Software Requirements Specification

Version 2.0

Submission Date : 17 January 2016

Automated Time Table

Submitted in partial fulfillment Of the requirements of

CS223 Software Engineering

This work is based upon the submissions of the course Software Engineering (CS223).

The students who submitted this team projects are Ayush Agrawal (B14CS010)

Akshay Chaudhari (B14CS013) Ganesh Patil (B14CS017) Varun Kumar (B14CS039)

Changes made from previous version :

1. Modified and Added new Use Cases ­ User LogIn, Update Data, Add Course, Delete Course, Edit Course, Add Classroom data, Verifying Timetable, Generation of Timetable, Log out , Notify Change , Display Timetable
2. Updated Glossary
3. Added one new Assumption

## No table of figures entries found.

**List of Figures**

* 1. **Introduction**

### *Purpose*

The purpose of this project is to ….

* + - Provide students/faculty/staff with efficient timetable.
    - Reduce the manual work of office of academics.
    - Reduce the clashes between the faculty timings.
    - Eradicate the slot clashes (Backlog/Elective courses).
    - Avoid the room clashes.
    - Make the application simple, easily understandable, efficient and portable.
    - Focus on optimization of resources i.e teachers, labs and rooms.

### *Scope of Project*

This software system will be designed to ….

* + - Give access to same online portal for all(students/faculties/staff).
    - Users can see their specific processed time table only.
    - To suffice Educational Institution Purpose only.
    - Provision of altering slots on request of students/faculty.
    - Add­Drop Courses: clash notifications
    - Modification is much easier and accurate than the manual system

### *Constraints:*

System constraints are divided into 2 categories:

1. Primary Constraints: The timetable is subjected to the following four types of primary constraints, which must be satisfied by a solution to be considered as a valid one:
   1. A student should have only one class at a time.
   2. An instructor should have only one class at a time.
   3. A room should be busy only for one class at a time.
   4. Some rooms require to have particular equipments. e.g. NKN,projectors.
2. Secondary Constraints: These are the constraints that are of no great concern but are still taken into consideration. They don't need to be satisfied but the solutions are generally considered to be good if they are satisfied.
   1. Courses must be evenly distributed.
   2. Students should not have any free time between two classes on a day.
   3. Scheduling of teachers should be well spread over the week.
   4. Faculty should not have consecutive classes.

### *Assumptions and Dependencies*

* + - Assumption:
      * Admin will provide the data of the courses.
      * Optimum number of rooms are available as per the number of courses and faculty.
      * Users are assumed to already have registered LDAP credentials and thus no need to sign up here.
      * Besides provision of “changing password” is not available here as it uses LDAP credentials.
      * Internet connection is always available.
    - Dependencies:
      * Details of the Courses (faculty, number of expected students)
      * Classroom Data (seating capacity, NKN facility)
      * Internet connection

### *Glossary*

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Schedule | An ordered list of times at which things are planned to occur |
| Database | The collection of the data to be used for the generation of Time Table |
| NKN | National Knowledge Network |
| User | Person who uses the software system(Admin, Faculty, Student) |
| Admin | Person in the administration who uses the software system to create Time Table |
| Faculty | Instructor for the courses mentioned in the Time Table |
| Student | Person registered for the courses mentioned in the Time Table |
| Master Timetable | one containing all courses’ timetable for all students and faculty ,not specific for a particular user |

* 1. ***References***

IEEE. *IEEE Std 830­1998 IEEE Recommended Practice for Software Requirements Specifications.* IEEE Computer Society, 1998.

* + - Research papers : Journal of Engineering Research and Applications
    - Web: [www.ijera.com](http://www.ijera.com/), [www.kent.ac.uk/timetabling](http://www.kent.ac.uk/timetabling)
    - Others : International Journal of Computer Applications Volume 127

### *Overview of Document*

The rest of the document is designed in the following way:

