



Audyogik Shikshan Mandal's
Institute of Management & Computer Studies (MCA Institute)
(IMCOST)

MASTER OF COMPUTER APPLICATION (MCA)

UNIVERSITY OF MUMBAI

A.Y. 2021-22

MINI PROJECT

ON

Web Based Claim Processing System

A PROJECT REPORT

Submitted by

Rohit Chandrakant Sulakhe

MC2021108

Certificate

This is to certify that :

Name - Rohit Chandrakant Sulakhe

Roll no. – MC2021108

of Master of Computer Application (MCA) as undertaken & completed the Mini project work titled “**Web Based Claim Processing System**” during the academic year 2021-22 .

Submitted to this college in partial fulfilment of the curriculum of Master of Computer Application, University of Mumbai.

Signature

(Course Coordinator)

Signature

(Director IMCOST)

Date:

Place:

Seal

TABLE OF CONTENTS

1. Problem Definition and scope of project	Error! Bookmark not defined.
1.1 Purpose	Error! Bookmark not defined.
1.2 Objective	Error! Bookmark not defined.
1.3 Project Scope	Error! Bookmark not defined.
1.3.1 Employee Module	Error! Bookmark not defined.
1.3.2 CPD Module	Error! Bookmark not defined.
1.4 Technologies	Error! Bookmark not defined.
1.4.1 Operating Environment	Error! Bookmark not defined.
1.4.2 Deployment Environment	Error! Bookmark not defined.
1.4.3 Development Tools and Technologies	Error! Bookmark not defined.
1.4.4 Development Environment	Error! Bookmark not defined.
2. Overall Description	Error! Bookmark not defined.
2.1 User Characteristics	Error! Bookmark not defined.
2.1.1 Employee	Error! Bookmark not defined.
2.1.2 CPD - Claims Processing Department	Error! Bookmark not defined.
2. 2.Assumptions	Error! Bookmark not defined.
3. System Features	Error! Bookmark not defined.
3. 1.System features	Error! Bookmark not defined.
3.1.1 .Description:	Error! Bookmark not defined.
3.1 .2.Constraints	Error! Bookmark not defined.
4. Requirement Analysis	Error! Bookmark not defined.
4.1 Business goals	Error! Bookmark not defined.
4.2 Design goals	Error! Bookmark not defined.
4.3 Feasibility Study	Error! Bookmark not defined.
4.3.1 Economic Feasibility	Error! Bookmark not defined.
4.3.2 Technical Feasibility	Error! Bookmark not defined.

4.3.3 Behavioural Feasibility	Error! Bookmark not defined.
4.4 Business Requirements.....	Error! Bookmark not defined.
4.5 <i>User Requirements</i>	Error! Bookmark not defined.
4.5.1 Employee.....	Error! Bookmark not defined.
4.5.2 CPD	Error! Bookmark not defined.
4.6 Operational requirements.....	Error! Bookmark not defined.
4.7 System Requirements	Error! Bookmark not defined.
4.8 System Constraints.....	Error! Bookmark not defined.
5. Nonfunctional Requirements.....	Error! Bookmark not defined.
5.1 Performance Requirements	Error! Bookmark not defined.
5.2 Availability Requirements	Error! Bookmark not defined.
5.3 Reliability Requirement.....	Error! Bookmark not defined.
5.4 Scalability Requirement	Error! Bookmark not defined.
5.5 Security Requirements.....	Error! Bookmark not defined.
Security Features of .NET Technologies.....	Error! Bookmark not defined.
5.6 Interoperability	Error! Bookmark not defined.
6. Project Management	Error! Bookmark not defined.
6.1 Development Methodologies	Error! Bookmark not defined.
6.1.1 Waterfall model.....	Error! Bookmark not defined.
6.1.2 Spiral model.....	Error! Bookmark not defined.
6.1.3 MSF Process Model:.....	Error! Bookmark not defined.
6.2 Project Development Life Cycle (PDLC)	Error! Bookmark not defined.
6.2.1 Envisioning	Error! Bookmark not defined.
6.2.2 Planning.....	Error! Bookmark not defined.
6.2.3 Developing and Designing the System	Error! Bookmark not defined.
6.2.4 Stabilizing.....	Error! Bookmark not defined.
7. Planning and Estimation	Error! Bookmark not defined.

7.1 Planning.....	Error! Bookmark not defined.
7.1 Application of the MSF model in our project:	Error! Bookmark not defined.
7.2 Estimation.....	Error! Bookmark not defined.
7.2. 1 Basic COCOMO.....	Error! Bookmark not defined.
7.2.2 Function Point Estimation	Error! Bookmark not defined.
7. Preliminary Design	Error! Bookmark not defined.
7.1 Use Case.....	Error! Bookmark not defined.
7.2 Specification of actors	Error! Bookmark not defined.
7.2.1 Employee.....	Error! Bookmark not defined.
7.2.2 CPD - Claims Processing Department.	Error! Bookmark not defined.
7.3 Specification of Use Cases.....	Error! Bookmark not defined.
7.3.1 Use Case Manage Employee.....	Error! Bookmark not defined.
7.3.2 Use Case. Apply Claim.....	Error! Bookmark not defined.
7.3.3 Use Case. Approve Claim	Error! Bookmark not defined.
7.3.4 Use Case. System Login.....	Error! Bookmark not defined.
7.4 Process Flow Chart	Error! Bookmark not defined.
7.5 ER Diagram 1.....	Error! Bookmark not defined.
7.6 ER Diagram 2.....	Error! Bookmark not defined.
7.7 Data Dictionary	Error! Bookmark not defined.
7.7.1 Table: emp_master	Error! Bookmark not defined.
7.7.2 Table: emp_claims.....	Error! Bookmark not defined.
7.7.3. Table: emp_security	Error! Bookmark not defined.
7.7.4 Table: department	Error! Bookmark not defined.
7.7.5. Table: designation	Error! Bookmark not defined.

7.8 Data Flow Diagram DFD	Error! Bookmark not defined.
7.8.1 Context Level	Error! Bookmark not defined.
7.8.2 Level 1 (High Level Diagram).....	Error! Bookmark not defined.
7.9 Planning.....	Error! Bookmark not defined.
7.10 User Interfaces	Error! Bookmark not defined.
7.11 Hardware Interfaces	Error! Bookmark not defined.
7.12 Software Interfaces.....	Error! Bookmark not defined.
7.13 Cost of Implementation.....	Error! Bookmark not defined.
7.14 Effort:.....	Error! Bookmark not defined.
7.14.1. Hardware cost:	Error! Bookmark not defined.
7.14.2 Training Cost:	Error! Bookmark not defined.
7.14.3 Project duration:	Error! Bookmark not defined.
7.14.4 With respect to the customer:.....	Error! Bookmark not defined.
7.15 Scheduling.....	Error! Bookmark not defined.
7.16 Gantt chart:.....	Error! Bookmark not defined.
7.17 PERT chart:.....	Error! Bookmark not defined.
7.17 Class Diagram.....	Error! Bookmark not defined.
7.18 WBS – Work Breakdown Structure of WCPS	Error! Bookmark not defined.
8 Coding and Web Pages	Error! Bookmark not defined.
8.1 WCPS Home Page.....	Error! Bookmark not defined.
8.2 WCPS – My Page.....	Error! Bookmark not defined.
8.3 WCPS – Employee.....	Error! Bookmark not defined.
8.4 WCPS – Add / Edit Employee.....	Error! Bookmark not defined.
8.5 WCPS – Approve Claim – View Status	Error! Bookmark not defined.
8.6 WCPS – Approve / Reject Claims....	Error! Bookmark not defined.
8.7 WCPS –View Status	Error! Bookmark not defined.
8.8 WCPS – Change Password	Error! Bookmark not defined.
9. Test Plan.....	Error! Bookmark not defined.
Introduction	Error! Bookmark not defined.
9.1 Test Scope	Error! Bookmark not defined.
9.2 Test Strategy.....	Error! Bookmark not defined.
9.3 Preconditions	Error! Bookmark not defined.
9.4 Test Priorities	Error! Bookmark not defined.

9.5 Test Techniques	Error! Bookmark not defined.
9.6 Test Organization.....	Error! Bookmark not defined.
Roles and Responsibilities	Error! Bookmark not defined.
9.7 Deliverables.....	Error! Bookmark not defined.
9.8 Test Environment.....	Error! Bookmark not defined.
Hardware and Software.....	Error! Bookmark not defined.
9.9 Testing Automation Software	Error! Bookmark not defined.
9.10 Application Configuration.....	Error! Bookmark not defined.
9.11 Test Management	Error! Bookmark not defined.
9.12 Testing Schedules.....	Error! Bookmark not defined.
9.13 Threats to Testing.....	Error! Bookmark not defined.
10. Future Enhancement.....	Error! Bookmark not defined.
11. Bibliography	Error! Bookmark not defined.
11.1 Websites.....	Error! Bookmark not defined.
11.2 Books	Error! Bookmark not defined.

1. Problem Definition and scope of project

1.1 Purpose

ABC is a company which is having employee strength of at least 1000. All the employees are provided medical reimbursement facility which means that the expenditure incurred by the employee for treatment is reimbursed by the company. For reimbursement, the employee needs to fill in a form detailing the treatment undertaken which includes the name & cost of medicines, laboratory tests, surgery. The form is duly signed by the employee and it will be sent to the concerned Claims Processing Department (CPD) by messenger for processing. CPD will process it and the order regarding the reimbursement is sent to the Cash counter (CC) where in the employee can come and receive the reimbursement amount.

Some of the problems associated with the above mentioned process are listed below:

- The form may miss during transit
- The form is prone to weariness due to which the company may not be able to read data in it after some years.
- The employees who claimed for medical reimbursement need to visit the CC from time to time enquiring about the status of their application. This results in enormous wastage of time of the employee.

To overcome these problems, it is proposed to develop a software titled **Web Based Claims Processing System (WCPS)** which is web based so that the employee can fill the form online and submit it so that the form is sent to CPD through Internet. At CPD, the form needs to be checked automatically by a program which will compute the amount that needs to be reimbursed to the employee for the treatment undertaken. Any excess amount claimed by the employee is ignored by the software. The amount computed will be routed to the e-mail account of the employee as well as to the Bank which holds the

accounts of all the employees of the company. The bank will credit the amount to the account of the employee based on the mail.

This project report describes the software functional and non-functional requirements for release 1.0 of the Web Based Claim Processing System. This document is intended to be used by the members of the project team that will implement and verify the correct functioning of the system. Unless otherwise noted, all requirements specified here is high priority and committed for release 1.0.

1.2 Objective

Web Based Claims Processing System (WCPS) is complete end to end solution to cover all aspects of online claim and reimbursement system.

The basic objective of developing this project is:

- Provides complete web site solution, including employee registration, enter new claim, approve/ reject claim. Complete web based administration.
- The Web Based Claims Processing System can automatically display pending claim count in the main page of employee of Claims Processing Department (CPD).
- Employee can view their claim status online and all claims can be treated as Pending, Approved and Rejected.
- Employee of CPD can add, edit employee details, all employees are identified by the employee no. Employee no will be unique throughout the system.

- System must store audit trail of all DML transaction, like it stores employee no, time stamp in the database for add, edit, delete operation.
- System should capture lat login time for all the employees.
- Complete web based system no installation required to run the application in client system.
- Employee must able to view his/her previous claim details.

1.3 Project Scope

The Web Based Claims Processing System (WCPS) will permit to enter new claim, track the claim status and maintaining master information. The main users of the project are Employee of all departments and Employee of CPD – Claim Processing Department.

From an end-user perspective, the Web Based Claims Processing System Project consists of two functional elements: enhanced Employee modules for Login, Apply for new claim, view status of already applied claim. And CPD module for approve/reject claim, view pending status, Add/Edit employee details.

1.3.1 Employee Module

An enhanced interface for employee to apply for new claim and view the claim status:

- **EM-1: Home** – It is the default page for the site. User of the system needs to login here; system will identified the employee type as per employee number.

- **EM -2: My Page** – After successful login, system will display this page with name of the employee, department and no of claim is pending for the approving.
- **EM -3: Application** – Employee can apply new claim from here, employee must select claim type like medicine, surgery. Claim amount and remarks.
- **EM -4: View Status** – It shows the details of previous claim like claim type, amount, date of apply, approval status, approval date.
- **EM -5: Change Password** – Employee can change his password from this link.
Student must type his old password to change the password with new password.
- **EM -6: Logout** – By clicking this link user logged out from this site all user session reset to default value.

