

# **AVITAS DATA ANALYSIS AND REPORTING PLATFORM**

**SREERAM V**

**SHIV NADAR UNIVERSITY  
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING  
SCHOOL OF ENGINEERING  
2015**

## B.Tech. Project Report

### Avitas Data Analysis and Reporting Platform

By  
Sreeram V

### Supervisors

Divya Lohani  
Assistant Professor, Computer Science And Engineering  
[Divya.Lohani@snu.edu.in](mailto:Divya.Lohani@snu.edu.in)

Suresh Sarojani  
Chief Technical Officer, HCL Avitas  
[Suresh.Sarojani@hcl.com](mailto:Suresh.Sarojani@hcl.com)

Submitted in the partial fulfillment of requirements for  
B. Tech. in Computer Science and Engineering

**SHIV NADAR UNIVERSITY**

Department of Computer Science and Engineering,  
School of Engineering, Shiv Nadar University,  
Gautam Buddha Nagar, U.P., India, 201314  
<http://www.snu.edu.in>

## **Approval Sheet**

This report entitled Avitas Data Analysis and Reporting Platform by Sreeram V is approved for the degree of B.Tech in Computer Science and Engineering.

### Project Advisor(s)

Name            Divya Lohani

Signature \_\_\_\_\_

Date \_\_\_\_\_

Name            Suresh Sarojani

Signature \_\_\_\_\_

Date \_\_\_\_\_

## **Declaration Sheet**

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources, which have thus not been properly cited, or from whom proper permission has not been taken when needed.

Name                    Sreeram V

Signature \_\_\_\_\_

Date \_\_\_\_\_

## **Abstract**

HCL Healthcare is the newest business venture of HCL Corporation, the parent company of HCL Technologies Ltd and HCL Infosystems. HCL Avitas, in affiliation with Johns Hopkins Medicine International, is the healthcare delivery arm of HCL Healthcare. Starting with the country's first nationwide-networked multispecialty clinics, HCL Avitas aims to provide the whole continuum of care for chronic and acute diseases.

Since its inception, Avitas has generated huge amounts of data, critical to the understanding and improvement of quality of services provided and also for accelerating the growth of the company. Thus, there is a need for a Management Information System (MIS) equipped with the required Business Intelligence (BI). The two main applications being used by Avitas for record management are products outsourced from two other companies. The much-needed data for analysis are in different formats on different servers. The project involves building a reporting server within Avitas, which gets real time updates of the required data from all the external servers, and then a BI over the data collected. The first step involves data cleaning, data integrity checks and database modeling and the next involves using the right set of tools and techniques to create a system that enables its end user to play with the data in every possible way. This stack when connected with a web based front end (not a part of this report) would enable the organization to visualize the vast data seamlessly and in real time to support decision making.

# Contents

	<i>PAGE NUMBER</i>
<b>1. Introduction</b>	<b>2</b>
<b>2. Requirement Analysis And Feasibility Study</b>	<b>5</b>
2.1. Project Description And Scope	5
2.2. Domain Analysis	5
2.3. Preliminary Requirement Analysis	6
2.4. Technical Feasibility	6
2.5. Software Development Process	7
2.6. Visibility Plan	8
2.7. Risk Analysis	9
2.8. Conclusion	9
<b>3. Software Requirements</b>	<b>11</b>
3.1. External Dependencies	11
3.2. Functions	11
3.3. Software System Attributes And Performance	13
<b>4. Project Schedule</b>	<b>14</b>
4.1. Timeline	16
4.2. Gantt Chart	18
4.3. Lines Of Code	21
<b>5. UML Diagrams and SQL DB Schemas</b>	<b>22</b>
5.1. UML Class Objects (Java Component) and Dependencies	22
5.2. SQL DB Schema (Main Tables and Relations)	23

*PAGE NUMBER*

<b>6. SQA Plan</b>	<b>24</b>
6.1. SQA Requirements and Implementation	24
6.2. SQA Testing	26
<b>7. Test Report</b>	<b>29</b>
<b>8. Conclusion</b>	<b>44</b>
<b>9. References</b>	<b>45</b>

## **List of Figures**

<b>Figure</b>	<b>Title</b>	<b>Page Number</b>
1.1	Project Requirements and Report Scope	3
2.1	Development Model	7
2.2	Calender View: Weekly Review Meeting	8
4.1	Development – Phase 1	20
4.2	Development – Phase 2	21
5.1	UML Class Objects (Java Component) and Dependencies	23
5.2	SQL DB Schema (Main Tables and Relations)	24
7.1	Tables in the Databases of the reporting server (MS SQL Server)	30
7.2	Update Job scheduled for everyday	30
7.3	Operational Analysis Dashboard	31
7.4	Clinical Analysis Dashboard	32
7.5	Marketing Analysis Reports	33
7.6	Management Dashboard	34
7.7	Feedback Data Analysis	35
7.8	Doctor Schedule Module	36
7.9	Sales Dasbboard	37
7.10	Care Coordination Dashboard	38
7.11	CRM Mailer	39
7.12	OLAP Cube	40
7.13	Code for Frequency of words	41
7.14	Pattern Analysis and Classification of Strings	42
7.15	DB in Reporting server – Feedback App support	43

## **List of Tables**

<b>Table</b>	<b>Title</b>	<b>Page Number</b>
3.1	Functions	11
4.1	Break up of Functions	14
4.2	Timeline	17
4.3	Lines of Code	22
6.1	Inspection Checklist for SQA Requirements and Implementation	25
6.2	Inspection Checklist for SQA Testing	27

# **Abbreviations, Notations and Nomenclatures**

## **Abbreviations**

- MIS – Management Information System
- BI – Business Intelligence
- SQL – Structured Query Language
- OLAP - online analytical processing
- MDA – Multi-dimensional analytical
- SDK – Software Development Kit
- SMTP - Simple Mail Transfer Protocol
- IP – Internet Protocol
- SDLC – Software Development life cycle
- DB – Database
- UML – Unified Modeling Language
- SQA – Software Quality Assurance
- NDA – Non Disclosure Agreement

# 1

## Introduction

HCL Healthcare is the newest business venture of HCL Corporation, the parent company of HCL Technologies Ltd and HCL Infosystems. HCL Avitas, in affiliation with Johns Hopkins Medicine International, is the healthcare delivery arm of HCL Healthcare. Starting with the country's first nationwide-networked multispecialty clinics, HCL Avitas aims to provide the whole continuum of care for chronic and acute diseases.

HCL Avitas currently has 9 clinics around NCR and they are divided into 4 clusters. Each clinic is classified either as an Advanced Medical Care (AMC) or a First Health Care (FHC) clinic based on the specialties of healthcare provided.

As a part of the final semester project course, CSD430, this internship at Avitas was arranged through the Career Development Center at Shiv Nadar University. The internship began on Jan 5,2015 and the mentor for the project from the organization is Mr. Suresh Sarojani, Chief Technical Officer. The internship includes a stipend of Rs 15000/- per month. The office is located in Noida Sector 16 and the working hours for the internship is from 9.30am to 6.30pm, Monday to Friday. (45 hours/week) The internship involves a Non Disclosure Agreement and hence the report will not include any vendor names and output/data.

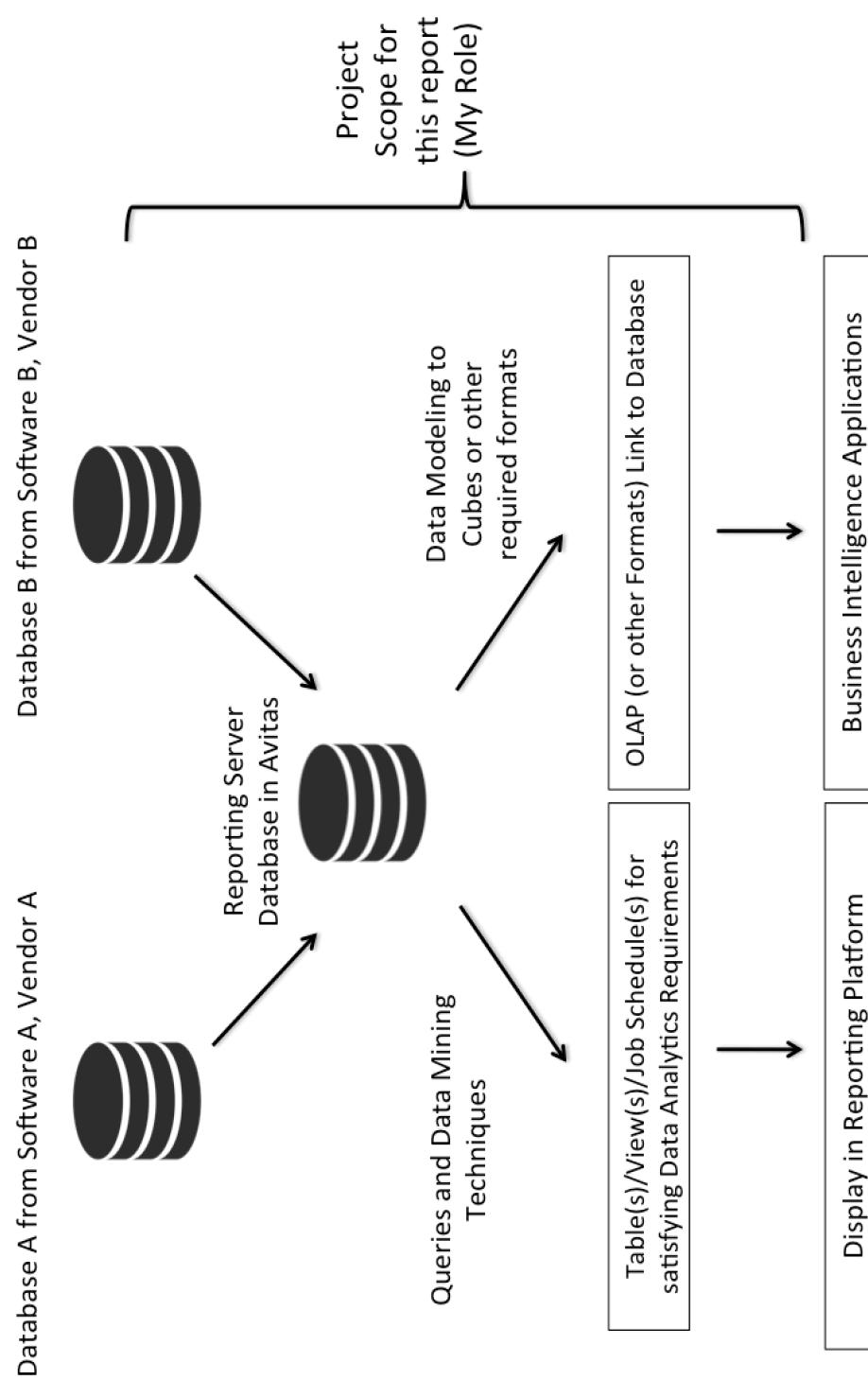
Since its inception, Avitas has generated huge amounts of data, critical to the understanding and improvement of quality of services provided and also for accelerating the growth of the company. Thus, there is a need for a Management Information System (MIS) equipped with the required Business Intelligence (BI). The two main applications being used by Avitas for patient record management are products outsourced from two vendors, Vendor A and Vendor B (names withheld to protect the confidentiality agreement). The much-needed data for analysis are in different formats on different servers.

