



GSCC HR SYSTEM

Mohamed Irshad Mohideen

BIT Reg No : R092391

Index No : 0923915

Supervisor : Mr. Hossam Al Masry

November 2017



**This dissertation is submitted in partial fulfilment of the requirement of the
Degree of Bachelor of Information Technology (external) of the
University of Colombo School of Computing**

Declaration

I certify that this dissertation does not incorporate, without acknowledgement, any material previously submitted for a degree or diploma in any university and to the best of my knowledge and belief, it does not contain any material previously published or written by another person or myself except where due reference is made in the text. I also hereby give consent for my dissertation, if accepted, to be made available for photocopying and for interlibrary loans, and for the title and abstract to be made available to outside organizations.

Signature of Candidate :  Date: 06-11-2017

Name of Candidate: MOHAMED IRSHAD MOHIDEEN

Countersigned by:

Signature of Supervisor :  Date: 6.11.2017

Name of Supervisor : MR. HOSSAM AL MASRY

Abstract

Gulf Stevedoring and Contracting Co. Ltd. (GSCC) is one of the sea port container terminal operators in Jeddah Islamic Port in Saudi Arabia. GSCC has been in this stream of business since 1985 and had held prominence in managing and operating sea port terminals.

To maintain human resources records, GSCC has been maintaining one of the legacy applications developed using Mumps (presently known as Caché). With the increase of operational and management demand for human resource (HR) related information, GSCC finds the features of the existing system is extremely inadequate due to its text-only based user interfaces, limitations in report generation, inability out spreadsheets from system etc.

In order to overcome these obstacles, new GSCC HR System has been proposed. By moving to an application with graphical user interfaces (GUI), users would be provided with all rich functions such as colours, ordering, shapes etc. Further, enhanced new system is expected to provide report generation by end users themselves instead of heavily depending IT professionals as structure and filters of the report change intermittently. On scrutiny of the requirements, Rational Unified Process (RUP) has been chosen as the suitable development methodology due to its flexibility.

A network based system accessible by multiple users has been decided to be developed using C# programming language running on windows operating systems on servers as well as clients. Microsoft visual studio is chosen to be development environment while Microsoft SQL Server is used for database.

GSCC is expected to lay the foundation for an advanced HR system with this proposed system and confidently steps in creating to become an emerging competitive container terminal operator in the arena of port operations.

Acknowledgement

I would like to extend my great gratitude to BIT coordinators and staffs in (University of Colombo School of Computing) UCSC for their continuous efforts in providing us the great source of knowledge and practice in completing the project work and then eventually the degree. I thank the Project Coordinators for all guidance and assistance provided throughout the course of the project.

My great appreciation goes to Mr. Hossam Al Masry, technical supervisor, for providing guidance and feedback in fulfilling project requirements. His timely review and advices have been excellent contributions in the completion of this project.

I thank Mr. Seif Raghieb Seif, HR Manager, for cordially cooperating in all stages of the project development. HR staffs' courteous assistance in our interviews and observations also is greatly valued.

Colleagues in our office cannot be forgotten. They were always willing to discuss and suggest the best avenues in fulfilling this time bound task.

My gratitude to all staffs in BIT EDC goes too with great sense of appreciation for their support in various stages of this degree program.

Contents

Declaration.....	i
Abstract.....	ii
Acknowledgement.....	iii
Contents.....	iv
List of Figures.....	viii
List of Tables.....	x
List of Acronyms.....	xi
Chapter 1 - Introduction.....	1
1.1 The Client.....	1
1.2 Problem Domain.....	1
1.3 Motivation for the project.....	1
1.4 Objectives of the project.....	3
1.5 Outline of Chapters.....	4
Chapter 2 - Analysis.....	5
2.1 Introduction.....	5
2.2 Facts Gathering Techniques.....	5
2.2.1 Interviews.....	5
2.2.2 Observation.....	5
2.3 Description of the Current System.....	6
2.4 Similar solutions with references.....	8
2.4.1 Paylite HRMS System.....	8
2.4.2 ADEPT HRMS Solution.....	10
2.4 Requirements.....	12
2.4.1 Functional Requirements.....	12
2.4.1 Non-Functional Requirements.....	12

2.4 Development Methodology	13
2.4.1 RUP – Rational Unified Process	13
2.4.2 Benefits in using RUP	14
Chapter 3 - System Design.....	15
3.1 Introduction	15
3.2 Alternative Solutions.....	15
3.2.1 Improve the existing text based system	15
3.2.2 Purchase packaged software	15
3.2.3 Develop a stand-alone system	16
3.2.4 Develop a network based system	16
3.3 Candidate matrix with alternate Solutions.....	17
3.3 Feasibility analysis matrix	18
3.4 Design Methodology	19
3.4.1 Design Diagrams	19
3.4.2 System Software.....	19
3.5 High-level use case diagram.....	20
3.6 Class Diagram	21
3.7 Sequence Diagram	22
3.8 Entity Relationship Diagram	23
3.9 Database Design	24
3.9 User Interface Design.....	25
3.9.1 Good user interface design principles	25
3.9.2 Designs used in the system to provide better interactions	25
3.9.3 Login Screen.....	29
3.9.4 Main Menu.....	29
3.9.5 User Access Management Form.....	30
3.9.6 Employee Form	31

3.9.7 Contracts Form	31
Chapter 4 - Implementation.....	33
4.1 Introduction	33
4.2 Implementation Environment	33
4.2.1 Hardware Environment	33
4.2.2 Software Environment	33
4.3 Development Tools	33
4.3.1 Microsoft Visual Studio 2012/2015	34
4.3.2 C# Language	34
4.3.3 Crystal Reports	34
4.3.4 Microsoft Office Excel.....	34
4.4 Platform Dependence.....	34
4.5 Data Import from Legacy Application.....	35
4.6 Major Code Structures	36
4.6.1 Connection to the database	36
4.6.2 Login with password encryption	36
4.6.3 Open the form as per the name defined in	37
4.6.4 Get a single field value from a table	38
4.6.5 Get a table as a result	38
4.6.6 Class structure for objects with database transactions	39
4.6.7 Class to call description of reference tables	40
4.7 Existing Re-usable Codes	41
4.7.1 Encrypt Function	41
Chapter 5 - Evaluation.....	42
5.1 Introduction	42
5.2 Test Plan	42
5.3 Test Procedures	43

5.3.1 Unit Testing	43
5.3.2 Integration Testing	43
5.3.3 System Testing	43
5.3.4 Acceptance Testing	43
5.4 - Test Cases.....	44
5.4.1 Test case for Login Screen.....	44
5.4.2 Test case for User Access Management	45
5.4.3 Test case for Nationality Reference Codes.....	46
5.5 User Evaluation	48
Chapter 6 - Conclusion	50
6.1 Introduction	50
6.2 Future Enhancements.....	51
6.3 Lessons Learnt.....	52
References.....	53
Appendix A - System Documentation	55
Appendix B - Design Documentation	57
Appendix C - User Documentation	64
Appendix D - Management Reports	68
Appendix E - Test Results.....	71
Appendix F - Code Listing	78
Appendix G - Client Certificate	83
Glossary	84
Index.....	85

List of Figures

Figure 1.1: Text-based Screen for Employee Data in Existing Application	2
Figure 1.2: Text-based Screen with Contract Details	2
Figure 2.1: High Level Use Case Diagram for Current System	6
Figure 2.2: Paylite HRMS – Simple screen layout	9
Figure 2.3: Paylite HRMS – Simple color scheme	10
Figure 2.4: ADEPT HRMS – Organizational Chart	11
Figure 2.5: ADEPT HRMS – Simple Grid List	11
Figure 2.6: Stages in Rational Unified Process	13
Figure 2.7: Phases of Rational Unified Process	14
Figure 3.1: High Leavel Use case Diagram	20
Figure 3.2: Design of Class Diagram	21
Figure 3.3: Sequence Chart for Contract Details Update	22
Figure 3.4: ER Diagram	23
Figure 3.5: Database Diagram	24
Figure 3.8: Error Message displaying for wrong date input	26
Figure 3.6: Design of a form with limited set of colours	26
Figure 3.7: Invalid Input Message for wrong character input	26
Figure 3.9: Message that comes when mandatory field is blank	27
Figure 3.10: Message that takes user confirmation	27
Figure 3.11: Confirmation message for record insert	28
Figure 3.12: Confirmation message for record update	28
Figure 3.13: Login Screen	29
Figure 3.14: Main Menu	29
Figure 3.15: User Access Management Form	30
Figure 3.16: Employees Form	31
Figure 3.17: Contract Form showing contracts in a table	31
Figure 3.18: Contract form showing details of a sing contract	32
Figure 4.1: Data Load from Text file to HR Database	35
Figure 4.2 : Class structure used for transactional objects	39
Figure 5.1: User Evaluation Summary	49

Figure B.1: Use case diagram for User Login	57
Figure B.2: Use case diagram for User Management	58
Figure B.3: Use case diagram for Employee Module	59
Figure B.4: Use case diagram for Contracts Module	60
Figure B.5: Use case diagram for Leave Module.....	61
Figure B.6: Use case diagram Sick Leave Module	62
Figure B.7: Use case diagram for Reports	63
Figure C.1: User Sign in Form	64
Figure C.2: Main Menu.....	64
Figure C.3: Option “Employee” to be clicked to Open	65
Figure C.4: Employee form is open	65
Figure C.5: Report Menu Selection.....	66
Figure C.6: Reference Menu Selection	66
Figure C.7: Nationality Reference List Displayed	67
Figure C.8: Adding a New Nationality	67
Figure D.1: Monthly Employee Status	68
Figure D.2: Active Employee Summary by Department.....	69
Figure D.3: Active Employee Summary by Nationality	70

List of Tables

Table 3.1: Candidate Matrix	17
Table 3.2: Feasibility Analysis Matrix.....	19
Table 5.1: Test Case for Login Screen.....	44
Table 5.2: Test case for User Access Management	45
Table 5.3: Test case for Nationality Reference Maintenance.....	47
Table 5.4: User Acceptance Testing	48
Table B.1: Use Case Narrative for Login	57
Table B.2: Use case narrative for User Management	58
Table B.3: Use case narrative for Employee	59
Table B.4: Use case narrative for Contract.....	60
Table B.5: Use case narrative for Leave	61
Table B.6: Use case narrative for Sick Leave	62
Table B.7: Use case narrative for Reports	63
Table E.1: Test case for Employee Module.....	71
Table E.2: Test case for Contracts Module.....	72
Table E.3: Test case for Leave Module	73
Table E.4: Test case for Disciplinary Module	74
Table E.5: Test case for Sick Leave Module	75
Table E.6: Test case for Accrued Leave	76
Table E.7: Test case for Termination	77

List of Acronyms

BIT	-	Bachelor of Information Technology
CPU	-	Central Processing Unit
EDC	-	External Degrees Center
ERP	-	Enterprise Resource Planning
GSCC	-	Gulf Stevedoring Contracting Company
GB	-	Giga Bytes
GP	-	Great Plains
GUI	-	Graphical User Interface
HR	-	Human Resources
IDE	-	Integrated Development Environment
IT	-	Information Technology
MS	-	Microsoft
NFR	-	Non-Functional Requirements
PC	-	Personal Computer
RAM	-	Random Access Memory
RUP	-	Rational Unified Process
RDBMS	-	Relational Database Management System
SQL	-	Structured Query Language
UI	-	User Interface
UML	-	Unified Modeling Language
TEU	-	Twenty Equivalent Units
UCSC	-	University of Colombo School of Computing
VPN	-	Virtual Private Network

Chapter 1 - Introduction

1.1 The Client

Gulf Stevedoring and Contracting Co. Ltd. (GSCC) has been in the business of sea port terminal operations since 1985. Initially having handled terminals in Dammam, GSCC expanded its business securing successful projects in Jeddah, Saudi Arabia as well.

During the last two decades GSCC established a strong hold in the Jeddah Islamic Port and GSCC became the first terminal in the port to have handled one million TEUs within a single year.

To support GSCC's prime business, Terminal Operating Systems had been once developed in-house and later moved towards world renowned software from Navis.

One of such systems that were developed in parallel for GSCC's human resources record maintenance was the HR System. Originally developed using a language & database package called Mumps, the same has been now upgraded to Caché from Intersystem Co. Ltd.

1.2 Problem Domain

With the development of the new HR system, users would be provided with a GUI based largely interactive system. The reporting mechanism would be improved by producing output in Spreadsheets for further analysis. HR users would have an alerting instrument through the new HR system.

1.3 Motivation for the project

The HR system presently runs on text based UIs. Therefore, users undergo all the difficulties that a system missing a GUI (Graphical User Interfaces).

Figure 1.1 below shows an existing screen that shows employee's basic data.

```

AAA                               Employee                               08/11/2017

Project      [4 ]                GCT (NORTH)
Payroll      [2400224]           Old No [    ]
Name Family  [BILAL              ]
  First      [HANI ZAIDH         ]
  Other      [                    ]
Date of Birth [02/07/1969]
Nationality  [SRL]              SRI LANKAN
Sponsorship  [1 ]              GSCCO
Religion     [1 ]              MUSLIM
Passport No. [N4353873          ] Issue Place [CMB] COLOMBO
Issue Date   [22/06/2011]       Expiry Date [22/06/2021]
Address      [                    ] [                ]
  [          ] [                ] [                ]
Telephone    [                    ]
Name of Father [                ]
  Mother     [                    ]
  Spouse     [                    ]
Project DOJ  [15/07/2011] 14/08/32(H)
Company DOJ  [15/07/2011] 14/08/32(H)
Contract Years [3]

<PF3><PF4>

```

Figure 1.1: Text-based Screen for Employee Data in Existing Application

Figure 1.2 below shows an existing screen that displays information of contract details.

```

AAM                               Contract                               08/11/2017

Project      [4 ]                GCT (NORTH)
Payroll      [2400224]           BILAL, HANI ZAIDH
Effective Date [15/07/2017]      Joined : 15/07/2011

Contract Status [N]            NEW
  End Date      [14/07/2020]
  Probation End [                ]

Department    [COMP]           COMPUTER
Sub Department [COMP]           COMPUTER
Job Code      [A53]            APPLICATIONS TEAM LEADER
Employment    [1]              CONTRACTUAL
Family Status (Y/N) [Y]
Car Entitled (Y/N) [Y]
Air Class     [Y]              ECONOMY
Vacation Destination [CMB]      COLOMBO
Entitlement Code [2 ]
  Leave       [ 42]
  Months per Ticket [ 6]
Remarks      [ASSESSMENT CONTRACT RENEWED.          ]

<?><PF3><PF4>

```

Figure 1.2: Text-based Screen with Contract Details

The reports in the present system could only be printed on A3 papers with dot matrix printers. Producing Graphically designed reports are extremely cumbersome.

The reports in the system are pre-defined and with limited number of output fields. This restricts users from inserting the columns they need in the report and as a result acquire the support of IT even on inclusion of fields in reports.

The database has hierarchical data structure which makes integrating the data with any other business applications impossible.

HR department requires notifications when important events occur. With the existing system, this feature can hardly be achieved.

With the determination of overcoming these shortcomings, the advent of a new HR system was proposed. The new HR system is expected to completely wipe out the said issues and pave the way for better scalability in the future. A much more user-friendly experience together with smart alerting and interaction are anticipated in this system.

1.4 Objectives of the project

The new HR system would help achieve below merits

- Users would not be confined to a text-only based UIs, but would be provided with much more interactive GUIs.
- New system should support generation of reports with user selectable fields added in reports.
- All reporting should be obtainable in MS Excel spreadsheet format so that it could be used for further analysis or could be shared with ease.
- For important events, system should provide notification to selected HR staffs in order to alert and take appropriate measures accordingly.

- Data would be stored in a high-end latest, RDBMS Microsoft SQL Server and helps integration with other latest apps and databases with much ease. All data in the present hierarchical format would be migrated to tabular data layout.

1.5 Outline of Chapters

Having mentioned the motivation and objectives, Chapter 2 provides facts gathering techniques, a high level use case diagram. Reviews of similar solutions are discussed here. Further this chapter details what the existing system is about and the functional and non-functional requirements of the new HR system is explained as well. Chapter 3 goes on to discuss alternative solutions. Candidate matrix and feasibility matrix are used to highlight the impact and justify the development platform. Further, design methodology, UML diagrams and user interfaces designs used in the application are discussed here. The implementation phase is discussed in Chapter 4. The hardware and software requirements, the development tools, major code structures and reusable codes are detailed in this chapter. The methods used to test the system are given in Chapter 5. All four testing procedures, namely unit, integration, system and acceptance tests are explained while providing details for test cases to ensure system works as expected. User acceptance testing is included as well in this chapter. In the final chapter, Chapter 6, a critical evaluation of the system is given together with details on future enhancements and lessons learnt during the implementation of this project.

Chapter 2 - Analysis

2.1 Introduction

In IT systems analysis can include looking at end-user implementation of a software package or product; looking in-depth at source code to define the methodologies used in building software; or taking feasibility studies and other types of research to support the use and production of a software product, among other things. [1]

2.2 Facts Gathering Techniques

To identify what exactly is available in the existing system and what is expected as improvement with the new HR system, below given fact finding techniques were used.

2.2.1 Interviews

Since the client is upgrading the HR system from legacy application to a new one, and primary knowledge resource of the systems were with experienced users, direct interviews has been identified and used as the major fact-finding technique.

HR Manager and HR clerks were talked to and they provided immensely useful piece of information in elaborating the features available in the existing system.

HR Manager highlighted the primary goals that are expected from the new system such as getting rid of text-based UIs, ability to generate spreadsheets directly from the application etc.

2.2.2 Observation

This technique also proved as one of the effective one as all the existing functionalities in the system were carried on daily tasks by the users.

2.3 Description of the Current System

Figure 2.1 displays the use case diagram for the current system.

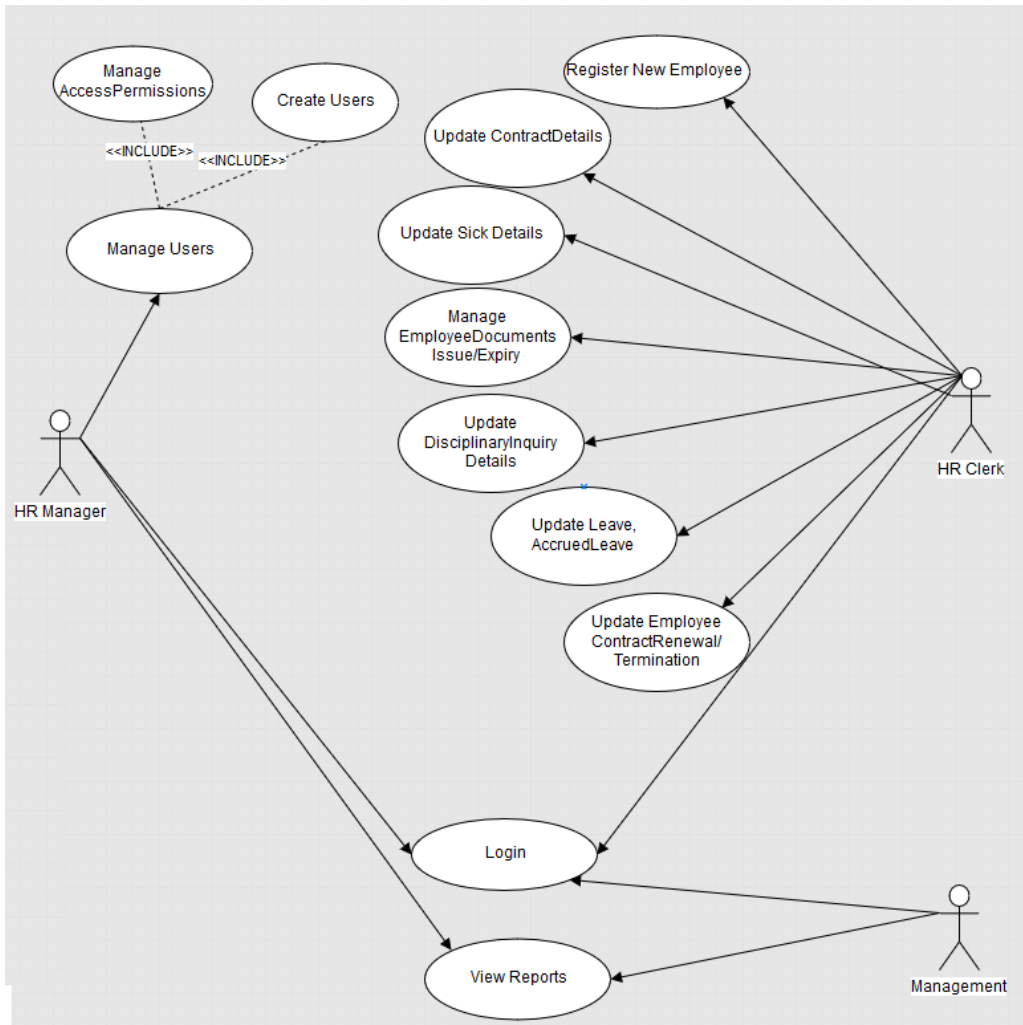


Figure 2.1: High Level Use Case Diagram for Current System

The existing legacy system has been in place for nearly two decades in GSCC. Though the system lacks nearly all the latest trends and technical aspects, features had been kept adding/amending as per growing and varying needs of the company. The changes in company policies and changes in government bodies had been incorporated into the system as they evolved.