1.3.2 CPD Module

CPD – Claim Processing Department module used to enter and edit employee details, approve/reject claim, view pending status; The Web-based CPD module will include the following features:

- **CPD-1: Login** – Login page for the CPD Employee.
- **CPD -2: Employee** – In this page system display details of all employees. From this page CPD can add new employee or edit information of old employee. System generates employee no for all the new employees. Employee no will be unique and it will be used to identify employee in the system.

- **CPD -3: Approve Claim** – This page list all the pending claim application, CPD employee can click on edit link and view the details and approved or reject the claim.
- **CPD -4: View Status** – It shows the details of previous claim like claim type, amount, date of apply, approval status, approval date.
- **CPD -5: Change Password** – Employee can change his password from this link. Student must type his old password to change the password with new password.
- **CPD -6: Logout** – By clicking this link admin user logged out from this site all user session reset to default value.

Both of these areas of functionality will be delivered as the first version of the Web Based Claims Processing System (WCPS). Functionality is described in more detail later in this document.

1.4 Technologies

1.4.1 Operating Environment

- **OE-1:** The WCPS web application will operate with the following Web Browsers: Microsoft Internet Explorer version 5.0, 6.0. 7.0
- **OE-2:** The WCPS web application shall operate on a server running the latest versions of IIS (Internet Information Server).
- **OE-3:** The WCPS web application shall permit user access from Internet connection

- **OE-4:** Operating System: Windows 2000. XP
- **OE-5:** Software requirements: SQL Server 2005, .net framework 2.0.
- **OE-6:** Languages used are asp.net using c# and scripting is done using JavaScript.
- **OE-7:** Hardware Requirements:
256(minimum)/512(recommended) MB RAM
- **OE-8:** Hard disc- nGB depending upon the requirement to store data minimum of 25GB.

1.4.2 Deployment Environment

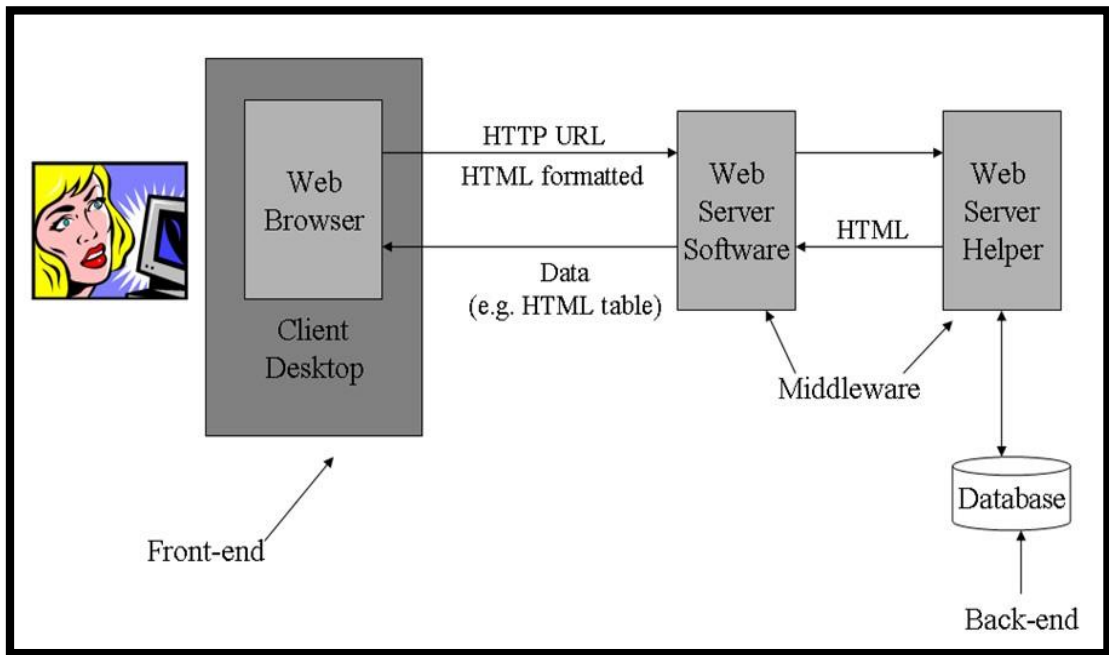
□ **DE-1: Database Server**

- OS – Win 2003 Enterprise Server
- SQL Server 2005
- HDD – Min 10 GB, Recommended 25 GB
- RAM – Min 2 GB, Recommended 4 GB
- Processor - Pentium Dual Xenon Processor

□ **DE-2: Application Server**

- OS – Win 2003 Enterprise Server
- IIS – Internet Information Server
- HDD – Min 5 GB, Recommended 10 GB
- RAM – Min 2 GB, Recommended 4 GB
- Processor - Pentium Dual Xenon Processor

- **DE-3:** The WCPS web application will operate with the following Web Browsers: Microsoft Internet Explorer version 5.0, 6.0, 7.0.



[Web Based Claims Processing System (WCPS) System Architecture]

1.4.3 Development Tools and Technologies

□ **DT-1: ASP.Net**

As we need to develop Web Application for Web Based Claims Processing System (WCPS). We will use ASP.Net as it is the new Microsoft technology to develop any application which support and integrate other server product for the internet. By ASP.Net we can develop in web application by .NET framework and manage environment with any .NET support language like VB.Net and C#.

□ **DT-2: C#**

.NET is built on the Windows Server System to take major advantage of the OS and which comes with a host of different servers which allows for building, deploying, managing and maintaining Web-based solutions. The Windows Server System is designed with performance as priority and it provides scalability, reliability, and manageability for the global, Web-enabled enterprise. The Windows Server System integrated software products are built for interoperability using open Web standards such as XML and SOAP.

.NET is a "Software Platform". It is a language-neutral environment for developing rich .NET experiences and building applications that can easily and securely operate within it. When developed applications are deployed, those applications will target .NET and will execute wherever .NET is implemented instead of targeting a particular Hardware/OS combination. The components that make up the .NET platform are collectively called the .NET Framework.

The .NET Framework is a managed, type-safe environment for developing and executing applications. The .NET Framework manages all aspects of program execution, like, allocation of memory for the storage of data and instructions, granting and denying permissions to the application, managing execution of the application and reallocation of memory for resources that are not needed.

The .NET Framework is designed for cross-language compatibility. Crosslanguage compatibility means, an application written in Visual Basic .NET may reference a DLL file written in C# (C-Sharp). A Visual Basic .NET class might be derived from a C# class or vice versa.

The .NET Framework consists of two main components:

Common Language Runtime (CLR)

Class Libraries

The advantage of C# includes

Powerful Windows-based Applications

Simplified Deployment

- Powerful, Flexible, Simplified Data Access
- Improved Coding
- Direct Access to the Platform
- Full Object-Oriented Constructs
- XML Web Services
- COM Interoperability

□ DT-3: SQL Server 2005

WCPS Application uses SQL Server 2005 as storing the data. Microsoft SQL Server 2005 as our database and it has so many features which is ideal for our dot net based application. Features Includes:

- Support for Multiple Platforms
- Integration with Windows Back office servers
- Integration with Microsoft .NET Enterprise Servers
- Scalability
- Replication
- Centralized Management
- Reliability
- Automating Tasks

1.4.4 Development Environment

□ DE-1: 1. Visual Studio 2005 IDE

Most advanced integrated development environment for developing .NET application. Microsoft Visual Studio is an integrated development environment (IDE) from Microsoft. It can be used to develop console and graphical user interface applications along with Windows Forms applications, web sites, web applications, and web services in both native code together with managed code for all platforms supported by Microsoft Windows, Windows Mobile, Windows CE, .NET Framework, .NET Compact Framework and Microsoft Silverlight.

Visual Studio includes a code editor supporting IntelliSense as well as code refactoring. The integrated debugger works both as a source-level debugger and a machine-level debugger. Other built-in tools include a forms designer for building GUI applications, web designer, class designer, and database schema designer. It allows plug-ins to be added that enhance the functionality at almost every level - including adding support for source control systems (like Subversion and Visual SourceSafe) to adding new toolsets like editors and visual designers for domain-specific languages or toolsets for other aspects of the software development lifecycle (like the Team Foundation Server client: Team Explorer).

Visual Studio supports languages by means of language services, which allow any programming language to be supported (to varying degrees) by the code editor and debugger, provided a language-specific service has been authored. Built-in languages include C/C++ (via Visual C++), VB.NET (via Visual Basic .NET), and C# (via Visual C#). Support for other languages such as Chrome, F#, Python, and Ruby among others has been made available via language services which are to be installed separately. It also supports XML/XSLT,

HTML/XHTML, JavaScript and CSS. Language-specific versions of Visual

Studio also exist which provide more limited language services to the user. These individual packages are called Microsoft Visual Basic, Visual J#, Visual C#, and Visual C++.

2. Overall Description

2.1 User Characteristics

2.1.1 Employee

Able to login, apply for new claim, view claim status, change password.

2.1.2 CPD - Claims Processing Department

CPD Employee can add new employee, edit employee details, approve / reject claim, view status, change his/her own password.

2.2. Assumptions

1) CPD and Employees communicate with each other via emails.

3. System Features

3.1. System features

3.1.1 .Description:

Proposed to develop a software titled Web Based Claims Processing System (WCPS) which is web based so that the employee can fill the form online and submit it so that the form is sent to CPD through Internet. At CPD, the form needs to be checked automatically by a program which will compute the amount that needs to be reimbursed to the employee for the treatment undertaken. Any excess amount claimed by the employee is ignored by the software. The amount computed will be routed to the e-mail account of the employee as well as to the

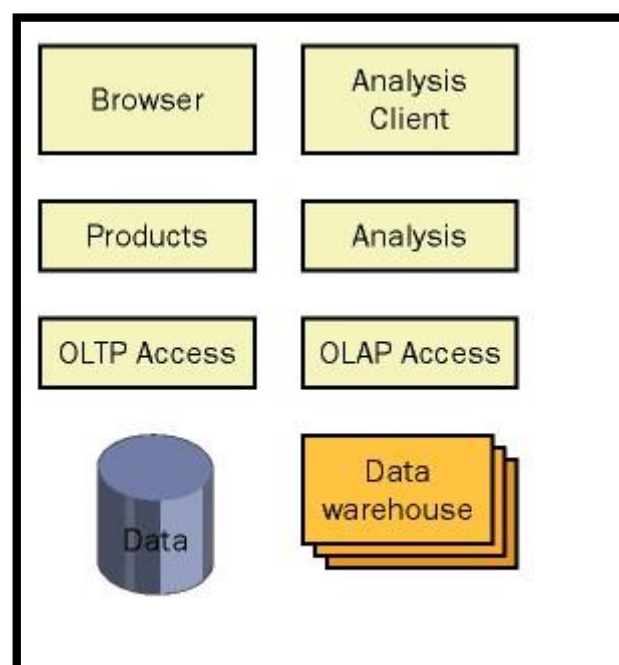
Bank which holds the accounts of all the employees of the company. The bank will credit the amount to the account of the employee based on the mail.

3.1 .2.Constraints

Linking and integration with other legacy system for data sharing. Connecting to third-party OLAP applications for generating reports. Etc.

4. Requirement Analysis

The requirement analysis outlines the approach the development team will take to meet the goals of the project and provides the basis for proceeding to the planning phase. After identifying the business problem and defining the vision and scope, the team creates the solution concept that explains in general terms how the team intends to meet the requirements of the project.



[Solution concept of Web Based Claims Processing System (WCPS)]

For a project to be successful, it is essential that we correctly identify the goals of the project. Project goals can be categorized as follows:

- Business goals
- Design goals

4.1 Business goals

Business goals represent what the organization wants to achieve with the solution. Business goals form the basis for determining the success criteria of the solution. The purpose of defining business goals is to clearly articulate the objectives for the project and to ensure that your solution supports those business requirements. The team needs to determine the best method for identifying the goals and agreeing on them.

Throughout the life of the project, the team makes tradeoffs among resources, schedule, and features. It is important that business goals are prioritized in a way that will allow the team to have a clear understanding about which ones the customer believes are most important, in case some of the goals cannot be achieved.