For carrying out the data analysis process, firstly, the data being generated by both the softwares need to be brought into one single server space, with the database modeled well for the future analytics need. The step involves building a reporting server within Avitas, which gets real time updates of the required data from all the external servers and it involves data cleaning activities, data integrity checks and database modeling.

The next involves using the right set of tools and techniques to create a system that enables its end user to play with the data in every possible way. The reporting server database created will be used to automate the current reporting processes and build other reports required by the organization. Few reports, for example - the number of patient visits daily, which were being populated manually by the cluster managers have to be automated and other reports, which are of utmost importance for the entire organization and not being generated from the two data sources because of lack of technical resources, would also be built and automated. A preliminary analysis of the requirements under this category shows that they can be built using any query language directly on the reporting server and by using various data analysis tools like text mining and pattern analysis. These reports would be built for several purposes like visualization from a front end dashboard application or as daily reports sent via email or available as downloads for the organization. This stack when connected with a web based front end (not a part of this report) would enable the organization to visualize the vast data seamlessly and in real time to support decision-making.

The final step involves providing the data in OLAP (online analytical processing) cube format or other formats as required for deeper Data Analytics in the dashboard/data analytics

softwares. An OLAP cube is an array of data understood in terms of its 0 or more dimensions and is an approach to answering multi-dimensional analytical (MDA) queries swiftly. The steps involved in the project and the scope of the project report are illustrated in the next image.



**Figure 1.1 Project Requirements and Report Scope**

# 2

## Requirement Analysis And Feasibility Study

### 2.1 Project Description And Scope

HCL Avitas manages the front end of their business through two softwares purchased from two vendors, Vendor A and Vendor B. The functions of these softwares are exclusive to each other and hence are the databases. Software A (from Vendor A) is used as a comprehensive **Patient Management System** by the front desk of each clinic, which includes functions like creating appointments, recording the visits history, managing follow up visits etc. of the patients during their visits. Software B (from Vendor B) is used by the nurses and doctors in each clinic as an **Electronic Health Record** (i.e. a record of the clinical data) of the patients visiting. Software A and Software B communicate using an international standard called Health Level 7 (HL7), which functions on the application layer. This is only a one-way data transfer from Software A to B, and is limited only to the Unique IDs being generated for each patient by Software A at the front desk.

After a year since its inception, the leadership and management of Avitas felt the need for data analytics and reports for further decision-making. Thus the requirement to be fulfilled by the internship was to build the business intelligence required for the organization. The reporting server and BI built would give the user all the flexibility and freedom to retrieve, slice and dice the data required to satisfy the analytics needs.

## **2.2 Domain Analysis**

There are many off the shelf analytics softwares like BIRT by Eclipse and Pentaho which provide good analytics tools, but the pre-requisites for them is a well modeled database, which, is yet to be created. Also, Avitas follows a business model which is not mainstream strategies seen in healthcare and thus the way the data is modeled within the outsourced softwares cannot be analysed accurately using these analytics tools. Thus, there arises a need for an analytics backend and dashboard to be built from scratch that will fit the data flow perfectly.

## **2.3 Preliminary Requirement Analysis**

After a thorough understanding of the business model and the data analysis requirements, the project requirements have been classified to three broad segments based on the technology to be used and the nature of the requirements. They are broadly:

1. Building a reporting server: The data being generated by both the softwares need to be brought into one single server space, with the database modeled well for the future analytics need.
2. Automate the current reporting processes and build other reports required by the organization: Few reports, for example - the number of patient visits daily, which were being populated manually (egs. by the cluster managers) have to be automated. Other reports, which are of utmost importance for the entire organization and not being generated from the two data sources because of lack of technical resources, were to be also built and automated. A preliminary analysis of the requirements under this category shows that they can be built using any query language directly on the reporting server from requirement 1. These reports would be built for several purposes like visualization from a front end dashboard application or as daily reports sent as email/sms or available as downloads for the organization.
3. Build advanced data analytics including, but not limited to trends, extrapolations, free text analysis and to also model the DB in the reporting server for use from a web based dashboard application/commercial report and analytics generating software.

DB should also provide backend support for other apps and provide analysis functions on the same data from the front-end application built.

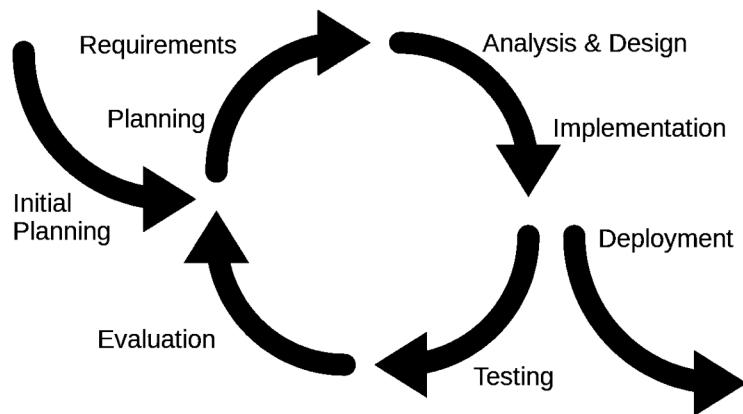
## 2.4 Technical Feasibility

The primary requirement is a Database Server. The server should meet the following requirements:

1. Microsoft SQL Server or My SQL: For database management and query processing. Vendor A uses MS SQL for their backend purposes and Vendor B uses My SQL. The recommended solution is MS SQL since it is more feasible to convert My SQL (an open source software) to MS SQL than doing it the other way around.
2. Visual Studio 2012 and Business Intelligence for Visual Studio 2012: For creating OLAP (Online Analytical Processing) Cube from the database being created. OLAP is known for its properties where in analytics operations on the data becomes more efficient.
3. Java Development Kit and a Java SE Compiler (e.g. Netbeans): For compiling Java codes for pattern/string analysis on the database.
4. Creating a reporting server requires a database server from which connectivity can be established with the vendor databases.
5. The capacity to serve the external applications with query processing. The database server must have good processing power to handle complex queries and pull requests from the web based/stand alone applications.
6. SMTP Server Connectivity: The server must be able to send emails through the SMTP server in which organization's email is configured.
7. Secure firewall and antimalware solutions: Data within the servers are highly confidential and when external IPs are allowed to connect to the database for the application, the server must be able to detect and block other malicious attempts to extract the data.

## 2.5 Software Development Process

The evolving nature of the requirements and the inherent property of the need for continuous improvement in analytics seems to show that an **Agile Model of Software Development** is best fit for this project. Agile is also iterative and incremental in definition and this makes sure that the deployment and modification of the application with each step of development or iteration happens in weeks rather than in months. This will make sure that the development process adapts to the changing requirements, is attentive to good design, values working software over comprehensive documents and the end product is of maximum usability.



**Figure 2.1 Development Model**

Source: Creative Commons

## 2.6 Visibility Plan

A weekly meeting for development review and demo (on Wednesdays) with the technology team comprising of the CTO, HCL Avitas and two Associate Managers will be held. One of the 12 principles of agile development methodology is that Face-to-face conversation is the best form of communication (co-location). Requirement related meetings happen atleast once in week and the changes are rolled out as increments. Collaboration tools from Microsoft Office 365 like Sharepoint are also used to keep track of the requirements and project development.



**Figure 2.2 Calender View: Weekly Review Meeting**

## 2.7 Risk Analysis

Since the project follows an Agile methodology of iterative and incremented development, many of the risks relating to change in requirements, change in technical requirements, change in project schedule etc. during development would be mitigated. Other risks are follows:

Risk: Change in software A and/or software B and hence the databases from which data are extracted.

Fall over Plan: The schema of the DB being designed for the reporting server should be generalized and any change in the parent databases should require only minimum level of code change. This code change would be limited only to the Jobs being scheduled in the SQL server for DB updating and hence any application being built presently or in the future will not be affected.

## **2.8 Conclusion**

After a detailed analysis of technical and operational feasibility, it has become clear that this project development is feasible. The 12 principles\* of Agile development methodologies will be strictly followed so that the development process is of the best experience to both the sides and will also result in the end solution solving all the requirements.

