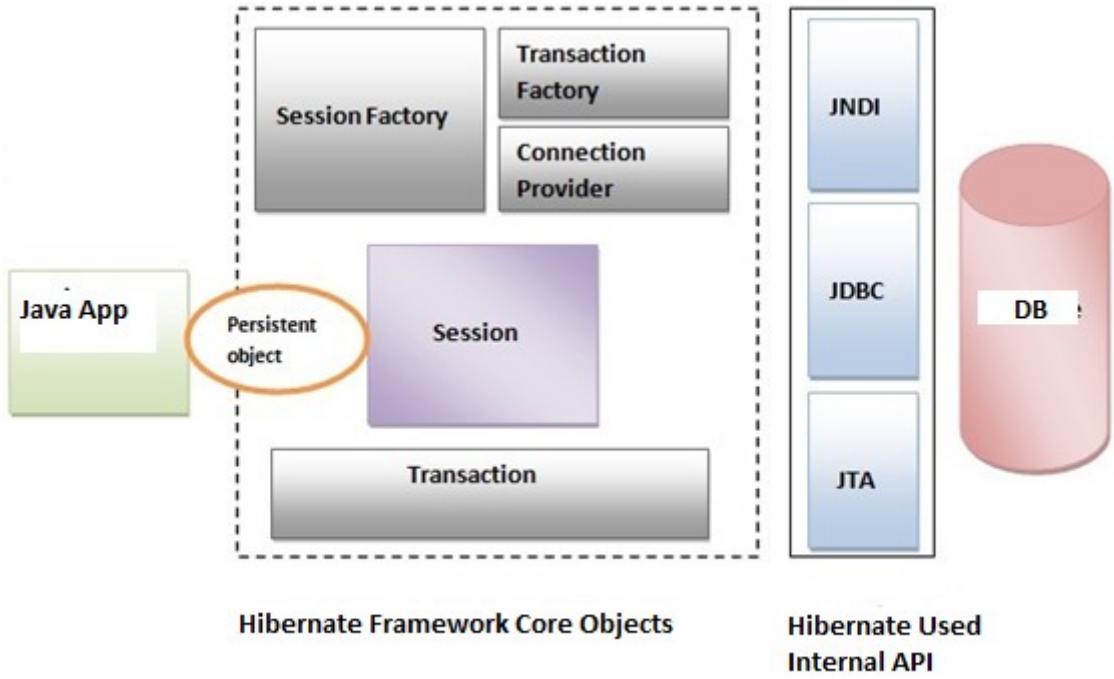

Figure 3.1.1 Hibernate Architecture

Use Case Diagram:

In software engineering, use case is a list of actions or event steps naturally defining the interactions between a role (known in the UML as an actor) and a system to achieve a goal. The actor can be a human or other external system.

The relationship between the Actors and the Use Cases of Travel Management System.

VISA & protocol Department Admin Entity: Use Cases of VISA & protocol Department Admin are manage Director Approval, provide enquiry details, manage VISA Assistance, manage Tracking No life cycle, Issue and manage Ticket information, do user management and view all reports.

System Admin Entity: Use Cases of System Admin are manage Director Approval, manage VISA Assistance, manage Tracking No life cycle, Issue and manage Ticket information, do user management and view all reports.

.

Employee/Staff Entity: Use Cases of Employee are view some reports, make enquiry, apply for director's approval, cancel application for director's approval, Apply for VISA & Protocol Department's assistance, Update VISA processing status.

Director Entity: Use Cases of Director are view some reports, Approve/Reject or Call for Discuss.

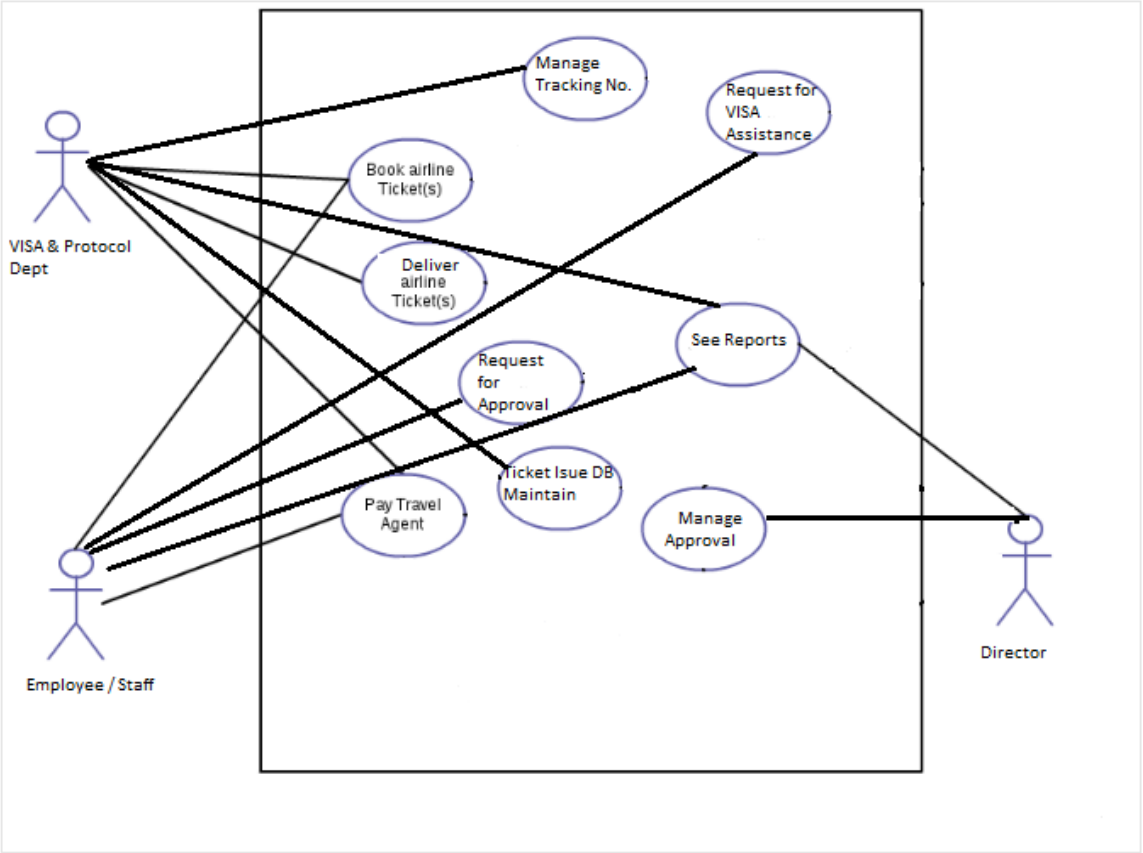

Figure 3.2.2 Use Case Diagram

Activity Diagram:

VISA & protocol Department Admin can manage Tracking No life cycle, manage Director Approval, provide enquiry details, manage VISA Assistance, Issue and manage Ticket information, do user management and view all reports.

System Admin can configure settings, modify data, approve, reject request, close ticket, Assign Roles, manage VISA Assistance, manage Director Approval, manage Tracking No life cycle, Issue and manage Ticket information, do user management and view all reports.
.

Employee/Staff can make enquiry, apply for director's approval, view some reports, cancel application for director's approval, Apply for VISA & Protocol Department's assistance, Update VISA processing status.

Director can Approve/Reject or Call for Discuss and view some reports. There is a Decision Node here.