For this Web Based Claims Processing System (WCPS) project, business goals might include the following:

- Expand the organization's geographic reach beyond the current physical location.
- Integrate all employees worldwide by using a online site process, and shorten the overall claim process.
- More transparent process of claim processing
- Online status updation, Employee can view the status of his/her claim online.
- No additional and specific system requirement to operate the WCPS site.
- It must be secure.
- Easy to use and no addition knowledge require to operate the system.
- Reliable and accurate and their should not be any manual intervention in claim process.

4.2 Design goals

Design goals are similar to business goals in many ways. The difference is that design goals focus more on the attributes of the solution and less on what the solution will accomplish for the business. Design goals address not only what the team wants to accomplish but also what the team is not trying to accomplish with the solution. As with business goals, we need to prioritize design goals so that the team knows which goals must be accomplished, in case the project cannot achieve all of them.

Consider the case of this Web Based Claims Processing System (WCPS) Web site. Some of the design goals for the online shopping cart might include:

- Improve the user experience by reducing page-download wait times to 5 seconds or less.
- Limit dependency on connectivity with the server.
- Reduce the time and level of effort required for employee to apply new claim.
- The service and all supporting applications must be localized for users worldwide.
- The service must have an availability of 99.99 percent.
- The service cannot lose data.
- The service must permit access only by authorized users.

4.3 Feasibility Study

4.3.1 Economic Feasibility

Economic analysis is most frequently used for evaluation of the effectiveness of the system. More commonly known as cost/benefit analysis the procedure is to determine the benefit and saving that are expected from a system and compare them with costs, decisions is made to design and implement the system. This part of feasibility study gives the top management the economic justification for the new system. This is an important input to the management the management, because very often the top management does not like to get confounded by the various technicalities that bound to be associated with a project of this kind. A simple economic analysis that gives the actual comparison of costs and benefits is much more meaningful in such cases. In the system, the

organization is most satisfied by economic feasibility. Because, if the organization implements this system, it need not require any additional hardware resources as well as it will be saving lot of time.

4.3.2 Technical Feasibility

Technical feasibility centres on the existing manual system of the test management process and to what extent it can support the system. According to feasibility analysis procedure the technical feasibility of the system is analyzed and the technical requirements such as software facilities, procedure, inputs are identified. It is also one of the important phases of the system development activities. The system offers greater levels of user friendliness combined with greater processing speed. Therefore, the cost of maintenance can be reduced. Since, processing speed is very high and the work is reduced in the maintenance point of view management convince that the project is operationally feasible.

4.3.3 Behavioural Feasibility

People are inherently resistant to change and computer has been known to facilitate changes. An estimate should be made of how strong the user is likely to move towards the development of computerized system. These are various levels of users in order to ensure proper authentication and authorization and security of sensitive data of the organization.

4.4 Business Requirements

The following preliminary lists are based on initial interviews and study of existing manual system, the business goal for the application is to support an increase the security, speed, accuracy and complete automation of existing manual or semi-automatic claim process. Business requirements are discussed in the Scope section, with the following additional detail:

- Provides complete web site solution, including employee registration, enter new claim, approve/ reject claim. Complete web based administration.
- The Web Based Claims Processing System can automatically display pending claim count in the main page of employee of Claims Processing Department (CPD).
- Employee can view their claim status online and all claims can be treated as Pending, Approved and Rejected.
- Employee of CPD can add, edit employee details, all employees are identified by the employee no. Employee no will be unique throughout the system.
- System must store audit trail of all DML transaction, like it stores employee no, time stamp in the database for add, edit, delete operation.
- System should capture lat login time for all the employees.
- Complete web based system no installation required to run the application in client system.
- Employee must able to view his/her previous claim details.
- The application should support the capability to use multi user environment.
- The Admin user should able to generate all type of reports as and when required by the management.
- Can be used anywhere any time as it is a web based application (user location doesn't matter).

4.5 User Requirements

User requirements are categorized by user type.

4.5.1 Employee

- Able to login into the site
- Able to Apply for new claim.
- Able to view status of old claim details

4.5.2 CPD

- Able to manage the employee, add, edit employee details.
- Must able to approve / reject claim.
- Must able to view status of claim details.

4.6 Operational requirements

The following requirements provide a high-level view of how the system will run:

- Processor usage should not exceed 80 percent during concurrent uses.
- Backups will occur incrementally throughout the day.
- A full weekly backup is required to WORM drives.
- Ensure that information is easy to access either, and meaningful for the system users and the company.

- Minimize the technical knowledge that system users and student need to access the data, generate ad hoc queries, search and view claim details.
- Any change to information must be reflected immediately, and the changes must be propagated to the search engine so that system users that perform searches see this new information.
- The application should work with the existing communications and networking infrastructure.
- The application should deploy with a minimum of additional operational processes, manual or otherwise.

4.7 System Requirements

These are additional constraints from a system perspective:

- The administrator must be able to monitor everything from the IT department.
- The information must be accessible by everyone in the organisation in intranet and in internet for the students and administrator as per the rights specify.
- All data especially question bank must be up to date.

4.8 System Constraints

Constraints indicate the parameters to which the final business solution must adhere. They are aspects of the business environment that cannot or will not be changed. Often, these constraints become design goals for the application. If constraints are not identified properly, the project team might design a product that cannot be deployed within the business.

Examples of possible constraints that you should document include:

- Budget limitations
- Characteristics of earlier supporting systems
- Network system architecture
- Security requirements
- Operating systems
- Planned upgrades to technologies
- Network bandwidth limitations
- Maintenance and support agreements and structures
- Knowledge level of development or support staff
- Learning limitations of users

5. Non-functional Requirements

5.1 Performance Requirements

An application's performance is defined by metrics such as transaction throughput and resource utilization. A user might define an application's performance in terms of its response time. No more than 5-percent degradation in average query response is allowed while all concurrent users are using the system. Processor utilization should not exceed 80 percent during all concurrent users are using the system.

We must define performance requirements before the team proceeds to the developing phase. To define a good performance requirement, we must identify project constraints, determine services that the application will perform, and specify the load on the application.

- **PR-1: Identifying constraints** - Constraints in the project include budget, schedule, infrastructure, and the choice of development tools or technologies. For example, we might need to deploy this Web Based Claims Processing System (WCPS) by a specific date. We might also need to use a specific development tool because the team has expertise in that tool only. We might not be able to design and develop applications that are processor intensive because the client computers do not have adequate hardware. We need to design an application so that it meets its performance goals within the limitations of the constraints. Instead of changing some aspects of a project to improve performance, you can modify aspects of the project that are not constrained to determine how we can improve performance. For example, can the team be trained so that they can create components by using a different tool? Can data access be improved by changing the data access technology?
- **PR-2: Determining features** - The features of this application correspond to use cases and usage scenarios. We can identify the usage scenarios that affect the performance of the application and, for each such scenario, specify what the user does and what the

application does in response, including how databases and other system services are accessed. In addition, you need to determine how often each feature will be used. This information can help you create tests for measuring performance that resemble actual usage of the application as closely as possible.

- **PR-3: Specifying the load** - We can specify the load of this Web Based Claims Processing System (WCPS) as the number of students that will use the application. In addition, we can examine how the load might vary over time. For example, the number of requests for this Web Based Claims Processing (WCPS) site will be higher during certain times of year. We can use the load to define the performance metrics of this application.

5.2 Availability Requirements

Availability is a measure of how often the application is available to handle service requests as compared to the planned run time. Availability also takes into account repair time because an application that is being repaired is not available for use.

Designing for availability includes anticipating, detecting, and resolving hardware or software failures before they result in service errors, faults, or data corruption, thereby minimizing downtime. To ensure availability, provide multiple routes to application services and data. Use only tested proven processes (both automated and people-based) that support the application throughout its life cycle.

In addition to unplanned downtime, planned downtime must be reduced. Planned downtime can include maintenance changes, operating system upgrades, backups, or any other activity that temporarily removes the application from service.

Availability of an application also depends on its reliability. For a highly available and reliable application, we need a reliable foundation: good application design, rigorous testing, and certification. Some of the techniques used for designing for availability include:

- **AR-1: Reduce planned downtime** - To avoid planned downtime, use rolling upgrades. For example, to update a component on a server, move the server's resource groups to another server, take the server offline, update the component, and then bring the server online. Meanwhile, the other servers handle the workload, and this application experiences no downtime. You can use this strategy in an application that scales out.
- **AR-2: Reduce unplanned downtime with clustering** - Clustering is a technology for creating high-availability applications. A cluster consists of multiple computers that are physically networked and logically connected using cluster software. By using clustering, a multiple server Web site can withstand failures with no interruption in service. When the active server fails, the workload is automatically moved to a passive server, current client processes are switched over, and the failed application service is restarted automatically. If a resource fails, customers connected to that server cluster might experience a slight delay, but the service will be completed. Cluster software can provide failover support for applications, file and print services, databases, and messaging systems that have been designed as cluster-aware and assigned to a cluster.
- **AR-3: Use network load balancing** - Network load balancing (NLB) is used to distribute traffic evenly across available servers. NLB also helps increase the availability of an application: if a server fails, you can use NLB to redefine the cluster and direct traffic to the other servers. NLB is especially beneficial for Web Based Claims Processing System (WCPS) that link external clients with transactions to data servers. As client traffic increases, you can scale out the Web server farm by adding up to 32 servers in a single cluster. NLB automatically detects server failures and redirects client traffic to the remaining servers, all the time maintaining continuous, unbroken client service.
- **AR-4: Use redundant array of independent disks (RAID) for data stores.** - RAID uses multiple hard disks to store data in multiple places. If a disk fails, the application is transferred to a mirrored data image

and the application continues running. The failed disk can be replaced without stopping the application.

- **AR-5: Isolate mission-critical applications** - An application is constantly performing tasks and requesting resources such as network communications, data access, or process threads. Each of these resource requests can affect the performance and availability of applications sharing the same resources. If an application shares these services on the same servers, the workload and throughput characteristics for these servers might change unfavorably. It is recommended that mission-critical applications use dedicated infrastructures and private networks.
- **AR-6: Use queuing** - Queuing enables your application to communicate with other applications by sending and receiving asynchronous messages. Queuing guarantees message delivery; it does not matter whether the necessary connectivity currently exists (with mobile applications, for example). Queuing removes a failure point from the application. Queuing is also a solution for managing peak workloads that can require a lot of hardware. In addition, by increasing the number of routes for successful message delivery, an application can increase the chances for successful and immediate message completion.

Calculation of availability

Measurement Types for Calculating Availability		
Name	Calculation	Definition
Mean Time Between Failure (MTBF)	Hours / Failure Count	Average length of time the application runs before failing
Mean Time To Recovery (MTTR)	Repair Hours / Failure Count	Average length of time needed to repair and restore service after a failure

The formula for calculating availability is:

$$\text{Availability} = (\text{MTBF} / (\text{MTBF} + \text{MTTR})) \times 100$$

For example, the typical availability requirement for this WCPS application is that the site is available 24 hours a day, 7 days a week. If you assume 1000 continuous hours as a checkpoint, two 1-hour failures during this time period results in availability of:

$$((1000 / 2) / ((1000 / 2) + 1)) \times 100 = (500 / 501) \times 100 = .998 \times 100 = 99.8\%.$$

A popular way to describe availability is by the nines, for example, three nines for 99.9 percent availability. However, the implication of measuring by nines is often misunderstood. We need to do the arithmetic to discover that three nines (99.9 percent availability) represent about 8.5 hours of service outage in a single year. The next level, four nines (99.99 percent), represents about 1 hour of service outage in a year. Five nines (99.999 percent) represent about 5 minutes of outage per year.

5.3 Reliability Requirement

The reliability of an application refers to the ability of the application to provide accurate results. Reliability and availability are closely related. While availability measures the capacity to handle all requests and to recover from a failure with the least loss of access to the application, reliability measures how long the application can execute and produce expected results without failing. Users bypass unreliable Web sites, resulting in lost revenue and reduced future sales. In addition, the expense of repairing corrupted data increases the cost of application failure. Unreliable systems are also difficult to maintain or improve because the failure points are typically hidden throughout the system. Because of the need no single point failure, automatic failover will be required. In addition, existing disaster recovery and backup plans and procedures must be revised to incorporate the Web Based Claims Processing System (WCPS).