In this aspect, following can be categorized as main business functions existing in the current system.

- **Employee Contract Maintenance Module**

Once an employee is registered, contract module is used to introduce all information pertaining to employees and organization agreement with respect to work.

- **Employee Disciplinary Record Maintenance Module**

This module assists in updating disciplinary records with details such as what caused the disciplinary action to take, what actions were taken, if employee was fined or so etc.

- **Employee Sick Details Maintenance Module**

Employee's sick records with period of sickness, medical clinic/hospital attended etc. is maintained here.

- **Employee Leave Earnings/Utilization Module**

This module covers recording of employee leave utilization. This option provides reviewing employees earned leave and eligibility to use leave. Leave types such as paid, unpaid, and accrued and various other types are handled in this module.

- **Issue/Expiry Document Maintenance Module**

Documents such as passport, Iqama (residence permit), labor card etc. that has issue and expiry are maintained in issue/expiry document module. This module assists in carrying out the renewal of documents.

- **Contract Renewal Module**

Employees in the company have varying contractual terms. As per company policy after the review of department manager, this module is used to record renewal of contracts for a similar period as of the existing or initial contract.

- **Service Termination Module**

Employee termination entry with details such as reason for termination, what type of benefits are paid are updated using the module.

- **Accrued Leave Maintenance**

Leaves other than monthly earned leaves are termed as accrued leave. When employee works on holidays, as per existing company policy, with the approval of HR and management such work days can be converted to leave earned days and maintained in accrued leave.

- **Reporting Module for all above Transactional Modules**

Almost all the above mentioned transactional modules has its counterpart in report. Users can use existing report to verify their entries and to output filtered reports.

2.4 Similar solutions with references

As like the saying goes, no two companies use similar HR system since the policies and procedures varies between them. Yet there exist generalized HR solutions attempting to resolve major areas used by majority of the companies. Such similar solutions that have similar features that GSCC existing software has are given below.

2.4.1 Paylite HRMS System

HRMS helps managing employee wise database, including their educational and professional profiles, skills, salary progression, assets tracking, important details for driving license, visa, work permit, passport etc. It features comprehensive and user-friendly leave and air-ticket management. It meets all regional requirements for UAE, Bahrain, Kuwait, Oman, Qatar and Saudi Arabia. Several organizations in Dubai, Abu Dhabi and Muscat are already enjoying its powerful and useful features. [2]

Paylite HRMS System was reviewed for its simple UI with concise layout of screens. This software immensely assisted in obtaining a cool color scheme and laying out basic employee transaction update screens. Figure 2.2 shows such an example with simple control layout.

Payroll Management > Leave Application > Add New Record >

Back to list

Leave Application

Application Date + 23/02/2015

Employee + Mitchell Johnson Look Up Clear

From Date + 16/03/2015

To Date + 18/03/2015

Full / Half Full Half

No of Days 3.00

Leave Type Annual Leave Compassionate Leave Death Leave Emergency Leave
 Examination Leave Extra Leave Haj Leave - Local Paternity Leave
 Sick Leave TEST2 TOIL Unpaid Leave

Reason

View Leave Status View TOILs Leave Calendar Post Leave Salary

Save

Figure 2.2: Paylite HRMS – Simple screen layout

The simple color scheme used in HRMS is shown below in Figure 2.3.

Employee Code	Name	Department	Designation	Active	History	Edit	Delete
CTU0001	Steve Waugh	Human Resource	HR Officer	Yes	🔍	✏️	✖️
CTU0002	Brett Lee	Marketing and Sales	Sales & Marketing Manager	No	🔍	✏️	✖️
CTU0003	Ricky Ponting	Marketing and Sales	Sales & Marketing Manager	Yes	🔍	✏️	✖️
CTU0004	Mark Jude Taylor	General Administration	Office Administrator	Yes	🔍	✏️	✖️
CTU0005	Justin Langer	General Administration	Heavy Vehicle Driver	Yes	🔍	✏️	✖️
CTU0006	Mark Waugh	Marketing and Sales	National Manager	Yes	🔍	✏️	✖️
CTU0007	Brad Hogg	Information Technology	Finance Director	No	🔍	✏️	✖️
CTU0008	Damien Martin	General Administration	Sales & Marketing Manager	Yes	🔍	✏️	✖️
CTU0009	Michle Clark	Finance	Finance Manager	Yes	🔍	✏️	✖️
CTU0013	Richard Hadly	Information Technology	Ast. IT Manager	Yes	🔍	✏️	✖️
CTU0014	Shaun Pollack	Marketing and Sales	Office Administrator	Yes	🔍	✏️	✖️
CTU0019	Mitchell Johnson	Information Technology	IT Manager	Yes	🔍	✏️	✖️
CTU0020	Peters Siddle	General Administration	Head of Budget	Yes	🔍	✏️	✖️
CTU0021	Doug Bollinger	Marketing and Sales	Global Business Development Director	Yes	🔍	✏️	✖️
CTU0022	Ben Hilfenhaus	Marketing and Sales	Project Director - Oil and Gas	Yes	🔍	✏️	✖️
CTU0024	Adam Craig Gilchrist	Finance	Finance Director	Yes	🔍	✏️	✖️
CTU0025	Shane Watson	Finance	Finance Manager	Yes	🔍	✏️	✖️
CTU0026	Glenn Maxwell	Accounts	Finance Manager	Yes	🔍	✏️	✖️

Figure 2.3: Paylite HRMS – Simple color scheme

2.4.2 ADEPT HRMS Solution

Though being a web based solution, ADEPT has evolved from employee management to strategic planning and resourcing. Some of the features given below in ADEPT has been identified that could be added in the GSCC HR system in future enhancements.

- Organizational Hierarchy
- Duty Roaster
- Salary Advance and Incentive Processing
- Company and Employee Document Handling
- Employee Recruitment

Figure 2.4 shows the organizational chart that is included in ADEPT HRMS system [3].



Figure 2.4: ADEPT HRMS – Organizational Chart

Figure 2.5 shows a simple grid list ADEPT HRMS that could be implemented in GCC HR.

The screenshot shows the 'Document Register' screen in ADEPT HRMS. It features a grid list of documents. The columns are: Document Type, Employee, Document Number, Date, Expiry Date, Custodian, Bin No, Issue Reason, Return Date, Remarks, and Document Status. The grid contains 15 rows of data, all with 'Passport' as the document type. The status for all entries is 'Good Con'. A 'Total Records: 31' indicator is shown at the bottom right of the grid. The interface includes a navigation bar at the top with icons for 'Trade License', 'Lease Agreement', 'Insurance', 'Passport', 'Visa', 'Driving License', 'Qualification', 'Emirates Card', 'Insurance Card', 'Labour Card', 'Health Card', 'Other Documents', and 'Stock Register'. A sidebar on the left has 'View' options for 'Document Register', 'Docs Available', and 'Docs Issued', along with search filters for 'Type', 'Document Type', and 'Document Number'. A 'Document Map' button is located at the bottom right.

Document Type	Employee	Document Number	Date	Expiry Date	Custodian	Bin No	Issue Reason	Return Date	Remarks	Document Status
Passport		M444000	04 Feb 2015	25 Nov 2018	SAAD(51)	Jordanian Pas...				Good Con
Passport		E69865460	22 Feb 2015	19 Dec 2018	SAAD(51)	Philippines Pa...				Good Con
Passport		M801274	22 Feb 2015	17 May 2019	SAAD(51)	Jordanian Pas...				Good Con
Passport		XX37916733	22 Feb 2015	06 Oct 2019	SAAD(51)	Philippines Pa...				Good Con
Passport		A06767779	22 Feb 2015	17 Apr 2019	SAAD(51)	Egyptain Pass...				Good Con
Passport		AC0178072	22 Feb 2015	27 Jun 2015	SAAD(51)	Pakistan Pas...				Good Con
Passport		XX0522419	22 Feb 2015	22 Sep 2019	SAAD(51)	Philippines Pa...				Good Con
Passport		AK2742022	22 Feb 2015	28 Feb 2016	SAAD(51)	Pakistan Pas...				Good Con
Passport		A00028516	22 Feb 2015	08 Feb 2015	SAAD(51)	Egyptain Pass...				Good Con
Passport		A00934900	22 Feb 2015	01 May 2018	SAAD(51)	Egyptain Pass...				Good Con
Passport		A00864255	22 Feb 2015	20 Jul 2016	SAAD(51)	Egyptain Pass...				Good Con
Passport		A02379495	22 Feb 2015	14 Aug 2017	SAAD(51)	Egyptain Pass...				Good Con
Passport		A03878721	22 Feb 2015	25 Jun 2018	SAAD(51)	Egyptain Pass...				Good Con
Passport		RL2696741	22 Feb 2015	26 Aug 2015	SAAD(51)	Labnese Pass...				Good Con
Passport		EP1699373	22 Feb 2015	04 Oct 2016	SAAD(51)	Ethiopiens Pa...				Good Con

Figure 2.5: ADEPT HRMS – Simple Grid List

2.4 Requirements

2.4.1 Functional Requirements

In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs. [4]

Below given have been identified and agreed by HR as functionalities that is required in the upgraded new system. Record update/transaction modules has to provide same features as in the existing legacy application.

- Employee registering and record maintenance module
- Employee Contract Maintenance Module
- Employee Disciplinary Record Maintenance Module
- Employee Sick Details Maintenance Module
- Employee Leave Earnings/Utilization Module
- Issue/Expiry Document Maintenance Module
- Contract Renewal Module
- Service Termination Module
- Accrued Leave Maintenance
- All reports should be producible in Spreadsheet for further review and analysis.
- Users should be able select what fields to be included in output
- For important events like document expiry, contract expiry etc., system should send a notification to the group of users as defined by HR department.

2.4.1 Non-Functional Requirements

In systems engineering and requirements engineering, a non-functional requirement (NFR) is a requirement that specifies criteria that can be used to judge the operation of a system, rather than specific behaviors. They are contrasted with functional requirements that define specific behavior or functions. [5]

Below specified are non-functional requirements of the new HR system.

- **User Friendliness:** System need to be developed with GUIs to provide better interaction with users compared to existing text based UI.
- **Security:** System should protect information security at a high level.
- **Performance:** The functions in the system should work without causing delays to users.

- **Scalability:** System has to support the addition of new features and should support the growing needs of the organization.

2.4 Development Methodology

Upon comparing the options in methodologies, such as Waterfall Methodology, Rational Unified Process RUP was chosen as the methodology for this project. The reasons for the selection RUP is described below.

2.4.1 RUP – Rational Unified Process

The Rational Unified Process (RUP) is an iterative software development process framework created by the Rational Software Corporation, a division of IBM since 2003. RUP is not a single concrete prescriptive process, but rather an adaptable process framework, intended to be tailored by the development organizations and software project teams that will select the elements of the process that are appropriate for their needs [6]

RUP consists of four important phases named Inception, Elaboration, Construction and Transition. Figure 2.6 shows the stages in RUP graphically [7].

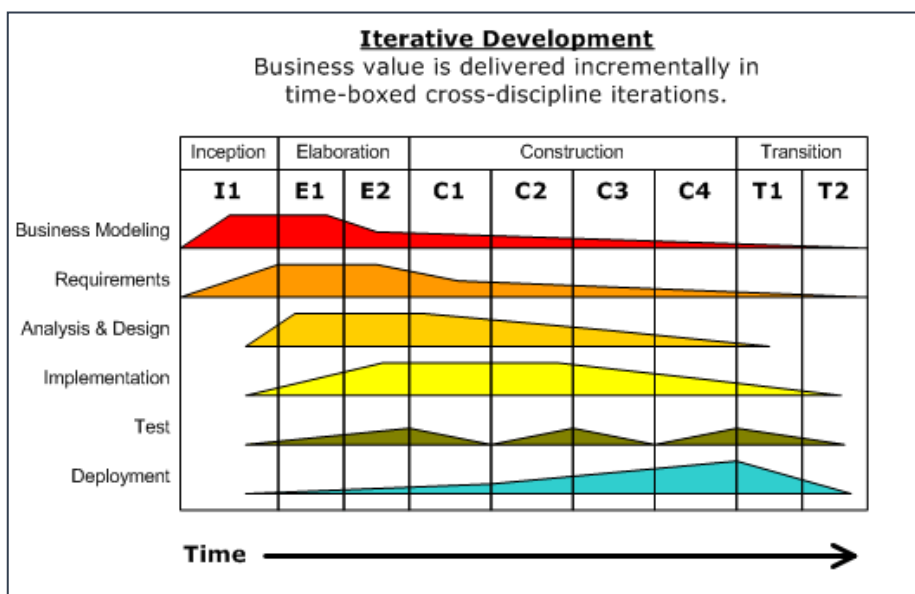


Figure 2.6: Stages in Rational Unified Process

Figure 2.7 below displays the phases and lifecycle of RUP [8].

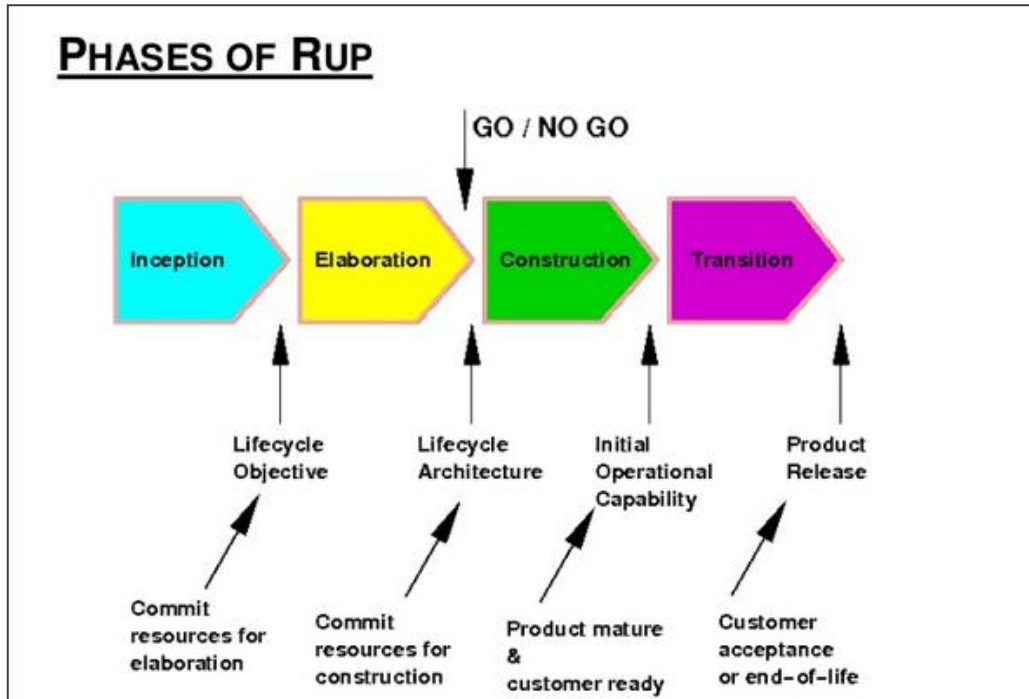


Figure 2.7: Phases of Rational Unified Process

2.4.2 Benefits in using RUP

RUP has been chosen since it allows below given benefits compared to Waterfall methodology.

- RUP has relatively high integration support with Object Oriented Design.
- RUP has incremental and iterative designing. Due to this feature, early releases can be done with limited features.
- Unlike Waterfall Methodology, RUP has got easy change management
- The risks can be identified earlier and as a result resolutions can be provided earlier.
- Due to iterative error correction routines, eventually RUP methodology helps produce a robust system.

With above mentioned advantages, this project was developed using RUP. Since new features were included while moving the programs from legacy application, using RUP with iterative development proved to be very effective.

Chapter 3 - System Design

3.1 Introduction

System design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering. [9]

3.2 Alternative Solutions

3.2.1 Improve the existing text based system

This could perhaps be considered the easiest as system is already running in this environment. But this would not resolve the system being ported to GUI environment with user friendly interfaces, spreadsheet outputs etc. since the prime objective is to port the application with rich features that exist in modern applications.

3.2.2 Purchase packaged software

Human resource system requirements are so unique to each company and GSCC is not having any exemption in this. This perception has made replacing the existing software with a purchased software is quite tough. A software named Prisma was evaluated but user evaluations proved the system wasn't equipped adequately enough to cater all needs.

3.2.3 Develop a stand-alone system

The existing software despite being legacy, it had the network support through telnet protocols. Several nodes of user workstations are connected in this existing setup. Therefore, replacing it with a stand-alone system is not feasible.

3.2.4 Develop a network based system

A network based system has been identified as the suitable system for GSCC HR System. With development time in consideration, client/server windows based solution as the most suitable system. All senior staffs in GSCC are privileged with VPN access to company network and therefore the system could even remotely be accessed when required. Since developed in-house, it would further allow customizations and enhancements with ease.

3.3 Candidate matrix with alternate Solutions

Table 3.1 below shows candidate matrix evaluating all four targeted systems.

System Characteristics	Maintain Existing System (A)	Purchase Package Software (B)	Develop Standalone (C)	Develop Network-based desktop app (D)	Develop Network-based web app (E)
Benefits	Running application	Quick implementation	Customizable	Customizable	Customizable
Achieving main goals (GUI, spreadsheets output, alerts.)	Not feasible due to text based screen.	Not applicable	Feasible	Feasible	Feasible
Customizations	Possible but still with text-based limitation	Feasible but costly	Feasible	Feasible	Feasible
Servers	Microsoft Server 2003 or above	Not applicable	Microsoft Server 2003 or above	Microsoft Server 2003 or above	Microsoft Server 2003 or above
Clients	Any OS supporting telnet	Not applicable	Desktop windows app	Desktop windows app	Browser
Data processing method	Client-Server with online & batch process.	Not applicable	Client-Server with online & batch process.	Client-Server with online & batch process.	Client-Server with online & batch process.
Software tools needed	Interysystem s Caché	Not applicable	Visual Studio Community Edition 2015	Visual Studio Community Edition 2015	Visual Studio Community Edition 2015
Development Schedule	High time consuming	Not applicable	Less time	Less time	More time
Customization Time-frame	High time consuming	Not applicable	Less time	Less time	More time
Inputs devices	Keyboard Only	Not applicable	Keyboard, Mouse, Tablet etc.	Keyboard, Mouse, Tablet etc.	Keyboard, Mouse, Tablet etc.
Output devices	Dot matrix printers primarily and text files	Not applicable	Laser printers, Excel spreadsheets etc.	Laser printers, Excel spreadsheets etc.	Laser printers, Excel spreadsheets etc.
Data Storage	Caché Database	Not applicable	Microsoft SQL Server 2012	Microsoft SQL Server 2012	Microsoft SQL Server 2012
Interfaces	Telnet based clients only	Desktop/Web app	Desktop app	Desktop app	Web app

Table 3.1: Candidate Matrix

3.3 Feasibility analysis matrix

This matrix would reveal how well each system could stand with respect to all feasibilities that are considered while developing a system. Table 3.2 provides the comparison and related scores with various feasibilities.

Feasibility Criteria	Weight	Maintain Existing System (A)	Purchase Package Software (B)	Develop Standalone (C)	Develop Network-based desktop app (D)	Develop Network-based web app (E)
Operational feasibility. Ability to support all main goals (GUI, spreadsheets output, alerts.)	20%	Due to client access via telnet protocol and text-only based UI, most of the features are not achievable. Score : 30	Can be implemented quickly but extremely difficult to ensure all existing features are supported. Score : 50	This system support development all required functionality but lacks network accessibility in multi user environment. Score : 75	This system support development of all required functional and non-functional requirements . Score : 100	This system support development of all required functional and nonfunctional requirements. Score : 100
Technical feasibility	25%	Required technology is not possessed in this system. Score : 25	Many of the solutions deliver a portion of the requirements . Score : 50	By developing custom applications, technically features can be achieved. Score : 100	By developing custom applications, technically features can be achieved. Score : 100	By developing custom applications, technically features can be achieved. Score : 100
Cultural or political feasibility	10%	There is a strong willingness to change this app as outdated. Score : 40	Management prefers quick implementation but wants the maintenance to be easy. Score : 65	Management disapproves this. Data & program cannot be just in one machine. Score : 60	Users expect rich features in app and in output. Therefore, desktop app is preferred. Score : 90	Relative less rich compared to desktop but users can adopt easily. Score : 75
Schedule feasibility	20%	App already exists. Changes can be done fast. Score : 100	Approx 2 months for implementation. Score : 50	2-3 months for development. Score : 80	2-3 months for development . Score : 80	4-5 months for development. Score : 60
Economic feasibility	15%	Development not required. Extremely	Estimated development cost is 3,000,000	Estimated development cost is 1,000,000	Estimated development cost is 1,000,000	Estimated development cost is 1,500,000

		high maintenance cost. Score : 50	rupees (75,000 Saudi riyals) Score : 10	rupees (25,000 Saudi riyals) Score : 60	rupees (25,000 Saudi riyals) Score : 60	rupees (37,000 Saudi riyals) Score : 40
Legal feasibility	10%	No issues in the present context. Score : 100	No issues in the present context. Score : 100	No issues in the present context. Score : 100	No issues in the present context. Score : 100	No issues in the present context. Score : 100
Weighted Score	100%	58	54	79	88	79

Table 3.2: Feasibility Analysis Matrix

Considering all above characteristics in candidate matrix and feasibility matrix, the decision is derived to implement a network based desktop application.