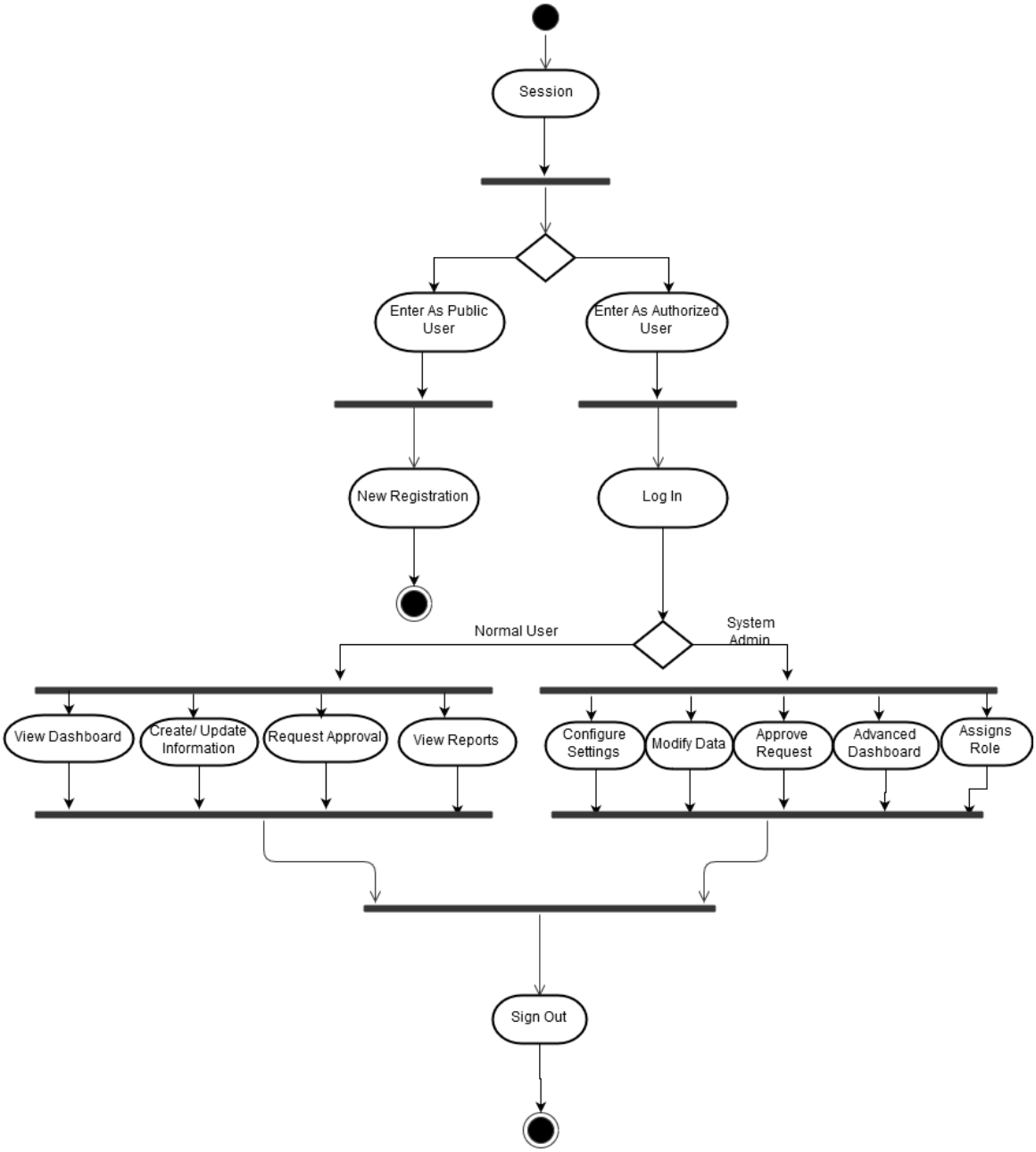

Figure 3.1.3 Activity Diagram

Process Flow: Below is the interdepartmental process flow.

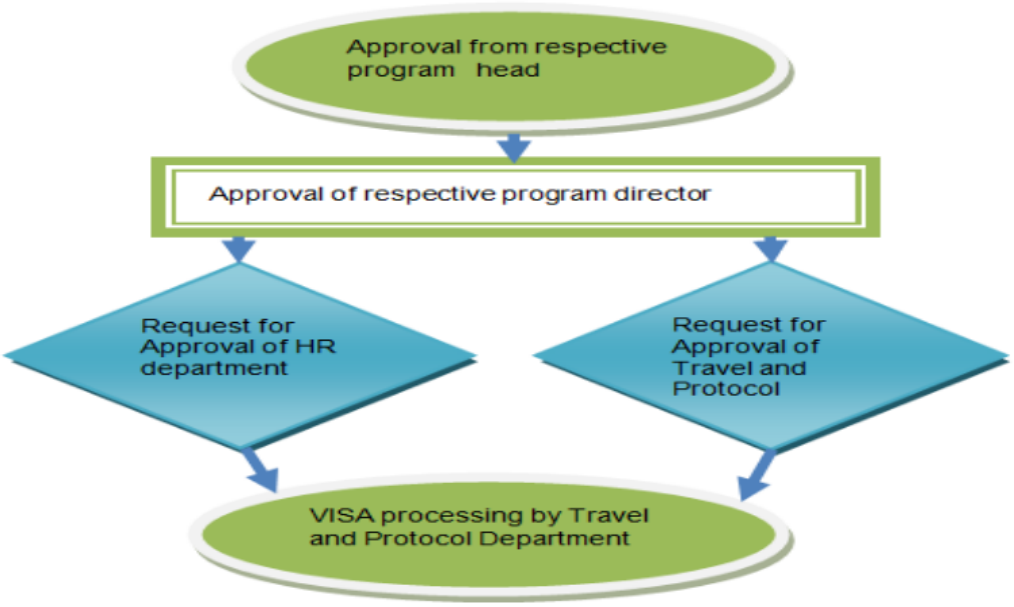

Figure: 3.1.4 Process Flow of TMS

Employee will take manual approval from respective program head. Then with program head's approval employee will apply to his/her program's director for approval. Director can Approve/Reject or Call for Discussion. After getting director's approval, an auto generated request will go to HR department for formalities processing. And in parallel, employee will apply again to VISA & Protocol Department for approval. After that, VPD will handle rest of the procedures.

SSO- Single Sign -on is a user authentication service and session that allows a user to use a set of credential for login to access multiple application. It is helpful for user logging activities as well as observing accounts. So I will use SSO here.

3.3 Technology Used

Language & Platform:

- Groovy on Grails
- JavaScript

Tools/IDE Summary:

- IntelliJ IDEA 2016.1.3

Report Tools Summary:

- Jasper Report 5.6

Database Technology Summary:

- PostgreSQL 9.5

Web Server Summary:

- Apache Tomcat 8.0

3.4 Hardware Requirement

RAM: 4GB or above
Processor: Core i3 or more
OS: Windows Server 2003 and above

3.5 Software Development

This is a way to introduce an automation process in Travel Management System. Through this application an employee can raise travel approval request, VISA request, Ticket request and Authorized person like Director can approve the employee approval request. And Authorized person can see the output of report by filtering specific item.

User Management

In User Management section, all of the application user related setting are available

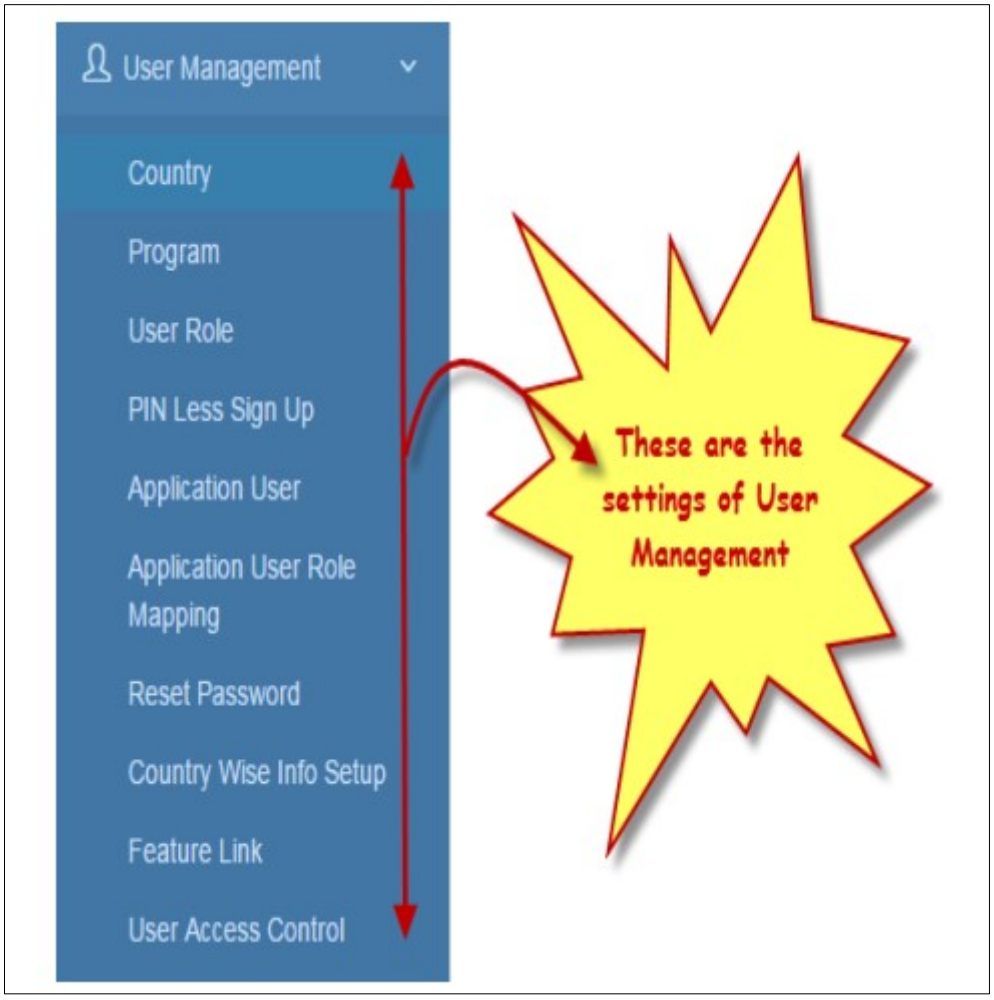

Figure 3.5.1

Country

Here, user can setup the country name. Just type Country Name in the following field & click on Save. Then Country Name will add in the country List.