To design for reliability, you need to examine the application's expected usage pattern, create a reliability profile, and create a solution that meets the profile. You must examine how a particular service is provided, evaluate failure scenarios, and design preferred alternatives. In addition, you need to consider the application's interactions with other applications.

It is difficult to identify reliability problems and solutions for a system that has not been developed. However, we can begin by analyzing the currently running applications in the organization. Such analysis reveals the failure frequency and distribution, root causes, and possible improvements for existing systems. We can use this information to design a reliable solution.

A reliable solution ensures error-free data input, data transformations, state management, and non-corrupting recovery from any failure conditions. Creating a high-reliability application depends on the entire software development lifecycle, from the planning phase, through development and testing, to deployment and stabilizing. The following tasks can help you create a reliable application:

- Putting reliability requirements in the specification
 - Using a good architectural infrastructure
 - Including management information in the application
 - Using redundancy
 - Using quality development tools
 - Using reliability checks that are provided by the application □
- Implementing error handling
- Reducing the application's functionality instead of completely failing the application

5.4 Scalability Requirement

Scalability is defined as the capability to increase resources to produce an increase in the service capacity. In a scalable application, you can add resources to manage additional demands without modifying the application itself.

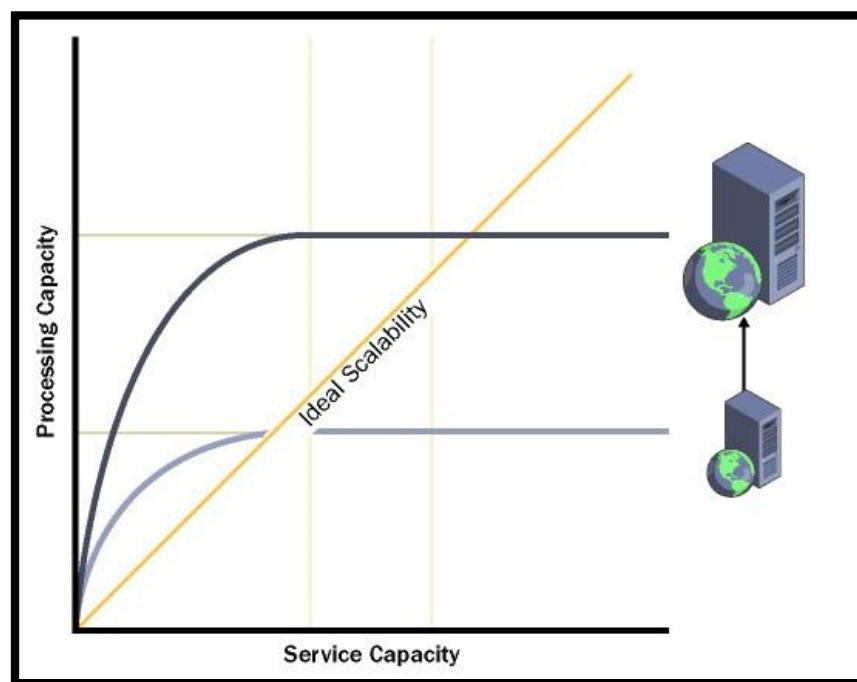
A scalable application requires a balance between the software and hardware used to implement the application. You might add resources to either software or hardware to increase the scalability of the application. Adding these resources might produce a benefit; however, it could also have a negative or null effect, with the application showing no significant increase in service capacity.

For example, you might implement load balancing in an application. This will help only minimally if the application has been written to make synchronous method calls or to retrieve lengthy datasets in response to a user's request.

Web Based Claims Processing System (WCPS) Application an average load of 1000 concurrent users after the system is fully operational, and expects that to grow by 25 percent each year for the next five years.

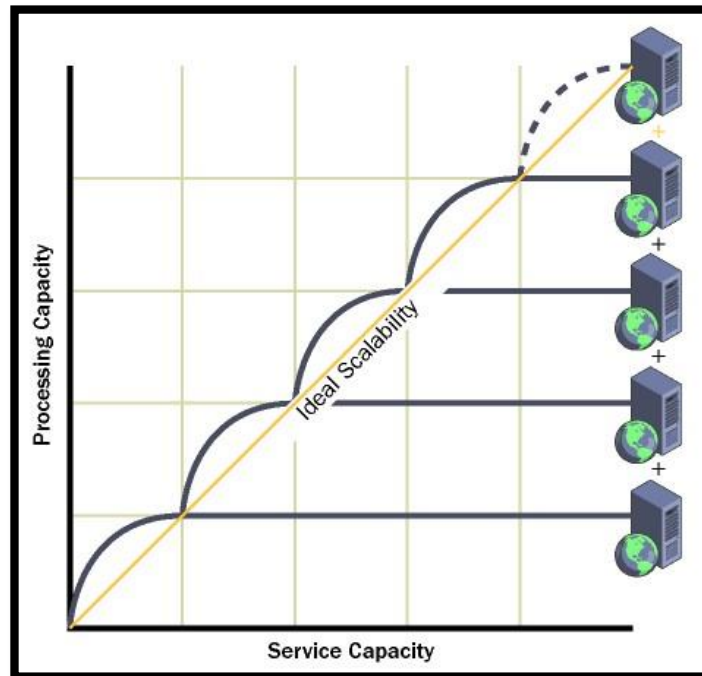
The two most common approaches to scalability are:

- **SR-1: Scaling up** - Refers to achieving scalability by improving the existing server's processing hardware. Scaling up includes adding more memory, more or faster processors, or migrating the application to a more powerful computer. The primary goal of scaling up an application is to increase the hardware resources available to the application. Typically, you can scale up an application without changing the source code. In addition, the administrative effort does not change drastically. However, the benefit of scaling up tapers off eventually until the actual maximum processing capability of the machine is reached.



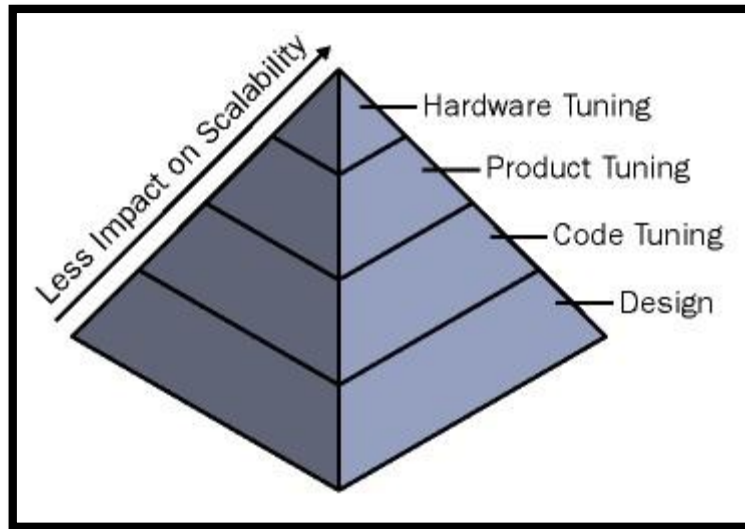
- **SR-2: Scaling out** - Refers to distributing the processing load across more than one server. Although scaling out is achieved by using multiple computers, the collection of computers continues to

act as the original device configuration from the end-user perspective. Again, the balance between software and hardware is important. The application should be able to execute without needing information about the server on which it is executing. This concept is called location transparency. Scaling out also increases the fault tolerance of the application.



Good design is the foundation of a highly scalable application. The planning phase has the greatest impact on the scalability of an application.

Bellow figure illustrates the role of design, code tuning, product tuning, and hardware tuning in the scalability of an application. Design has more impact on the scalability of an application than the other three factors. As you move up the pyramid, the impact of various factors decreases. The pyramid illustrates that effective design adds more scalability to an application than increased hardware resources.



To design for scalability, we need to following guidelines:

- **SR-3: Design processes such that they do not wait** - A process should never wait longer than necessary. A process can be categorized as synchronous or asynchronous. A synchronous process waits for another process to complete before it continues. Such processes must wait for another process to succeed or fail completely before performing another operation. Applications that implement synchronous processes encounter bottlenecks for resources. These bottlenecks affect both the performance and the scalability of the application. One way to achieve scalability is to implement asynchronous processes. In applications that have asynchronous processes, long-running operations can be queued for completion later by a separate process.
- **SR-4: Design processes so that processes do not compete for resources** - One of the biggest causes of scalability problems is competition for resources such

as memory, processor cycles, bandwidth, or database connections. Plan your resource usage to minimize these problems:

- Sequence resource usage to use the most plentiful resources first and the least plentiful resources last.
 - Acquire resources as late as possible. The shorter the amount of time a process uses a resource, the sooner the resource becomes available to another process.
- **SR-5: Design processes for commutability** - Two or more operations are called commutative if they can execute in any order and still obtain the same result.
 - **SR-6: Design components for interchangeability** - An interchangeable component is designed to release its resources, move into a pool managed by a resource manager, and be re-initialized for use by a new client. Design the component so that no client-specific state persists from client to client. In addition, the component should support aggregation and not be bound to a specific thread. Resource pooling schemes such as COM+ component pooling and Open Database Connectivity (ODBC) connection pooling use interchangeable resources. For example, you can use the Component Services Administration tool to enable object pooling, set minimum and maximum pool size, and create timeout settings.
 - **SR-7: Partition resources and activities** - Minimize relationships between resources and activities by partitioning them. This helps you avoid the risk of bottlenecks. Partitioning activities can also ease the load on critical resources such as the processor and bandwidth. For example, using Secure Sockets Layer (SSL) to provide a secure connection results in significant overhead. Therefore, you might decide to use SSL only for pages that require a high level of security and use dedicated Web servers to handle SSL sessions. You can also partition resources and activities by creating many small components rather than a few large components, and by limiting cross-device communication. However, partitioning can make a system more complex. Dividing resources that have dependencies can add significant overheads to an operation.

5.5 Security Requirements

Malicious attackers use various methods to exploit system vulnerabilities to achieve their goals. Vulnerabilities are weak points or loopholes in security that an attacker exploits to gain access to an organization's network or to resources on the network. Some vulnerabilities, such as weak passwords, are not the result of application or software development design decisions. However, it is important for an organization to be aware of such security weaknesses to better protect its systems. Common vulnerabilities of applications include:

- **SR-1: Weak passwords** - A weak password might give an attacker access not only to a computer, but to the entire network to which the computer is connected.
- **SR-2: Misconfigured software** - Often the manner in which software is configured makes the system vulnerable. If services are configured to use the local system account or are given more permissions than required, attackers can exploit the services to gain access to the system and perform malicious actions on the system.
- **SR-3: Social engineering** - A common form of discovering passwords that generally occurs when users are not aware of security issues and can be deceived into revealing their passwords. For example, an attacker posing as a help desk administrator might persuade a user to reveal his or her password under the pretext of performing an administrative task.
- **SR-4: Internet connections** - The default installation of Internet Information Services (IIS) version 5.0 often enables more services and ports than are necessary for the operation of a specific application. These additional services and ports provide more opportunities for potential attacks. For example, modem connections bypass firewalls that protect networks from outside intruders. If an intruder can identify the modem telephone number and password, the intruder can connect to any computer on the network.
- **SR-5: Unencrypted data transfer** - If the data sent between a server and the users is in clear text, there is a possibility that the data can be intercepted, read, and altered during transmission by an attacker.

- **SR-6: Buffer overrun** - Malicious users probe applications looking for ways to trigger a buffer overrun because they can use a buffer overrun to cause an application or an operating system to crash. They can then find more security weaknesses by reading error messages.
- **SR-7: SQL injection** - SQL injection occurs when developers dynamically build SQL statements by using user input. The attacker can modify the SQL statement and make it perform operations that were not intended.
- **SR-8: Secrets in code** - Many security problems are created when a malicious user is able to find secrets that are embedded in code, such as passwords and encryption keys. To design a secure Web Based Claims Processing System (WCPS), we should be familiar with the following principles of security and employ them when creating security strategies:
- **SR-9: Rely on tested and proven security systems** - Whenever possible, we should rely on tested and proven security systems rather than creating your own custom solution. Use industry-proven algorithms, techniques, platform-supplied infrastructure, and vendor-tested and supported technologies. If we decide to develop a custom security infrastructure, validate our approach and techniques with expert auditing and security review organizations before and after implementing them.
- **SR-10: Never trust external input** - We should validate all data that is entered by users or submitted by other services.
- **SR-11: Assume that external systems are not secure** - If our application receives unencrypted sensitive data from an external system, assume that the information is compromised.
- **SR-12: Apply the principle of least privilege** - Do not enable more attributes on service accounts than those minimally needed by the application. Access resources with accounts that have the minimal permissions required.
- **SR-13: Reduce available components and data** - Risk will increase with the number of components and amount of data you have made

available through the application, so you should make available only the functionality that you expect others to use.