3.4 Design Methodology

The use of modeling to define and analyze the requirements necessary for success of a system. Object-oriented analysis is a process that groups items that interact with one another, typically by class, data or behavior, to create a model that accurately represents the intended purpose of the system as a whole [10].

For this project RUP (Rational Unified Process) has been chosen. The important reasons are:

- Iterative Software Development Process
- Object Oriented Methodology support

3.4.1 Design Diagrams

Unified Modeling Language (UML) is used in object oriented designing. UML assist design how the objects used in the system would be represented.

3.4.2 System Software

GSCC has invested in its existing infrastructure with Microsoft operating system software for Domain Controllers, database servers, clients etc... As such, the development of the new GSCC HR system also is developed on Microsoft OS.

3.5 High-level use case diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors) [11].

The high-level use case diagram used in the development GSCC HR system is given below in Figure 3.1.

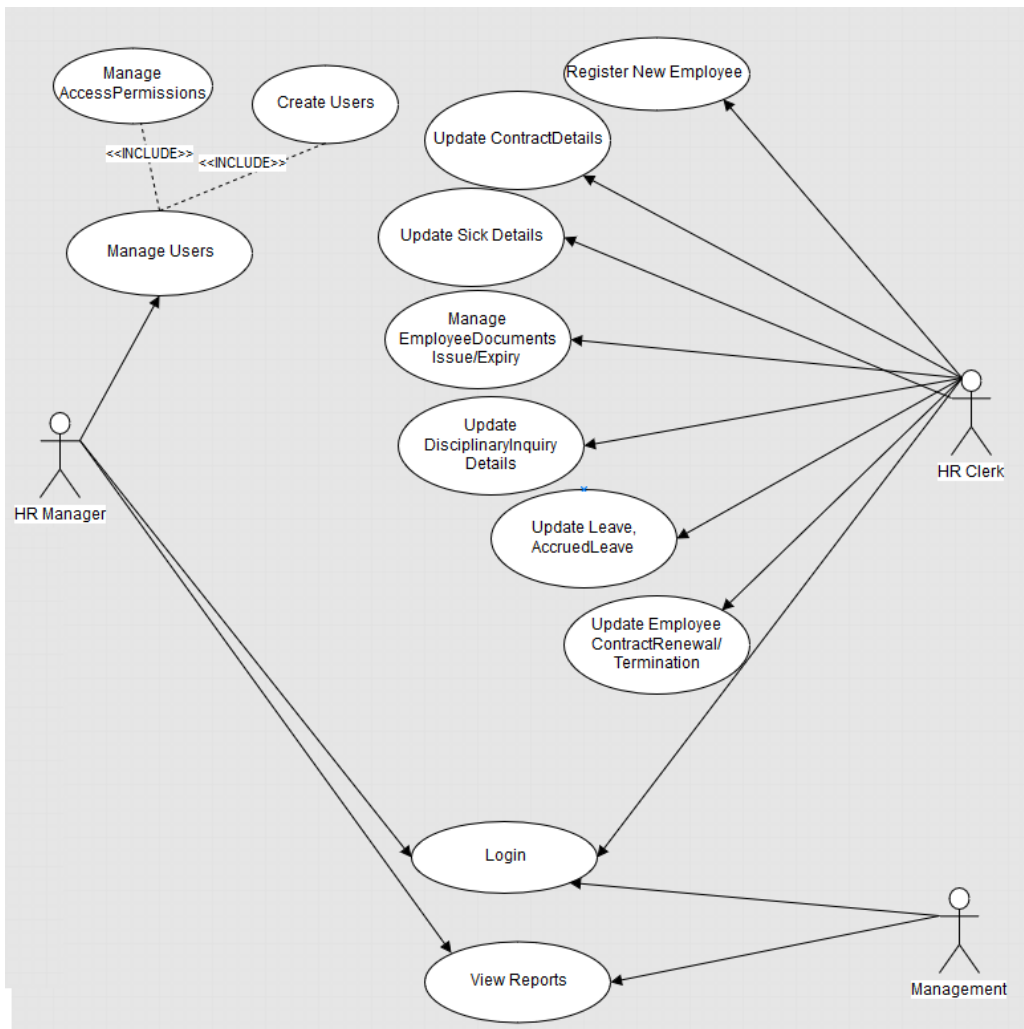


Figure 3.1: High Level Use case Diagram

3.6 Class Diagram

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects [12].

Figure 3.2 shows the class diagram designed for GSCC HR project.

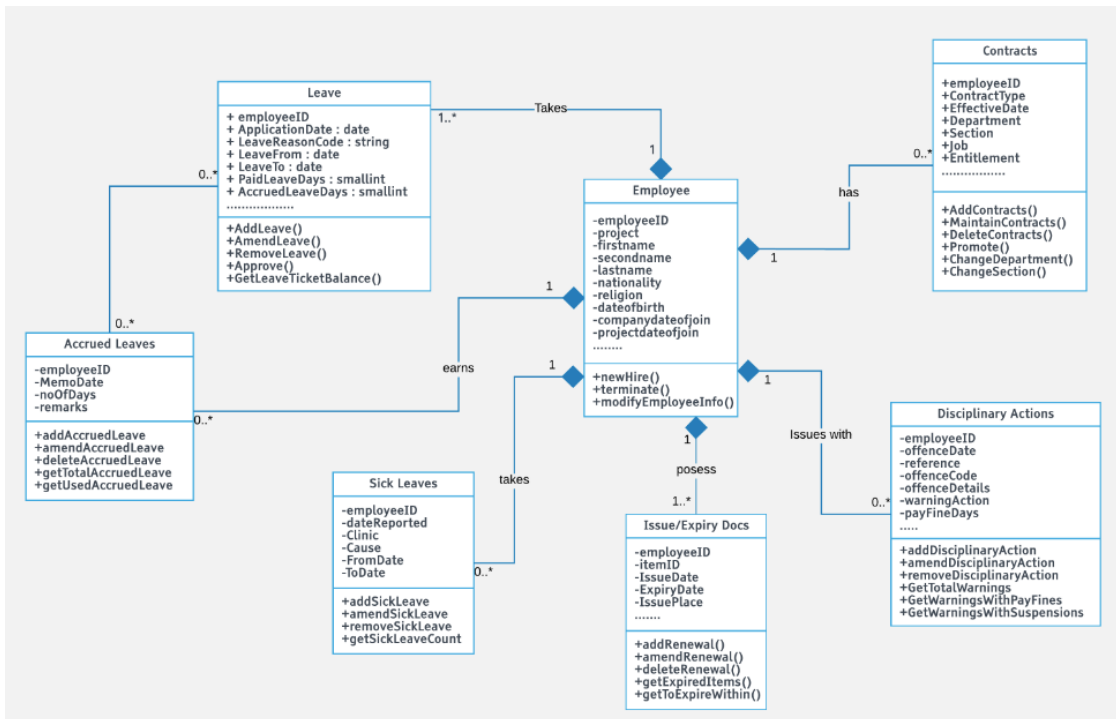


Figure 3.2: Design of Class Diagram

3.7 Sequence Diagram

It is a construct of a message sequence chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario [13].

A Sequence diagram showing the steps in updating the contract details of an employee is given in Figure 3.3 below.

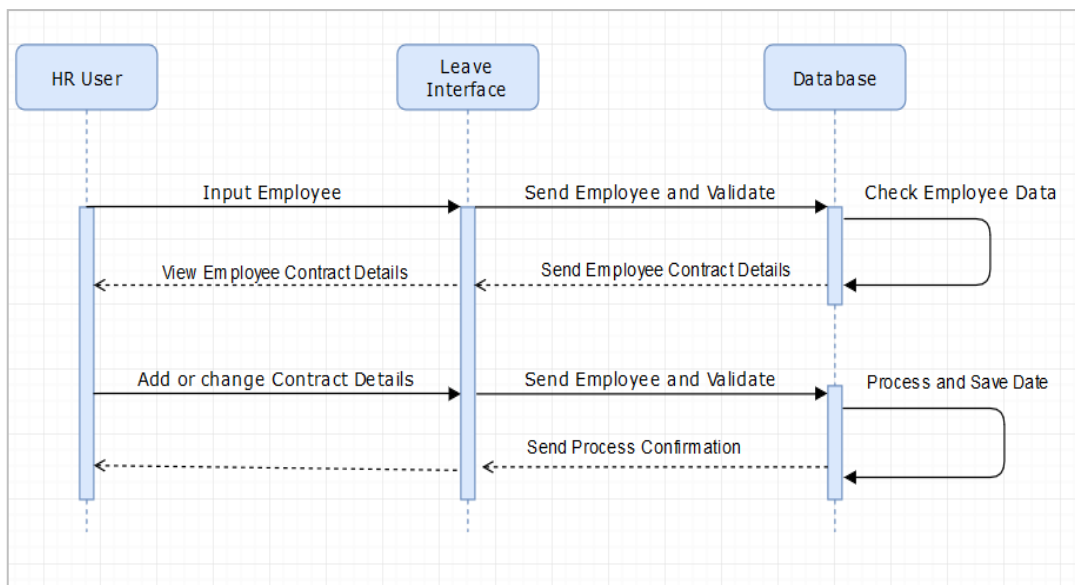


Figure 3.3: Sequence Chart for Contract Details Update

3.8 Entity Relationship Diagram

An entity-relationship (ER) diagram is a graphical representation of entities and their relationships to each other, typically used in computing in regard to the organization of data within databases or information systems. An entity is a piece of data-an object or concept about which data is stored [14].

Figure 3.4 shows the ER diagram with all entities used in GSCC HR System.

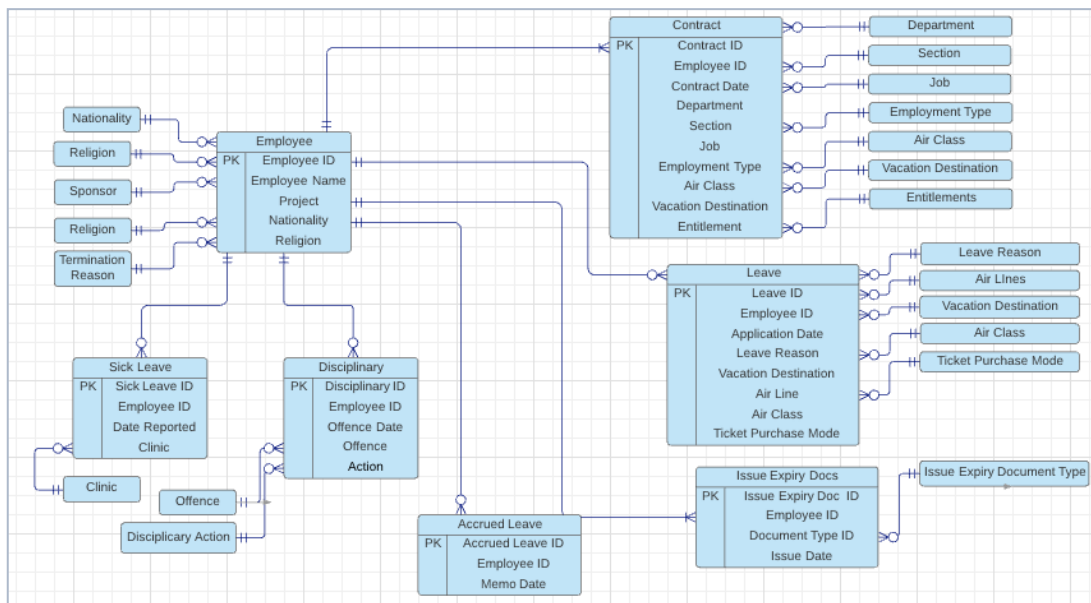


Figure 3.4: ER Diagram

3.9 Database Design

Database design is the process of producing a detailed data model of database [15]. Database design is an important part of system design. A well-designed database avoids data redundancy. Further, maintaining the data consistency also is well achieved by a good database design. Normalization rules are applied in designing the database to ensure such data integrity and consistency.

Figure 3.5 displays the database design diagrams developed for this system.

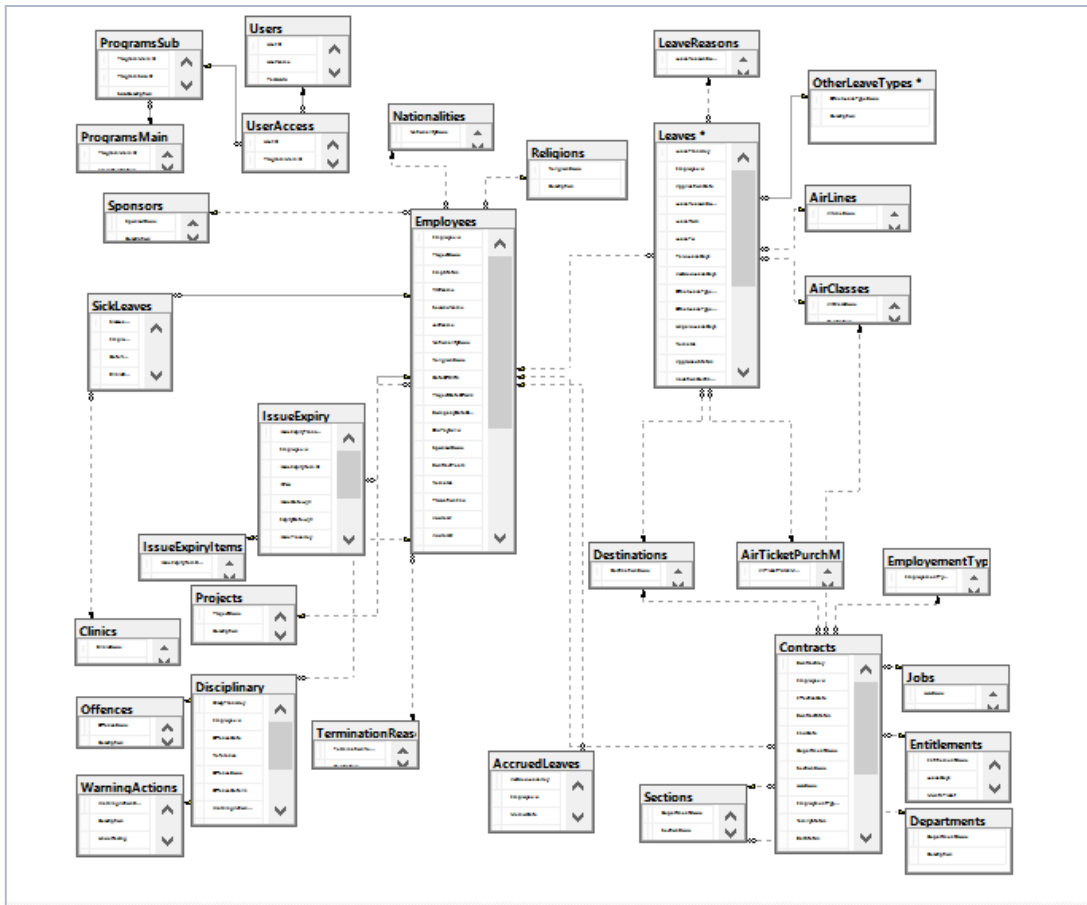


Figure 3.5: Database Diagram

3.9 User Interface Design

One of the very important parts in designing a system is the design of user interfaces. User acceptance for the system greatly depends on how the interfaces have been designed. A poorly designed user interface can easily lose its value even if contains loads of complex business functions internally.

3.9.1 Good user interface design principles

Following are some of the aspects that need to be considered while designing a good user interface.

- Consistency -The look and function of all the interfaces should have consistency. Providing different designs and looks at each screen could easily confuse the user.
- Familiarity – It should be ensured the designs are done keeping the user's participation in mind.
- Recoverability – If any errors occur, the interface design should help users recover from such situations.
- User Guidance – When errors occur, system has to provide relevant meaningful messages.
- User Diversity – Based on the type of system users, system has to provide suitable interactions.

3.9.2 Designs used in the system to provide better interactions

In order to provide satisfying experience to the users, below implementations has been included in the system.

- To provide better look and feel, a limited set of colours have been used. Controls are placed spaciouly without clutter.

Figure 3.6 shows an example of an entry screen that has limited set of colours.

The screenshot shows a web form titled "Employees" with a blue header bar. The form is organized into two columns of input fields. The left column includes fields for Employee ID (2400224), Project (Jeddah North Terminal), First Name (Mohamed), Second Name (Irshad), Last Name (Mohideen), Nationality (Sri Lanka), Religion (Islam), Date of Birth (02/05/1969), Project Date of Join (16/09/2011), Company Date of Join (14/09/2011), Old Payroll No (11111), Sponsor (GSCC), Contract Years (3), and Remarks (new remarks updated for testing). The right column includes fields for Probation End (16/11/2011), Address (aaa, bbb, ccc, eee), Contact Phone (0509639830), Name of Father (AA), Name of Mother (BB), Name of Spouse (CC), Summer Student (checkbox), and Marine Trainee (checkbox). At the bottom right, there is a "button1" label. At the bottom left, there are four buttons: "Query", "New", "Save", and "Delete".

Figure 3.6: Design of a form with limited set of colours

- Visual clues are provided when user enters invalid characters in the input. For example, when user tries to enter any alpha characters in date field, system shows help with a pop up balloon message as shown below in Figure 3.7.

The screenshot shows three date input fields: "Date of Birth", "Project Date of Join", and "Company Date of Join". Each field contains a date with slashes and underscores (e.g., / / _ _). A light blue error message box is overlaid on the right side of the fields, containing the text "Invalid Input" in bold and "Please input only digits." below it.

Figure 3.7: Invalid Input Message for wrong character input

- Messages are instantly displayed, and focus is kept in the fields when user enters invalid value for the input. For example, if user enters and invalid date that does not exist, then system shows message as shown below in Figure 3.8 to input a valid date.

The screenshot shows three date input fields: "Date of Birth", "Project Date of Join", and "Company Date of Join". The "Date of Birth" field contains "01/01/1990". The "Project Date of Join" field contains "15/15/2005". The "Company Date of Join" field contains "/ / _ _". A light blue error message box is overlaid on the right side of the "Project Date of Join" field, containing the text "Invalid Date Value" in bold and "Please input a valid date" below it.

Figure 3.8: Error Message displaying for wrong date input

- When mandatory inputs are not entered and user attempts to save the transaction, as given in Figure 3.9 below, system shows a message to input the details.

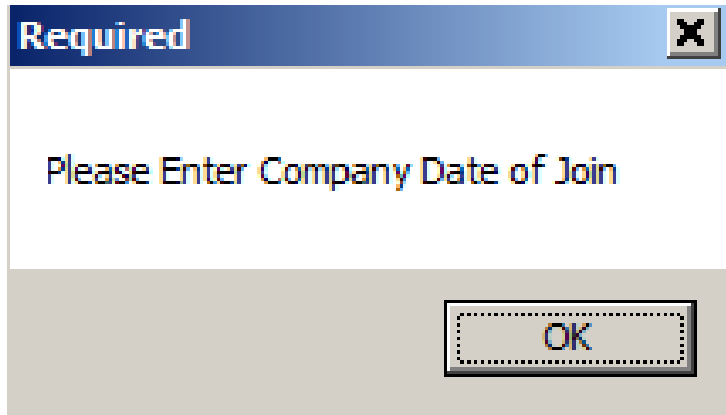


Figure 3.9: Message that comes when mandatory field is blank

- Prior to changing any critical information especially prior to deleting information, system prompts the user to confirm if user would like to proceed with the desired deletion. Also when user closes a form, input prompts up asking user whether details should be saved or not to avoid entered data getting lost. Such a prompt is shown below in Figure 3.10.

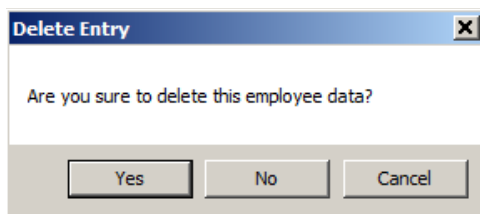


Figure 3.10: Message that takes user confirmation

- When transactions are completed, system shows informative messages to confirm that the desired transaction has been carried out. Figure 3.11 shows an example of such an instance after inserting a record.