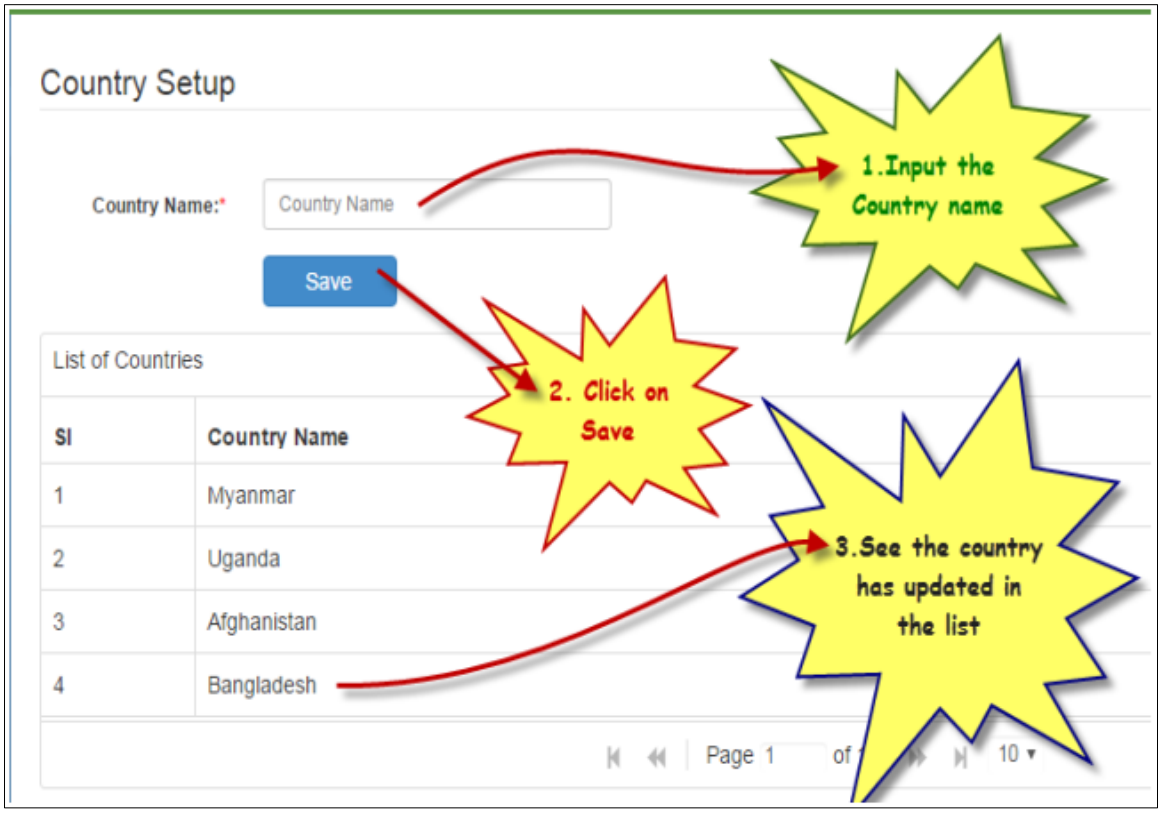

Figure 3.5.2

If user want to update the country then select any country from list & see the above Save button already changed to update & selected country name showing in the Country Name field. Now you can update the country Name & just click on Update for save. And you can click on Clear for clear the fields.

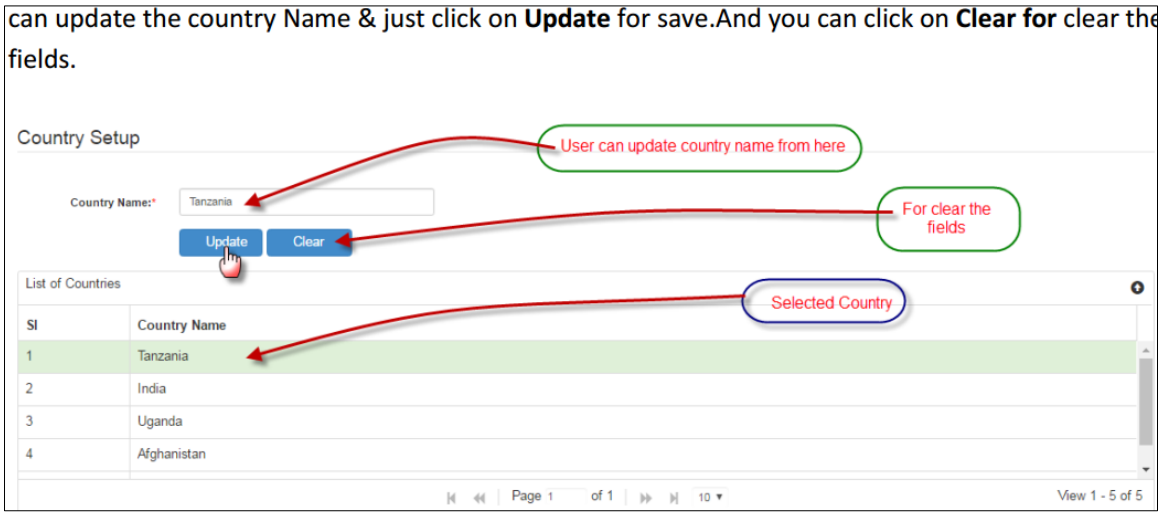

Figure 3.5.3

Program

Here, user can setup the Program name. Just type Program Name in the following field & click on Save. Then Program Name will be added in the Program List.

User Role

User can setup the User Role. Just type Role name in the following field & click on Save. Then Role Name will be added in the User Role List

Application User

User can setup the Application user details. Fill up all the required fields & click on Save. Then Application User information added in the list. User can edit his/her information also.

Application User Role Mapping

User can be mapped with the Role. First select any User which are available in the drop down. In the meantime select any Role against Selected User. And then click on Save. A successful message will appear in the Right down corner after saved.

User Access Control

In here user can create new role for user account. For create role, first click on Create New Role button & then Type the Role name & click on Save. After saved user can check from the User Role list.

User can also permit the feature accessibility in specific role wise. So that first select the created role from the drop down list. Then give permission for each feature click on check box & a tick mark will appear there & then click on Submit. After that, only check marked feature will be accessible by using the role name.

Approval

This is the place where all of the approval process will be completed. An Employee will apply to request for approval to travel anywhere for his/her official task purpose. Then Assigned person can approve or reject that request. Or after request employee can also cancel his/her request.

Dept. Wise Director Assign

Here, Department wise director can be assigned. And only Assigned person can approve or reject the travel request of employee.

Director Approval Request

Here, an employee can request for official travel. For this, Employee have to fill the given fields first then click on Save & automatically this request will appear in the assigned department wise director's approval list.