### **\* AGILE PRINCIPLES**

The Agile Manifesto is based on 12 principles:

1. Customer satisfaction by rapid delivery of useful software
2. Welcome changing requirements, even late in development
3. Working software is delivered frequently (weeks rather than months)
4. Close, daily cooperation between business people and developers
5. Projects are built around motivated individuals, who should be trusted
6. Face-to-face conversation is the best form of communication (co-location)
7. Working software is the principal measure of progress
8. Sustainable development, able to maintain a constant pace
9. Continuous attention to technical excellence and good design
10. Simplicity—the art of maximizing the amount of work not done—is essential
11. Self-organizing teams
12. Regular adaptation to changing circumstance

(Source: <http://www.agilemanifesto.org/principles.html>)

# 3

## **Software Requirements**

### **3.1 External Dependancies**

The Database connectivity/interface from the reporting server to the two external vendor servers are of utmost importance. For the database being created to reflect the updates in the master/parent databases (i.e. vendor databases) in the time intervals deemed fit for analytics, the databases needs to seamlessly connect and maintain data integrity.

The firewall configured in the organization's network should allow incoming trusted requests from applications hosted in external servers and should also allow outgoing connections from the reporting server (hosted within the network) for data transfer after query processing.

### **3.2 Functions**

- The reporting server built shall allow external trusted applications being built to access the data.
- The queries being built for requirement 2 (refer to section 2) will allow external applications to access and format the data needed for reports.
- The reporting server will update itself from the parent databases and will also automate processes like emailing daily reports.

- The reporting server shall also contain the database in multiple formats (Databases and OLAP Cubes (or other formats) and in well designed schemas for use in different types of applications.

The detailed list of functions from the Database server being created, along with the requirement and the development phase they fall under are mentioned in Table 3.1

**TABLE 3.1 Functions**

Function	Part of Requirement (as listed in Section 2)	Phase of Project Development
Unified and Central Database	1	1
Provide flexibility for the organization to add or delete data like Discount Categories being offered in the clinics, or Clinics in operation etc (cont.) (cont.) seamlessly and without effort. These changes should reflect in the data being generated from the product with no code change required at all.	1	1
Provide data in the visual format and operational conditions required for the following in the dashboard (Dashboard not a part of the project) <ol style="list-style-type: none"> <li>1. Operational Analysis</li> <li>2. Quality Dashboard</li> <li>3. Marketing Analysis</li> <li>4. Management Dashboard</li> <li>5. Feedback Data</li> <li>6. Doctor Schedule</li> <li>7. Sales Dashboard</li> <li>8. Care-Coordination Dashboard</li> <li>9. Customer Relations Module Mailer</li> </ol>	2	1,2

Send Automated emails and SMS to different sections of the organization with data required on daily basis	2	1
Provide data in OLAP Cube format (or other formats) for deeper Data Analytics in the dashboard/data analytics softwares	3	2
Text mining of free text	3	2
Pattern Analysis for formulary/non formulary drugs entered in Plan of care text for patients	3	2
Data Storage Functionality for other apps (e.g. Feedback App) being built	3	2

### 3.3 Software System Attributes And Performance

- i. RELIABILITY: It should be ensured that the server responds to data and analysis requests from multiple users with complete reliability and accuracy. For this criteria, the queries written for data analysis are completely dynamic, thus ensuring there is no transaction clashes resulting in data loss or application crashes. Backups of the data is also periodically taken.  
In case of failure in connecting to the parent databases, it has been made sure that data will be pulled for the entire down time by the scheduled jobs once the connectivity is restored.
- ii. AVAILABILITY: The server will be available depending on whether the system hosting the server has continuous power and also on data connectivity with the parent databases (only for data updates)

- iii. SECURITY: The connectivity to the server for data access from external IPs are filtered through the internal firewall and scanned for any malicious attempts. All SQL processing and data requests are also written into a log file.
- iv. MAINTAINABILITY: Maintainability of the codes is high since complex transactions are broken down to multiple source codes and tables during their transactions. Also, any changes in the business components need only appropriate updates in the database and no change in the schema or the codes are required.
- v. PORTABILITY: All the codes and scheduled jobs for automation can be exported to any Windows Server running SQL Server. The code cannot be ported to any other OS or any other Database or Query Language.

# 4

## Project Schedule

The three broad requirements mentioned in Requirement Analysis in Section 2 of this report is assigned to two phases of the project. This will help in carrying out the iterative and incremented approach of the project and to also support the evolving nature of the specific requirements under the three categories. The two phases are:

- Phase 1: Requirement 1 (Creating Reporting Server)  
Requirement 2 (Reporting Processes)
- Phase 2: Requirement 3

Please refer to table below for the detailed break up of functions in each phase.

**TABLE 4.1 Break up of Functions**

SNO	Requirement name for future references	Description	Requirement	Phase of Project Development
1	Reporting Server	Unified and Central Database	1	1
2	Scalability requirements for reporting server	Provide flexibility for the organization to add or delete data like Discount Categories being offered in the clinics, or Clinics in	1	1

		operation etc (cont.) (cont.) seamlessly and without effort. These changes should reflect in the data being generated from the product with no code change required at all.		
3	Operational Analysis Reports	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of the project)	2	1
4	Clinical Analysis Reports	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	1
5	Marketing Analysis Reports	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	1
5	Management Dashboard for Trends	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	1
6	Feedback Data Analysis	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	2

		Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)		
7	Doctor Schedule	Send Automated emails/sms to different sections of the organization with data required on daily basis	2	2
8	Sales Dashboard	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	2
9	Care Coordination Dashboard	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	2
10	CRM Mailer	Provide data in the visual format and operational conditions required in the dashboard (Dashboard not a part of this project)	2	2
11	OLAP Cube	Provide data in OLAP Cube format for deeper Data Analytics in the dashboard/data analytics softwares	3	2
12	Text mining	Text mining of free text	3	2
13	Pattern Analysis	Pattern Analysis for formulary/non formulary drugs entered in Plan of care text for patients	3	2
14	Backend Support for other Apps	Data Storage functionality by the reporting server for data from other apps being built	3	2

## 4.1 Timeline

**TABLE 4.2 Timeline**

DATE RANGE	DEVELOPMENT ACTIVITIES	REQUIREMENTS/REVIEW MEETINGS
Jan 5 – Jan 9	1. Initial Requirements Analysis 2. Reporting Server Design Analysis	Jan 7 – Requirements Meet
Jan 12 – Jan 23	1. Reporting Server Development and Testing 2. Operational Analysis Reports Development,Testing and Deployment 3. Scalability requirements design for reporting server	Jan 14 – Weekly Review Meet Jan 16 – Requirements Meet for Operational Analysis Jan 21 – Weekly Review Meet
Jan 27 – Feb 6	i. Incremental changes: Operational Analysis - Development,Testing and Deployment ii. Clinical Analysis Reports- Development,Testing and Deployment	Jan 28 – Weekly Review Meet Jan 30 – Requirements Discussion with Clinical Quality Team Feb 4 – Weekly Review Meet Feb 6 – Requirements Discussion with Marketing Team
Feb 9 – Feb 13	1. Marketing Analysis Reports - Development,Testing and Deployment 2. Incremental changes: Clinical Analysis - Development,Testing and Deployment	Feb 11 – Weekly Review Meet; Requirements meet for Trends in Management Dashboard Feb 13 – Marketing Requirements Review
Feb 16 – Feb 27	1. Incremental changes: Marketing Analysis - Development,Testing and Deployment	Feb 18 – Weekly Review Meet Feb 25 – Weekly Review Meet Feb 26 – Incremental

	<p>2. Management Dashboard for trends - Development, Testing and Deployment</p> <p>3. Daily Reports - Development, Testing and Deployment</p>	<p>requirements meet for Clinical Analysis</p> <p>Feb 27 – Requirements Meet for Phase 2</p>
Mar 2 – Mar 20	<p>1. Pattern Analysis: Development, Testing and Deployment</p> <p>2. Text Mining: Development, Testing and Deployment</p> <p>3. Incremental changes: Clinical Analysis - Development, Testing and Deployment</p> <p>3. Care Coordination Dashboard - Development, Testing and Deployment</p>	<p>Mar 11 – Weekly Review Meet</p> <p>Mar 18 – Weekly Review Meet</p>
Mar 23 – April 10	<p>1. Incremental changes: Marketing Analysis - Development, Testing and Deployment</p> <p>2. Incremental Changes: Daily Reports - Development, Testing and Deployment</p> <p>4. Feedback Data - Development, Testing and Deployment</p> <p>5. Doctor Schedule - Development, Testing and Deployment</p> <p>6. Evaluation of Open Source Analytics tools for BI application</p>	<p>Mar 25 – Weekly Review Meet</p> <p>April 1 – Weekly Review Meet</p> <p>April 8 – Weekly Review Meet</p>

April 13 – April 24	1. OLAP Cubes – Feasibility Analysis and Development. Fall back options if not feasible. 2. External App DB support and report/analysis service development 3. Sales Dashboard - Development, Testing and Deployment 4. CRM Mailer - Development, Testing and Deployment	April 15 – Weekly Review Meet April 22 – Weekly Review Meet
April 27 – May 8	1. Final Testing, SQA and Optimization 2. Documentation; Code Handover	April 29 – Weekly Review Meet May 6 – Weekly Review Meet;