- **SR-14: Default to a secure mode** - Do not enable services, account rights, and technologies that you do not explicitly need. When we deploy the application on client or server computers, its default configuration should be secure.
- **SR-15: Do not rely on security by obscurity** - Encrypting data implies having keys and a proven encryption algorithm. Secure data storage will prevent access under all circumstances. Mixing up strings, storing information in unexpected file paths, and so on, is not security.
- **SR-9: Follow STRIDE principles** - Each letter in the STRIDE acronym specifies a different category of security threat: spoofing identity, tampering, repudiation, information disclosure, denial of service, and elevation of privilege. These are classes of security vulnerabilities a system needs to protect itself against.

Security Features of .NET Technologies

.NET Web applications implement one or more of the logical services by using technologies such as Microsoft ASP.NET, Enterprise Services, XML Web services, remoting, Microsoft ADO.NET, and Microsoft SQL Server. To create effective security strategies, we need to understand how to fine-tune the various security features within each product and technology area, and how to make them work together.

Authentication

Authentication is the process of discovering and verifying the identity of a user by examining the user's credentials and then validating those credentials against some authority. A variety of authentication mechanisms are used, some of which can be used with .NET Framework rolebased security.

Examples of commonly used authentication mechanisms include the operating system, Passport, and application-defined mechanisms, such as NTLM and Kerberos version 5 authentication.

Authorization

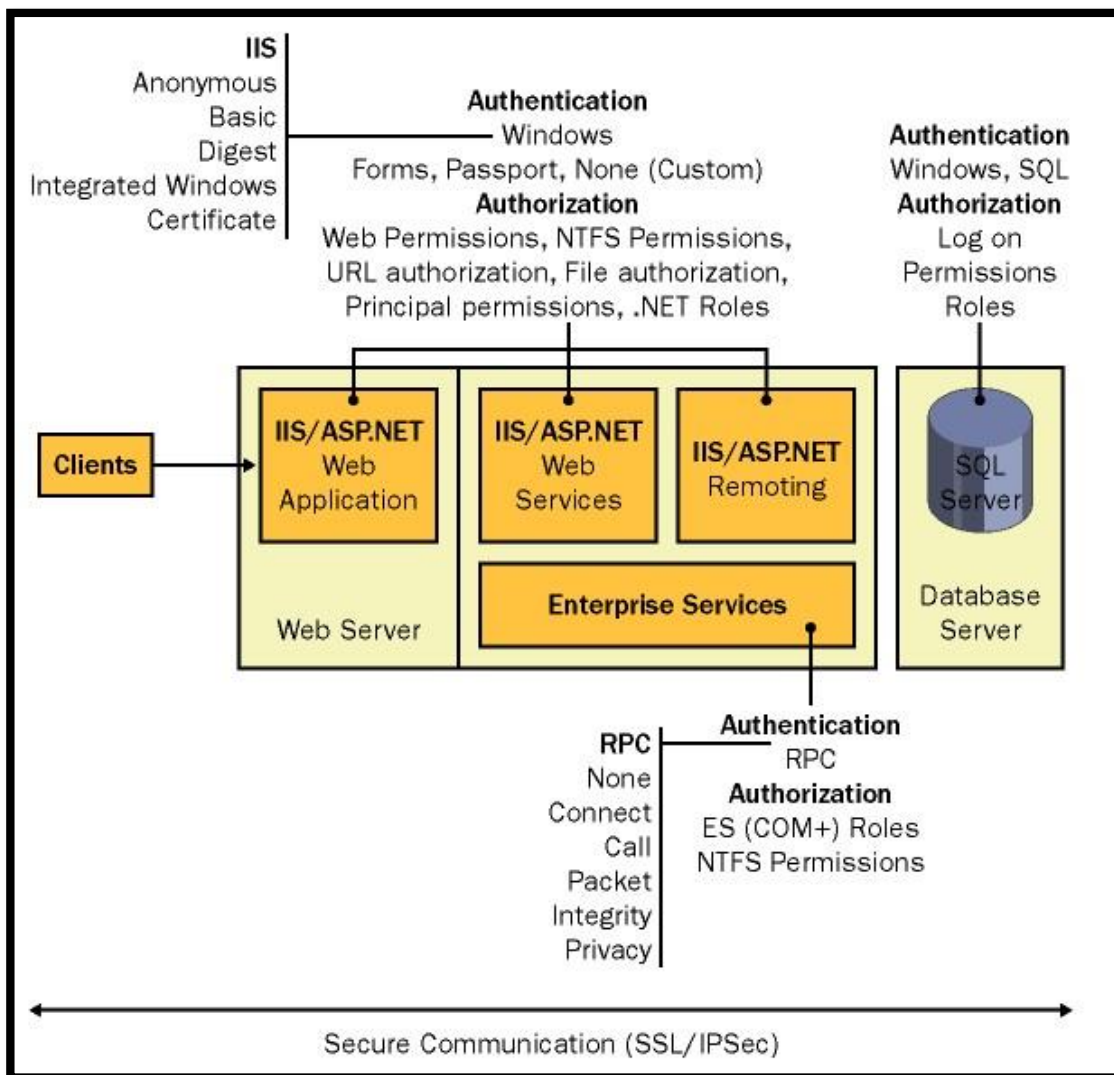
Authorization is the process of determining whether a user is allowed to perform a requested action. Authorization occurs after authentication and uses information about a user's identity and roles to determine what resources that user can access. You can use .NET Framework rolebased security to implement authorization.

ASP.NET security

ASP.NET provides a useful tool for application developers to use to create Web pages. When a Web site records a user's credit card information, the file or database that stores such information must be secured from public access. ASP.NET, in conjunction with IIS, can authenticate user credentials such as names and passwords

ADO.NET and SQL Server

ADO.NET provides data access services. It is designed for distributed Web applications, and supports disconnected scenarios. When we build Web-based applications, it is essential that we must use a secure approach to accessing and storing data. ADO.NET and SQL Server provide several security features that can be used to ensure secure data access.



5.6 Interoperability

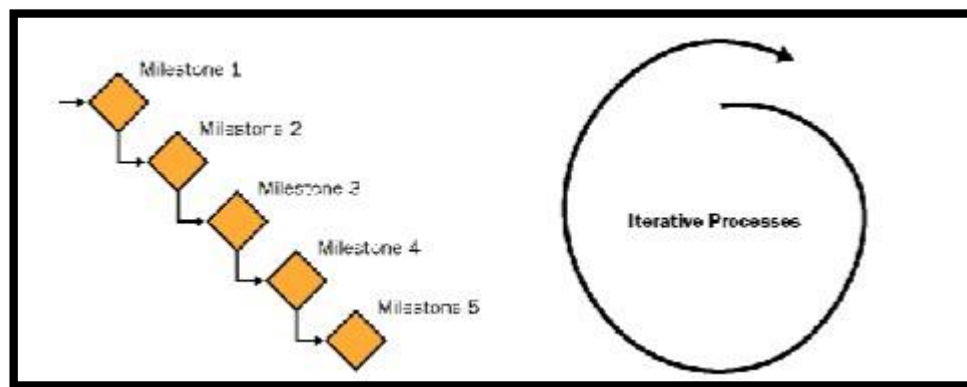
Typically, medium and large organizations have heterogeneous computing environments. For example, many organizations deploy distributed n-tier client/server applications that require access to data or transactions on existing systems. In addition, our application might need to interact with applications that have been developed using proprietary or third-party software. In Version 1.0 of the Web Based Claims Processing System (WCPS), there are no requirements for interoperability with other systems.

6. Project Management

6.1 Development Methodologies

What Are Process Models and development methodology?

A process model guides the order of project activities and represents the life cycle of a project. Historically, some process models are static and others do not allow checkpoints. Two such process models are the waterfall model and the spiral model.



[The waterfall model and the spiral model]

These models provide two different approaches to the project life cycle. The preceding illustration shows the waterfall model's cascading checkpoints and the spiral model's circular approach to process.

6.1.1 Waterfall model.

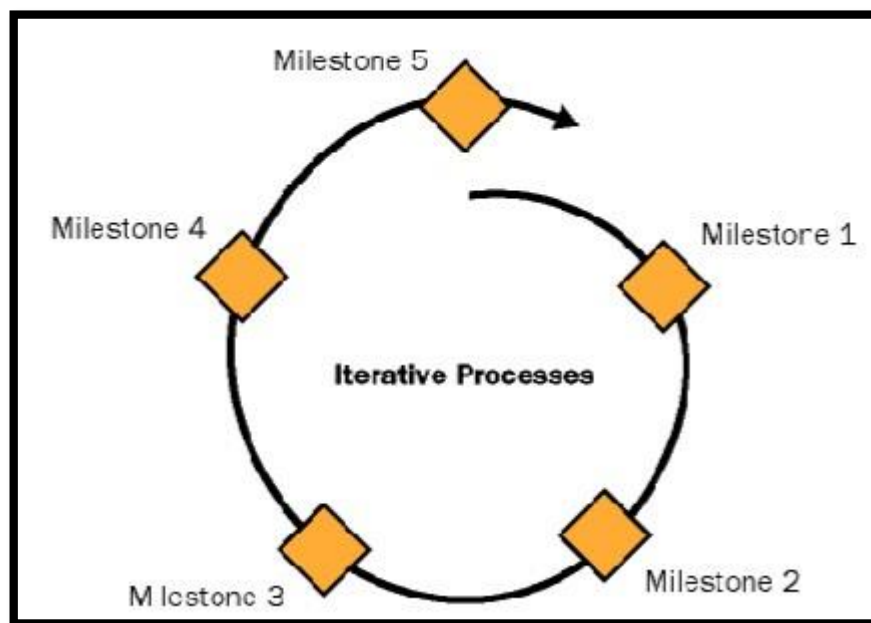
This model uses milestones as transition and assessment points. When using the waterfall model, you need to complete each set of tasks in one phase before moving on to the next phase. The waterfall model works best for projects in which the project requirements can be clearly defined and are not liable to modifications in the future. Because this model has fixed transition points between phases, you can easily monitor schedules and assign clear responsibilities and accountability.

6.1.2 Spiral model.

This model is based on the continual need to refine the requirements and estimates for a project. The spiral model is effective when used for rapid application development of very small projects. This approach can generate great synergy between the development team and the customer because the customer is involved in all stages by providing feedback and approval. However, the spiral model does not incorporate clear checkpoints. Consequently, the development process might become chaotic.

6.1.3 MSF Process Model:

The MSF – Microsoft Solution Framework Process Model combines the best principles of the waterfall and spiral models. It combines the waterfall model's milestone-based planning and resulting predictability with the spiral model's benefits of feedback and creativity.



[MSF Process Model]

In Web Based Claims Processing System (WCPS) devolvment process we have used MSF Process Model. This is a combination of spiral and waterfall process.

6.2 Project Development Life Cycle (PDLC)

The project will be completed by this four distinct phases as per MSF Process Model.

6.2.1 Envisioning

Envisioning is gathering the requirement of the project from different sources. Some techniques for gathering information are interviewing, shadowing, user instructions, and prototyping. Creating and identify the project scope. The scope of the project specifies what will and will not be included in the project. In this phase we have created the scope document. It includes information about the team and project structure, the problem statement, the vision statement, the scope of the project, the solution concept, user profiles, and project goals.

6.2.2 Planning

The planning phase results in the architecture and design of the solution, the plans to accomplish the development and deployment of the solution, and the schedules associated with tasks and resources. There are three design processes in the planning phase: conceptual, logical, and physical design.