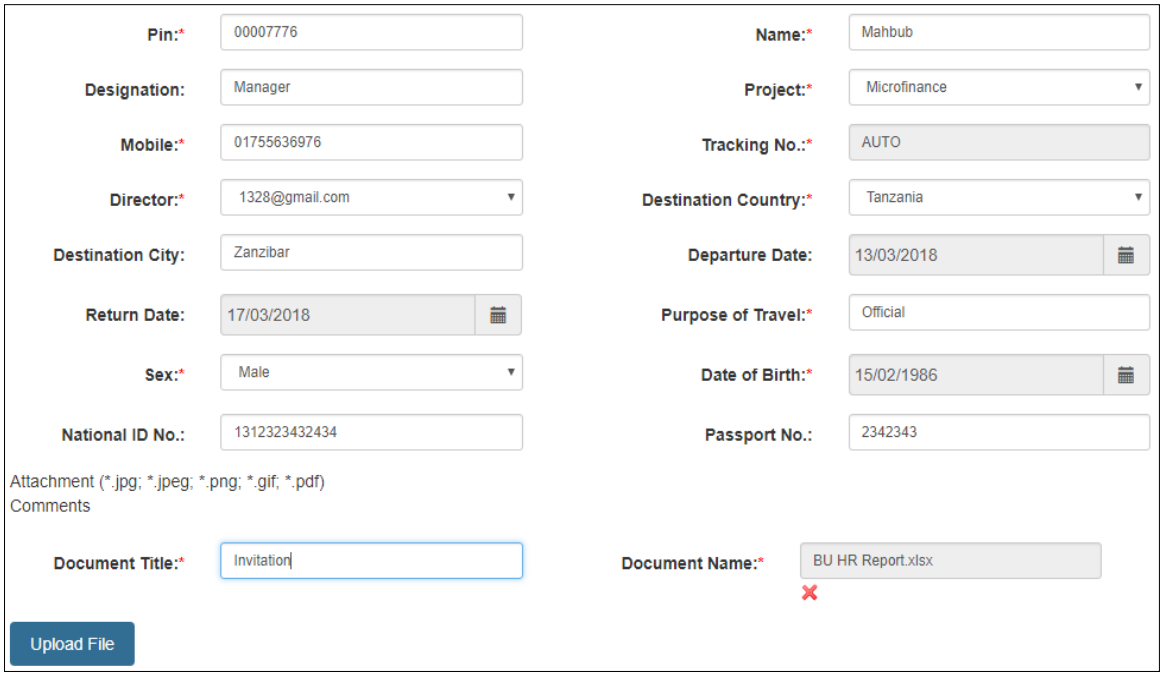

Figure 3.5.4

Cancel Approval Request

After submitting the approval request for travel, an employee can cancel his/her approval request by selecting Tracking ID & check mark the Cancel checkbox & then just click on Update. Also Employee can clear the information by clicking "Clear" button

Approve or Reject Travel Request

Here, all of the travel request from employee will be appeared here. And Director is able to approve/reject/call for discuss that approval request from here. See the following process.

View of after click on Details

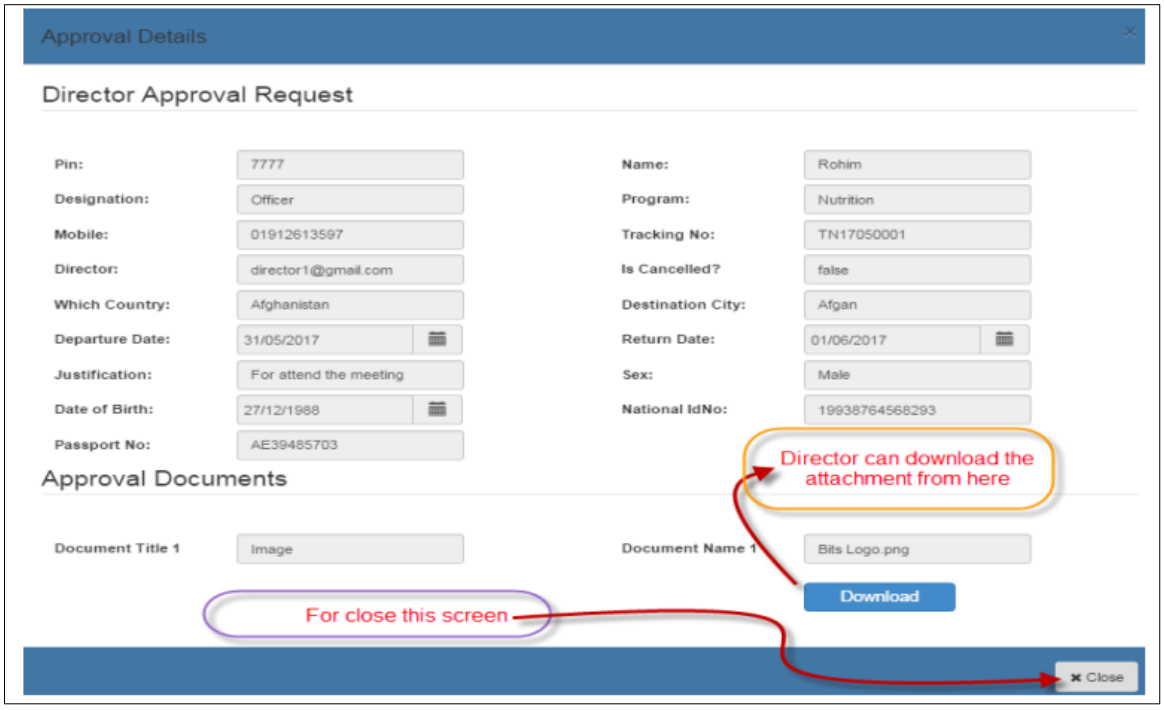

Figure 3.5.5

After a click on approve then you will get the following pop up for your positive & negative confirmation.

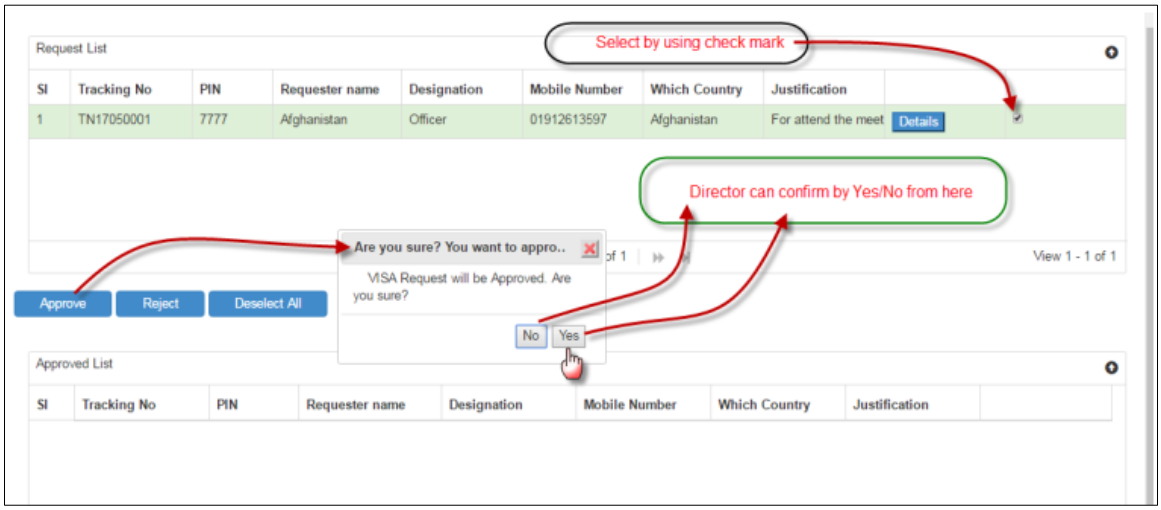

Figure 3.5.6

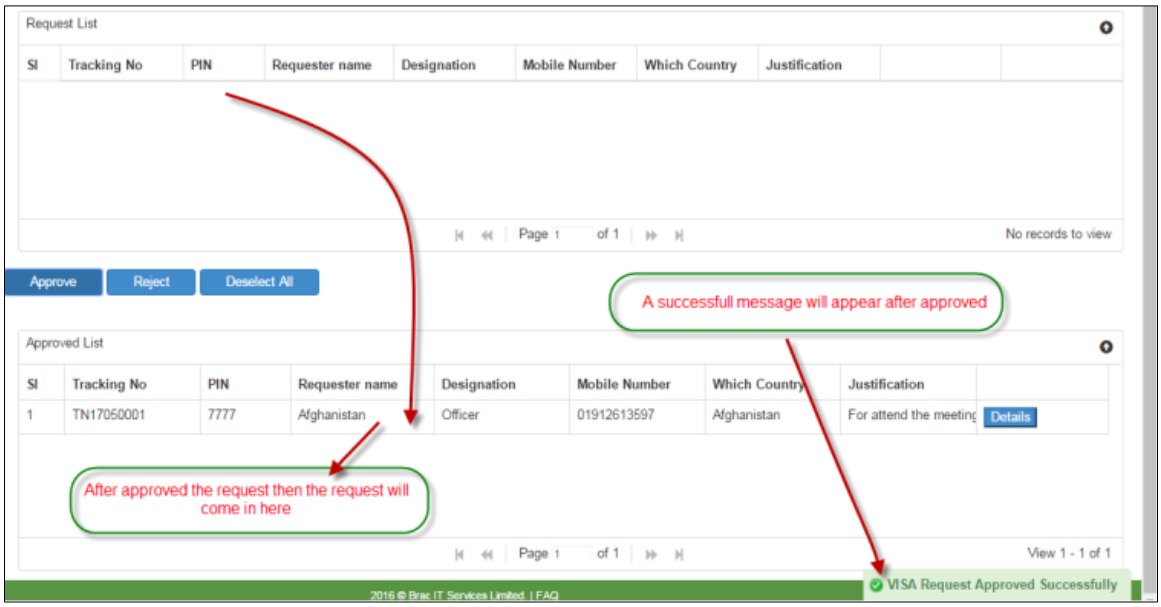

Figure 3.5.7

After click on Reject then you will get the following pop up for your positive & negative confirmation.