# 2.0. Overall Description

### *System Environment*

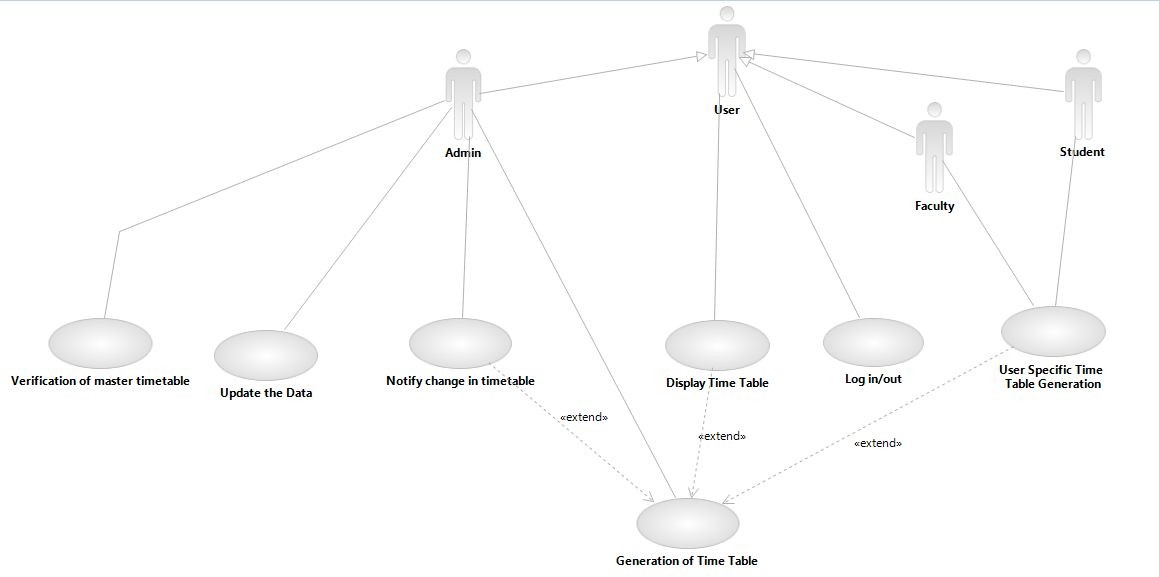
<< Keep blank for the time being >>

### *Functional Requirements Specification*

<< Keep blank for the time being >>

* + 1. Use case 1

Use case: **Generation of Timetable (Interface) Diagram:**