6.2.3 Developing and Designing the System

Design of any application is not complete without a way for users to interact with the system. User interaction takes place through the application's presentation layer. The presentation layer is the part of the application that

provides a communication mechanism between the user and the business service layer of the system. The most simple presentation layers contain user interface components, such as Windows Forms or ASP.NET Web Forms. For more complex user interactions, we can design user process components to orchestrate the user interface elements and control the user interaction. User interface components display data to users, acquire and validate data from user input, and interpret user gestures that indicate the user wants to perform an operation on the data. Additionally, the user interface should filter the available actions to let users perform only the operations that are appropriate at a certain point in time.

6.2.4 Stabilizing

The purpose of the stabilizing phase is to reduce the risks of releasing the solution to production. A successful stabilizing phase requires that the team make the transition from a mindset focused on building features to one focused on getting the solution to a known state of quality. Deliverables of the deploying phase are operations and support information systems, repository of all versions of documentation and code, and project closeout reports.

7. Planning and Estimation

7.1 Planning

The MSF – Microsoft Solution Framework Process Model describes a generalized sequence of activities for building and deploying enterprise solutions. This process is flexible and can accommodate the design and development of a broad range of enterprise projects. The MSF Process Model is a phase-based, milestone-driven, and iterative model that can be applied to developing and deploying traditional applications, enterprise solutions for e-commerce, and Web-distributed applications.

MSF guidance includes disciplines for managing the people, processes, and technology elements that most projects encounter. The three key MSF disciplines are risk management, readiness management, and project management.

7.1 Application of the MSF model in our project:

7.1.1 Envisioning process:

Each phase in the MSF Process Model has interim milestones and a major milestone. Interim milestones are associated with the various activities that are performed in a phase, such as creating a team and creating a vision/scope document. The major milestone indicates that the team can progress to the next phase in the MSF Process Model. For example, the major milestone of the envisioning phase is the vision/scope approved milestone. When the team reaches this milestone, the team can progress to the planning phase of the MSF Process Model. The team creates deliverables for each task in the envisioning phase. Together, these deliverables provide context and direction for the team for the remainder of the project, and communicate the project vision and scope to the customer. The deliverables that the team creates during the envisioning phase include:

- Vision/scope - Problem statements and business objectives, A review of the existing processes, A broad definition of user requirements.

- Project structure - A project structure and process standards for the team to follow
- Risk assessment - A preliminary risk assessment, Plans for mitigating or eliminating the identified risks

7.1.2 Planning process:

During the planning phase, the team determines what to develop and plans how to create the solution. The team prepares the functional specification, creates a design of the solution, and prepares work plans, cost estimates, and schedules for the various deliverables.

The planning phase involves the analysis of requirements. These requirements can be categorized as business requirements, user requirements, operational requirements, and system requirements. These requirements are used to design the solution and its features and to validate the correctness of the design.

After gathering and analyzing the requirements, the team creates the design of the solution. The team creates user profiles that specify the various users of the solution and their roles and responsibilities. The team then creates a series of usage scenarios. A usage scenario specifies the activity performed by a particular type of user. Therefore, the team needs to create usage scenarios for all user profiles. After creating usage scenarios, the team creates use cases for the usage scenarios. A use case specifies the sequence of steps that a user will perform in a usage scenario.

The planning phase deliverables provide the basis for making future tradeoff decisions. The following deliverables are produced during the planning phase:

- Functional specification
- Risk management plan
- Master project plan and master project schedule

7.1.3 Design Process:

The three design stages are:

- Conceptual design, in which we view the problem from the perspective of the users and business requirements and define the problem and solution in terms of usage scenarios.
- Logical design, in which we view the solution from the perspective of the project team and define the solution as a set of services.
- Physical design, in which you view the solution from the perspective of the developers and define the technologies, component interfaces, and services of the solution.

We document the solution design in the functional specification. The functional specification describes the behavior and appearance of each feature of the solution. It also describes the architecture and the design for all features.

7. 1.4 Development process:

During the developing phase, the project team creates the solution. This process includes creating the code that implements the solution and documenting the code. In addition to developing code, the team also develops the infrastructure for the solution.

The deliverables of the developing phase include:

- Source code and executable files
- Installation scripts and configuration settings for deployment
- Finalized functional specification
- Performance support elements
- Test specifications and test cases

7. 1.5 Stabilization process:

During the stabilizing phase, the team performs integration, load, and beta testing on the solution. In addition, the team tests the deployment scenarios for

the solution. The team focuses on identifying, prioritizing, and resolving issues so that the solution can be prepared for release. During this phase, the solution progresses from the state of all features being complete as defined in the functional specification for this version to the state of meeting the defined quality levels. In addition, the solution is ready for deployment to the business

The deliverables of the stabilizing phase are as follows:

- Final release
- Release notes
- Performance support elements
- Test results and testing tools
- Source code and executable files
- Project documents
- Milestone review

7. 1.6 Deployment process:

During this phase, the team deploys the solution technology and site components, stabilizes the deployment, transfers the project to operations and support, and obtains final customer approval of the project. After deployment, the team conducts a project review and a customer satisfaction survey. The deploying phase culminates in the deployment complete milestone.

The deliverables of the deploying phase are as follows:

- Operation and support information systems
 - Procedures and processes
 - Knowledge base, reports, and logbooks
- Documentation repository for all versions of documents and code developed during the project
- A training plan
- Project completion report

- Final versions of all project documents
- Customer satisfaction data
- Definition of next steps

7.2 Estimation

7.2. 1 Basic COCOMO

The COConstructive COnstructive COSt MOdel (COCOMO) is an algorithmic Software Cost Estimation Model developed by Barry Boehm. The model uses a basic regression formula, with parameters that are derived from historical project data and current project characteristics. Constructive Cost Model: It is a hierarchy of estimation models that address: Application composition model: Used during the early stage of software engineering, when prototyping of user interfaces, consideration of software and system interaction, assessment of performance, and evaluation of technology maturity are paramount..

COCOMO consists of a hierarchy of three increasingly detailed and accurate forms. The first level, Basic COCOMO is good for quick, early, rough order of magnitude estimates of software costs, but its accuracy is limited due to its lack of factors to account for difference in project attributes

The basic COCOMO equations take the form

$$\text{Effort Applied} = a_b(\text{KLOC})b^b \text{ [man-months]}$$

$$\text{Development Time} = c_b(\text{Effort Applied})d^b \text{ [months]}$$

$$\text{People required} = \text{Effort Applied} / \text{Development Time} \text{ [count]}$$

The coefficients a_b , b_b , c_b and d_b are given in the following table.

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38

Semi Detached	–	3.0	1.12	2.5	0.35
Embedded		3.6	1.2	2.5	0.32

$$E = 3.0(KLOC)^{1.12}$$

$$= 3.0(5000)^{1.12}$$

$$= 18 \text{ person months}$$

$$D = 2.5E^{0.35}$$

$$= 2.5(18)^{0.35}$$

$$= 6.8 \text{ months}$$

Therefore No. of persons required = $18/6.8 = 3$ persons.

7.2.2 Function Point Estimation

A function point is a unit of measurement to express the amount of business functionality an information system provides to a user. Function points are the units of measure used by the IFPUG Functional Size Measurement Method. The IFPUG FSM Method is an ISO recognised software metric to size an information system based on the functionality that is perceived by the user of the information system, independent of the technology used to implement the information system.

The method of measuring the size of an information system and expressing it in a number of function points is called function point analysis (FPA). The method is kept up to date by worldwide cooperating FPA user groups like NESMA and IFPUG. A function point analysis expresses the functional size of an information system in a number of function points (for example: the size of a system is 314 FPs). There are many uses and benefits of function points and the functional size may be used as input into many types of project and organization decisions including determining the:

- Budget for application development or enhancement costs.
- Budget for the annual maintenance costs of the application portfolio.
- Project productivity after completion of the project.
- Software Size for cost estimating.

Function-Oriented Metrics

$$FP = \text{count_total} * [0.65 + 0.01 * \text{sum of } F_i]$$

1. Does the system require reliable backup and recovery=5
 2. Are data communications required=4
 3. Are there distributed processing functions=2
 4. Is performance critical=5
 5. Will the system run an existing, heavily utilized operational environment=5
 6. Does system requires online data entry=5
 7. Does online data entry req. input transaction to be build on multiple screens=3
 8. Are master files updated online =4
 9. Are I/ps , O/ps, files and inquires complex=3
 - 10.Is essential processing complex=5
 11. Is code reusable=4
 - 12.Are conversion and installation included in design=2
 - 13.Is system supports multiple installations =2
 14. Is application designed to facilitate change and ease of use by user=5
- sum of $F_i = 54$**

Measurement Parameter	Count		Simple	Average	Complex		
Number of	12	*	3	4	6	=	36

User Inputs							
Number of User outputs	4	*	4	5	7	=	16
Number of User inquires	1	*	3	4	6	=	6
Number of files	5	*	7	10	15	=	35
Number of External interface	3	*	5	7	10	=	15
Count Total							108

$$FP = \text{Count_total} * [0.65 + 0.01 * \text{sum of } Fi]$$

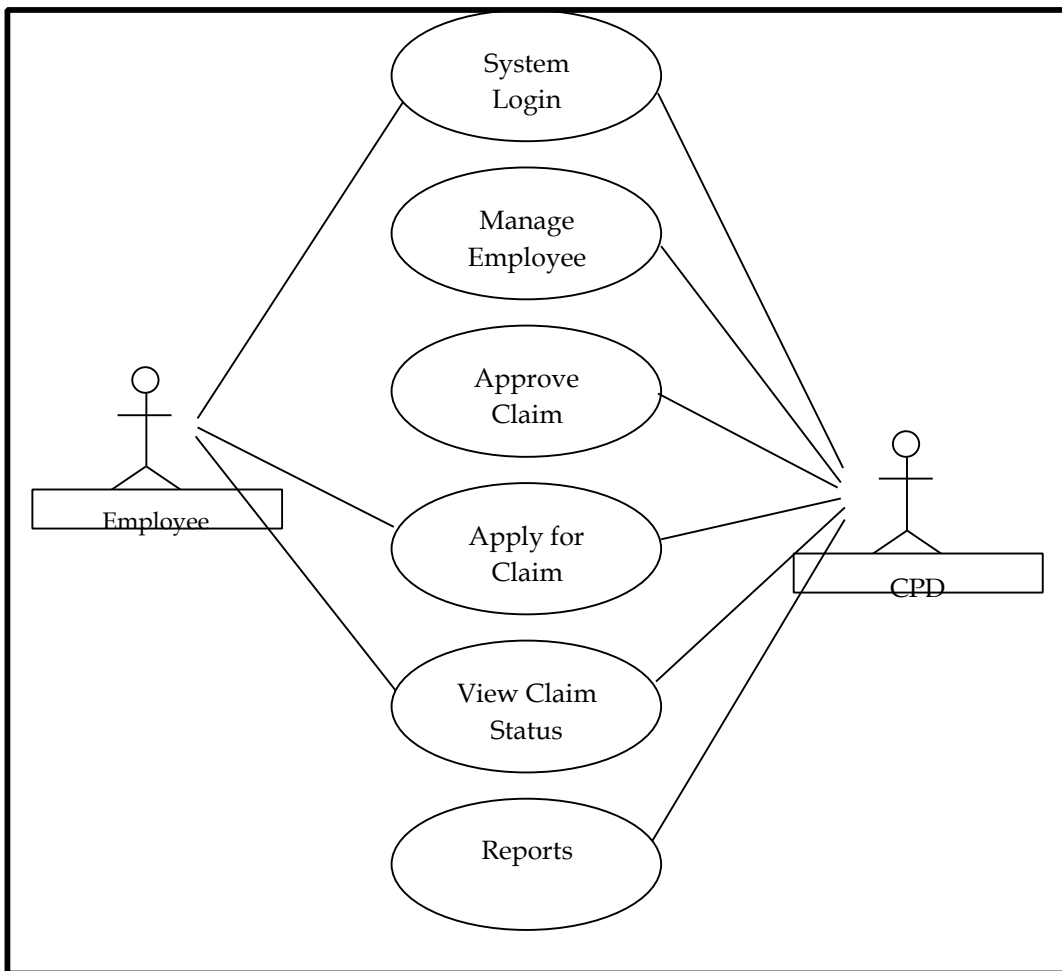
$$FP = 108 * [0.65 + 0.01 * 54]$$

$$FP = 129$$

8. Preliminary Design

8.1 Use Case

Web Based Claims Processing System (WCPS) 1.0 will address the following use cases. The complete usage scenarios will be completed during the information-gathering process. Use cases will be created and prioritized. Selected use cases will be expanded into usage scenarios and features that are derived from both use cases and the usage scenarios, as represented in the following diagram:



[Web Based Claims Processing System (WCPS) Usage Scenario – This usage scenario, or scenario for short, describes a real-world example of how one or more people or organizations interact with Web Based Claims Processing System (WCPS). It describes the steps, events, and/or actions which occur during the interaction. This Usage scenarios indicating exactly how someone works with the user interface, or reasonably high level describing the critical business actions but not the indicating how they’re performed.]