## 4.2 Gantt Chart

Development - Phase 1: Jan 5 – Feb 27

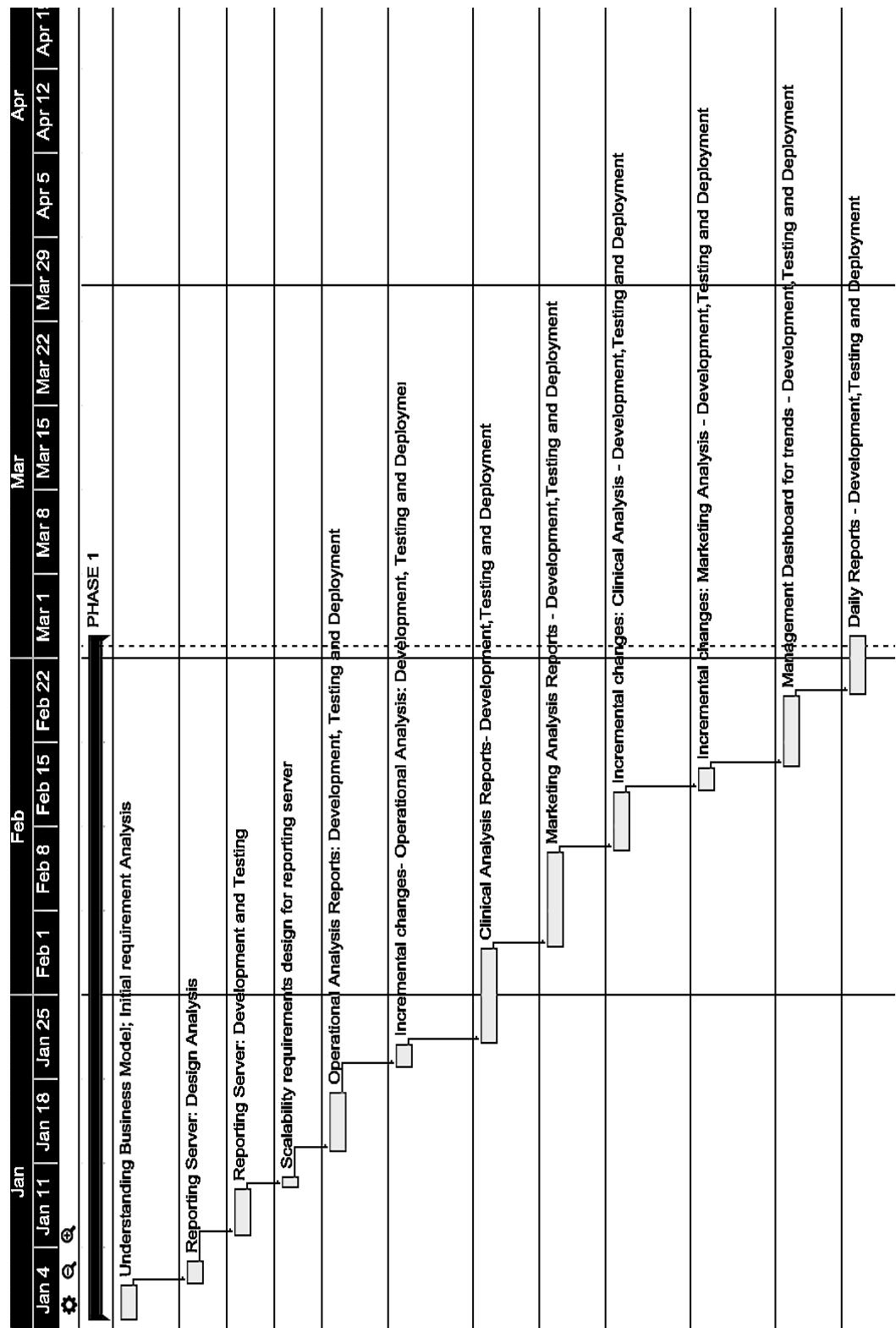


Figure 4.1 Development - Phase 1

Development – Phase 2: Mar 2 – May 8

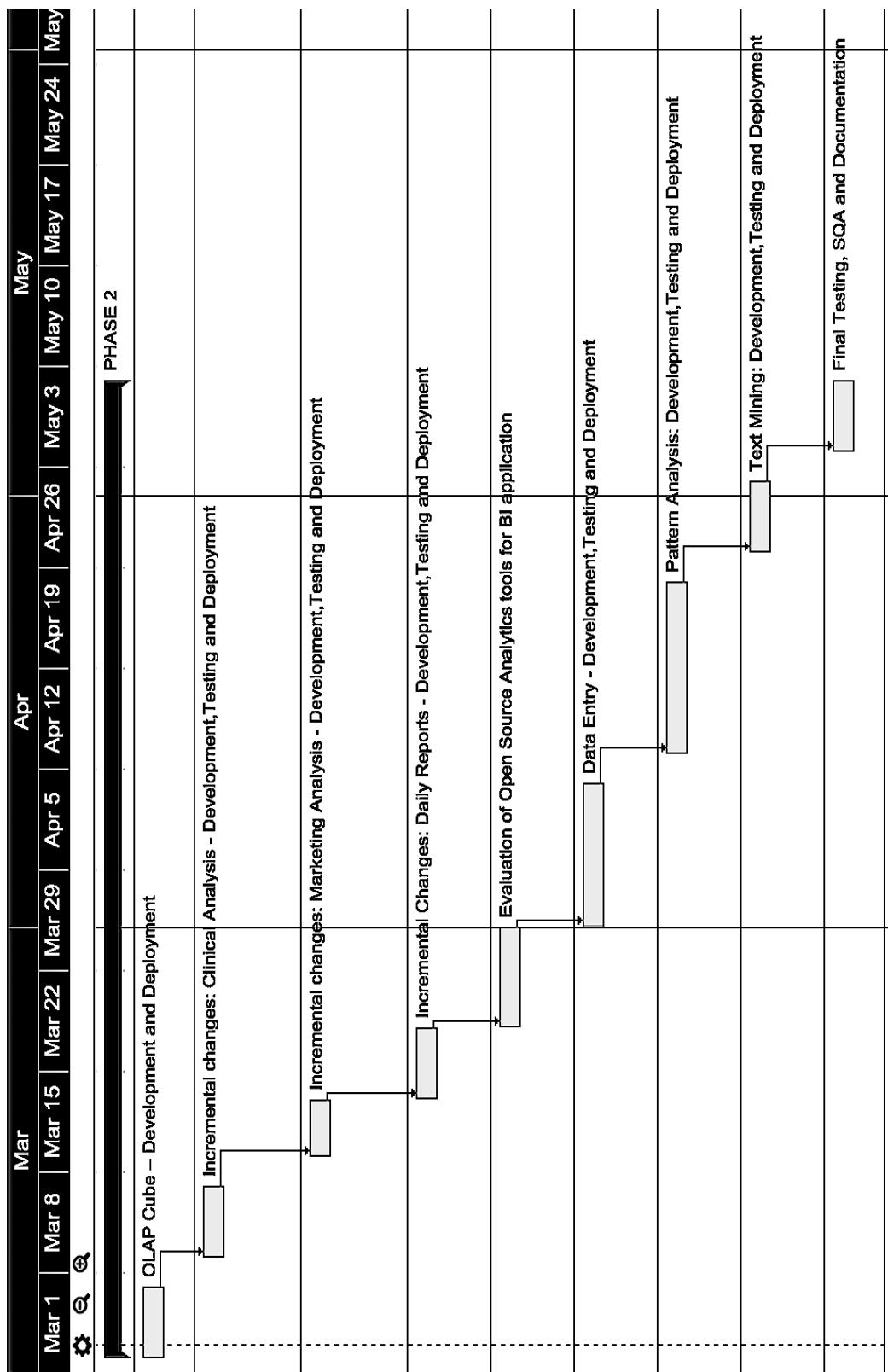


Figure 4.2 Development - Phase 2

### 4.3 Lines Of Code

**Table 4.3 Lines of Code**

SNO	Requirement name	Requirement	Phase of Project Development	Lines of Code
1	Reporting Server	1	1	415
2	Operational Analysis Reports	2	1	1306
3	Clinical Analysis Reports	2	1	882
4	Marketing Analysis Reports	2	1	1420
5	Management Dashboard for Trends	2	1	219
6	Feedback Data Analysis	2	2	1200
7	Doctor Schedule	2	2	325
8	Sales Dashboard	2	2	1175
9	Care Coordination Dashboard	2	2	768
10	CRM Mailer	2	2	126
11	Daily Reports/SMS	2	1	813
12	Text mining	3	2	180
13	Other Reports	3	2	780
14	Backend Support for other Apps	3	2	45

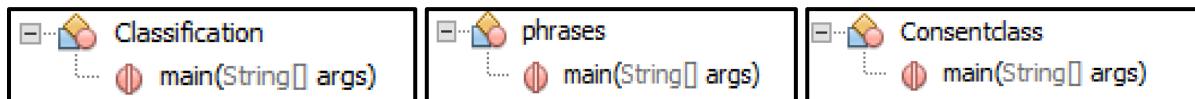
**TOTAL LOC: 9654**

# 5

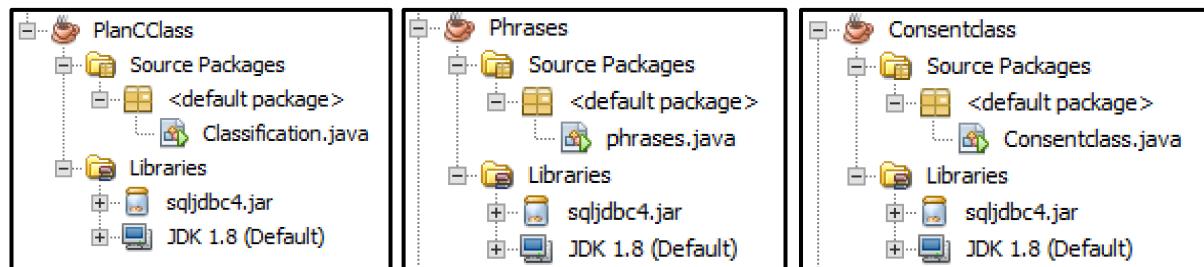
## UML Class Objects and SQL DB Schemas

### 5.1 UML Class Objects (Java Component) and Dependencies

Class Objects (No dependancies)