Pending & Approved List

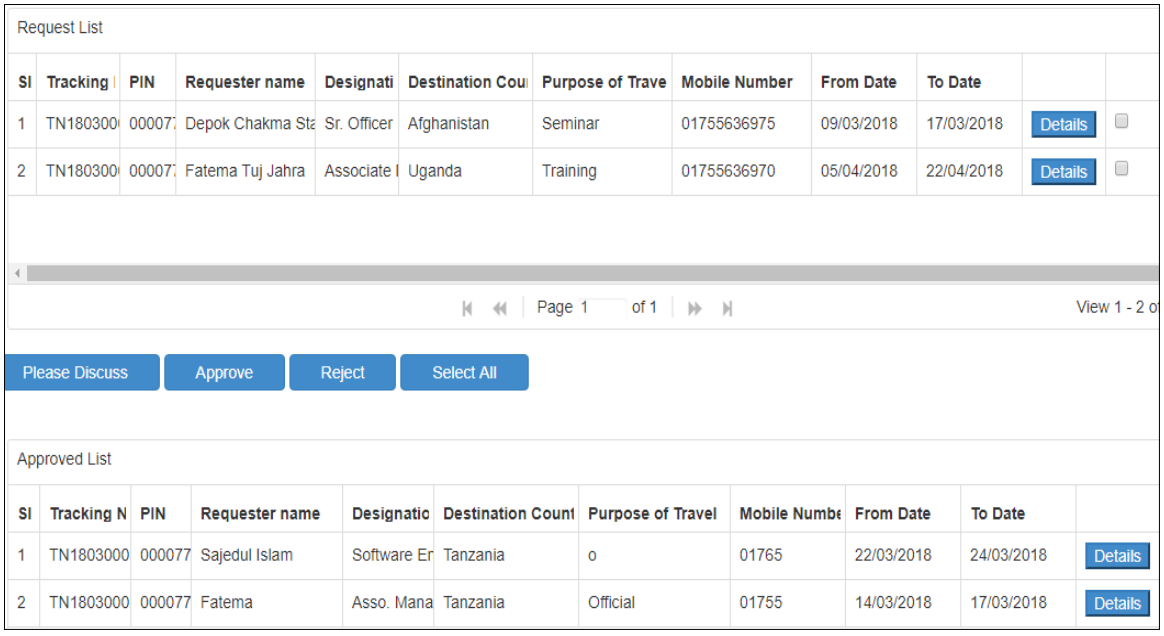

Figure 3.5.8

VISA Services

Here user will get the VISA request window & Ticket request process. VISA & Ticket request can be done only after getting director's approval.

VISA Request

By this feature an employee can apply for VISA assisting which will be available after getting approval from his/her Director. Just select the approved tracking id from Tracking No field & then some fields information automatically displayed. Now fill up only rest of the fields & click on Save for VISA assistance request. By this feature an employee can apply for VISA which will be available after approved his/her Director Approval request. Just select the approved tracking id from Tracking No field & then some fields information automatically displayed. Now fill up only rest of the fields & click on Save for VISA request. After selecting the Tracking ID

Figure 3.5.9

Successful message will appear after clicking on Save.

VISA Status Update

Here, employee will update the VISA status. Suppose an employee apply for VISA in any specific Embassy. After getting the VISA, employee needs to update specific fields in the VISA status Update feature.

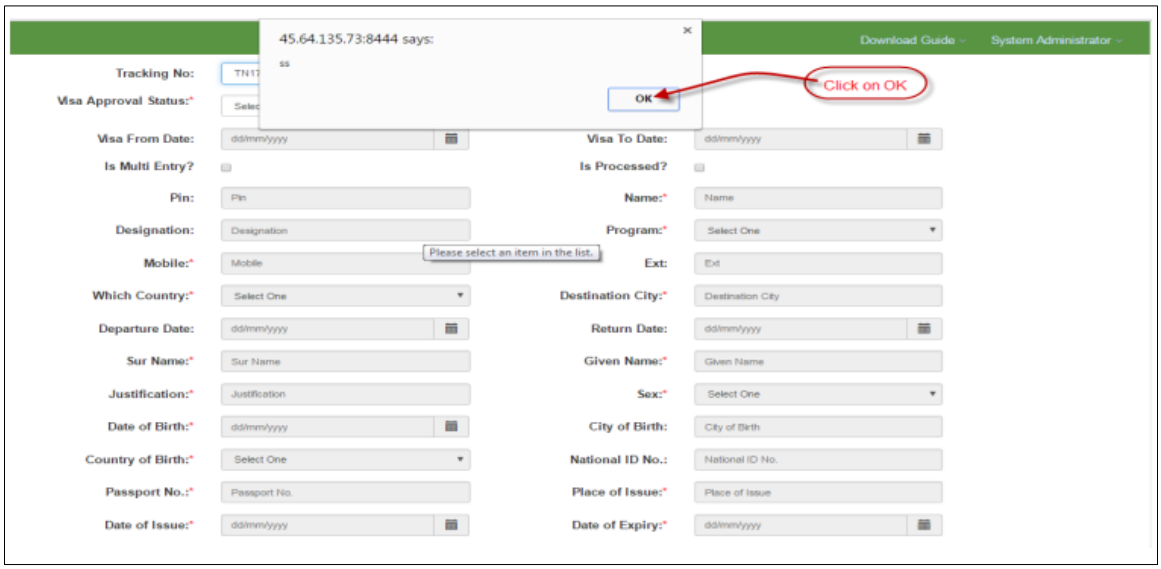

Figure 3.5.10

Ticket Issue

Every Ticket Details will be entered here.

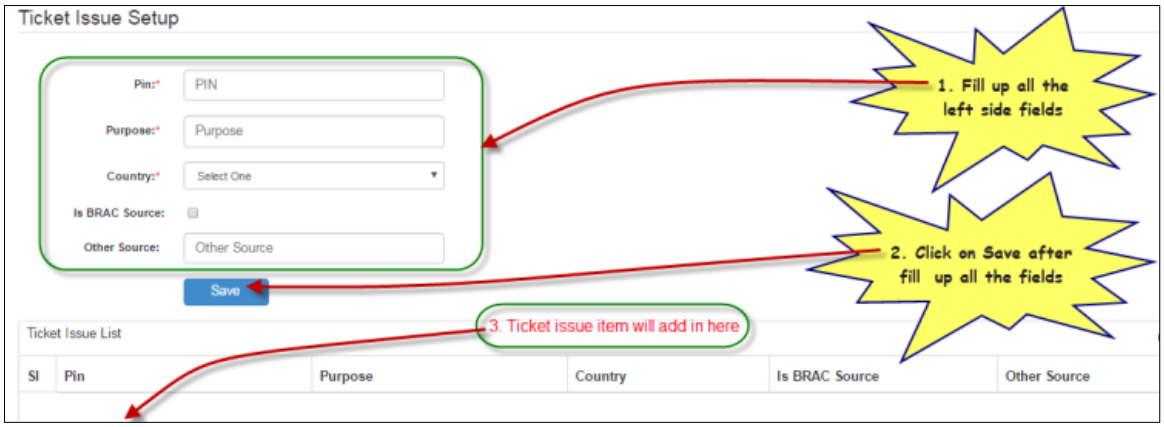

Figure 3.5.11

VISA Unprocessed

Unprocessed VISA ticket request can be updated from here by providing VISA details & checking Is Processed checkbox.