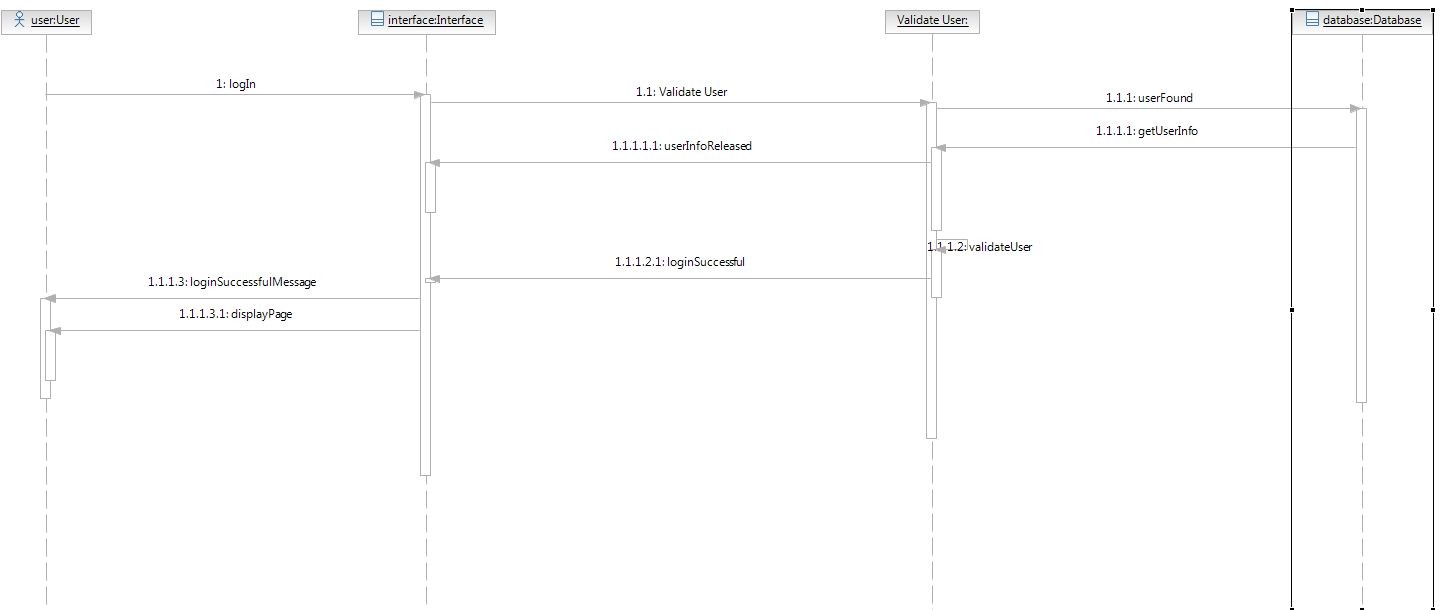


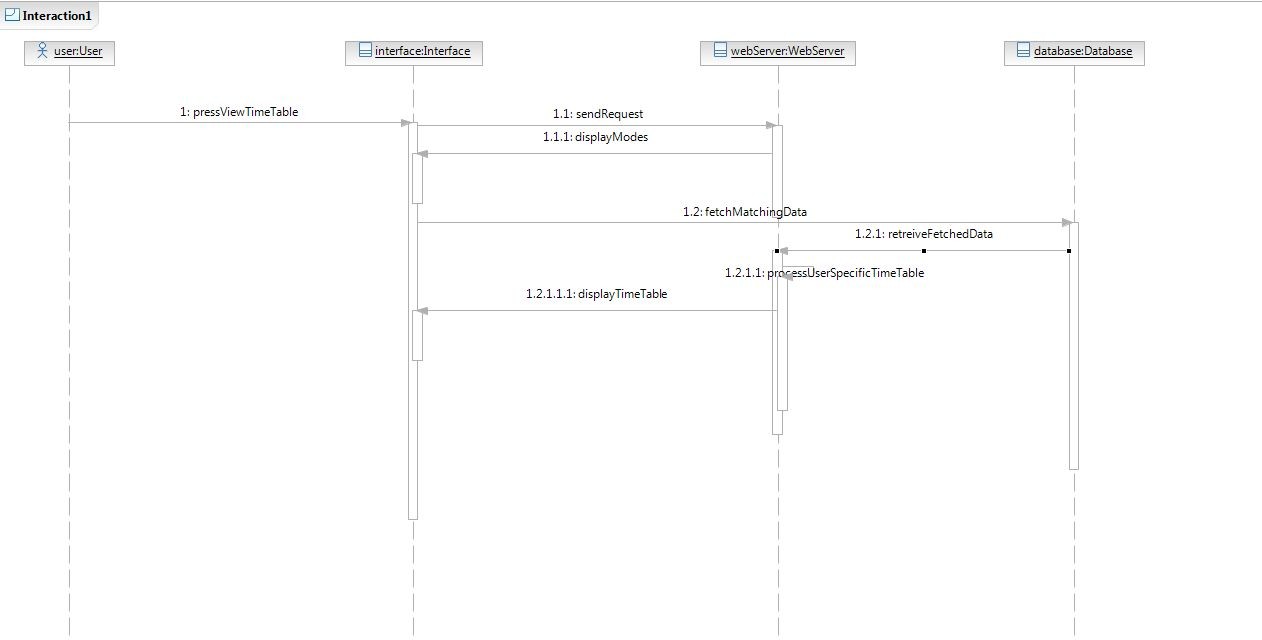
**Brief Description**

This use case diagram describes what are the functionalities a user can have in the system.It considers all users point of view.