Class Files and Libraries



**Figure 5.1 UML Class Objects (Java Component) and Dependencies**

## 5.2 SQL DB Schema (Main Tables and Relations)

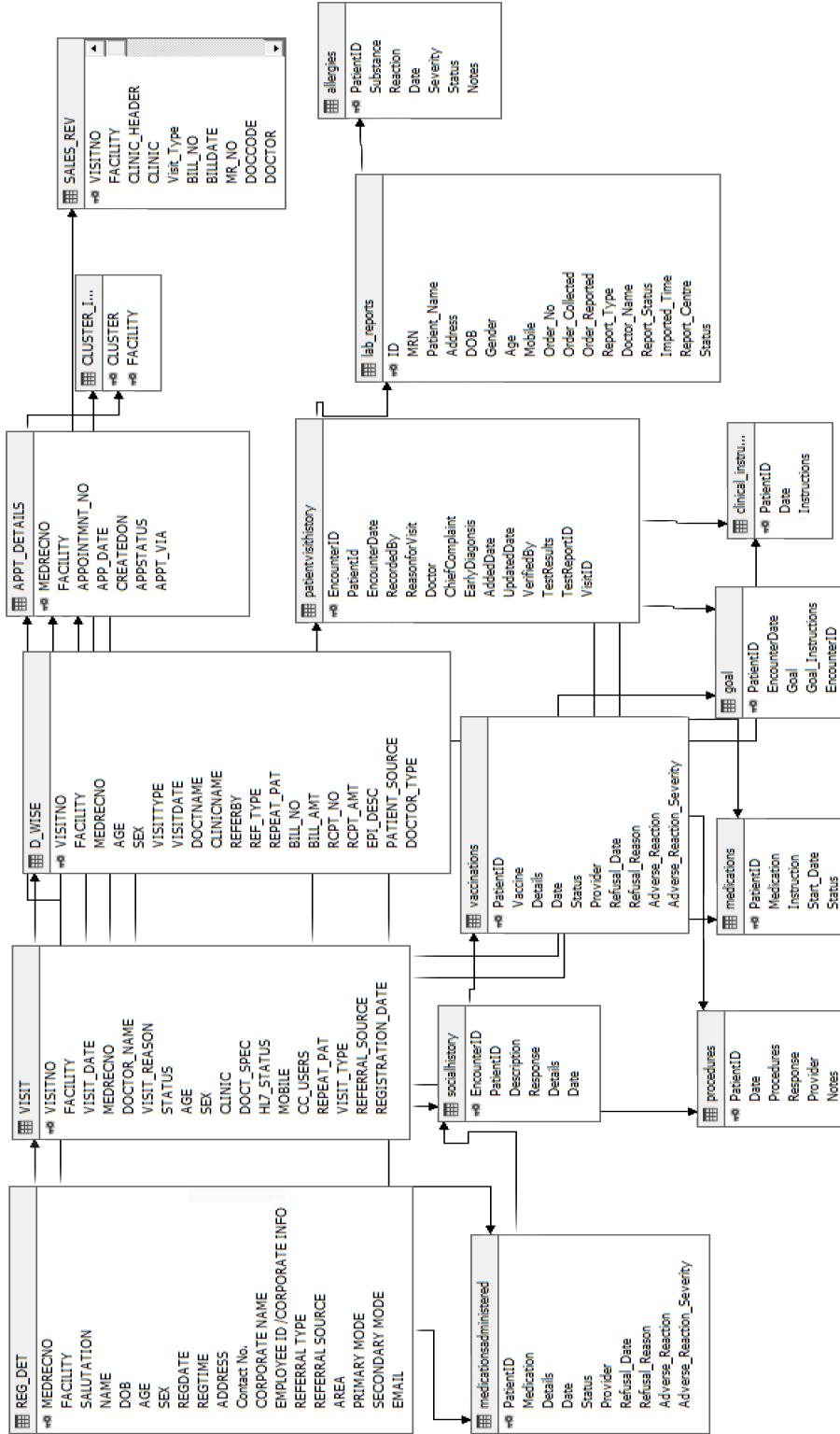


Figure 5.2 SQL DB Schema (Main Tables and Relations)

# 6

## SQA Plan

### 6.1 SQA Requirements and Implementation

Software Quality Assurance plan is essential to make sure that the development process is in sync with the requirements and the implementation conforms to the standards, is scalable and easy to maintain. The checklist of criterion for this segment is listed in the table below.

**Table 6.1     Inspection Checklist for SQA Requirements and Implementation**

Inspection Checklist	Yes/No
<b>Support Material</b>	
Feasibility Study	Yes
Requirements Specifications	Yes
Domain Analysis	Yes
Technical Feasibility	Yes
<b>Documents Standard Compliance</b>	
Identification of Standards	Yes
Compliance to Standards	Yes
Requirements and Feasibility Report comply to standards	Yes

Changes to Reports	<b>Yes</b>
<b>Feasibility Management</b>	
Domain Analysis	<b>Yes</b>
Timeline Realistic and realisable	<b>Yes</b>
Existense of Process review plan	<b>Yes</b>
Flexibility for incremental approach	<b>Yes</b>
Process for follow up and closure of risks	<b>Yes</b>
<b>Requirements Management</b>	
Approved set of requirments	<b>Yes</b>
SDLC planned and designed according to requirements	<b>Yes</b>
<b>Changes in Requirements</b>	
Process to manage change in requirements	<b>Yes</b>
Changes in timeline planned according to change in requirements	<b>Yes</b>
<b>Implementation Structure</b>	
Changes cause run/structure errors to other components of the software	<b>No</b>
Presence of unwanted procedures	<b>No</b>
Code well structured	<b>Yes</b>
Flexibility for organization to add more function parameters effortlessly	<b>Yes</b>
<b>Variables</b>	
Intuitive names for variables, SQL tables and rows	<b>Yes</b>
Proper type consistency and casting	<b>Yes</b>
Unused Variables	<b>No</b>

## 6.2 SQA Testing

Testing ensures the software meets all the requirements in hand, has no syntactical or logical bugs and the interface is user friendly. The checklist of criterion for this segment is listed in the table below.

**Table 6.2 Inspection Checklist for SQA Testing**

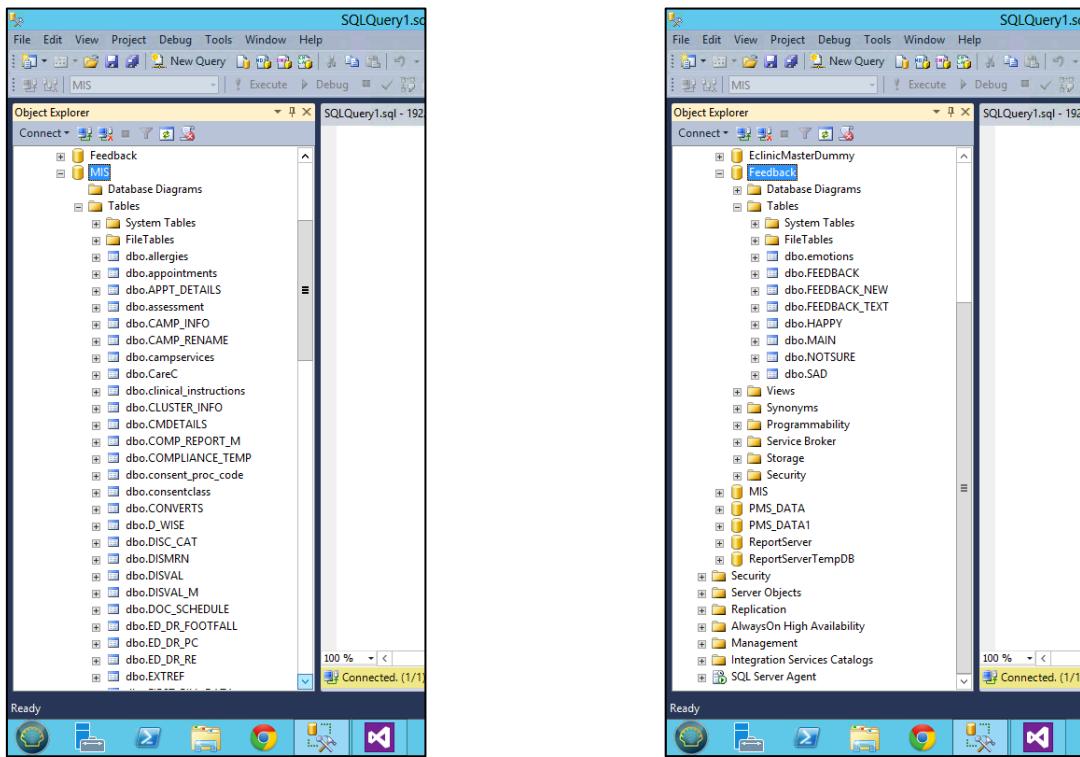
Inspection Checklist	Yes/No
<b>Requirements</b>	
All requirements solved	Yes
Organization's requirements align with requirement specification in the document	Yes
Testability of Code	Yes
<b>Functional Design</b>	
Process to manage change in requirements	Yes
Changes cause run/structure errors to other components of the software	No
Presence of unwanted procedures	No
Flexibility for organization to add more function parameters effortlessly	Yes
Proper type consistency and casting	Yes
Crashes during testing	No
Intuitive User Interface	Yes
Maintain Functionality even during multiple instances/requests of codes running	Yes

<b>(cont.)</b>	
<b>Testing</b>	
Black Box Testing	<b>Yes</b>
White Box Testing	<b>Yes</b>
Unit Testing	<b>Yes</b>
Incremental Integration Testing	<b>Yes</b>