8.2 Specification of actors

The following actors are defined so far in the analysis phase of the Web Based Claims Processing System (WCPS) development process.

8.2.1 Employee

Employee	
Element	Details
Description	An employee is a user of the Web Based Claims Processing System (WCPS) system.
Examples	An Employee apply for new claim and view the status of his/her own claim.

8.2.2 CPD - Claims Processing Department

CPD - Claims Processing Department	
Element	Details
Description	The CPD is the person who can approve / reject claim applied by the employee.
Examples	CPD updates the claim status and his/her remark for a claim.

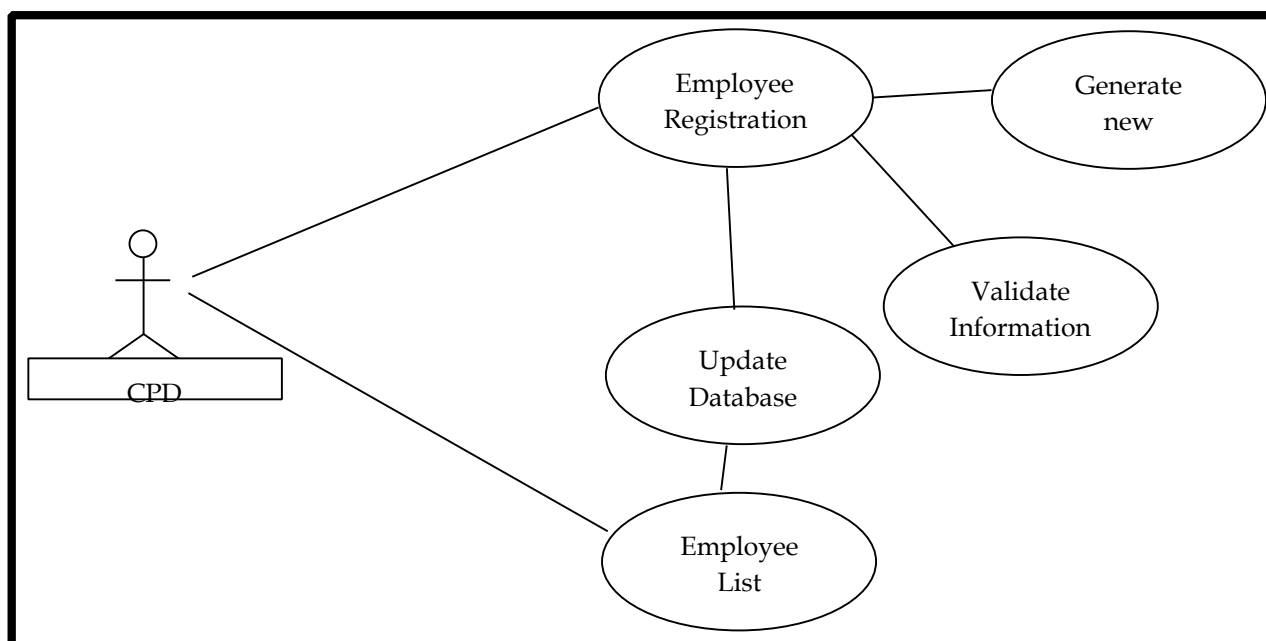
8.3 Specification of Use Cases

8.3.1 Use Case Manage Employee

Manage Employee	
Element	Details
Actor	CPD

Trigger	Employee is not registered in the System and the CPD wish to Add new employee into the WCPS.
Pre Conditions	Employee no is not listed in the system. CPD opens the new employee page, and the new employee page is displayed.
Post Conditions	Employee is registered in the system, new employee no is generated.
Normal course	<ol style="list-style-type: none"> 1. New Employee page appears on the screen 2. System increment the last registered Employee No by 1 to get the No for the new employee. 3. CPD fill in the Employee's information including Name, Department, Designation, Date of Birth, Email ID, Adress(es), Phone number(s), 4. System update
Alternative courses	3a. Not all mandatory data fields are filled 3a1.Employee fills in the missing data fields

Use Case Diagram : Manage Employee



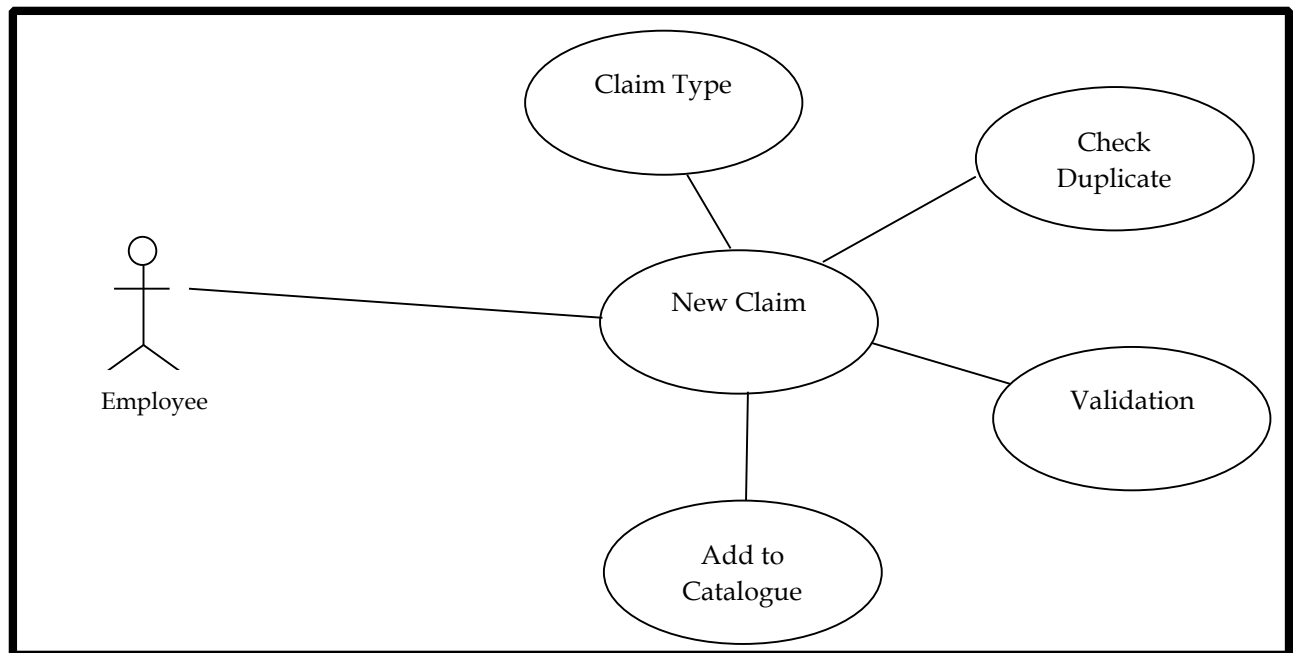
[Use Case: Manage Employee – This use case scenario, or scenario for short, describes how CPD will add new employee into Web Based Claims Processing System (WCPS). It describes the steps, events, and/or actions which occur

during the interaction. This Usage scenarios indicating exactly how someone works with the employee registration interface.]

8.3.2 Use Case Apply Claim

Mange Question Banks	
Element	Details
Actor	Employee
Trigger	Apply for new claim
Pre Conditions	Employee wish to apply for new claim and having login rights to WCPS.
Post Conditions	New claim is saved, and details are listed in pending claim details.
Normal Event Flow	<ol style="list-style-type: none"> 1. The employee open the new claim page 2. Employee select claim type. Enter amount, and remark. 3. Employee update the System by confirming the data entered into the new claim form. 4. System saved the claim and display claim details in pending lists.
Variations	<p>3a. Mandatory fields in the claim form are missing</p> <p>3a1. The system reject the entry and system update with an error message about missing mandatory fields.</p>

Use Case Diagram: Apply Claim



[Use Case: Apply Claim – This use case scenario, or scenario for short, describes how employee will apply new claim into Web Based Claims Processing System (WCPS). It

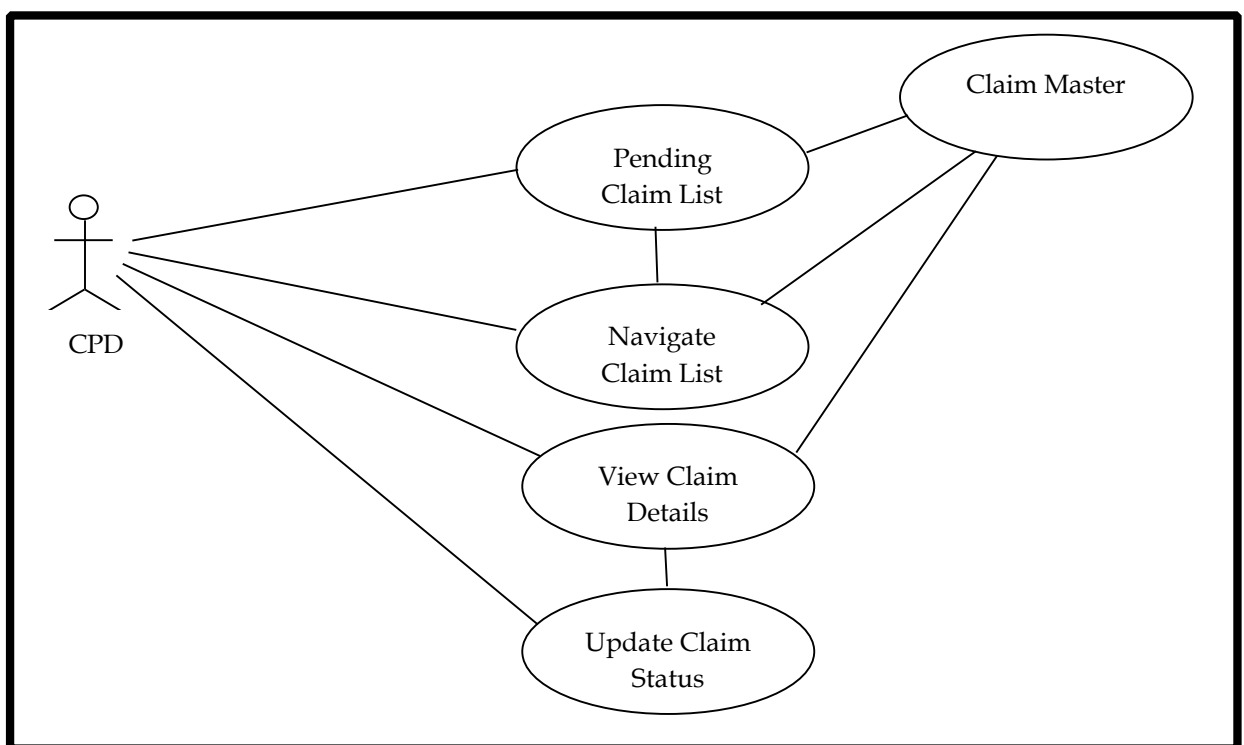
describes the steps, events, and/or actions which occur during the interaction.]

8.3.3 Use Case Approve Claim

Approve Claim	
Element	Details
Actor	CPD
Trigger	CPD staff wants to approve claim
Pre Conditions	The CPD staff must register into the system, the CPD is logged into the system, and the system display pending list for claim need to approve.
Post Conditions	Claim status is updated and Employee can view the status of his / her claim.