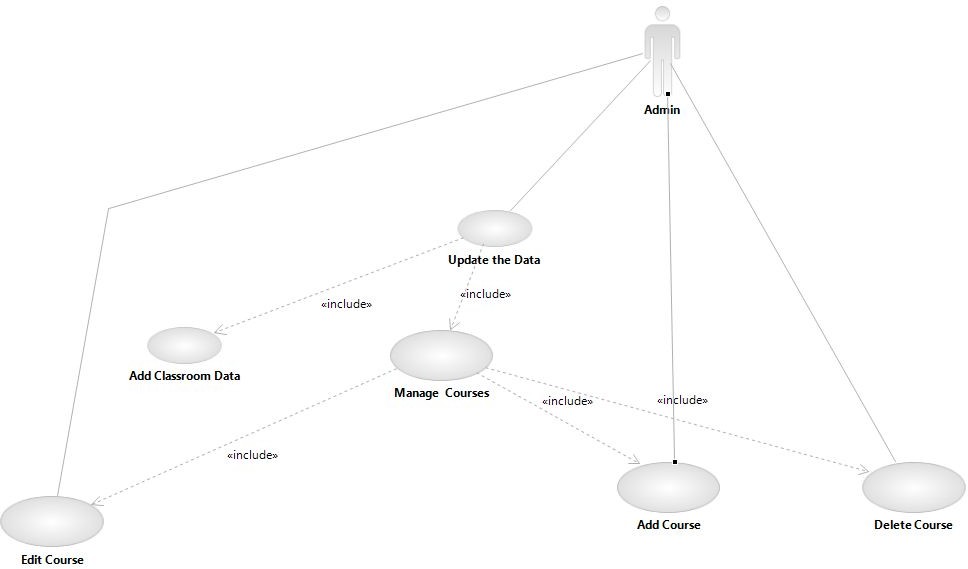
## Initial Step­By­Step Description Sequence Diagrams

1. Login



1. Display Time Table
   * 1. Use case

Use case: **Update Data Diagram:**

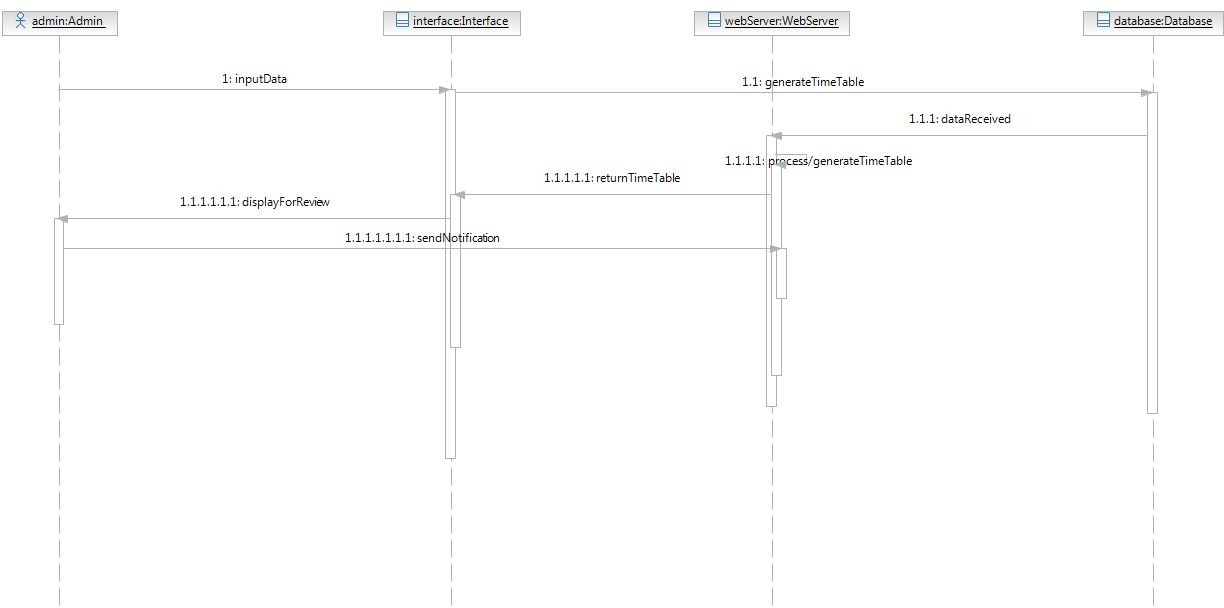


## Brief Description

This use case diagram describes about updating course and Classroom data in given database .Only admin can update this data.