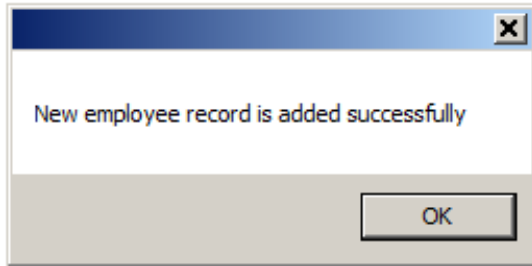


Figure 3.11: Confirmation message for record insert

Figure 3.12 below displays a confirmation message after a record update.

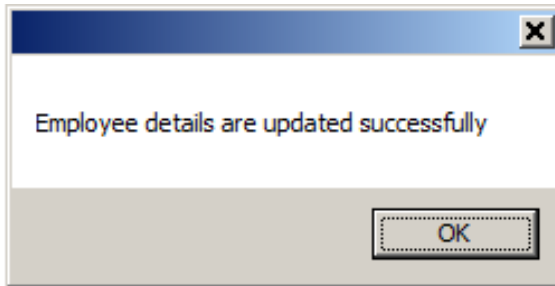


Figure 3.12: Confirmation message for record update

- Keyboard shortcuts have been added to ease the interaction, such to add record, to save record, to query record etc. so that user does not have to switch between mouse and keyboard inputs. Examples are function key F10 can be pressed to save the records and F7 key is for querying the record.

3.9.3 Login Screen

User starts their actions with the login screen as shown below in Figure 3.13. This screen helps authenticate user and gets into the main menu.



Figure 3.13: Login Screen

3.9.4 Main Menu

The main menu, shown in Figure 3.14, helps user select the required program and open them.

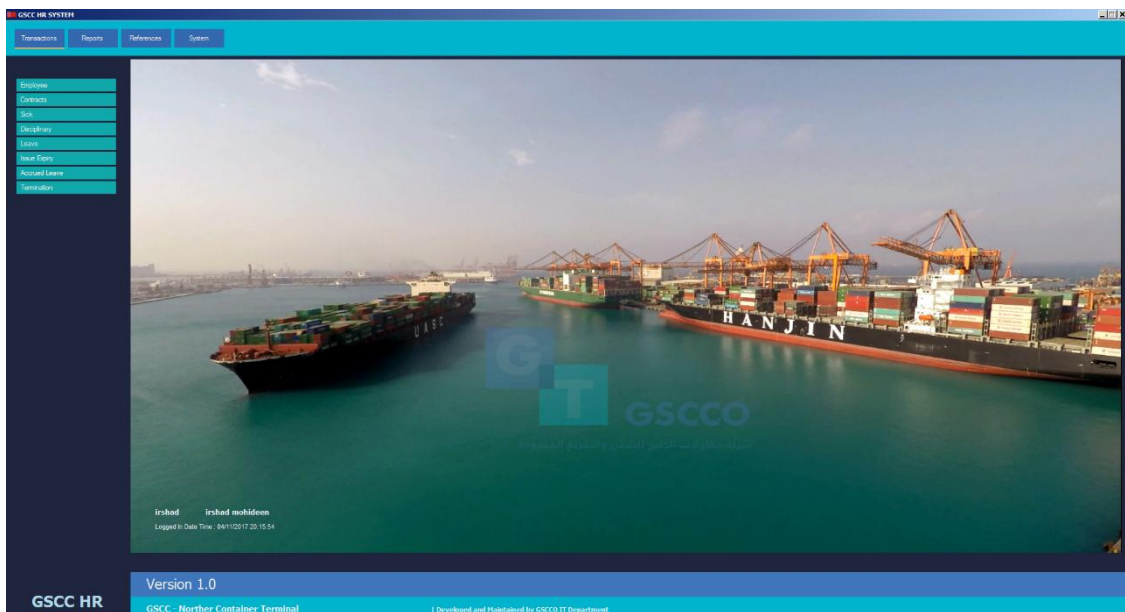


Figure 3.14: Main Menu

3.9.5 User Access Management Form

This form is used to create new users for the GSCC HR system. This form further assists in updating the access for the screens. This form is generally expected to be used by the Super user preferably like HR manager. Permission for update, reports or reference can be categorically given using this program. Figure 3.15 displays the layout of the user management form.

The screenshot shows a window titled "User Access Management" with the following fields and controls:

- User ID:
- Name of Employee:
- Password:
- Status:
- Select/Unselect All

	Menu Name	Option Name	Access
▶	Transactions	Employee	<input checked="" type="checkbox"/>
	Transactions	Contracts	<input checked="" type="checkbox"/>
	Transactions	Sick	<input checked="" type="checkbox"/>
	Transactions	Disciplinary	<input checked="" type="checkbox"/>
	Transactions	Leave	<input checked="" type="checkbox"/>
	Transactions	Issue Expiry	<input checked="" type="checkbox"/>
	Transactions	Accrued Leave	<input checked="" type="checkbox"/>
	Transactions	Termination	<input checked="" type="checkbox"/>
	Reports	Manpower Summary	<input checked="" type="checkbox"/>
	Reports	Leave Report	<input checked="" type="checkbox"/>
	Reports	Active Employee List	<input checked="" type="checkbox"/>

Buttons: Query, New, Save, Delete

Figure 3.15: User Access Management Form

3.9.6 Employee Form

Figure 3.16 displays the employee form used to update new hires as well to maintain employee basic information.

The screenshot shows a form titled "Employees" with the following fields and values:

- Employee ID: 2400224
- Project: NCT
- First Name: Bilal
- Second Name: Hani
- Last Name: Zaidh
- Nationality: SRILANKAN
- Religion: ISLAM
- Date of Birth: 02/07/1975
- Project Date of Join: 15/07/2011
- Company Date of Join: 15/07/2011
- Old Payroll No: (empty)
- Sponsor: GSCC
- Contract Years: 3
- Remarks: TEST
- Probation End: 11/10/2011
- Address: bani malik, jedah, saudi arabia
- Contact Phone: 0509693830
- Name of Father: (empty)
- Name of Mother: (empty)
- Name of Spouse: (empty)
- Summer Student:
- Marine Trainee:

Figure 3.16: Employees Form

3.9.7 Contracts Form

Figure 3.17 shows the contract form that is used to create and maintain employee's contracts.

The screenshot shows a form titled "Contracts" with the following fields and values:

- Employee ID: 2400224
- Name: BILAL HANI ZAIDH
- Department: COMPUTER
- Project Date of Join: 16/09/2011
- Company Date of Join: 14/09/2011

Effective Date	Contract Status	End Date	Department Name	Section Name	Job name	Employment Type	Family Status	Car Status	Air Class	Vacation Destination
16/09/2011	NEW	15/09/2014	COMPUTER	PROGRAMMING	CLERK	CONTRACTUAL	X	X	ECONOMY	COLOMBO

Figure 3.17: Contract Form showing contracts in a table

Contract detail shown in below Figure 3.18, appears upon double clicking any of the row in the contracts grid.

Contracts			
Employee ID	<input type="text" value="2400224"/>	Project Date of Join	<input type="text" value="16/09/2011"/>
Name	<input type="text" value="BILAL HANI ZAI DH"/>	Company Date of Join	<input type="text" value="14/09/2011"/>
Department	<input type="text" value="COMPUTER"/>		

Effective Date	<input type="text" value="16/09/2011"/>	Air Class	<input type="text" value="ECONOMY"/>
Contract Status	<input type="text" value="NEW"/>	Vacation Destination	<input type="text" value="COLOMBO"/>
End Date	<input type="text" value="15/09/2014"/>	Entitlement Code	<input type="text" value="30 DAYS LEAVE + 1 YEAR TICKET"/>
Department	<input type="text" value="COMPUTER"/>	Leave Per Annum	<input type="text" value="30"/>
Section	<input type="text" value="PROGRAMMING"/>	Months Per Ticket	<input type="text" value="12"/>
Job	<input type="text" value="CLERK"/>	Remarks	<input type="text" value="CHECK"/>
Employment Type	<input type="text" value="CONTRACTUAL"/>		
Family Status	<input checked="" type="checkbox"/>		
Car Status	<input type="checkbox"/>		

Figure 3.18: Contract form showing details of a sing contract

All design requirements have been fulfilled as detailed above. This paved way to move to next phase where the implementation work related to hardware and software are done.

Chapter 4 - Implementation

4.1 Introduction

The design that had been done so far is transformed into computer executable code in the implementation phase. Even though the weight of coding in the system development process is comparatively less, a good design can easily be spoiled by choosing an inappropriate coding environment. Thus choosing better implementation procedures is very important for the success of the system.

4.2 Implementation Environment

The implementation environment can be classified in two areas, hardware and software. The details of these environments are given below.

4.2.1 Hardware Environment

- Requirement for Server
 - Processor : Intel® Xeon CPU E5-2680 V2
 - RAM : 8 GB
 - Hard Disk : 500 GB
- Requirement for Client
 - Processor : Intel® Core™ i5 CPU 3.3 GHz
 - RAM : 4 GB
 - Hard Disk : 500 GB

4.2.2 Software Environment

- Requirement for Server
 - Operating System : Windows Server 2012 or above
 - Database : Microsoft SQL 2012 or above
- Requirement for Client
 - Operating System : Windows 7 Professional
 - .Net Framework : Version 3.5
 - Reporting Tool : Crystal Reports
 - Office Client : Microsoft Office 2007

4.3 Development Tools

Below development tools have been used in developing GSCC HR system.

4.3.1 Microsoft Visual Studio 2012/2015

This integrated development tool provides great deal of benefits in increasing developer productivity. Some of the main features are highlighted as follows

- Error highlighting as you code – As soon as a line of code with errors is written, visual studio highlights it with squiggly lines catching the attention of developer.
- IntelliSense – This feature drops down the possible inputs when the first few letters are entered.
- Fill all references – Visual studio displays all the code lines that refers the element right clicked. Provides developer the feature to spot all places of reference in one go.

4.3.2 C# Language

C# is an object oriented programming language from Microsoft with rich development features. This project is developed using C# language.

4.3.3 Crystal Reports

A product from SAP can be integrated with visual studio IDE. Crystal report for visual studio, which is provided free, is widely used in developing reports in applications. A couple of management reports has been done using this software.

4.3.4 Microsoft Office Excel

Since GSCC HR department requires the ability to output all the reports to excel, Excel runtime objects is used as object in the visual studio.

4.4 Platform Dependence

GSCC has its IT network with Microsoft Windows based servers and work stations. Therefore this project runs only on Microsoft windows based platforms.

4.5 Data Import from Legacy Application

In order to port the data from legacy application, all data in the previous app were exported to delimited text files. Delimited files were later imported to HR database using a custom build .net application.

The application reads the files available in the given location and writes the data identified in the connection string. Figure 4.1 shows the screen that appears indicating the status of data loading.

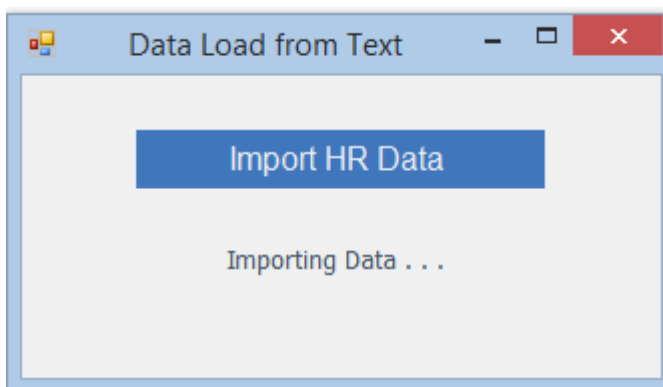


Figure 4.1: Data Load from Text file to HR Database

Details of the code developed to import data have been further discussed in Appendix F.

4.6 Major Code Structures

4.6.1 Connection to the database

Below code segment is used to connect to the SQL database. The connection string required is provided in app.config file.

```
// Connect to the SQL server
public static void connectDb(SqlConnection sqlDb)
{
    string connectionString = getConnectionString();
    sqlDb.Close();
    sqlDb.ConnectionString = connectionString;
    sqlDb.Open();
}
```

4.6.2 Login with password encryption

Below code is executed to validate if the user and password entered by user matches what is saved in DB encrypted.

```
private string validateLogin(string username, string password)
{
    // get the encrypted password
    string encPassword = gsCommon.Encrypt(password,gsCommon.EnKey) +
    "";
    string validateResult = "";

    try
    {
        using (SqlConnection myConnection = new SqlConnection())
        {
            gsCommon.connectDb(myConnection);
            // validate if the user and encrypted password input
            matches with the one in database
            SqlCommand myCommand = myConnection.CreateCommand();
            SqlDataReader myDataReader;
            string myString = "select userid from users where userid='"
            + username + "' and password='" + encPassword + "'";
            myCommand.CommandText = myString;
            myDataReader = myCommand.ExecuteReader();
            if (!myDataReader.Read())
            {
                validateResult= "Invalid user or password";
            }
            myDataReader.Close();
            return validateResult;
        }
    }
    catch (Exception ex)
    {
        return ex.Message;
    }
}
```


4.6.3 Open the form as per the name defined in

Below code is executed to open a form by passing the form's name instead of form object. The form name comes from the "ProgramsSub" table and is saved in the button's tag property. Upon user clicking the button, the tag property is read and Activator.CreateInstance() method is called to open the form dynamically.

```
try
{
    Button button = (System.Windows.Forms.Button)sender;
    subButtonTag = button.Tag.ToString();
    string formToCall="";
    formToCall = gsCommon.getDbValue("select FormName from ProgramsSub
where ProgramMainID = " + mainButtonTag + " and ProgramSubID = " +
subButtonTag + "");

    if (formToCall != "")
    {
        var form =
(Form)Activator.CreateInstance(Type.GetType("GscchRApp." +
formToCall));
        form.Tag = "" + mainButtonTag + "," + subButtonTag + "";
        form.ShowInTaskbar = false;
        form.Show(this);
    }
    else
    {
        MessageBox.Show("Form name is not defined, Please contact IT -
Application Team for Support !", "Form Not Defined",
MessageBoxButtons.OK, MessageBoxIcon.Warning);
    }
}
catch (Exception ex)
{
    MessageBox.Show("Error : " + ex.Message + "", "Error Infor",
MessageBoxButtons.OK, MessageBoxIcon.Error);
}
```

4.6.4 Get a single field value from a table

Often it requires getting a single value from table passing SQL statements. This functionality is added in the `gsCommon.getDbValue` function.

```
public static string getDbValue(string sql)
{
    string result = "";
    object ret;
    try
    {
        using (SqlConnection sqlCon = new SqlConnection())
        {
            gsCommon.connectDb(sqlCon);

            SqlCommand myCommand = sqlCon.CreateCommand();
            myCommand.CommandText = sql;
            {
                ret = myCommand.ExecuteScalar();
                if (ret != null)
                    result = ret.ToString();
                else
                {
                    result = "";
                }
            }
        }
        return result;
    }
}
```

4.6.5 Get a table as a result

When a set of values is required or rows of data is required, `getDbTable` function is used.

```
public static DataTable getDbTable(string sql)
{
    DataTable dt = new DataTable();
    try
    {
        using (SqlConnection sqlCon = new SqlConnection())
        {
            gsCommon.connectDb(sqlCon);

            using (SqlDataAdapter reader = new SqlDataAdapter(sql,
sqlCon))
            {
                dt.Clear();
                reader.Fill(dt);
                reader.Dispose();
                return dt;
            }
        }
    }
}
```

4.6.6 Class structure for objects with database transactions

Below code structure, shown in Figure 4.2, displays the layout used to define classes that interacts with database transaction updates. Primarily it provides details about the instance being queried, in this case the details of Employee. A further carries out database transactions such as insert, update and delete.

```
namespace GscCHrApp
{
    internal class Employee
    {
        // constructor
        internal Employee(string EmpId)...

        // class attributes
        class attributes

        // check if
        internal bool EmployeeExists()...

        // Extract employe information
        internal void GetEmployeeInfo()...

        // Get brief info about employee.
        internal void GetBriefEmployeeInfo()...

        // Insert new employee
        internal bool NewHire()...

        // update employee info
        internal bool UpdateEmployee()...

        // delete employee details
        internal bool DeleteEmployee()...
    }
}
```

Figure 4.2 : Class structure used for transactional objects

4.6.7 Class to call description of reference tables

Below code shows a portion of gsReference class that helps supplying the description of reference tables when the code is passed.

```
public class gsReferences
{
    public static string getProjectDescription(string code)
    {
        return gsCommon.getDbValue("select description from
projects where projectcode='" + code + "'");
    }

    public static string getNationalityDescription(string code)
    {
        return gsCommon.getDbValue("select Description from
Nationalities where NationalityCode='" + code + "'");
    }

    public static string ReligionDescription(string code)
    {
        return gsCommon.getDbValue("select Description from
Religions where ReligionCode='" + code + "'");
    }

    public static string getSponsorDescription(string code)
    {
        return gsCommon.getDbValue("select Description from
Sponsors where SponsorCode='" + code + "'");
    }

    public static string getDepartmentDescription(string code)
    {
        return gsCommon.getDbValue("select Description from
Departments where DepartmentCode='" + code + "'");
    }

    public static string getSectionDescription(string
deptcode,string sectcode)
    {
        return gsCommon.getDbValue("select Description from
Sections where DepartmentCode='" + deptcode + "' and sectioncode='" +
sectcode + "'");
    }

    public static string getJobDescription(string code)
    {
        return gsCommon.getDbValue("select Description from Jobs
where JobCode='" + code + "'");
    }

    public static string getEmployemntTypeDescription(string code)
    {
        return gsCommon.getDbValue("select Description from
EmployementTypes where EmploymentTypeCode='" + code + "'");
    }
}
```

4.7 Existing Re-usable Codes

Below existing code listing were taken from online resources

4.7.1 Encrypt Function

Below code encrypts a given string. This function has been used to encrypt the password prior to saving in database so that users having access to database would not be able to view the original password [16].

```
public static string Encrypt(string clearText, string EncryptionKey)
{
    byte[] clearBytes = Encoding.Unicode.GetBytes(clearText);
    using (Aes encryptor = Aes.Create())
    {
        Rfc2898DeriveBytes pdb = new
Rfc2898DeriveBytes(EncryptionKey, new byte[] { 0x49, 0x76, 0x61, 0x6e,
0x20, 0x4d, 0x65, 0x64, 0x76, 0x65, 0x64, 0x65, 0x76 });
        encryptor.Key = pdb.GetBytes(32);
        encryptor.IV = pdb.GetBytes(16);
        using (MemoryStream ms = new MemoryStream())
        {
            using (CryptoStream cs = new CryptoStream(ms,
encryptor.CreateEncryptor(), CryptoStreamMode.Write))
            {
                cs.Write(clearBytes, 0, clearBytes.Length);
                cs.Close();
            }
            clearText = Convert.ToBase64String(ms.ToArray());
        }
    }
    return clearText;
}
```

As per the afore-mentioned details then implementation work were carried on. The product generated should be subjected to thorough testing to ensure the system works as expected. The following chapter describes the activities done to achieve this.

Chapter 5 - Evaluation

5.1 Introduction

Any evaluation has pragmatically chosen goals. This goal aims at the determination of the degree of desired qualities of a finished system. The evaluation of the system with respect to “Usability-Goals” is one of the application of this goal. Other examples are the certification of software, and the check on conformity with given standards [17].

In order to ensure that developed software meets the requirement and controls the process flow, software need to be tested and evaluated well.

5.2 Test Plan

Having a properly laid out test plan is very critical to verify and validate the system. Verification can be performed in two ways

- Dynamic verification
Involves checking the software behaviour by running the executable.
- Static verification
This is related to analyzing the static system to discover problems. This process requires a great deal of scrutiny and concentration to achieve effective results.

Given the time constraints, dynamic testing process was used in evaluating GSCC HR software.

5.3 Test Procedures

Dynamic testing processes are briefly discussed as given below.

5.3.1 Unit Testing

This test depends on the design and test plan should be done prior to coding. The developer of the code generally takes the responsibility of writing the plan and then carry out the test.

5.3.2 Integration Testing

This test is written by the system specification writers. This is carried out after unit testing to ensure the system behaves as expected in functional specification.

5.3.3 System Testing

This test is done once the integration testing is over. Test cases and plans are made based on requirement of the system, which can either functional or non-functional.

5.3.4 Acceptance Testing

This test includes the major stakeholder customer, to ensure the functions behave as expected by the customer. When written by system design or development team, the test plan should be agreed with customer.

In GSCC, above four testing phases were carried out as per the design, development, and implementation progress. For acceptance testing, tests were done by HR users and HR manager as per the agreed functions that the system should provide. Satisfactory results were yielded upon this testing.

5.4 - Test Cases

Some of the test cases used in testing are given below. Please refer Appendix E for more details.