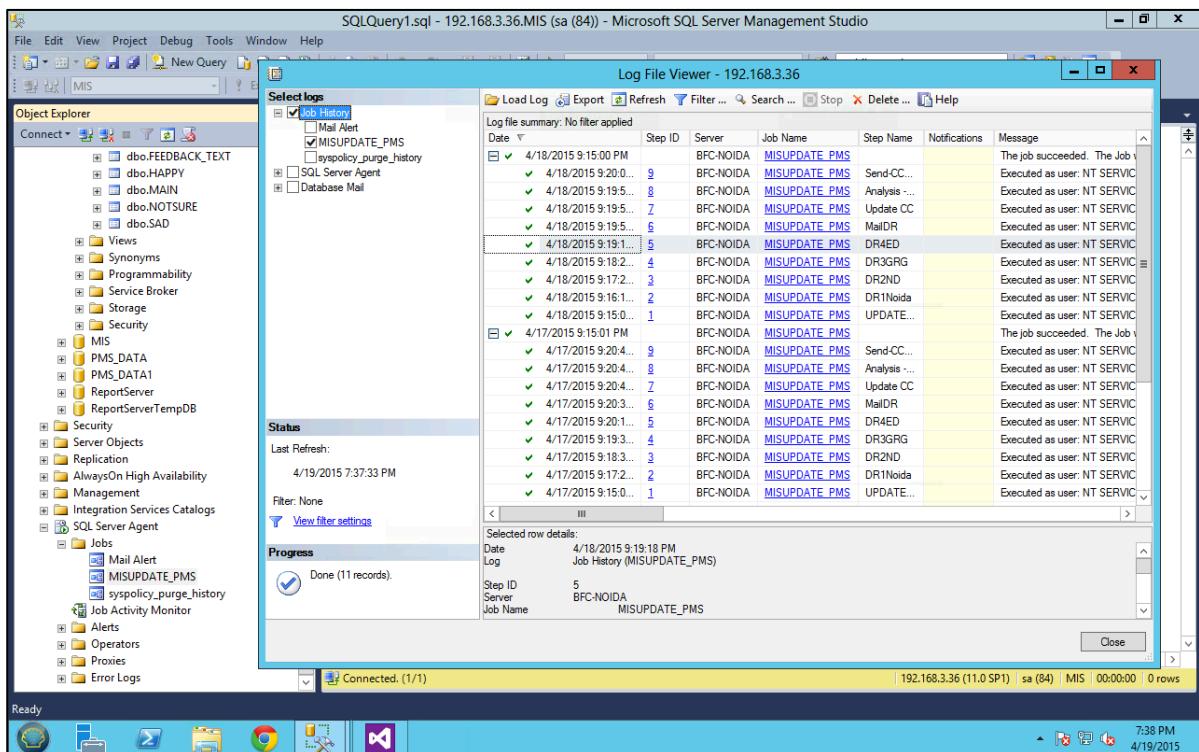
## Testing

Each functional module went through Black box, White Box and Unit and Incremental Integration Testing (after each increment). The results are listed here (for each module, only final iteration). Refer to Table 4.1 for details of the functions. The screenshots that show the web application front end are added for reader's understanding of the output of the back end processes. The front end/dashboard is not a part of the project. Data output is blurred to protect the NDA.

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Reporting Server	PASS	PASS	PASS	PASS



**Figure 7.1 Tables in the Databases of the reporting server (MS SQL Server)**



**Figure 7.2 Update Job scheduled for everyday**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Scalability Requirements	PASS	PASS	PASS	PASS

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Operational Analysis Reports	PASS	PASS	PASS	PASS

The screenshot displays a web-based operational analysis dashboard for HCL Avitas. The URL in the browser is 123.136.138.226/dataanalytics/dataanalytics/index.php#ajax/operational\_analysis.php. The interface includes a top navigation bar with a user profile for 'Sreeram' and a 'Welcome, administrator' message. On the left, a vertical sidebar menu lists several operational analysis modules: Operational Analysis, Quality Dashboard, Marketing Analysis, Management Dashboard, Feedback Data, Doctor Schedule, Sales Dashboard, Care Coordination Dashboard, Clinical Dashboard, and CRM Mailer. The main content area is titled 'Report' and shows a table for 'Unique Footfalls (U)'. The table has columns for 'Type' (which lists 'Numbers', 'East Delhi', 'Gurgaon', 'Noida', 'North Delhi', and 'All Clusters') and 'Services Availed (S)' (which lists 'Medical', 'Cardiac', 'Care Coordinator', 'Dental', 'Dermatology', 'Diabetology', and 'Dietician'). A search bar at the top of the main content area is set to 'Search :'. To the right of the main content, there are two small panels: one labeled 'Report' and another labeled 'Charts'.

**Figure 7.3      Operational Analysis Dashboard (only one functionality shown)**

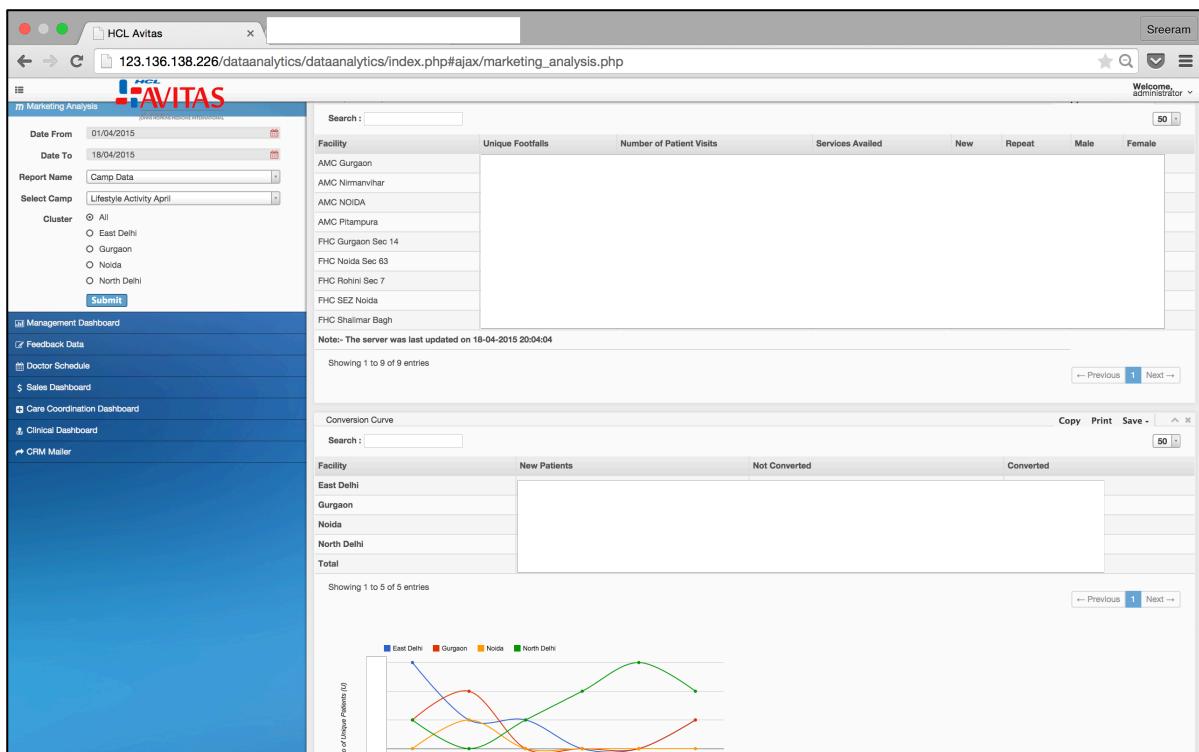
<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Clinical Analysis Reports	PASS	PASS	PASS	PASS

The screenshot displays the HCL Avitas Clinical Analysis Dashboard. On the left, a sidebar menu lists various operational and analytical dashboards. The main content area is titled "Clinical Report Downloads" and contains a table with two columns: "Download Link" and "Indicator SNOs". The table entries are as follows:

Download Link	Indicator SNOs
JHI Reports (Data)	10,11,20,21,22,23,24,25,26,27,28,45,46
Compliance Reports	10,11,20,21,22,23,24,25,26,27,28,45,46,31,32,35,45,46
Registration Details	10,11,14,15,16,18,58
<b>Self Care and Community Resource</b>	
JHI Reports (Data)	10,11,20,21,22,23,24,25,26,27,28,45,46
Compliance Reports	10,11,20,21,22,23,24,25,26,27,28,45,46,31,32,35,45,46
Radiology Order Reports	44
Lab Order Reports	36,39
Care Coordination Rate	38
<b>Service Quality</b>	
Registration Details	10,11,14,15,16,18,58
<b>Accessibility</b>	

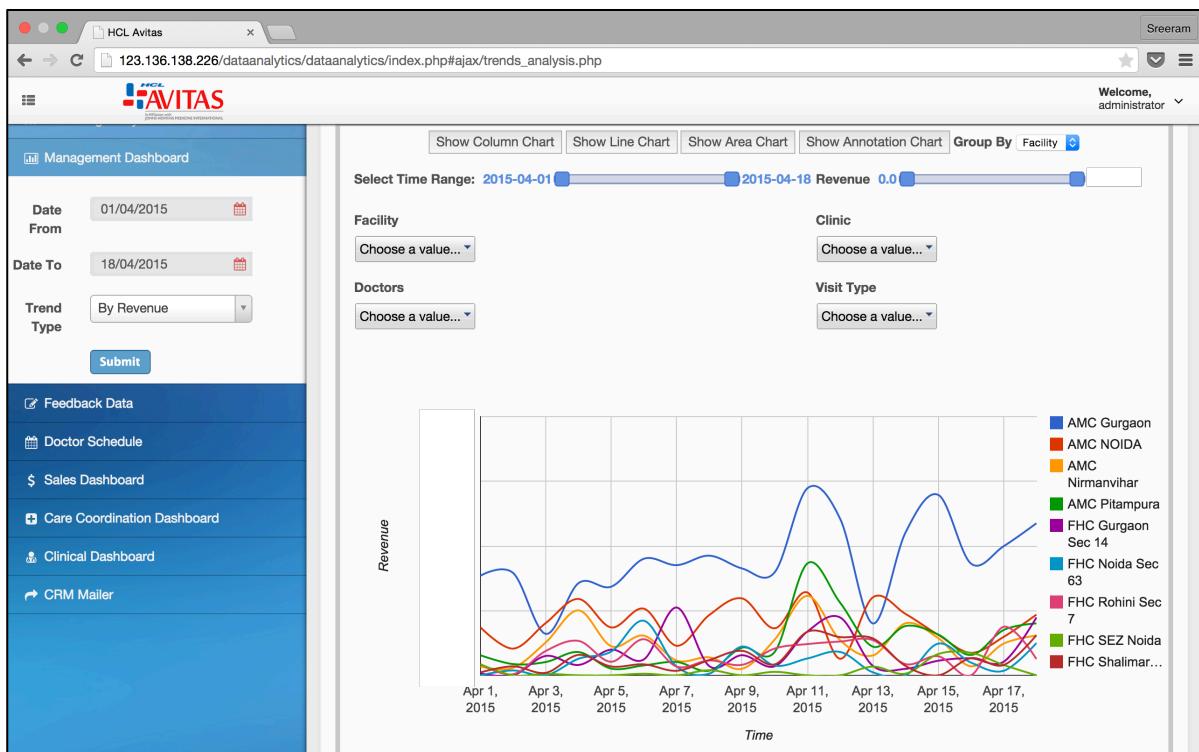
**Figure 7.4 Clinical Analysis Dashboard (only one functionality shown)**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Marketing Analysis Reports	PASS	PASS	PASS	PASS