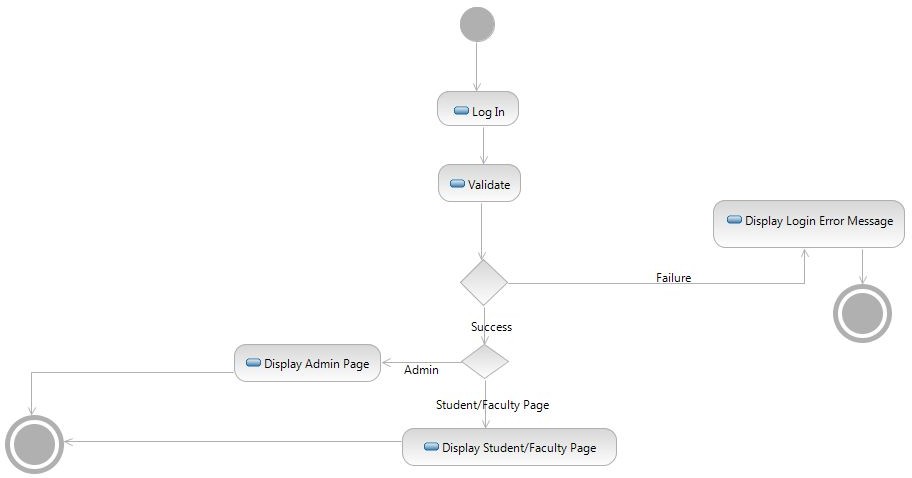
## Initial Step­By­Step Description

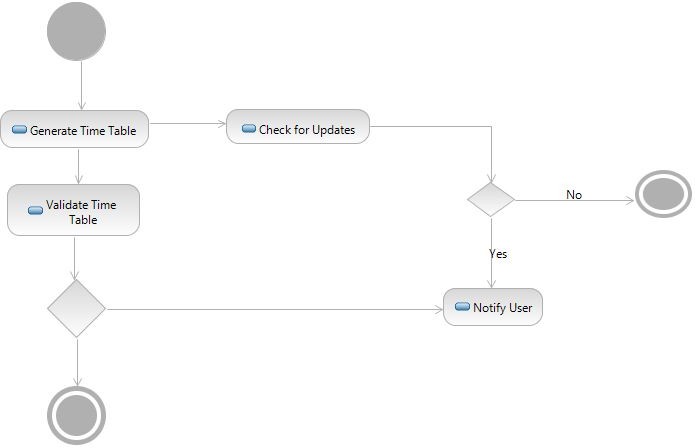
**Sequence Diagram**

1. Update Data

## Activity Diagrams

* 1. Log in



* 1. Generate timetable and notify

### *User Characteristics*

The user of this software system requires the following skills to use this software:

* General Internet skills.

### *Non­Functional Requirements*

* Safety
* Reliability
* Availability

# 3.0. Requirements Specification

### *Functional Requirements*

* + 1. User Log in

|  |  |
| --- | --- |
| **Use Case Name** | User log in |
| **Trigger** | Nil |
| **Precondition** | User has already registered |
| **Basic Path** | 1. User will enter the required credentials and system will validate it. 2. If the credentials are correct it will redirect it to   concerned interface. |
| **Alternative Paths** |  |
| **Postcondition** | User will be logged in. |
| **Exception Paths** | If the credentials are wrong, show an error message |
| **Other** |  |

* + 1. Update Data

|  |  |
| --- | --- |
| **Use Case Name** | Updating the data |
| **Trigger** |  |
| **Precondition** | System has the Database |
| **Basic Path** | Admin enters the insert Query to add course in the Database |
| **Alternative Paths** |  |
| **Postcondition** | Course is added |
| **Exception Paths** | Course already exists |
| **Other** |  |

MANAGE COURSE :­

* + 1. Add Course

|  |  |
| --- | --- |
| **Use Case Name** | Adding Course to the Database |
| **Trigger** | Admin presses the “Add Course” key |
| **Precondition** | System has the Database |
| **Basic Path** | Admin enters the insert Query to add course in the Database |
| **Alternative Paths** |  |
| **Postcondition** | Course is added |
| **Exception Paths** | Course already exists |
| **Other** |  |

* + 1. Delete Course

|  |  |
| --- | --- |
| **Use Case Name** | Deleting Course from the Database |
| **Trigger** | Admin presses the “Delete Course” key |
| **Precondition** | System has the Database |
| **Basic Path** | Admin enters the delete Query to delete course from the Database |
| **Alternative Paths** |  |
| **Postcondition** | Course Deleted |
| **Exception Paths** |  |
| **Other** |  |

* + 1. Edit Course

|  |  |
| --- | --- |
| **Use Case Name** | Editing the existing Course in the Database |
| **Trigger** | Admin presses the “Edit Course” key |
| **Precondition** | Course already exist in System Database |
| **Basic Path** | Admin enters the alter Query to edit course in the Database |
| **Alternative Paths** |  |
| **Postcondition** | Course Edited |
| **Exception Paths** | Course does not exist |
| **Other** |  |