5.4.1 Test case for Login Screen

Test cases and results for Login Screen is shown in below Table 5.1.

Test Case No	Test Case	Expected Result	Success/Fail
1	Open application and ensure login form comes with empty user and password.	Login screen opened. Username and password fields are empty.	✓
2	Leave username and password blank and click on Sign-In button.	Message should appear "Please enter User ID"	✓
3	Enter valid user name and leave password blank.	Message should appear "Please enter your password"	✓
4	Enter valid user name and incorrect password.	Message should appear "Invalid user or password"	✓
5	Enter invalid user name and a valid password	Message should appear "Invalid user or password"	✓
6	Enter valid username and valid password.	Authentication to be successful. Login screen should disappear, and Main menu has to be displayed	✓
7	Enter a valid username and password for user who has no menu access to system.	Message should appear "No program access has been granted to user. Login denied."	✓
8	Enter a valid username and password for a user with inactive status.	Message should appear "User status is not active. Login denied."	✓

Table 5.1: Test Case for Login Screen

5.4.2 Test case for User Access Management

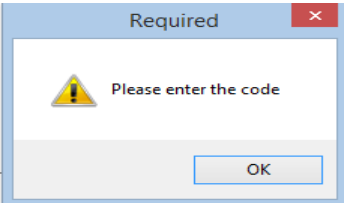
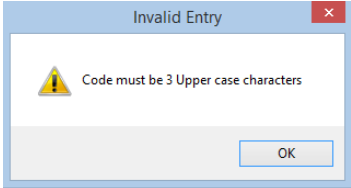
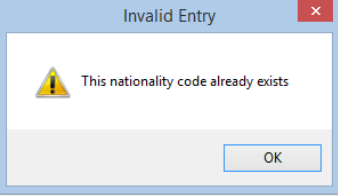
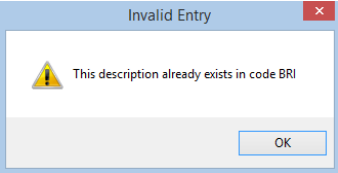
Table 5.2 shows the test case for user access Management Screen.

Test Case No	Test Case	Expected Result	Success/Fail
1	Click and open Users and Access Management Screen from Left Menu Panel	User Access Management screen should be opened. Username and other input should be empty. Only new button should be enabled. Focus should be on username.	✓
2	Enter an invalid username	Message should appear "Username is not found"	✓
3	Enter an existing username and press Enter.	User details and user access should be displayed. Query, New, Save and Delete buttons should be enabled.	✓
4	Press F7 to re-enter the query mode. Or click Query button.	All fields should be cleared, and focus should be on username.	✓
5	While displaying an existing username details, press F6 to enter add mode. Or click New button.	All fields should be cleared, and focus should be on username.	✓
6	While displaying an existing username details, leave employee name or password blank, press F10 to save or click Save button.	Error message should appear "Please enter username" or "Please enter password"	✓
7	While displaying an existing username details, enter employee name and password etc., press F10 to save or click Save button.	Error message should appear "Please enter username" or "Please enter password"	✓

Table 5.2: Test case for User Access Management

5.4.3 Test case for Nationality Reference Codes

Table 5.3 below displays test cases and results for Nationality reference codes.

Test Case No	Test Case	Expected Result	Success/Fail
1	Open Nationality Form from the Main Menu	Nationality form should be opened. All presently existing codes should be displayed.	✓
2	Click 'New' Button.	Nationality list should disappear. New panel should appear asking to input nationality code and description.	✓
3	Click save button without entering code and description.	Message should appear as "please enter the code" or "please enter the description"	✓
			
4	Input a nationality code with a format other than 3 alpha characters; such as 3 numbers etc.	Message should appear "Code must be 3 upper case characters"	✓
			
5	Input a nationality code that has been already added.	Message should appear as "This nationality code already exists"	✓
			
6	Input a nationality description that has been already entered for another code.	Error message should appear "This description already exists".	✓
			

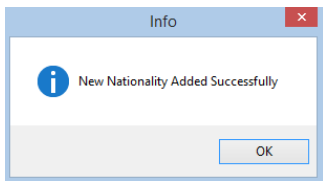
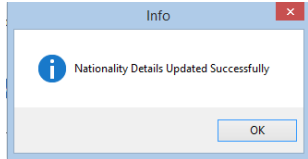
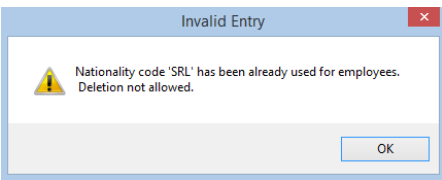
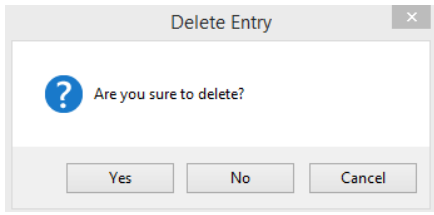
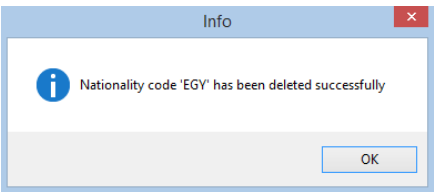
7	Feed in all information correctly and press F10 to save.	Notification message should appear to confirm record is saved 	✓
8	From initial nationality list, double click a row having data.	Nationality edit panel should be opened.	✓
9	Modify the description and save	Nationality description should be saved and a message should appear to confirm. 	✓
10	Double click a row in nationality list which has been already used for employees and click delete button.	Message should appear indicating the selected code has been already used. 	✓
11	Double click a row in nationality list which has been already used for employees and click delete button.	User should be prompted to confirm the deletion. 	✓
12	Click "Yes" in deletion confirmation.	Nationality code should be deleted and a message should appear to confirm. 	✓
13	Click "No" or "Cancel" in deletion confirmation.	No records are deleted. Selected nationality should retain.	✓

Table 5.3: Test case for Nationality Reference Maintenance

Further test cases are provided in Appendix E.

5.5 User Evaluation

Upon completing the system testing, users from HR department including HR manager performed testing on system. Each of the module that existed in the legacy system was comparatively testing in the new GSCC application. Feedback was collected from HR users. Table 5.4 shows a sample collected from a user in this regard.

GSCC HR PROJECT System Evaluation Questionnaire					
Date		: 28 th October 2017			
Performed by		: Azmath Shafi			
Position		: Senior Supervisor HR			
Signature		:			
Evaluation done on	Rating				
	Strongly Agree	Agree	Average	Disagree	Strongly Disagree
Functionality : System does what was proposed.	✓				
Functionality : Software has secure access through login and authentication.		✓			
Reliability : Software runs without errors in doing regular functions.		✓			
Reliability : Software provides appropriate messages.	✓				
Usability : It is easy to use the system.	✓				
Usability : It is easy to learn how to use.	✓				
Efficiency : System response are quick.		✓			
Efficiency : Process execution times are appropriate.	✓				

Table 5.4: User Acceptance Testing

Results were collected using the questionnaire mentioned above in Table 5.4 from seven different users. The results taken are summarized and presented graphically in Figure 5.1.

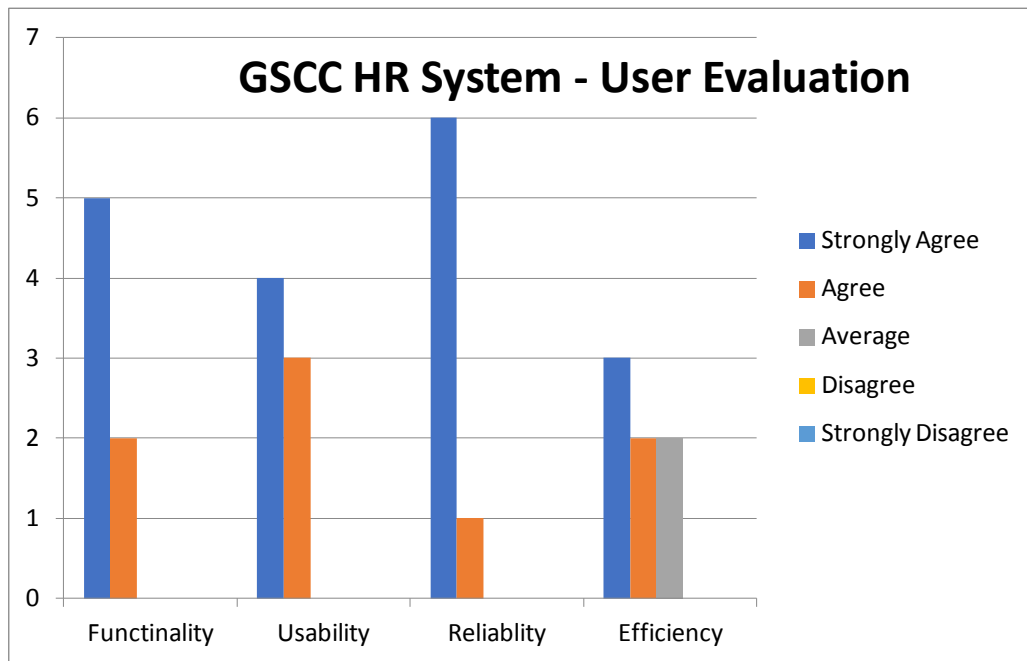


Figure 5.1: User Evaluation Summary

An overall acceptance has been achieved as per the evaluation done. However, it has been recognized that improvements need to be done in the area of efficiency of the system.

Chapter 6 - Conclusion

6.1 Introduction

GSCC had been using the HR system for nearly two decades with the outdated legacy text-only based system. In the old system, data entry was tedious, copy-paste functions were totally missing, button clicks and multi selects options were not possible and so on.

With the implementation of the new HR system with GUI based windows application, GSCC users were able to avail below listed features.

- All rich features that a windows system provides are achieved such as dropdown and grid view, multi select, usage of pointing device for clicks and selections etc.
- Previously reports were printed only on dot-matrix printers. With the new application, users are able to generate reports with graphical layout.
- Another major facility given by the new GSCC HR system is the ability to view reporting data in a grid and the ability export the data to spreadsheets in Microsoft Excel. This allows user to do further analysis and share the data with much ease via email attachments etc.
- The notification system for major events proved to be another highlighted feature of the GSCC HR system. Previously users had to print and identify on daily basis what contacts are to expire. With the email based alert system, users would be notified what contracts expire without the hassle of going through daily reports.

With the inclusion of above features, new GSCC HR application has been of great value to the users. HR Manager and HR clerks alike liked the features and have already begun providing suggestions for future enhancements.

6.2 Future Enhancements

A major leap has been done and now this system has laid the foundation to upgrade and enhance with the emerging wealth of technologies such as enhancements given below

- **Integration with Financial Package**
GSCC uses Microsoft GP for financial account and employee payroll. With the new SQL server database in place, the integration processes now can be commenced.
- Data can be accessed via web services. Therefore, mobile applications could be build allowing users to view their personal info as well as analysis summaries for top management.
- Build dashboard and BI analytical screen and reports to present data in much more visualized manner than the traditional tabular reports.
- Develop modules to allow employees to directly interact with system such as leave application, contract renewal etc., and bring the concept of paperless office a reality in GSCC.
- Include options to set a level for each employee within organization and allow generation of organization chart from the system.

6.3 Lessons Learnt

The learning and experience processes in the developing this system were many. Below given are a few of them

- The need came to discuss with many staffs in HR to access the requirements. Staffs with varying background and experiences were met and built a mutual understanding and a bond to collaboratively work towards the success of this project.
- Development of this system introduced many tools such UML diagram software authoring and that help develop the skills in system analysis, system design etc.
- Data migration from Caché database to MS SQL server was a challenging task. This system development assisted a good grip of these migrations and helped develop a utility that does the insertion from text file to SQL database.
- This was a time-bound objective. Therefore, we learnt how to handle the pressure and work to attain with set targets and schedules.
- Many of the emerging revolutionary technologies were noted during the study to implement this system. Thus, it created a thirst to invest more time in learning the new techs and enhance the existing software platforms.

References

- [1] Techopedia System Analysis. [Online] Available:
<https://www.techopedia.com/definition/9611/systems-analysis>.
[Accessed 28-Jul-2017]
- [2] Paylite Human Resources Management System. [Online] Available:
<http://www.paylitehr.com/>. [Accessed – 31-Jul-2017]
- [3] ADEPT HRMS Solution. [Online] Available:
<http://adeptsol.com/products/hrms-dubai-hr-payroll-software-dubai-sharjah-abudhabi-soudi-bahrain-middle-east-qatar-kuwait-ksa-oman-uae/>.
[Accessed – 08-Jan-2018]
- [4] Wikipedia – Functional Requirements. [Online] Available:
https://en.wikipedia.org/wiki/Functional_requirement.
[Accessed – 01-Aug-2017]
- [5] Wikipedia – Non-Functional Requirements. [Online] Available:
https://en.wikipedia.org/wiki/Non-functional_requirement
[Accessed – 01-Aug-2017]
- [6] Wikipedia – Rational Unified Process. [Online] Available:
https://en.wikipedia.org/wiki/Rational_Unified_Process
[Accessed – 09-Feb-2018]
- [7] Wikipedia – Rational Unified Process Chart. [Online] Available:
https://en.wikipedia.org/wiki/Rational_Unified_Process#/media/File:Development-iterative.png. [Accessed – 08-Jan-2018]
- [8] Slideshare - Phases in RUP. [Online] Available:
<https://www.slideshare.net/zoimustafa/rup-model>
[Accessed – 08-Jan-2018]
- [9] Wikipedia – System Design. [Online] Available:
https://en.wikipedia.org/wiki/Systems_design.
[Accessed – 01-Aug-2017]

- [10] Object Oriented Analysis. [Online] Available:
www.businessdictionary.com/definition/object-oriented-analysis.html
[Accessed – 03-Aug-2017]
- [11] Use case diagrams. [Online] Available:
www.uml-diagrams.org/use-case-diagrams.htm
[Accessed – 02-Sep-2017]
- [12] Wikipedia Class Diagram. [Online] Available:
https://en.wikipedia.org/wiki/Class_diagram
[Accessed – 02-Sep-2017]
- [13] Sequence Diagram. [Online] Available:
https://en.wikipedia.org/wiki/Sequence_diagram
[Accessed – 02-Sep-2017]
- [14] ER Diagrams – Webopedia. [Online] Available:
https://www.webopedia.com/TERM/E/entity_relationship_diagram.html
[Accessed – 11-Feb-2018]
- [15] Database Design. [Online] Available:
https://en.wikipedia.org/wiki/Database_design. [Accessed – 15-Sep-2017]
- [16] Encrypt a plain string. [Online] Available:
<https://stackoverflow.com/questions/10168240/encrypting-decrypting-a-string-in-c-sharp>. [Accessed – 15-Sep-2017]
- [17] Software Evaluation. [Online] Available:
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.87.8362&rep=rep1&type=pdf>. [Accessed – 02-Nov-2017]

Appendix A - System Documentation

Below given steps explain the steps required to deploy the system. In GSCC the intended deployment was to setup as a client server network based application. Yet this could be deployed and executed as a standalone application as well in a single PC.

Ensure the pre-requisites

Hardware

- Intel® Core™ i5 CPU 2.8 GHz or above
- 8GB RAM
- 500GB Hard Disk
- Inkjet or LaserJet Printer
(Optional if required to produce hard copy)

Software

- Microsoft Windows 7 or above
- Visual Studio Community Edition 2015 or above
- Crystal Reports
- SQL Server 2012 or above
- Microsoft Office 2010

Install and configure SQL Server

- Install Microsoft SQL Server 2012 or above.
- Run the “DB Create and Data Insert.sql” script provided in the CD. This script would perform the following
 - Create a new database named “gscchr”
 - Create all related tables
 - Insert data into tables required to run the application.
 - Create a database user name “gscchr”. By default, the password would be set to 123. If desired, password could be changed by choosing properties of the user “gscchr”.
 - Create an application user named “admin1” with full privilege to the application. Password of admin1 user is 123.
 - Add an application user named “admin” with password set to “123”.

Setup the application environment

- Take a copy of the application folder and paste in a desired location in the PC.
- Navigate to the folder Application\GscHrApp\ GscHrApp
- Open “App.config” file using any text editor
- Modify the connection string
 - Set Data Source = {Server Name} where SQL server is installed
 - Set the password = to the password set above. If password was not changed, leave it as 123.

Open the application in visual studio and execute the application.

- Navigate to folder Application\GscHrApp
- Open the solution file “GscHrApp.sln”
- Click the Start button in tool bar or press F5 to run
- Login screen would be prompted.
- Enter username and password as admin1 and 123 respectively.

Appendix B - Design Documentation

This section includes the use case diagrams and use case narratives used in each Module.

- **User Login Module**

Figure B.1 shows the use case diagram for user login module.

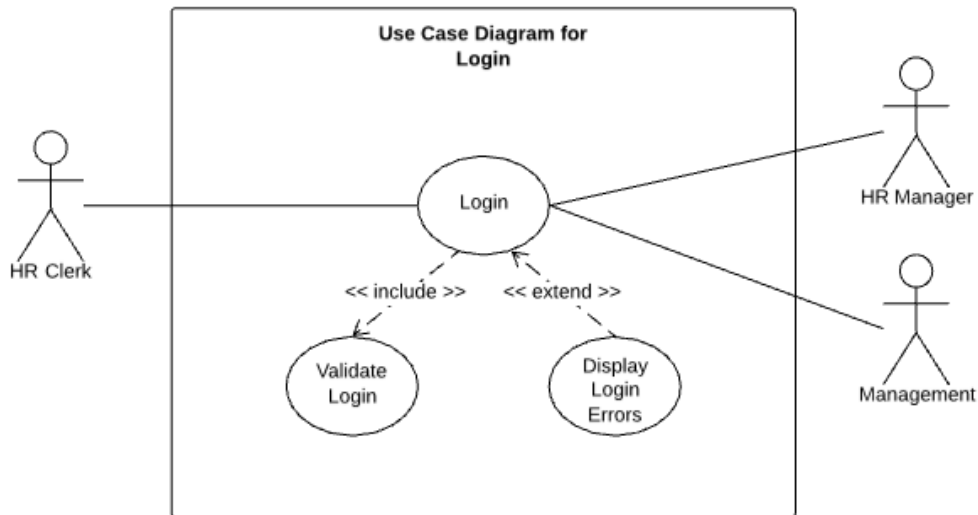


Figure B.1: Use case diagram for User Login

Table B.1 below displays the use case narrative for user login module.

Use Case Name	Login
Actors	HR Clerk, HR Manager, Management Staffs
Description	Login to GSCC HR Application
Pre-Conditions	Application is installed in users workstation
Flow of Events	<ol style="list-style-type: none"> 1. User opens the application 2. User types username and password 3. System should allow login if account is validated 4. System should deny and show error message if account is not validate
Post-Conditions	User should be able to view main screen and be able to choose options carry on tasks.

Table B.1: Use Case Narrative for Login

- **User Management Module**

Figure B.2 shows the use case diagram for user management.

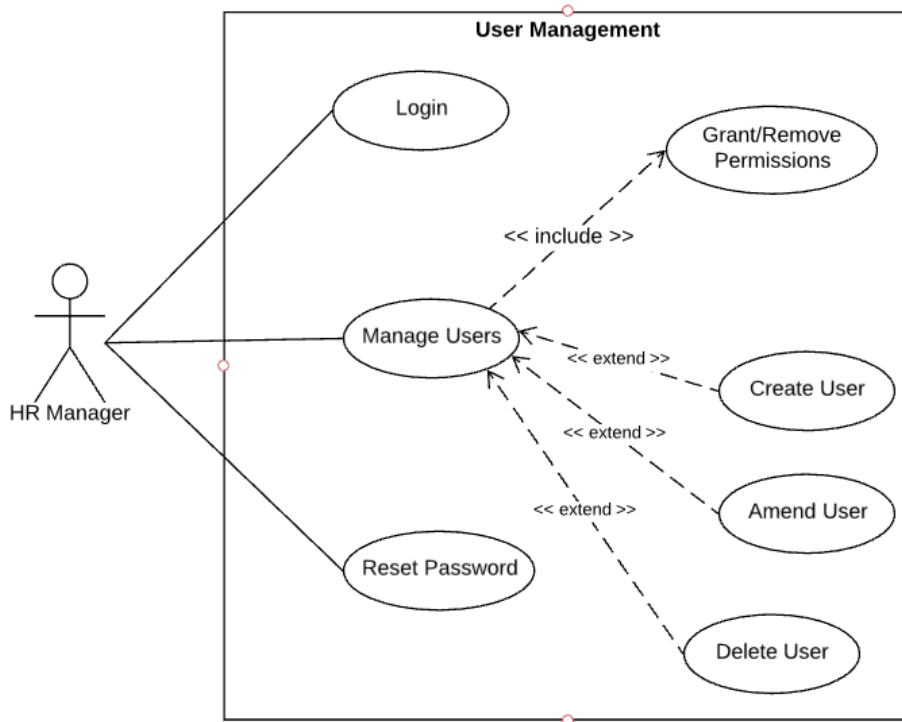


Figure B.2: Use case diagram for User Management

Table B.2 below displays the use case narrative for user management.