**Figure 7.5 Marketing Analysis Reports (only one functionality shown)**

Function	Black Box Testing	White Box Testing	Unit Testing	Incremental Integration Testing
Management Dashboard for Trends	PASS	PASS	PASS	PASS



**Figure 7.6 Management Dashboard (only one functionality shown)**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Feedback Data Analysis	PASS	PASS	PASS	PASS

The screenshot shows a web-based application titled 'HCL Avitas' with a URL of 123.136.138.226/dataanalytics/index.php#ajax/feedback\_analysis.php. The page is titled 'Feedback Data Report'. On the left, there is a sidebar with navigation links: Operational Analysis, Quality Dashboard, Marketing Analysis, Management Dashboard, and Feedback Data (which is currently selected). The main area displays a table header with columns: Unique Code, Patient Name, First Visit Date, Last Visit Date, Number of visits, Age, Gender, Contact No, and Address. Below the table, there is a search bar labeled 'Search :', a dropdown menu set to '10', and a date range selector from 'Date From: 10/04/2015' to 'Date To: 18/04/2015'. A 'Cluster' section includes radio buttons for 'All', 'East Delhi', 'Gurgaon', 'Noida', and 'North Delhi', with 'All' selected. A 'Submit' button is located below the cluster selection. The bottom of the sidebar also lists Doctor Schedule, Sales Dashboard, Care Coordination Dashboard, and Clinical Dashboard.

**Figure 7.7     Feedback Data Analysis**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Doctor Schedule	PASS	PASS	PASS	PASS

The screenshot shows a web-based application titled "HCL Avitas". The main header includes the logo "AVITAS" and the text "DATA & INSIGHTS FOR BETTER PATIENT EXPERIENCES". The top right corner shows a welcome message for "Sreeram" and "administrator". The left sidebar contains navigation links for "Operational Analysis", "Quality Dashboard", "Marketing Analysis", "Management Dashboard", "Feedback Data", and "Doctor Schedule". The "Doctor Schedule" link is highlighted. The main content area is titled "Doctor Schedule Report" and features a search bar and a table with 10 rows visible. The table columns include Facility, Clinic Name, Doctor Name, Schedule Date, Time Slot, Slot Duration, Breaks, Blocks, Cluster Manager, Mobile Number, Landline Number, and Email. Below the table, a note states "Note:- The server was last updated on 18-04-2015 20:04:04". At the bottom, it says "Showing 1 to 3 of 914 entries" with a navigation bar for "Previous", "1", "2", "3", "4", "5", and "Next".

**Figure 7.8     Doctor Schedule Module**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Sales Dashboard	PASS	PASS	PASS	PASS

The screenshot shows a web-based application interface for HCL Avitas. On the left, there is a sidebar with various menu options: Operational Analysis, Quality Dashboard, Marketing Analysis, Management Dashboard, Feedback Data, Doctor Schedule, and Sales Dashboard. The Sales Dashboard is currently selected. Below these are filters for Date From (01/04/2015), Date To (18/04/2015), Corporate (Walmart selected), Report Name (Walmart selected), and Cluster (XL selected). The main content area displays two reports. The top report is titled "Top 5 Services Data Report" and includes a search bar, a table header with columns "Service Name", "Count", "Total Discount Given", and "Total Bill Paid", and a table body showing data for 6 entries. The bottom report is titled "Revenue Summary" and includes a search bar, a table header with columns "Facility", "U", "P", "S", "New", "Repeat", "Discounts", and "Paid Amount", and a table body showing data for 6 entries. Both reports have "Copy", "Print", and "Save" buttons at the top right.

**Figure 7.9 Sales Dashboard (only one functionality shown)**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Care Coordination Dashboard	PASS	PASS	PASS	PASS

The screenshot shows a web-based dashboard titled "Care Coordination Dashboard". On the left, there is a sidebar with various menu items: Operational Analysis, Quality Dashboard, Marketing Analysis, Management Dashboard, Feedback Data, Doctor Schedule, Sales Dashboard, and Care Coordination Dashboard (which is currently selected). Below these are date range filters (Date From: 01/04/2015, Date To: 18/04/2015), a report name dropdown (Status by Staff), and a cluster selection (All, East Delhi, Gurgaon, Noida, North Delhi). The main content area is titled "Status Table by Staff" and displays a table with columns: Staff, Total Completed Actions, Completed by Due Date, and Completed after Due Date. The table shows 28 entries. At the bottom, it says "Showing 1 to 10 of 28 entries" and has navigation buttons for Previous, Next, and page numbers 1, 2, 3.

**Figure 7.10 Care Coordination Dashboard (only one functionality shown)**

Function	Black Box Testing	White Box Testing	Unit Testing	Incremental Integration Testing
CRM Mailer	PASS	PASS	PASS	PASS

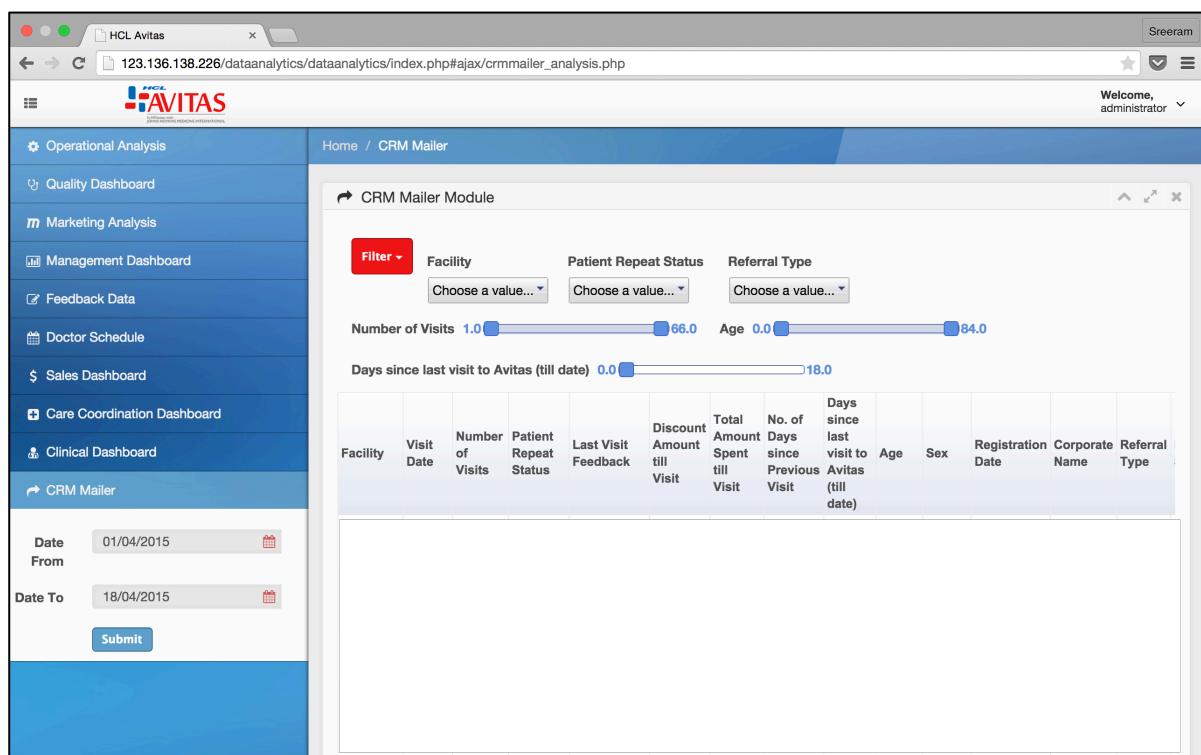
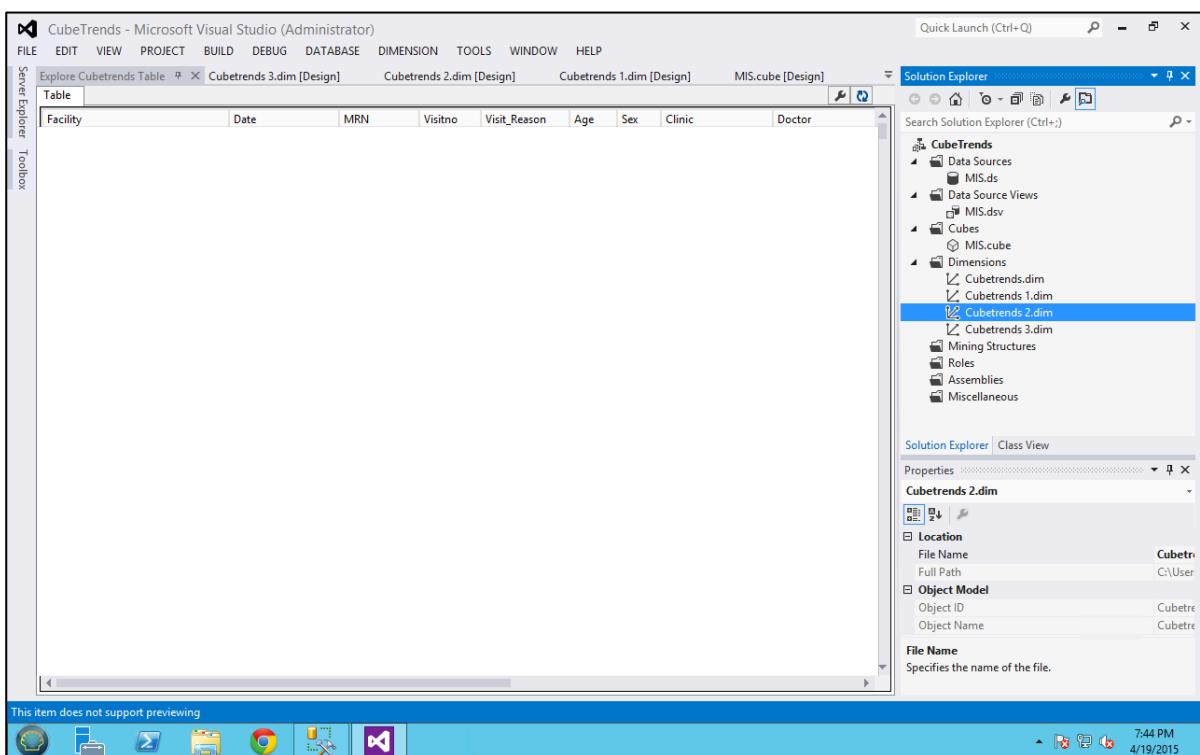