* + 1. Add classroom data

|  |  |
| --- | --- |
| **Use Case Name** | Add classroom data |
| **Trigger** | Admin presses the “Add classroom data” key |
| **Precondition** | Admin has the classroom data |
| **Basic Path** | Admin enters the insert Query to add classroom data in the Database |
| **Alternative Paths** |  |
| **Postcondition** | Classroom data added |
| **Exception Paths** |  |
| **Other** |  |

* + 1. Verification of master timetable

|  |  |
| --- | --- |
| **Use Case Name** | Verifying master timetable |
| **Trigger** | Master timetable is displayed to the admin |
| **Precondition** | Master timetable is generated |
| **Basic Path** | Admin verifies the master timetable |
| **Alternative Paths** |  |
| **Postcondition** | Master timetable is verified |
| **Exception Paths** |  |
| **Other** |  |

* + 1. Generation of Timetable

|  |  |
| --- | --- |
| **Use Case Name** | Generating timetable |
| **Trigger** | Admin presses the “Generate Time Table” key |
| **Precondition** | System has the data of all the courses and faculties in the Database |
| **Basic Path** | Software will use algorithm on the Data to create the optimum Timetable. |
| **Alternative Paths** | Nil |
| **Postcondition** | Automated timetable is generated |
| **Exception Paths** |  |
| **Other** |  |

* + 1. User Log out

|  |  |
| --- | --- |
| **Use Case Name** | User Log Out |
| **Trigger** | Presses the logout button |
| **Precondition** | User has to be already logged in |
| **Basic Path** | 1. If the user presses the log |
| **Alternative Paths** | logout based on time limit crossed without any activity or time out |
| **Postcondition** | Successfully Logged out |
| **Exception Paths** | Network problem |
| **Other** |  |

* + 1. Notify change in Timetable

|  |  |
| --- | --- |
| **Use Case Name** | Notifying the change in timetable |
| **Trigger** | A change is made in time table |
| **Precondition** | Timetable exists |
| **Basic Path** | Admin sends e­Mail to every user having the new Timetable |
| **Alternative Paths** |  |
| **Postcondition** | Users are notified about the change in timetable |
| **Exception Paths** |  |
| **Other** |  |

* + 1. User specific Timetable generation

|  |  |
| --- | --- |
| **Use Case Name** | User specific Timetable generation |
| **Trigger** | General timetable is confirmed by the admin. |
| **Precondition** | A general timetable has been generated |
| **Basic Path** | 1. System takes data of every student and the respective courses. 2. According to the courses and students, generate their   timetable. |
| **Alternative Paths** |  |
| **Postcondition** | User specific timetable is generated |
| **Exception Paths** |  |
| **Other** |  |

* + 1. Display Timetable

|  |  |
| --- | --- |
| **Use Case Name** | Display the timetable |
| **Trigger** | User presses the “View Timetable” button |
| **Precondition** | User logged in and generated Timetable exists |
| **Basic Path** | Software fetched the generated Timetable from saved memory |
| **Alternative Paths** |  |
| **Postcondition** | Timetable is displayed to the user |
| **Exception Paths** |  |
| **Other** |  |

### *Detailed Non­Functional Requirements*

* + - Reliability: As the system provide the right tools for discussion, problem solving it must be made sure that the system is reliable in its operations and for securing the sensitive details.
    - Safety: Information transmission should be securely transmitted to server without any changes in information.
    - Availability: If the internet service gets disrupted while sending information to the server, the information can be send again for verification.

### *Logical Structure of the Data*

<< Keep this blank for the time being>>

### *Database Tables*

|  |  |
| --- | --- |
| **Tables** | Description |
| Faculty and Student Tables | Store Faculty and student details, which include Username and Password. It also stores administrator's Username and Password.Faculty and student will be able to view own details, while the administrator can update and edit all the lecturer and student details(except username and password). |
| Classroom Table | Store all the class details of. Only the administrator are allowed to insert, update and delete the data in the tables |
| Course Table | Store all the course details of. Only the administrator are allowed to insert, update and delete the data in the tables |

* 1. **Supporting information**
  2. ***Table of contents and index***

***Index:***

|  |  |
| --- | --- |
| **Content** | Page No |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

* 1. ***Appendices***