Use Case Name	User Management
Actors	HR Manager
Description	Manage HR users and permissions
Pre-Conditions	User should have a valid account. Permission must have been granted to user management User should have successfully logged into system. User should have privilege to manage users.
Flow of Events	1. Enter a new user. 2. View and modify an existing user. 3. Save the details. 4. Delete an existing user. 5. Grant and revoke permission for the user
Post-Conditions	New user account should be created and new user should be able to login. After user logs in, permission granted/revoked should be correctly reflected.

Table B.2: Use case narrative for User Management

- **Employee Module**

Figure B.3 shows the use case diagram for employee module.

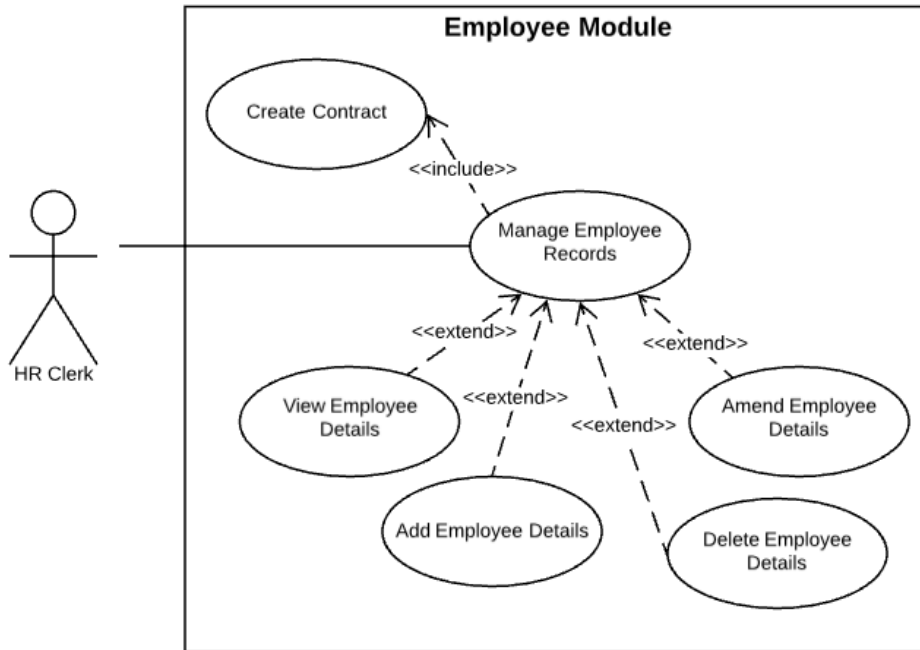


Figure B.3: Use case diagram for Employee Module

Table B.3 below displays the use case narrative for employee module.

Use Case Name	Employee Module
Actors	HR Clerk
Description	Manage Employee Details in HR System
Pre-Conditions	User should have a valid account. Permission must have been granted. User should have logged into account. User should have opened Employee Form.
Flow of Events	1. Add a new number. 2. View and existing number and amend some details. 3. Save the details. 4. Delete an existing employee and confirm.
Post-Conditions	New employee record should be viewable in employee reports. Deleted employee records should not appear in employee reports.

Table B.3: Use case narrative for Employee

- **Contract Module**

Figure B.4 shows the use case diagram for contract module.

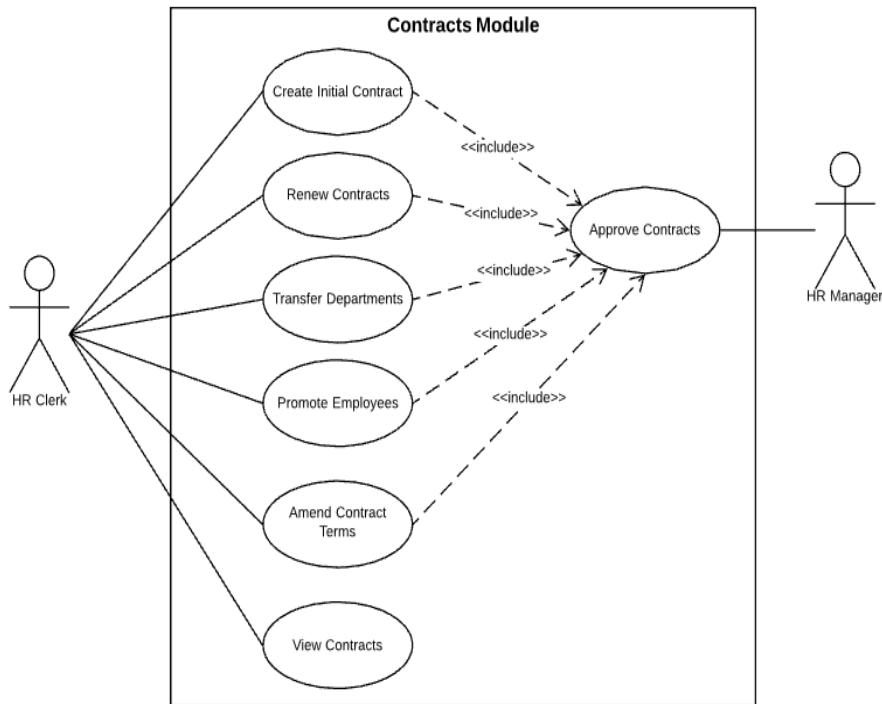


Figure B.4: Use case diagram for Contracts Module

Table B.4 below displays the use case narrative for contract module.

Use Case Name	Contracts Module
Actors	HR Clerk, HR Manager
Description	Manage Employee Contract Details
Pre-Conditions	User should have a valid account. Permission must have been granted. User should have logged into account. User should have opened Contract Form
Flow of Events	1. Enter employee number 2. View contract details 3. Add New Contracts for Promotion, Transfer of Department etc. 4. Add new contract for renewal 5. Modify contract details 6. Save contract details. 7. Delete an existing contract
Post-Conditions	Once saved or modified and saved, the latest information should be seen on contract reports.

Table B.4: Use case narrative for Contract

- **Leave Module**

Figure B.5 shows the use case diagram for employee leave module.

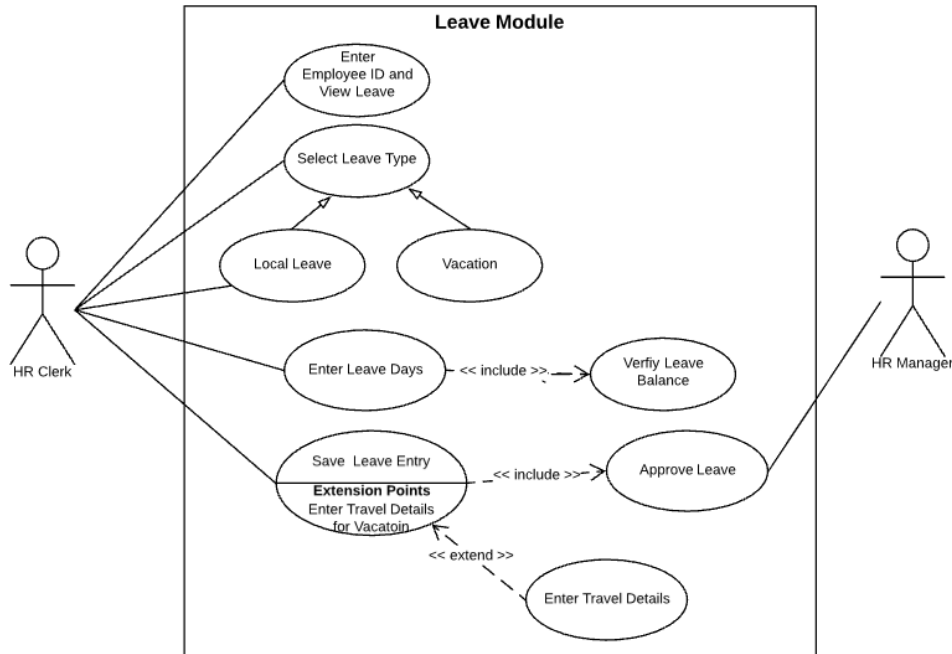


Figure B.5: Use case diagram for Leave Module

Table B.5 below displays the use case narrative for leave module.

Use Case Name	Leave Module
Actors	HR Clerk, HR Manager
Description	Manage Employee Leave Details
Pre-Conditions	User should have a valid account. Permission must have been granted for leave program User should have logged into account. User should have opened Leave Form.
Flow of Events	1. Enter employee number 2. View Leave details 3. Select Leave Type 4. Save Leave Detail 5. Approve Leave
Post-Conditions	* Once inserted or modified, the latest information should be seen on leave report. * Employee's leave balance should be correctly reflected.

Table B.5: Use case narrative for Leave

- **Sick Leave Module**

Figure B.6 shows the use case diagram for sick leave module.

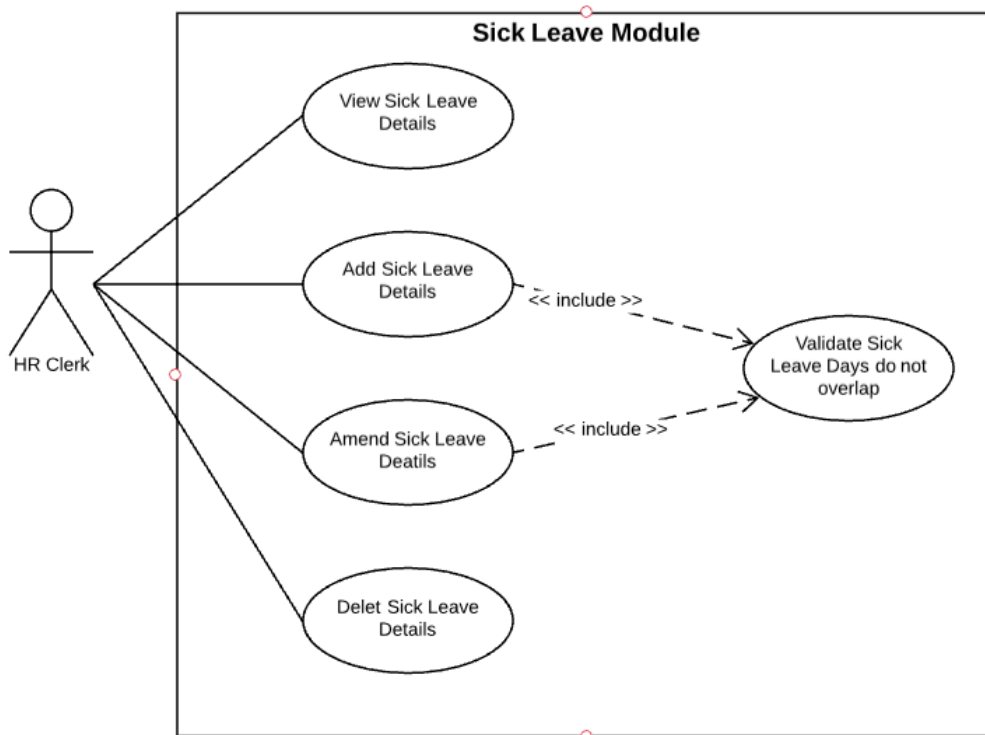


Figure B.6: Use case diagram Sick Leave Module

Table B.6 below displays the use case narrative for sick leave module.

Use Case Name	Sick Leave Module
Actors	HR Clerk
Description	Manage Sick Leave Details
Pre-Conditions	User should have a valid account. Permission must have been granted to transact entry in Sick Leave. User should have logged into account. User should have opened the sick leave form.
Flow of Events	1. Enter Employee Number 2. Enter new sick leave entry 3. Modify an existing sick leave entry 4. Delete an existing sick leave entry
Post-Conditions	When successfully entered, the entry should correctly reflect in sick leave entry screen and in reports.

Table B.6: Use case narrative for Sick Leave

- **HR Reports**

Figure B.7 shows the use case diagram for HR reports module.

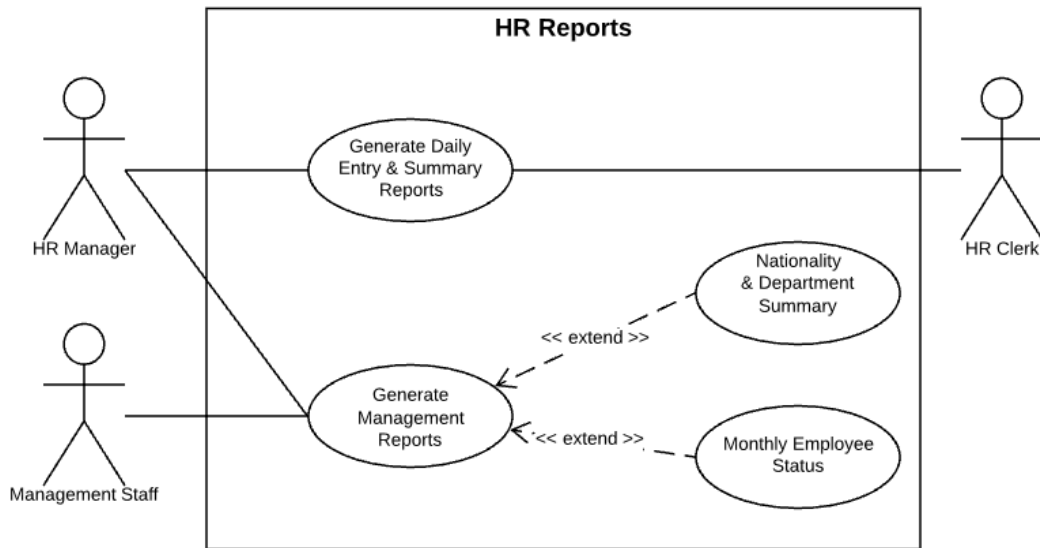


Figure B.7: Use case diagram for Reports

Table B.7 below displays the use case narrative for HR reports module.

Use Case Name	HR Reports Module
Actors	HR Clerk, HR Manager, Management Staff
Description	Generate Reports
Pre-Conditions	User should have a valid account. Permission must have been granted for designated reports. User should have logged into account. User should have opened the specific report.
Flow of Events	1. Enter related criteria in report such as date range 2. Generate/Print Daily Entry and Employee Report 3. Print Management Report
Post-Conditions	* When daily report generated, output should be saved to excel file and should be displayed for review. * Management reports should be generated as per the specific period selected.

Table B.7: Use case narrative for Reports

Appendix C - User Documentation

Login to Application

When GSCC HR application is run, below screen would appear. Enter the user and password correctly and log into the system. Figure C.1 displays the login screen that appears when the application is started.

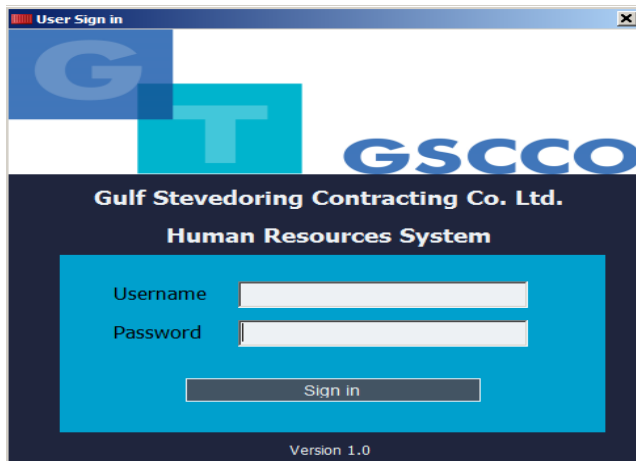


Figure C.1: User Sign in Form

Main Menu

Once user log in, System would display the Main Windows. On the top panel the menu options would be displayed. On the left panel appears the option for the selected main menu. Click on any of the option on the left panel to start the screen for it. For example, click on “Employee” button to start the “Employees” maintenance form. Figure C.2 shows the main menu that appears on successful logon.

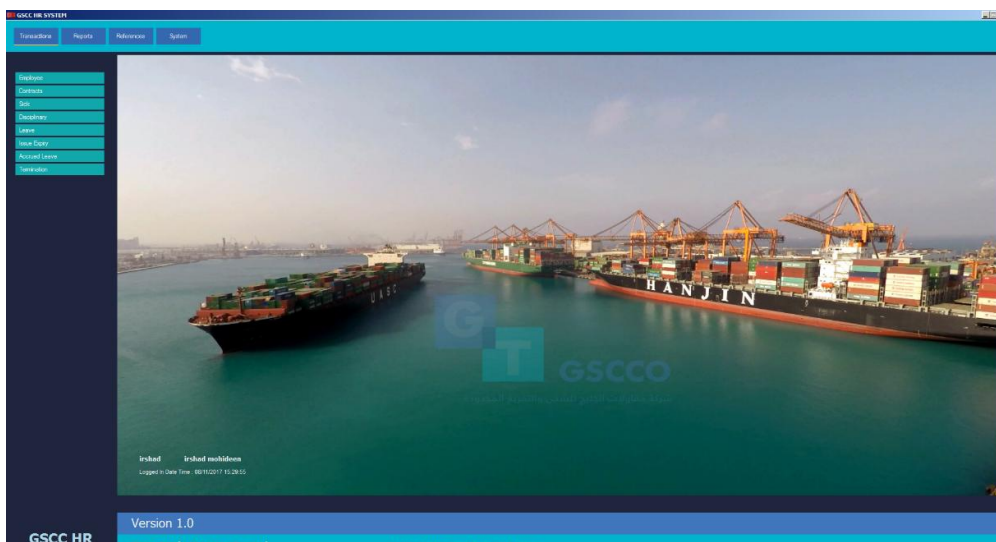


Figure C.2: Main Menu

Opening Update Forms

To open Employee Form click “Transactions” on the top panel. Then click on the “Employee” button. Figure C.3 highlights what options to be used to open an update screen.

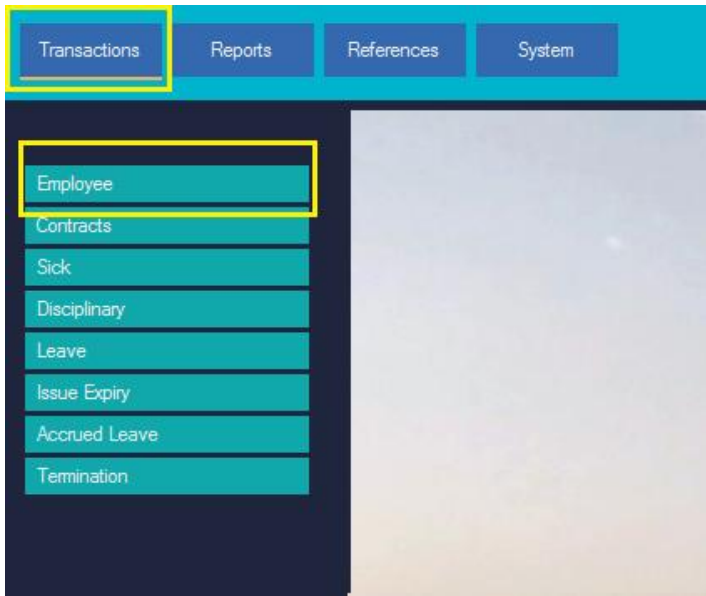
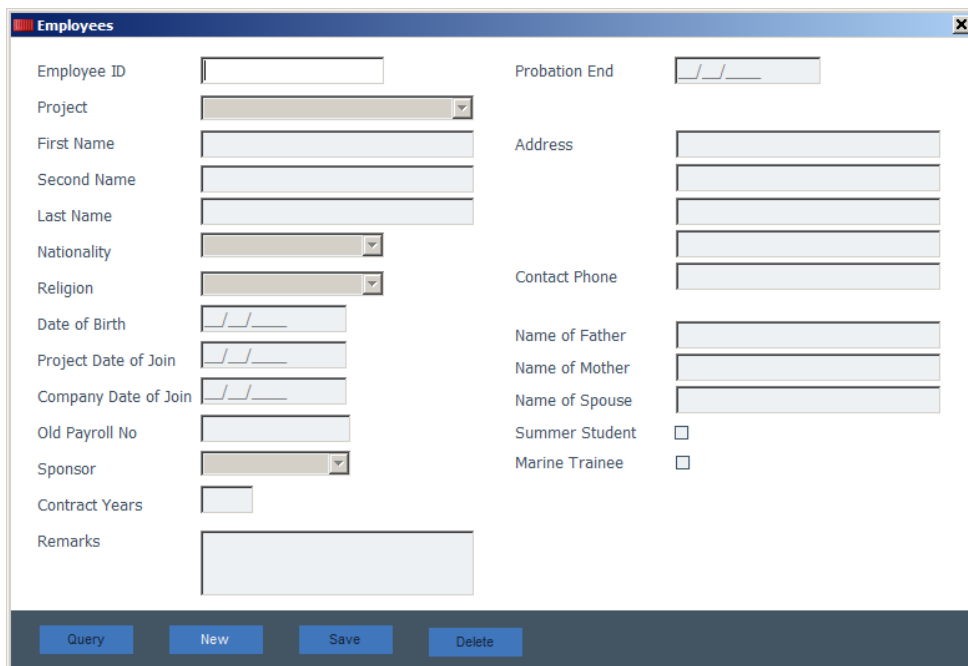


Figure C.3: Option “Employee” to be clicked to Open

Figure C.4 displays the employee screen is open and users can start entering data.