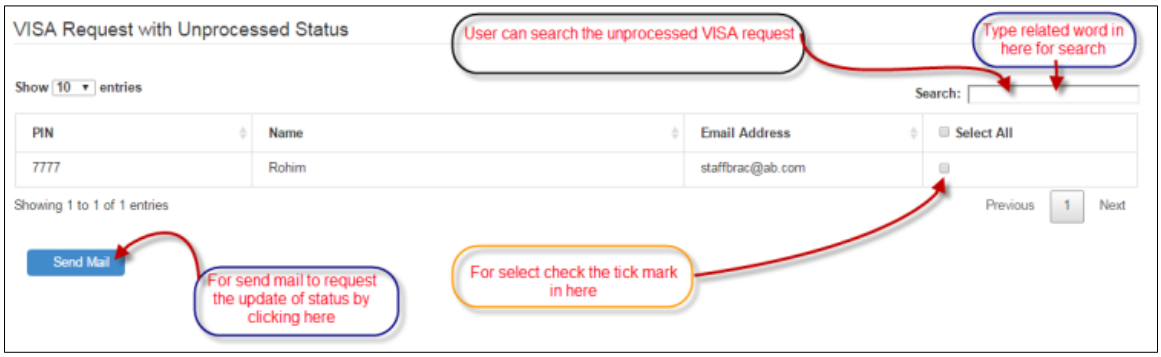

Figure 3.5.12

First select any unprocessed VISA request & then click on Send mail. Now a confirmation pop will appear.

Figure 3.5.13

Reports

It's the stage where user can find the reports.

Year Wise Visit Summary

From here user can get the Year wise visit summary report. First select the country name & then select the year. Now click on Preview & get the report in new tab. Report will appear in the new tab after click on Preview button.

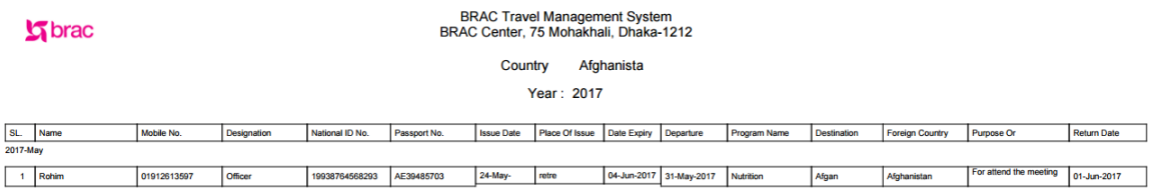

Figure 3.5.14

Program Wise Visit Summary
From here user can get the Program wise visit summary report. First select the Program name & then select the Date range. Now click on Preview & get the report in new tab.

Month Wise Employee Departures

From here user can get the Month wise Employee Departure Report. Select the Month Name & then click on Preview & get the report in new tab.

Tracking No. Wise Report

From here user can get the Tracking No. Wise Report. Type the Tracking No. & then click on Preview & get the report in new tab.

System Integration

Now all HR information will be fill upped by user. Later I will do integration with HR Database. After writing PIN all required employee information like Name, DOB, Passport NID etc will automatically be loaded here.

Also I will develop the Ticket Management Module later.

3.6 Reliability, Availability, & Serviceability (RAS)

The features of this application considering RAS will be,

Processor:
Processor instruction error detection (e.g. residue checking of results) with instruction retry e.g. substitute processor recovery in IBM mainframes, or "Instruction replay technology" in Itanium systems. Processors running in lock-step to accomplish master-checker or voting schemes. Machine check architecture to echo errors to the OS.

Memory:
Parity or ECC (including single device correction) security of memory components (cache and system memory), as well as memory bus; bad cache line disabling; memory cleaning; memory sparing; bad page offlining; redundant bit steering; redundant array of independent memory (RAIM).

I/O:
Cyclic redundancy check checksums for data retry/transmission and data storage, e.g. PCIe Advanced Error Reporting, redundant I/O paths.

Storage:
RAID configurations for magnetic disk storage. Journaling file systems for file repair after crashes. Checksums on both data and metadata, and background scrubbing.

Power/cooling:
Replication of components to evade single points of failure (for example power-supplies). Over-designing the structure for the stated operating ranges of clock frequency, temperature, voltage, vibration. Temperature sensors to throttle operating frequency when temperature goes out of description. Surge protector, uninterruptible power supply, and supplementary power.

System:
Hot exchange of components. Predictive failure study to calculate which alternating correctable errors will lead eventually to hard non-correctable errors. Partitioning of computer components to allow one large system to act as several smaller systems. Virtual machines to decrease the severity of operating system software errors. Redundant I/O domains or I/O partitions for providing virtual I/O to guest virtual machines. Computer clustering capability with failover capability, for complete redundancy of hardware and software. Lively software updating to dodge the need to reboot the system for a kernel software update. Independent service processor for serviceability: remote monitoring, notifying and control.

Chapter 4

Results

From the first UAT, I got some feedback. Basically those are UI level changes, changes in design. Also they give some Change Request for adding few new features.

User Management

In User Management section, all of the application user related setting are available.

Here, user can setup the country name. Just type Country Name in the following field & click on Save. Then Country Name will add in the country List.
If user want to update the country then select any country from list & see the above Save button already changed to update & selected country name showing in the Country Name field. Now you can update the country Name & just click on Update for save. And you can click on Clear for clear the fields.

There is Setup page for Program name. Just type Program Name in the following field & click on Save. Then Program Name will be added in the Program List.

User can create the User Role. Just type Role name in the following field & click on Save. Then Role Name will be added in the User Role List

User can create, update the Application user details. Fill up all the required fields & click on Save.
Then Application User information added in the list. User can edit his/her information also.

Application User can be mapped with specific Role. First select any User which are available in the drop down. In the meantime select any Role against Selected User. And then click on Save. A successful message will appear in the Right down corner after saved.

Here user can create new role for user account. For create role, first click on Create New Role button & then Type the Role name & click on Save. After saved user can check from the User Role list.

User can also permit the feature accessibility in specific role wise. So that first select the created role from the drop down list. Then give permission for each feature click on check box & a tick mark will appear there & then click on Submit. After that, only check marked feature will be accessible by using the role name.

Approval

This is the place where all of the approval process will be completed. An Employee will apply to request for approval to travel anywhere for his/her official task purpose. Then Assigned person can approve or reject that request. Or after request employee can also cancel his/her request.
Here Department wise director can be assigned. And only Assigned person can approve or reject the travel request of employee.

Employee can request for official travel. For this, Employee have to fill the given fields first then click on Save & automatically this request will appear in the assigned department wise director's approval list.

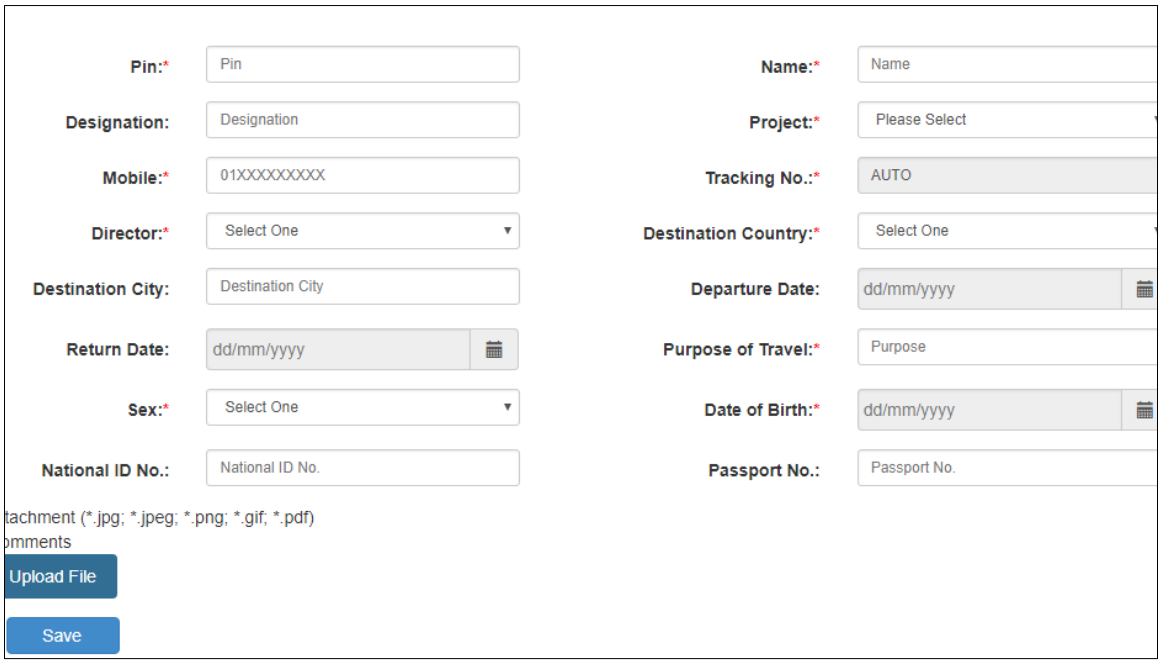

Figure 4.1 Director Approval Request

After submitting the director approval request for travel, an employee can cancel his/her approval request by selecting Tracking ID & check mark the Cancel checkbox & then just click on Update. This can be done only before the respective director has been approved/rejected. Also Employee can clear the information by clicking "Clear" button

Here, all of the travel request from employee will be appeared here. And Director is able to approve/reject/call for discuss that approval request from here. See the following process.