Figure 7.11 CRM Mailer

Function	Black Box Testing	White Box Testing	Unit Testing	Incremental Integration Testing
OLAP Cube	PASS	PASS	PASS	PASS



**Figure 7.12 OLAP Cube – Visual Studio 2012 with Business Intelligence**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Text Mining	PASS	PASS	PASS	PASS

```

Phrases - NetBeans IDE 8.0.2
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Project Files Services <default config> Search (Ctrl+F)
Source Page Classification.java phrases.java Consentclass.java
Source History <default package> Consentclass.java
Consentclass
Source Packages <default package>
Libraries sqjdbc4.jar JDK 1.8 (Default)
Phrases PlanCClass
Source Packages <default package> Classification.java
Libraries sqjdbc4.jar JDK 1.8 (Default)

phrases - Navigator x
Members <empty>
phrases main(String[] args)

Output - PlanCClass (run) x
PlanCClass (run) 17:4 7:21 PM 4/19/2015

public class phrases {
    public static void main(String[] args) throws SQLException, ClassNotFoundException {
        Class.forName("com.microsoft.sqlserver.jdbc.SQLServerDriver");
        Connection conn = DriverManager.getConnection("jdbc:sqlserver://localhost;user=sa;password=pms@123;databaseName=tempdb");
        Statement sta = conn.createStatement();

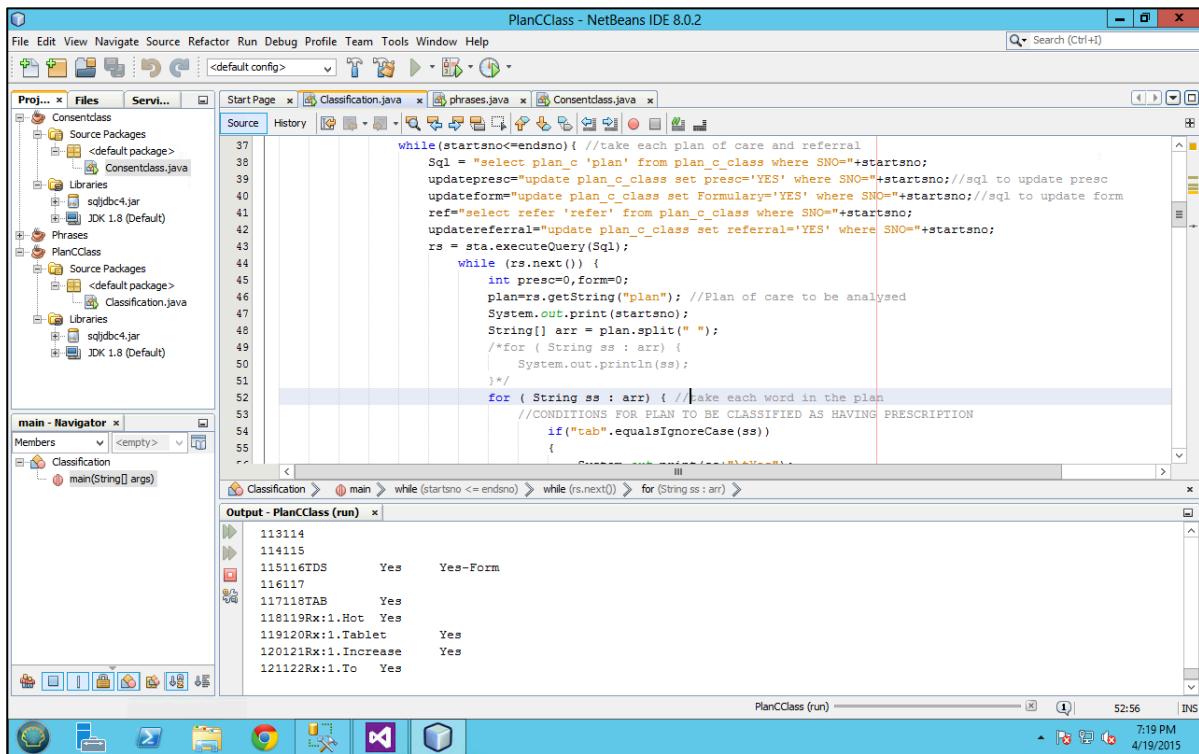
        String Sql="SELECT Plan_c 'plan' FROM tempass";
        ResultSet rs = sta.executeQuery(Sql);

        while(rs.next())
        {
            int i=0;
            String plan=rs.getString("plan"); //Plan of care to be analysed
            String[] arr = plan.split(",|\\.|\\n| ");
            for ( String ss : arr)
            {
                System.out.println(ss);
            }
        }
    }
}

```

**Figure 7.13 Code for Frequency of words in Plan of Care entered by Doctors and assign weights – Java Code - Netbeans**

Function	Black Box Testing	White Box Testing	Unit Testing	Incremental Integration Testing
Pattern Analysis	PASS	PASS	PASS	PASS



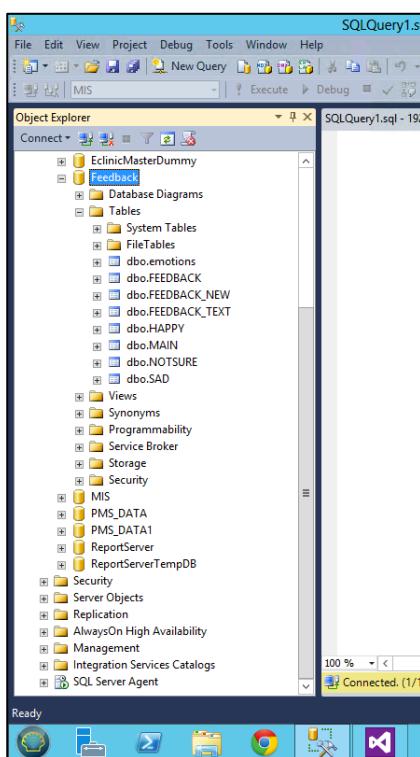
The screenshot shows the NetBeans IDE interface with the following details:

- Project Explorer:** Shows two source packages: `Consentclass` and `PlanClass`, each containing a `Classification.java` file.
- Code Editor:** Displays the `Classification.java` code for the `PlanClass` package. The code performs pattern analysis and classification of strings based on SQL queries and string manipulation logic.
- Output Window:** Shows the results of the run, titled "PlanCClass (run)". It lists various entries with their classifications:
 

Classification	Condition
113114	Yes
114115	Yes
115116TDS	Yes-Form
116117	
117118TAB	Yes
118119Rx:1.Hot	Yes
119120Rx:1.Tablet	Yes
120121Rx:1.Increase	Yes
121122Rx:1.To	Yes
- Bottom Bar:** Includes standard NetBeans icons for file operations, search, and help.

**Figure 7.14 Pattern Analysis and Classification of Strings – Java Code – Netbeans**

<b>Function</b>	<b>Black Box Testing</b>	<b>White Box Testing</b>	<b>Unit Testing</b>	<b>Incremental Integration Testing</b>
Backend Support for other Apps	PASS	PASS	PASS	PASS



**Figure 7.15 DB in Reporting server showing Feedback App tables**

# **8**

## **Conclusion**

The project has been successful overall and the requirements have been satisfied while complying without major changes to the scheduled timeline. The development process was smooth and a good experience for both the organization and for me as a developer. The feedback from the Organization and its key players were excellent.

A deeper way of intelligence and analytics was not possible during the span of development due to the constraints from the structures in the data sources. But with the reporting server built successfully, and the reporting systems and analytics, it can be said that it is just a matter of more increments on the built foundation for the organization to achieve even powerful levels of analytics (e.g. predictions of revenue). These increments, if needed, can be done effortlessly since the codes have been written with readability, maintainability and scalability in mind.

# 9

## References

1. Microsoft Developer Network Documentation. *EXECUTE (Transact-SQL)*. URL: <https://msdn.microsoft.com/en-IN/library/ms188332.aspx>
2. MySQL. *Step-by-step Guide to Connecting to a MySQL Database through Connector/ODBC*. URL: <http://dev.mysql.com/doc/connector-odbc/en/connector-odbc-examples-walkthrough.html>
3. Microsoft Developer Network Documentation. *Linked Servers (Database Engine)*. URL: <https://msdn.microsoft.com/en-in/library/ms188279.aspx>
4. Microsoft Developer Network Documentation. *Openquery (Transact-SQL)*. URL: <https://msdn.microsoft.com/en-us/library/ms188427.aspx>
5. Microsoft Developer Network Documentation. *Create and process an OLAP cube*. URL: <https://msdn.microsoft.com/en-us/library/aa548673%28v=ax.10%29.aspx>
6. Microsoft Developer Network Documentation. *sp\_send\_dbmail (Transact-SQL)*. URL: <https://msdn.microsoft.com/en-IN/library/ms190307.aspx>
7. Microsoft Developer Network Documentation. *Using the JDBC Driver*. URL: <https://msdn.microsoft.com/en-us/library/ms378526%28v=sql.110%29.aspx>