The image shows a window titled 'Employees' with a close button in the top right corner. The form contains the following fields:

- Employee ID: text input
- Project: dropdown menu
- First Name: text input
- Second Name: text input
- Last Name: text input
- Nationality: dropdown menu
- Religion: dropdown menu
- Date of Birth: date input (format: / / - -)
- Project Date of Join: date input (format: / / - -)
- Company Date of Join: date input (format: / / - -)
- Old Payroll No: text input
- Sponsor: dropdown menu
- Contract Years: text input
- Remarks: large text area
- Probation End: date input (format: / / - -)
- Address: three stacked text input fields
- Contact Phone: text input
- Name of Father: text input
- Name of Mother: text input
- Name of Spouse: text input
- Summer Student: checkbox
- Marine Trainee: checkbox

At the bottom of the window, there are four buttons: 'Query', 'New', 'Save', and 'Delete'.

Figure C.4: Employee form is open

Running Reports

To run a report click on the report button available on the top panel. Then select the interested report from the left panel. Figure C.5 below displays the menu and option that appear for reports.

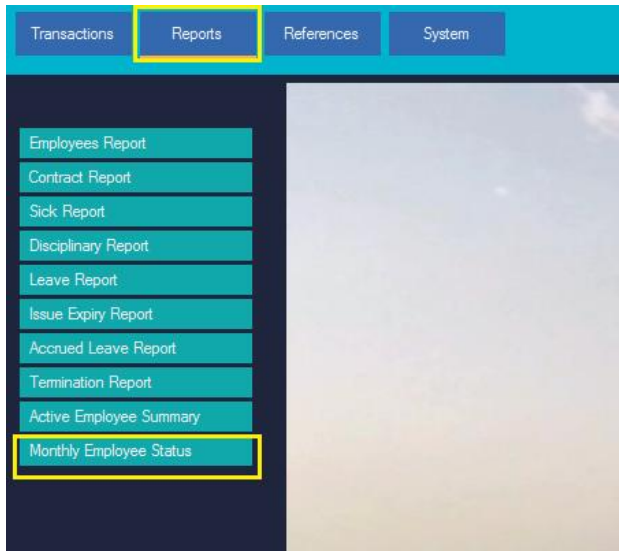


Figure C.5: Report Menu Selection

Maintaining References

HR system consists of reference details such as Nationalities, Departments, and Jobs. Using the “Reference” menu and then picking the required option, references can be maintained. Figure C.6 shows how reference options can be accessed.

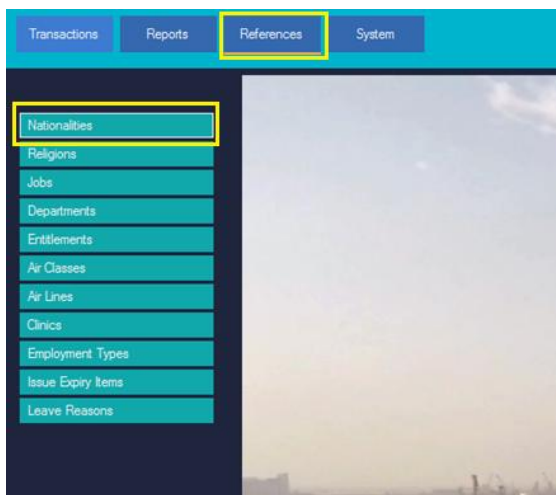


Figure C.6: Reference Menu Selection

Once any of the reference option is clicked from the menu, the relevant screen would be opened. Figure C.7 shows the nationality reference screens displaying the existing nationality codes in a grid.

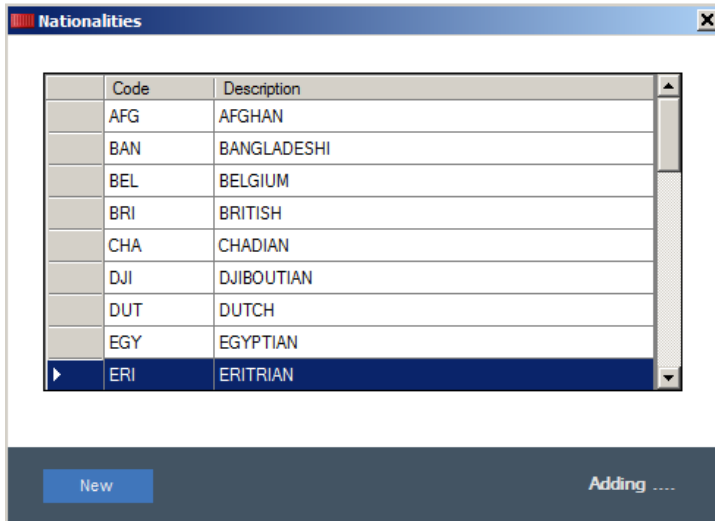


Figure C.7: Nationality Reference List Displayed

In order to edit any of the items in the grid, double-click the desired code. Figure C.8 shows such an editing screen for nationality reference.

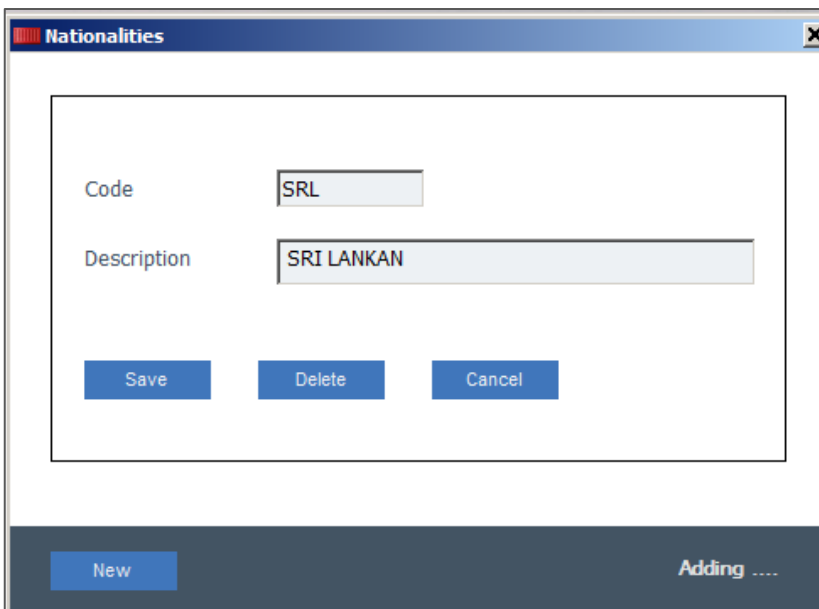
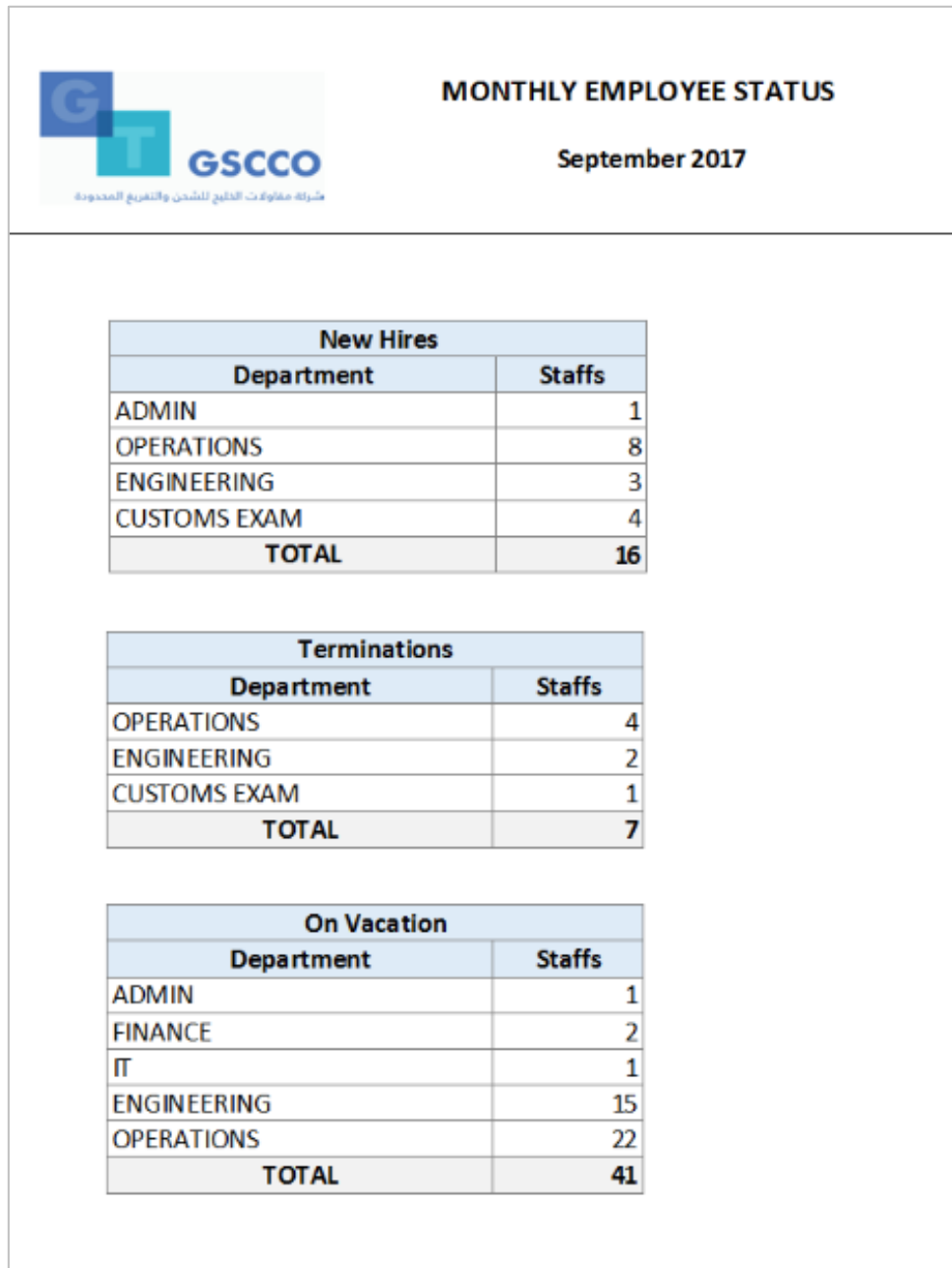


Figure C.8: Adding a New Nationality

Appendix D - Management Reports

Monthly Employee Status

This is one of the key reports run by management on monthly basis. Figure D.1 displays the output of the report.



MONTHLY EMPLOYEE STATUS
September 2017

New Hires

Department	Staffs
ADMIN	1
OPERATIONS	8
ENGINEERING	3
CUSTOMS EXAM	4
TOTAL	16

Terminations

Department	Staffs
OPERATIONS	4
ENGINEERING	2
CUSTOMS EXAM	1
TOTAL	7

On Vacation

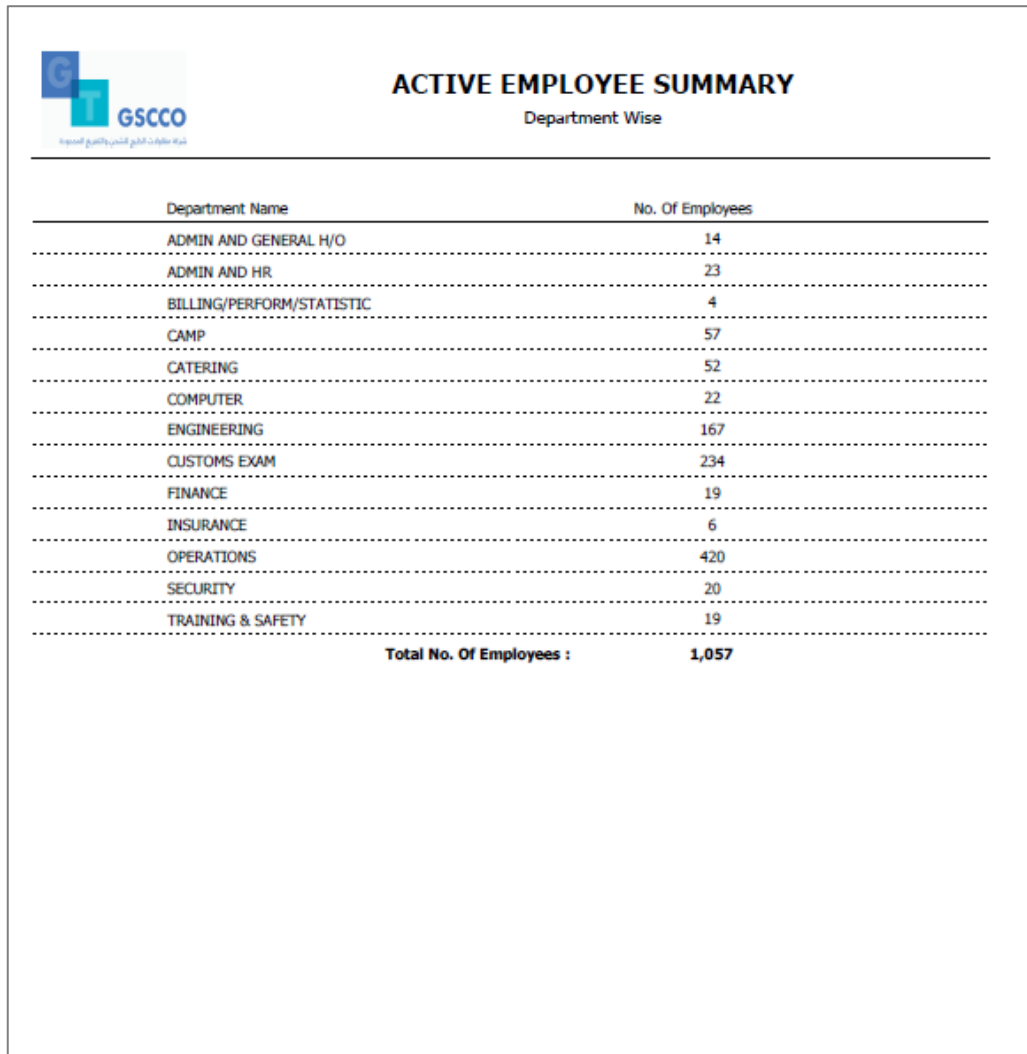
Department	Staffs
ADMIN	1
FINANCE	2
IT	1
ENGINEERING	15
OPERATIONS	22
TOTAL	41

Figure D.1: Monthly Employee Status

Active Employee Summary by Department

Active employee count at any given time can be retrieved from this report.

Figure D.2 shows a sample of Active Employee Report by Department.



The image shows a screenshot of a report titled "ACTIVE EMPLOYEE SUMMARY" with the subtitle "Department Wise". In the top left corner, there is a logo for GSCCO (General Security Company of Oman) with the text "شركة أمن عمان العامة" below it. The report is a table with two columns: "Department Name" and "No. Of Employees". The table lists 13 departments and their respective employee counts, with a total of 1,057 employees. The departments and their counts are: ADMIN AND GENERAL H/O (14), ADMIN AND HR (23), BILLING/PERFORM/STATISTIC (4), CAMP (57), CATERING (52), COMPUTER (22), ENGINEERING (167), CUSTOMS EXAM (234), FINANCE (19), INSURANCE (6), OPERATIONS (420), SECURITY (20), and TRAINING & SAFETY (19). The total number of employees is 1,057.

Department Name	No. Of Employees
ADMIN AND GENERAL H/O	14
ADMIN AND HR	23
BILLING/PERFORM/STATISTIC	4
CAMP	57
CATERING	52
COMPUTER	22
ENGINEERING	167
CUSTOMS EXAM	234
FINANCE	19
INSURANCE	6
OPERATIONS	420
SECURITY	20
TRAINING & SAFETY	19
Total No. Of Employees :	1,057

Figure D.2: Active Employee Summary by Department

Active Employee Summary by Nationality

Active employee count can also be seen by Nationality which is another important dimension for management. Figure D.3 shows a sample of Active Employee Report by Nationality.


 ACTIVE EMPLOYEE SUMMARY Nationality Wise	
Nationality Name	No. Of Employees
BANGLADESHI	35
BRITISH	3
EGYPTIAN	10
ETHIOPIAN	1
INDIAN	657
JORDANIAN	4
NEPALESE	4
PAKISTANI	4
FILIPINO	90
SAUDI	85
SOMALI	18
SRI LANKAN	97
SUDANESE	3
SYRIAN	1
TANZANIAN	1
YEMENI	44
Total No. Of Employees :	1,057

Figure D.3: Active Employee Summary by Nationality

Appendix E - Test Results

More of the test cases and the results are provided in this Appendix.

Test case for Employee Data

Test case for employee module is provided in Table E.1.

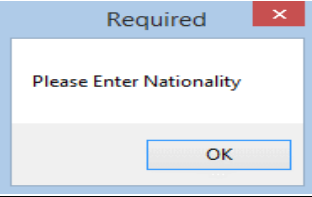
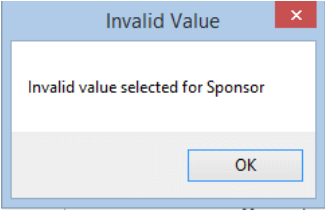

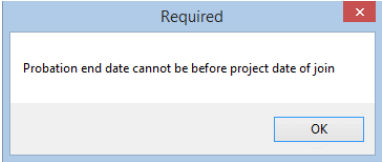
Test Case No	Test Case	Expected Result	Success/Fail
1	Open Employee Form from the Menu	Form should be opened. All the drop downs should be filled with valid values.	✓
2	Enter only partial information. Leave some mandatory fields such as Nationality empty and save.	Message should appear "Please enter Nationality". 	✓
3	Input invalid value that does not exist in combo box.	Error message should appear "Invalid value selected for Sponsor" 	✓
4	Input invalid value in date of birth and press tab to leave the field.	Instant tool tip message should appear "Invalid Date Value" 	✓
5	Enter probation end date with date before date of join.	Error message should appear "Probation end ate cannot be before date of join" 	✓
6	Feed in all information correctly and press F10 to save.	Notification message should appear "Employee details are updated successfully"	✓

Table E.1: Test case for Employee Module

Test case for Contracts Module

Test case for contract module is provided below in Table E.2.

Test Case No	Test Case	Expected Result	Success/Fail
1	Open Contracts Form from the Menu	Contract form should be opened and program has to wait for user input for Employee ID	✓
2	Enter an Employee ID that does not exist.	Message should appear "Employee ID not found".	✓
3	Enter an Employee ID that is already entered.	System should display employee summary info and in the grid, system has to show all the existing contracts.	✓
4	Double click a row in the contracts grid.	System has to show all the details of the contract that is selected and double clicked. User can update or delete contract data.	✓
5	Modify any data and click Save button.	System should save data, display a confirmation message and go back to screen with contracts grid.	✓
6	Click New Button.	All the existing contract data should be copied and shown. Effective data should be blank. End data should be same as the previous contract.	✓
7	Enter a new effective data and click Save.	System has to add a new contract, show a confirmation message and show all the contracts in grid. A new contract record should be shown.	✓
8	Click Renew Button after entering a valid employee	A new effective date should be set by system. This date has to be the next date of the last contract. Contract type should be set to NEW.	✓
9	Select a valid contract and click delete	System should prompt for confirmation. If selected "Yes", contract data should be deleted.	✓

Table E.2: Test case for Contracts Module

Test case for Leave Module

Test case for leave module is provided below in Table E.3.

Test Case No	Test Case	Expected Result	Success /Fail
1	Open Leave Form from the Menu	Leave form should be opened and program has to wait for user input for Employee ID	✓
2	Enter an Employee ID that does not exist.	Message should appear "Employee ID not found".	✓
3	Enter an Employee ID that is already entered.	System should display employee summary info and fill the table with all leave entries.	✓
4	Double click a row in the leave table.	System has to show all the details of the leave that is selected and double clicked. User can update or delete leave data.	✓
5	Modify any data and click Save button.	System should save leave details, display a confirmation message and go back to screen that shows all leave entries in the table.	✓
6	Click New Button.	Leave details panel should be opened with empty values.	✓
7	Input an application data that has already been added in leave.	System has to show message "This date has already been added in leave".	✓
8	Input from date and to date of leave that does not match with total paid and unpaid leave.	Error message should appear "Total leave applied does not match with dates"	✓
9	Select other leave types and leave the other leave days blank.	Error message should appear "Other leave days should be entered if type is selected"	✓
10	Input valid values and click Save button.	All data for leave should be saved. Message should appear "Leave entry updated successfully"	✓
11	Select a leave entry and click delete button.	Selected entry should be removed. System should prompt a confirmation message "Are you sure to delete this leave data?" If clicked yes, the leave entry should be removed. If clicked No, leave entry should retain.	✓

Table E.3: Test case for Leave Module

Test case for Disciplinary Module

Test case for disciplinary module is provided below in Table E.4.