There is a list of Rejected Request List also. Whenever needed, Director can see the previously Rejected Request list.

Pending & Approved Request List:

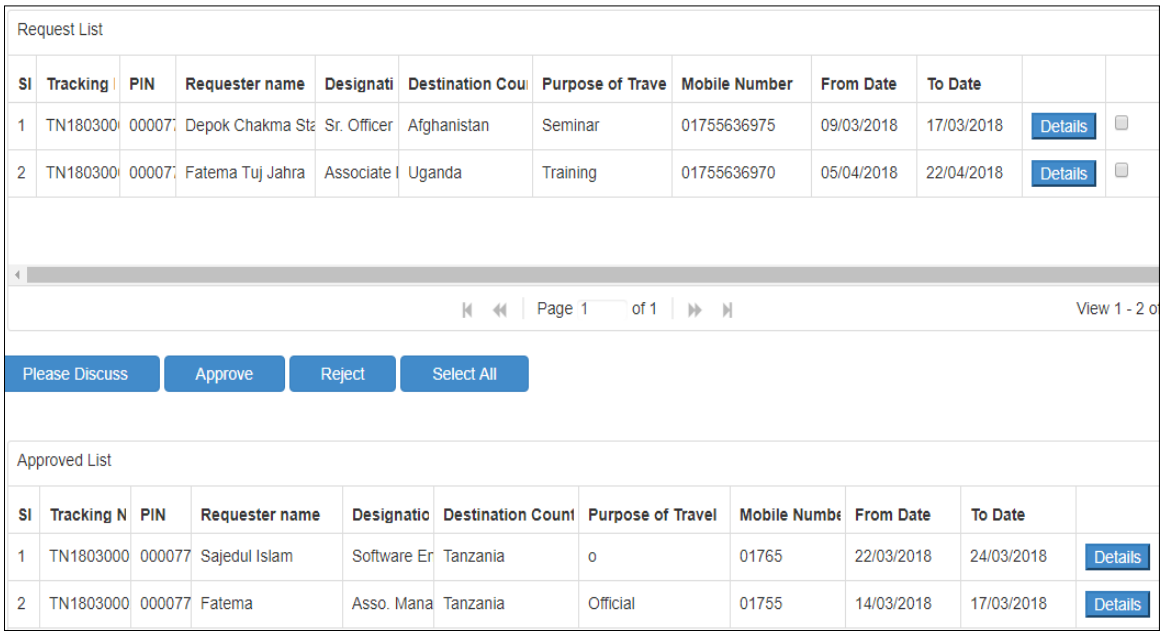

Figure 4.2 Pending & Approved Request List

VISA Services

Here user will get the VISA request window & Ticket request process. VISA & Ticket request can be done only after getting director's approval.

By the feature VISA Request, an employee can apply for VISA assisting which will be available after getting approval from his/her Director. Just select the approved tracking id from Tracking No field & then some fields information automatically displayed. Now fill up only rest of the fields & click on Save for VISA assistance request. By this feature an employee can apply for VISA which will be available after approved his/her Director Approval request. Just select the approved tracking id from Tracking No field & then some fields information automatically displayed. Now fill up only rest of the fields & click on Save for VISA request.

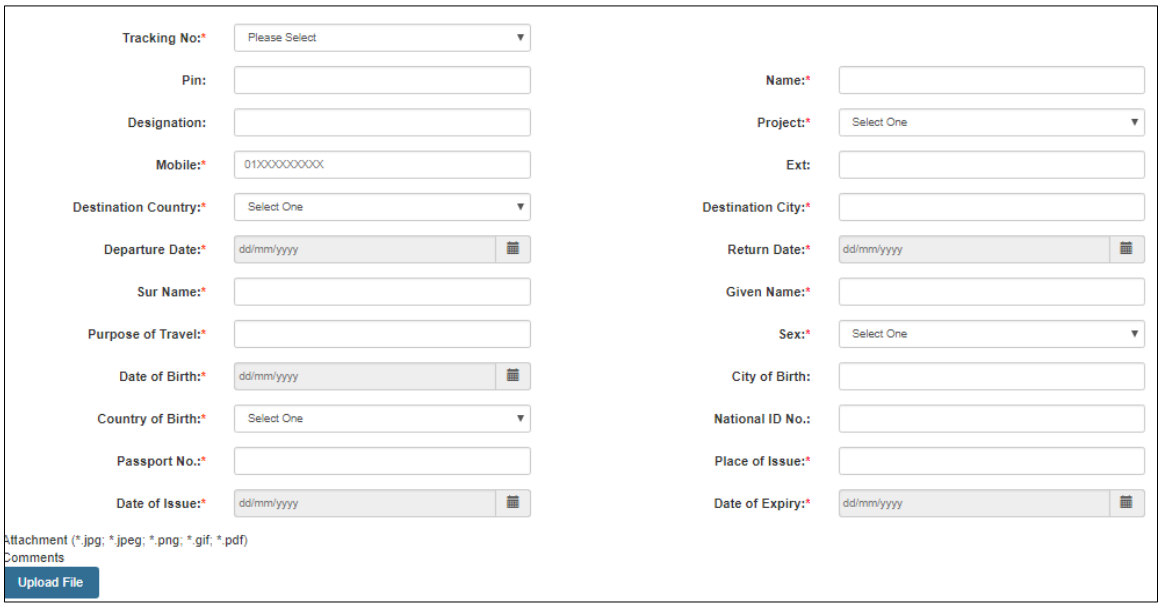

Figure 4.3 VISA Request

Here, employee will update the VISA status through the page VISA Status Update. Suppose an employee apply for VISA in any specific Embassy. After getting the VISA, employee needs to update specific fields in the VISA status Update feature.

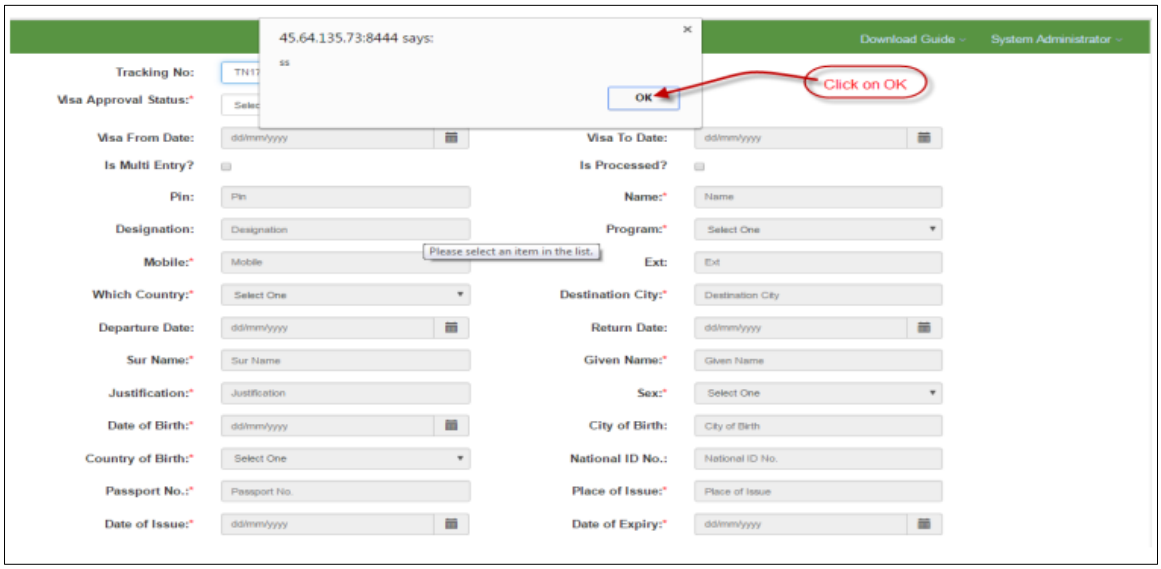

Figure 4.4

Every Ticket Details will be entered here.

Unprocessed VISA ticket request can be updated from here by providing VISA details & checking Is Processed checkbox.

First select any unprocessed VISA request & then click on Send mail. Now a confirmation pop will appear.