Test Case No	Test Case	Expected Result	Success/Fail
1	Open Disciplinary Form from the Menu	Disciplinary form should be opened and program has to wait for user input for Employee ID	✓
2	Enter an Employee ID that does not exist.	Message should appear "Employee ID not found".	✓
3	Enter an Employee ID that is already entered.	System should display employee summary info and fill the table with all disciplinary details entered so far.	✓
4	Double click a row in the disciplinary table.	System has to show all the details of the selected disciplinary entry. User can update or delete in this screen.	✓
5	Modify any data and click Save button.	System should save disciplinary details, display a confirmation message and go back to screen that shows all disciplinary entries in the table.	✓
6	Click New Button.	Disciplinary details panel should be opened with empty values in all fields. User can start entering values here.	✓
7	Input an offence date that already exists.	System has to show message "This offence date has already been added in disciplinary details".	✓
8	Input valid values and click Save button.	All data for disciplinary should be saved. Message should appear "Disciplinary entry added/updated successfully"	✓
9	Select a disciplinary entry and click delete button.	Selected entry should be removed. System should prompt a confirmation message "Are you sure to delete this disciplinary data?" If clicked yes, the disciplinary entry should be removed. Message should be shown "Disciplinary data has been deleted successfully".	✓

Table E.4: Test case for Disciplinary Module

Test case for Sick Leave Module

Table E.5 below provides the test case for sick leave module.

Test Case No	Test Case	Expected Result	Success/Fail
1	Open Sick Leave Form from the Menu	Sick leave form should be opened and program has to wait for user input for Employee ID	✓
2	Enter an Employee ID that does not exist.	Message should appear "Employee ID not found".	✓
3	Enter an Employee ID that is already entered.	System should display employee summary info and fill the table with all Sick Leave details entered so far.	✓
4	Double click a row in the Sick Leave table.	System has to show all the details of the selected Sick Leave entry. User can update or delete in this screen.	✓
5	Modify any data and click Save button.	System should save sick leave details, display a confirmation message and go back to screen that shows all sick leave entries in the table.	✓
6	Click New Button.	Sick leave details panel should be opened with empty values in all fields. User can start entering values here.	✓
7	Input a reported date that already exists.	System has to show message "This reported date already exists".	✓
8	Input valid values and click Save button.	All data for sick leave should be saved. Message should appear "sick leave entry added/updated successfully"	✓
9	Select a sick leave entry and click delete button.	Selected entry should be removed. System should prompt a confirmation message "Are you sure to delete this sick leave data?" If clicked yes, the sick leave entry should be removed. Message should be shown sick leave data has been deleted successfully".	✓

Table E.5: Test case for Sick Leave Module

Test case for Accrued Leave Module

Table E.6 below provides the test case for accrued leave module.

Test Case No	Test Case	Expected Result	Success/ Fail
1	Open Accrued Leave Form from the Menu	Accrued Leave form should be opened and program has to wait for user input for Employee ID	✓
2	Enter an Employee ID that does not exist.	Message should appear "Employee ID not found".	✓
3	Enter an Employee ID that is already entered.	System should display employee summary info and fill the table with all accrued leave details entered so far.	✓
4	Double click a row in the accrued leave table.	System has to show all the details of the selected accrued leave entry. User can update or delete in this screen.	✓
5	Modify any data and click Save button.	System should save accrued leave details, display a confirmation message and go back to screen that shows all accrued leave entries in the table.	✓
6	Click New Button.	Accrued leave details panel should be opened with empty values in all fields. User can start entering values here.	✓
7	Input a Memo Date that already exists.	System has to show message "This Memo Date already exists".	✓
8	Input any alpha characters in No of Days Field	Message should appear "Invalid Entry. Please input only numbers."	✓
9	Input valid values number of days and remark and click Save button.	All data for accrued leave should be saved. Message should appear "accrued leave entry added/updated successfully"	✓
10	Select an accrued leave entry and click delete button.	Selected entry should be removed. System should prompt a confirmation message "Are you sure to delete this accrued leave?" If clicked yes, the accrued leave entry should be removed. Message should be shown accrued leave data has been deleted successfully".	✓

Table E.6: Test case for Accrued Leave

Test case for Termination Module

Test case for termination module is provided below in Table E.7.

Test Case No	Test Case	Expected Result	Success/Fail
1	Open Terminations screen from the Menu	Termination form should be opened and program has to wait for user input for Employee ID	✓
2	Enter an Employee ID that does not exist.	Message should appear "Employee ID not found".	✓
3	Enter a valid employee ID that is active.	System has to show name, department etc., and should allow entry of Termination Date and Termination Reason.	✓
4	Input an invalid value for date such as alpha characters.	System should show message "Invalid Entry. Input only valid numbers".	✓
5	Input an invalid date such as "55/55/5555"	System should show message "Invalid date value. Please input a valid".	✓
6	Input valid date, select a termination reason from drop down and click Save.	Termination details should be updated. Employee status should be made inactive.	✓
7	Close form, input the same Employee ID, remove the termination date and reason, and click save.	Termination entry should be removed. Employee status should be made active again.	✓

Table E.7: Test case for Termination

Appendix F - Code Listing

Import data from text file to SQL database

Below code is used to import data exported from legacy system to SQL database.

```
private void LoadData(string zFile, string tableName, Int16 fieldCount)
{
    string sql = "INSERT INTO " + tableName + " values (";
    string textLine = "";
    for (int i = 1; i <= fieldCount; i++)
    {
        if (i > 1) sql = sql + ",";
        sql = sql + "@v" + i.ToString();
    }
    sql = sql + ");";

    //very if the file exists
    if ((System.IO.File.Exists(zFile) == true))
    {
        System.IO.StreamReader objReader = new
System.IO.StreamReader(zFile);
        while ((objReader.Peek() != null))
        {
            textLine = (objReader.ReadLine());
            if (textLine == "" || textLine == null) break;

            textLine = "~" + textLine;
            string[] values = textLine.Split(new Char[] { '~' });

            label1.Text = "updating " + values[1];
label1.Refresh();
            Application.DoEvents(); Application.DoEvents();

            SqlCommand cmd = new SqlCommand(sql, conn);
            for (int i = 1; i <= fieldCount; i++)
            {
                cmd.Parameters.AddWithValue("@v" + i.ToString(),
(values[i] ?? (object) DBNull.Value));
            }
            cmd.ExecuteNonQuery();
        }
    }

    // notify that import is done.
    label1.Text = "import completed for " + tableName;
    label1.Refresh();
}
}
```

Code used in saving module of transaction program

```
// call the save module and display message appropriately
if (save_data())
{
    if (mode == FormMode.AddMode)
    {
        MessageBox.Show("New employee record is added successfully");
        DisplayMode();
    }
    else
    {
        MessageBox.Show("Employee details are updated successfully");
    }
    return true;
}
else { return false; }
```

Code used to get brief information of employee

```
// Get brief info about employee.
internal void GetBriefEmployeeInfo()
{
    DataTable dt = gsCommon.getDbTable("select emp.* " +
        ",(select description from sections where
        departmentcode=emp.departmentcode and sectioncode=(select sectioncode
        from contracts where employeeid='" + EmployeeId + "' and
        effectivedate=(select max(effectivedate) from contracts where
        employeeid='" + EmployeeId + "')) SectionName " +
        "from (" +
        "select (FirstName + ' ' + SecondName + ' ' + LastName) FullName
        " +
        ",(select departmentcode from contracts where employeeid='" +
        EmployeeId + "' and effectivedate=(select max(effectivedate) from
        contracts where employeeid='" + EmployeeId + "')) departmentcode" +
        ",(select description from departments where departmentcode =
        (select departmentcode from contracts where employeeid = '" + EmployeeId
        + "' and effectivedate = (select max(effectivedate) from contracts where
        employeeid ='" + EmployeeId + "')) DepartmentName " +
        ",(case when empStatus='1' then 'ACTIVE' else 'INACTIVE' end)
        employeeStatus " +
        ",ProjectDateOfJoin,CompanyDateOfJoin    from employees where
        employeeid='" + EmployeeId + "' ) emp");
    FullName = dt.Rows[0]["FullName"].ToString();
    DepartmentName = dt.Rows[0]["DepartmentName"].ToString();
    SectionName = dt.Rows[0]["SectionName"].ToString();
    ProjectDateOfJoin =
    Convert.ToDateTime(dt.Rows[0]["ProjectDateOfJoin"]).ToString("dd/MM/yyyy"
    );
    CompanyDateOfJoin =
    Convert.ToDateTime(dt.Rows[0]["CompanyDateOfJoin"]).ToString("dd/MM/yyyy"
    );
}
```

Code used in saving a new hire transaction to database

```
// Insert new employee
internal bool NewHire()
{
    string connectionString = gsCommon.getConnectionString();

    // insert new employee details
    try
    {
        using (SqlConnection myConnection = new SqlConnection())
        {
            myConnection.Close();
            myConnection.ConnectionString = connectionString;
            myConnection.Open();
            SqlCommand myCommand = myConnection.CreateCommand();

            myCommand.CommandText = "INSERT INTO [dbo].[Employees] " +
                "([EmployeeId],[ProjectCode],[EmpStatus],[FirstName],[SecondName],[LastNa" +
                "me],[NationalityCode]" +
                ", [ReligionCode],[DateOfBirth],[ProjectDateOfJoin],[CompanyDateOfJoin],[O" +
                "ldPayrollId]" +
                ", [SponsorCode],[ContractYears],[Remarks],[ProbationEnd],[Address1],[Addr" +
                "ess2],[Address3],[Address4]" +
                ", [ContactPhone],[NameOfFather],[NameOfMother],[NameOfSpouse],[SummerStud" +
                "ent],[MarineTrainee])" +
                " VALUES " +
                "(@EmployeeId, @ProjectCode," +
                "@EmpStatus, @FirstName, @SecondName, @LastName, @NationalityCode" +
                ", @ReligionCode, @DateOfBirth," +
                "@ProjectDateOfJoin, @CompanyDateOfJoin, @OldPayrollId" +
                ", @SponsorCode, @ContractYears," +
                "@Remarks, @ProbationEnd, @Address1, @Address2, @Address3, @Address4" +
                ", " +
                "@ContactPhone,@NameOfFather,@NameOfMother,@NameOfSpouse,@SummerStudent,@M" +
                "arineTrainee)";

            myCommand.Parameters.AddWithValue("@EmployeeId", EmployeeId);
            // initially in the new hire employee status is set to "2".
            // This would be made to "1" when contract is created.
            // status "1" means employee is active
            myCommand.Parameters.AddWithValue("@EmpStatus", "2");

            myCommand.Parameters.AddWithValue("@ProjectCode",
            ProjectCode);
            myCommand.Parameters.AddWithValue("@FirstName", FirstName);
            myCommand.Parameters.AddWithValue("@SecondName", SecondName);
            myCommand.Parameters.AddWithValue("@LastName", LastName);
            myCommand.Parameters.AddWithValue("@NationalityCode",
            NationalityCode);
            myCommand.Parameters.AddWithValue("@ReligionCode",
            ReligionCode);

            myCommand.Parameters.AddWithValue("@DateOfBirth",
            DateOfBirth);
            myCommand.Parameters.AddWithValue("@ProjectDateOfJoin",
```

```

ProjectDateOfJoin);
    myCommand.Parameters.AddWithValue("@CompanyDateOfJoin",
CompanyDateOfJoin);
    myCommand.Parameters.AddWithValue("@OldPayrollId",
OldPayrollId);
    myCommand.Parameters.AddWithValue("@SponsorCode",
SponsorCode);
    myCommand.Parameters.AddWithValue("@ContractYears",
ContractYears);
    myCommand.Parameters.AddWithValue("@Remarks", Remarks);

    myCommand.Parameters.AddWithValue("@ProbationEnd",
ProbationEnd);

    myCommand.Parameters.AddWithValue("@Address1", Address1);
    myCommand.Parameters.AddWithValue("@Address2", Address2);
    myCommand.Parameters.AddWithValue("@Address3", Address3);
    myCommand.Parameters.AddWithValue("@Address4", Address4);
    myCommand.Parameters.AddWithValue("@ContactPhone",
ContactPhone);
    myCommand.Parameters.AddWithValue("@NameOfFather",
NameOfFather);
    myCommand.Parameters.AddWithValue("@NameOfMother",
NameOfMother);
    myCommand.Parameters.AddWithValue("@NameOfSpouse",
NameOfSpouse);
    myCommand.Parameters.AddWithValue("@SummerStudent",
(SummerStudent == true ? "X" : ""));
    myCommand.Parameters.AddWithValue("@MarineTrainee",
(MarineTrainee == true ? "X" : ""));
    myCommand.ExecuteNonQuery();
    return true;
}
}
catch (Exception ex)
{
    throw new Exception(ex.Message);
}
}

```

Code used to retrieve information of employee

```
// Extract employee information
internal void GetEmployeeInfo()
{
    string connectionString = gsCommon.getConnectionString();
    DataTable dt=new DataTable();

    using (SqlConnection myConnection = new SqlConnection())
    {
        myConnection.Close();
        myConnection.ConnectionString = connectionString;
        myConnection.Open();
        string qry = "select * from employees where employeeID='" +
EmployeeId + "'";
        using (SqlDataAdapter reader = new SqlDataAdapter(qry, myConnection))
        {
            dt.Clear();
            reader.Fill(dt);
            if (dt.Rows.Count > 0)
            {
                ProjectCode =
gsReferences.getProjectDescription(dt.Rows[0]["ProjectCode"].ToString());
                FirstName = dt.Rows[0]["FirstName"].ToString();
                SecondName = dt.Rows[0]["SecondName"].ToString();
                LastName = dt.Rows[0]["LastName"].ToString();
                NationalityCode =
gsReferences.getNationalityDescription(dt.Rows[0]["NationalityCode"].ToString
());
                ReligionCode =
gsReferences.ReligionDescription(dt.Rows[0]["ReligionCode"].ToString());
                DateOfBirth =
Convert.ToDateTime(dt.Rows[0]["DateOfBirth"]).ToString("dd/MM/yyyy");
                ProjectDateOfJoin =
Convert.ToDateTime(dt.Rows[0]["ProjectDateOfJoin"]).ToString("dd/MM/yyyy");
                CompanyDateOfJoin =
Convert.ToDateTime(dt.Rows[0]["CompanyDateOfJoin"]).ToString("dd/MM/yyyy");
                OldPayrollId = dt.Rows[0]["OldPayrollId"].ToString();
                SponsorCode =
gsReferences.getSponsorDescription(dt.Rows[0]["SponsorCode"].ToString());
                ContractYears =
Convert.ToInt16(dt.Rows[0]["ContractYears"].ToString());
                Remarks = dt.Rows[0]["Remarks"].ToString();

                ProbationEnd =
Convert.ToDateTime(dt.Rows[0]["ProbationEnd"]).ToString("dd/MM/yyyy");
                Address1 = dt.Rows[0]["Address1"].ToString();
                Address2 = dt.Rows[0]["Address2"].ToString();
                Address3 = dt.Rows[0]["Address3"].ToString();
                Address4 = dt.Rows[0]["Address4"].ToString();
                ContactPhone = dt.Rows[0]["ContactPhone"].ToString();
                NameOfFather = dt.Rows[0]["NameOfFather"].ToString();
                NameOfMother = dt.Rows[0]["NameOfMother"].ToString();
                NameOfSpouse = dt.Rows[0]["NameOfSpouse"].ToString();

                SummerStudent = (dt.Rows[0]["SummerStudent"].ToString() ==
"X" ? true : false);
                MarineTrainee = (dt.Rows[0]["MarineTrainee"].ToString() ==
"X" ? true : false);
            }
        }
    }
}
```

Appendix G - Client Certificate

<p>شركة مقاولات الخليج للشحن والتفريغ المحدودة مقاول عقد تأجير محطة الحاويات الشمالية مس.ب. ١٩٨٦٠ - جدة ٢١٤٤٥ - المملكة العربية السعودية تليفون : ٢٨٤٦٦٦٠ / ٠١٢ ٢٨٤٦٦٦٦ فاكس : ٢٤٧٩٦٠٦ / ٠١٢ ٢٨٤٦٨٨٥ GULF STEVEDORING CONTRACTING CO. LTD., LEASE CONTRACTOR FOR NORTH CONTAINER TERMINAL P.O. Box 19860 Jeddah 21445, Kingdom of Saudi Arabia Tel. 012 2856686 / 2856780 Fax. 012 2856885 / 6479606</p>			<p>المملكة العربية السعودية KINGDOM OF SAUDI ARABIA المؤسسة العامة للموانئ SEAPORT AUTHORITY ميناء جدة الإسلامي JEDDAH ISLAMIC PORT</p>
---	---	--	---

06th November 2017

Project Examination Board
University of Colombo School of Computing
221/2A, Dharmapala Mawatha
Colombo 7

Dear Sir/Madam

LETTER OF CERTIFICATION

This is to certify that Mohamed Irshad Mohideen (R092391), has developed a HR system as one of the requirements of Bachelor of Information Technology Degree. This system is a migration from previous text-only based legacy system.

We are pleased to state that this system has provided us many rich features that were lacking in the previous legacy program. This system meets our requirements and allows us to further enhance in the future.

Thank you



Seif Raghieb Seif
HR Manager





<p>Company with Limited Liability Paid Capital 2,000,000 S.R. HEAD OFFICE : P. O. Box : 1153 Al-Khobar 31952 - King Abdul Aziz Street Kingdom of Saudi Arabia - C. R. : 2051011075 Tel. : (013) 3566993 - 3566929 Fax : (013) 3628144 - 3566905</p>	<p>شركة ذات مسؤولية محدودة / رأس المال ريال سعودي مدفوع بالكامل المكتب الرئيسي : ص.ب. ١١٥٣ الدبر ٣١٩٥٢ - شارع الملك عبد العزيز المملكة العربية السعودية - مس.ب. ٣٠١١٠٧٥ تليفون : ٣٥٦٦٩٩٣ (١٢) - ٣٥٦٦٩٢٩ فاكس : ٣٦٢٨١٤٤ (١٢) - ٣٥٦٦٩٠٥</p>
--	---

Glossary

- MS SQL - SQL DATABASE FROM MICROSOFT WITH RELATIONAL MANAGEMENT FUNCTION.
- SQL - STRUCTURED QUERY LANGUAGE USED TO SEND INSTRUCTIONS TO DATABASE AND EXTRACT INFORMATION OR COMMIT CHANGES TO DATA.
- UML - KNOWN AS UNIFIED MODELLING LANGUAGE. IT HAS COLLECTION OF TOOLS TO DEVELOP DIAGRAMS.
- RUP - RATIONAL UNIFIED PROCESS. A COMMONLY KNOWN DESIGN METHODOLOGY
- SAP - AN ERP SYSTEM FROM A GERMANY SOFTWARE COMPANY. SAP STANDS FOR SYSTEM, APPLICATION AND PRODUCTS IN DATA PROCESSING.

Index

A

Acceptance Testing, 43
access, 30
Accrued Leave, 8
Application, 64

C

Caché, ii, 1, 17, 52
class, 40
class diagram, 21
Client, 83
Consistency, 25
Contract, 7
Contracts, 31
Crystal report, 34

D

Department, 69
deployment, 55
Design Methodology, 19
development, 33
Development, 52
Disciplinary, 7
Documentation, 57
Dynamic, 42

E

encrypts, 41
enhancements, 10
Enhancements, 51
environment, 15
evaluation, 42
Excel, 34
experience, 52

F

Familiarity, 25
Feasibility, 18
Feedback, 48
Functional, 12

G

GSCC, ii, 1, 6

I

instance, 39
Integration Testing, 43
IntelliSense, 34
interactions, 25

L

Language, xi, 19, 21, 34
Leave, 7
login, 29
Login, 36, 44, 64

M

management, 8
matrix, 17
Microsoft, 34
Modules, 8
Motivation, 1

N

Nationality, 70, 71
Normalization, 24

P

panel, 64
Performance, 12
Processor, 33
property, 37

R

Recoverability, 25
References, 66
Reports, 66, 68
RUP, ii

S

Scalability, 13
Security, 12
Sequence Diagram, 22
SQL, 36
SQL Server, 55
Static, 42

Summary, 70
System, ii, 64
System design, 15

T

Termination, 8
Test Case, 44
transaction, 39, 79
Transactions, 65

U

Unit Testing, 43
User Diversity, 25
User Friendliness, 12
User Guidance, 25
user interface, 25

V

Visual, 26