Reports

It's the stage where user can find the reports.

Year Wise Visit Summary

From here user can get the Year wise visit summary report. First select the country name & then select the year. Now click on Preview & get the report in new tab. Report will appear in the new tab after click on Preview button.

BRAC Travel Management System
BRAC Center, 75 Mohakhali, Dhaka-1212

Country Afghanista

Year : 2017

SL.	Name	Mobile No.	Designation	National ID No.	Passport No.	Issue Date	Place Of Issue	Date Expiry	Departure	Program Name	Destination	Foreign Country	Purpose Or	Return Date
2017-May														
1	Rohim	01912613597	Officer	19938764568293	AE39485703	24-May-	retre	04-Jun-2017	31-May-2017	Nutrition	Afgan	Afghanistan	For attend the meeting	01-Jun-2017

Figure 4.5

Month Wise Employee Departures

From here user can get the Month wise Employee Departure Report. Select the Month Name & then click on Preview & get the report in new tab.

Program Wise Visit Summary

From here user can get the Program wise visit summary report. First select the Program name & then select the Date range. Now click on Preview & get the report in new tab.

Tracking No. Wise Report

From here user can get the Tracking No. Wise Report. Type the Tracking No. & then click on Preview & get the report in new tab.

Chapter 5

Conclusion

The main purpose of developing this application is to keep record of every employee and department's abroad visit status and generate automated reports. So that VISA & Protocol department can track employees and generate yearly reports easily from every required aspect. Also I will integrate this system with HR Database and with Accounts Database. Airline ticket management system has been a tough ask for the management. Certain element of this project leaves scope for development. So there is plan for developing Ticket Management Module in future.

References

To get domain related knowledge I analyzed following sites

[1] VFS Global. Site url- http://www.visathing.com/schengen/

[2] VISA Thing. Site url- http://www.visathing.com/

[3] IVAC. Site url- http://www.ivacbd.com/

Appendix

Single Sign –on (SSO) is a user authentication service and session that allows a user to use a set of credential for login to access numerous application.

Following code block receives encrypted return message from authentication service. After decrypting using simple AES algorithm, I got the result of authentication and then alow or deny login depending on the result.

```
def index() {
    try {
        flash.default="";
        response.setHeader("Access-Control-Allow-Origin", "*");
        GrailsParameterMap grailsParameterMap = (GrailsParameterMap) params
        String encrypted = grailsParameterMap.multipass;
        String key = "com.www.https.google";
        SimpleAesEncryption simpleAesEncryption = new SimpleAesEncryption();

        String decrypted = simpleAesEncryption.decrypt(encrypted, key);
        String output = decrypted.substring(key.length())
        def resultList = output.tokenize('|')
        String authenticatedVal = resultList[3].substring(14);
        String uName = resultList[0].substring(5);
        if (!springSecurityService.isLoggedIn() &&
authenticatedVal.equalsIgnoreCase('true')) {
            render(view: '/login/auth_bypass', model: [username: uName])
        }
        else if (!springSecurityService.isLoggedIn() &&
authenticatedVal.equalsIgnoreCase('false')) {
            flash.error = "Username & Password doesn't match.";
            render(view: '/login/auth_bypass', model: [username: uName])
        }else {
            flash.error = "Username & Password doesn't match.";
            render(view: '/login/auth')
        }


    } catch (Exception ex) {
        log.error(ex.message)
        render(ex.message)
    }
}
```

In this Simple AES Encryption, PKCS5Padding has been used. UTF-8 character Encoding used.

```
public SimpleAesEncryption() {
}
private final String characterEncoding = "UTF-8";
private final String cipherTransformation = "AES/CBC/PKCS5Padding";
private final String aesEncryptionAlgorithm = "AES";
```

```
public byte[] decrypt(byte[] cipherText, byte[] key, byte[] initialVector)
throws NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
InvalidAlgorithmParameterException, IllegalBlockSizeException,
BadPaddingException {
    Cipher cipher = Cipher.getInstance(cipherTransformation);
    SecretKeySpec secretKeySpecy = new SecretKeySpec(key,
aesEncryptionAlgorithm);
    IvParameterSpec ivParameterSpec = new IvParameterSpec(initialVector);
    cipher.init(Cipher.DECRYPT_MODE, secretKeySpecy, ivParameterSpec);

   cipherText = cipher.doFinal(cipherText);
    return cipherText;
}

public byte[] encrypt(byte[] plainText, byte[] key, byte[] initialVector) throws
NoSuchAlgorithmException, NoSuchPaddingException, InvalidKeyException,
InvalidAlgorithmParameterException, IllegalBlockSizeException,
BadPaddingException {
    Cipher cipher = Cipher.getInstance(cipherTransformation);
    SecretKeySpec secretKeySpec = new SecretKeySpec(key,
aesEncryptionAlgorithm);
    IvParameterSpec ivParameterSpec = new IvParameterSpec(initialVector);
    cipher.init(Cipher.ENCRYPT_MODE, secretKeySpec, ivParameterSpec);
    plainText = cipher.doFinal(plainText);
    return plainText;
}

private byte[] getKeyBytes(String key) throws UnsupportedEncodingException {
    byte[] keyBytes = new byte[16];
    byte[] parameterKeyBytes = key.getBytes(characterEncoding);
    System.arraycopy(parameterKeyBytes, 0, keyBytes, 0,
Math.min(parameterKeyBytes.length, keyBytes.length));
    return keyBytes;
}

public String encrypt(String plainText, String key) throws
UnsupportedEncodingException, InvalidKeyException, NoSuchAlgorithmException,
NoSuchPaddingException, InvalidAlgorithmParameterException,
IllegalBlockSizeException, BadPaddingException {
    String saltedPlainText  =  key + plainText;
    byte[] plainTextbytes = saltedPlainText.getBytes(characterEncoding);
    byte[] keyBytes = getKeyBytes(key);
    byte[] encryptedByte = encrypt(plainTextbytes, keyBytes, keyBytes);
    String encodedString = Base64.encodeBase64String(encryptedByte);
    encodedString = URLEncoder.encode(encodedString, "UTF-8");
    return encodedString;
}

public String decrypt(String encryptedText, String key) throws KeyException,
GeneralSecurityException, GeneralSecurityException,
InvalidAlgorithmParameterException, IllegalBlockSizeException,
BadPaddingException, IOException {
    encryptedText = encryptedText.replaceAll("_", "/");
    encryptedText = encryptedText.replaceAll("\\-", "+");
    encryptedText += "=";
    byte[] cipheredBytes = Base64.decodeBase64(encryptedText);
    byte[] keyBytes = getKeyBytes(key);
    return new String(decrypt(cipheredBytes, keyBytes, keyBytes),
characterEncoding);
}

public static void main(String arg[]) {
    String secreateKey =
"name:raju|password:brac1234|authenticated:true|authorization:true|login:true|pl
atform:null";
```

```
    String key = "com.bits.btts.wb.project";
    SimpleAesEncryption anotherTest = new SimpleAesEncryption();
    try {
        String encrypted =
"BkqY1MhTCtpLJ%2Fkae%2FxPutw0Hxyr5%2B6sfS0mzzBEnxwC09twnt%2BUK2elcxvLQsWcVMvEd9w
KIH2fj%2FaSbVnpOwEd6k%2FI%2BEmMZS1H8oKCAn9J1mztcb6MHo2Nu9fcgy3wPYRm%2BFzrsWxlpNa
piZNNVC187XH4bgvCMaaZzK6rGVlUiqXtVTxGKHrMHfGO3mfEgmOipsX6E%2ByN8O7t6h%2BVX%2Fw%2
BCl1L%2BrhoGzI8X4LGKqI44mN9%2Bve%2BAUR%2B%2BP2ySD30%2B1OoyTc5wJebC7RF3Ai%2F2XZBL
w227HhQOiFBrzjwvjkHAOtaEDrvULlvmwT3wn74rBpaz4%2BZkfsd%2BibYlI8a8w%3D%3D";
        System.out.println("Encrypted : " + encrypted);
        String decrypted = anotherTest.decrypt(encrypted, key);

        String v = decrypted.substring(key.length());

        System.out.println("Decrypted : " + v);
    } catch (KeyException ex) {
        Logger.getLogger(SimpleAesEncryption.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (GeneralSecurityException ex) {
        Logger.getLogger(SimpleAesEncryption.class.getName()).log(Level.SEVERE,
null, ex);
    } catch (IOException ex) {
        Logger.getLogger(SimpleAesEncryption.class.getName()).log(Level.SEVERE,
null, ex);
    }
}
```