Normal event flow	<ol style="list-style-type: none"> 1. CPD logged in. 2. System displays all pending claim details. 3. CPD selects the claim, view the details 4. CPD updated the claim status, like approved / reject, enters remarks. 5. CPD logged out from the system.
Variations	NA

Use Case Diagram: Approve Claim

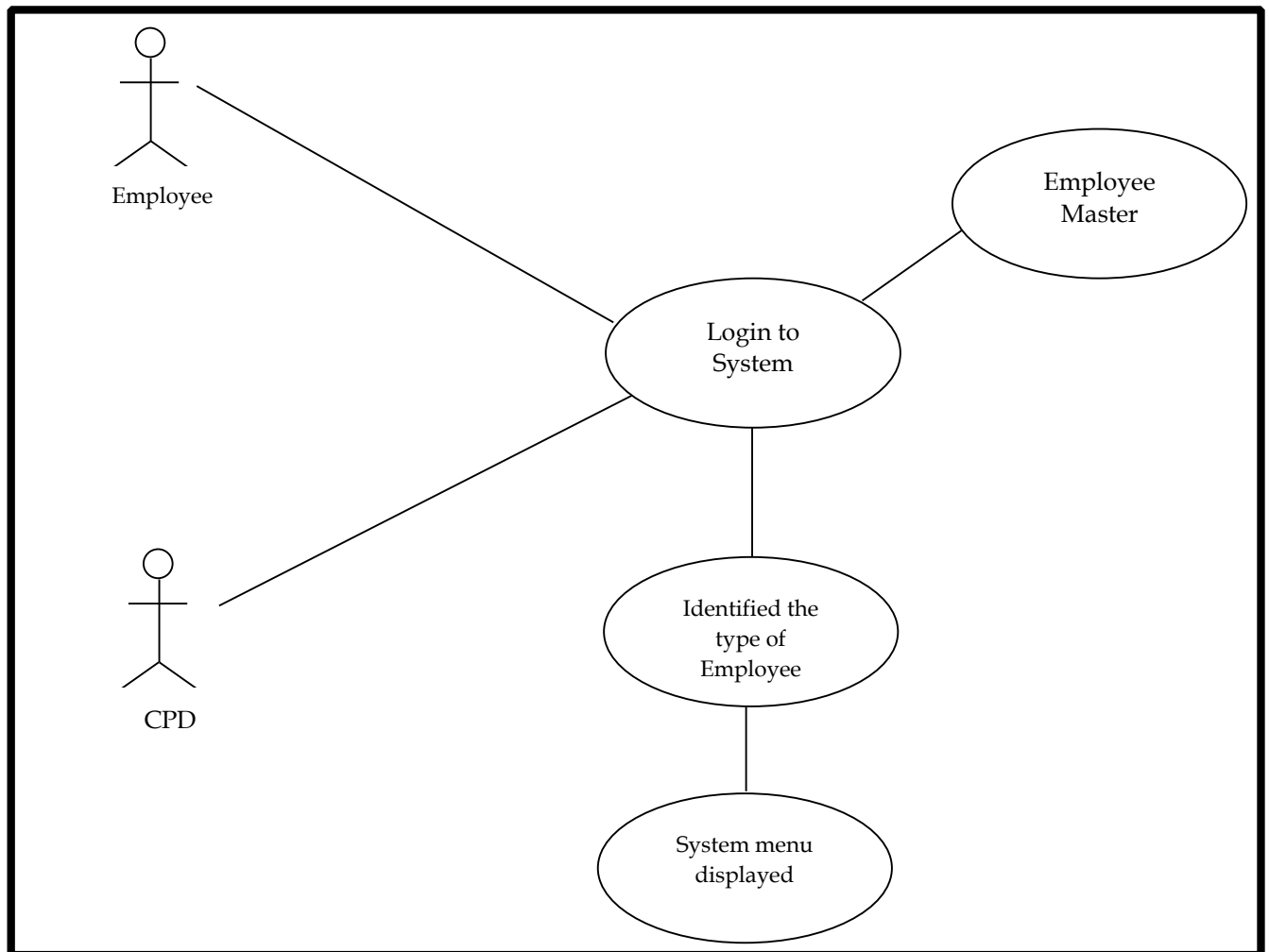


[Use Case: Approve Claim – This use case scenario, or scenario for short, describes how CPD staff approved claim in Web Based Claims Processing System (WCPS). It describes the steps, events, and/or actions which occur during the interaction.]

8.3.4 Use Case. System Login

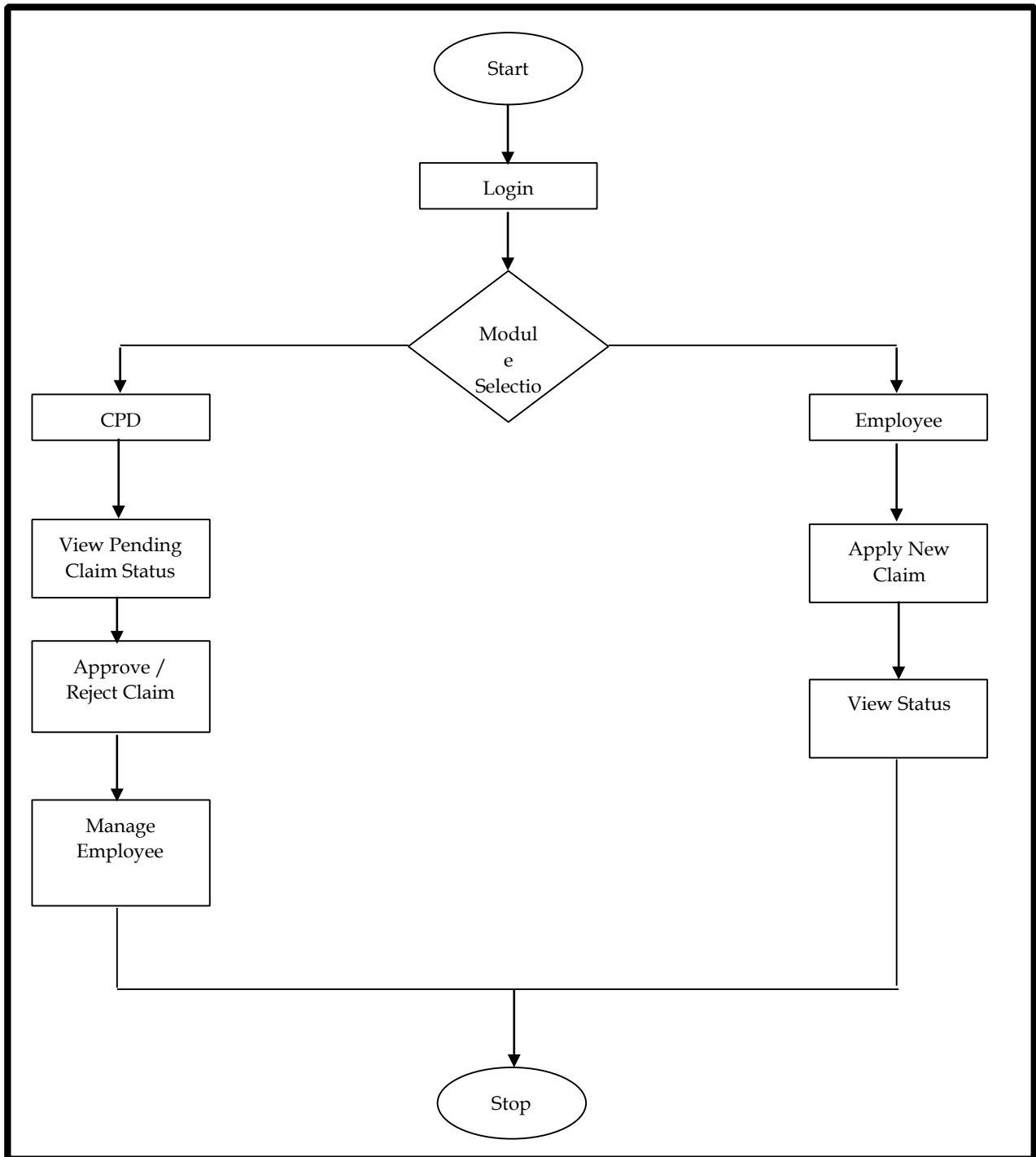
System Login	
Element	Details
Actor	Employee, CPD
Trigger	The user wish to start using the system.
Pre Conditions	The user is not logged into the system.
Post Conditions	The user is logged into the system, and the system menu is displayed.
Normal course	<ol style="list-style-type: none">1. The user click the link for theWCPS and a login page appear on the screen.2. The user types his username and password into the form and press the login button.3. The system confirms that the user is logged on.
Alternative courses	<p>2a. The user is not a valid user or the user name or the password is mistyped.</p> <p>2a1. The system reject login with an error message that express wrong Login name or password.</p>

Use Case Diagram: System Login

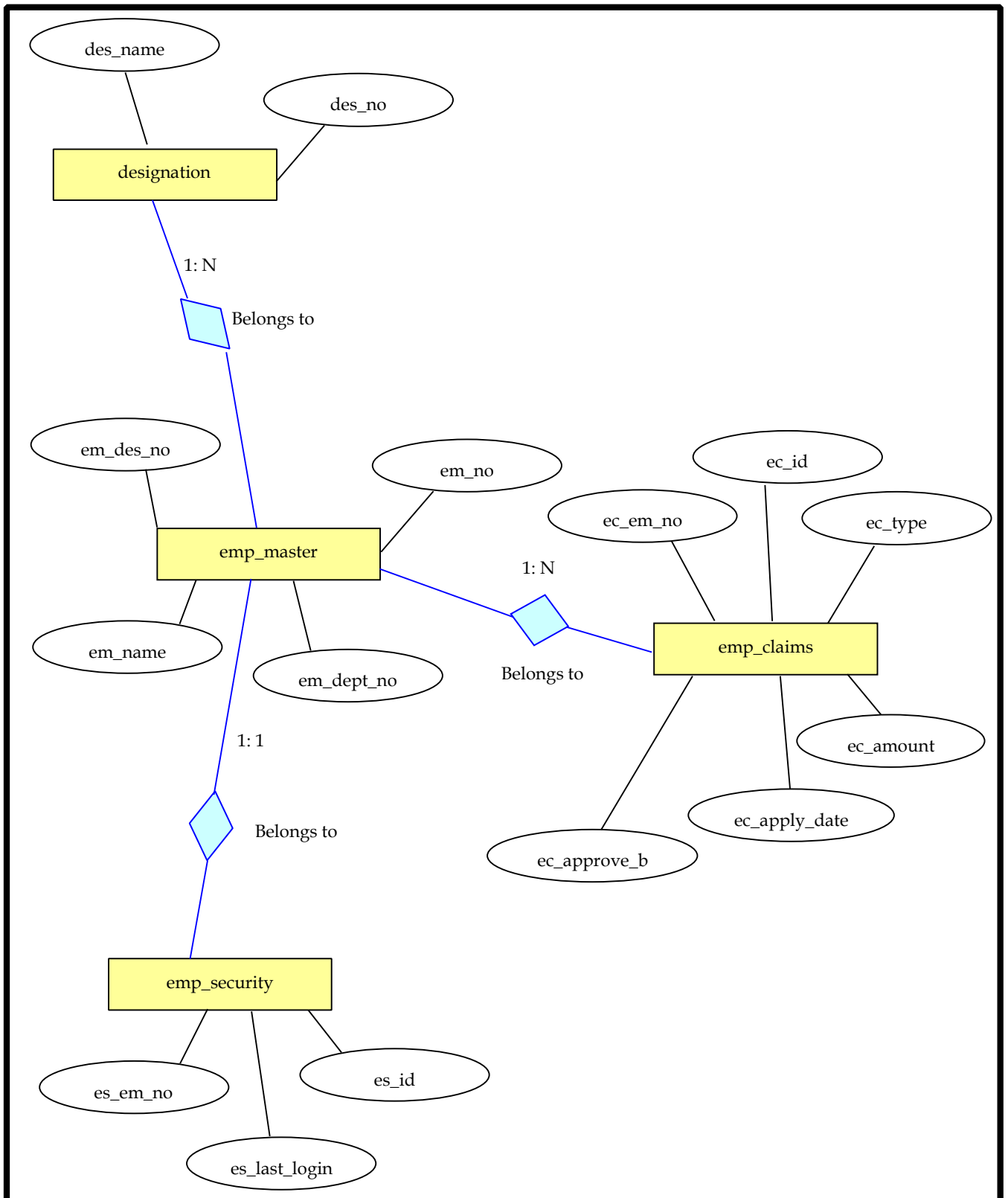


[Use Case: System Login – This use case scenario, or scenario for short, describes how actors will perform login Web Based Claims Processing System (WCPS). It describes the steps, events, and/or actions which occur during the interaction.]

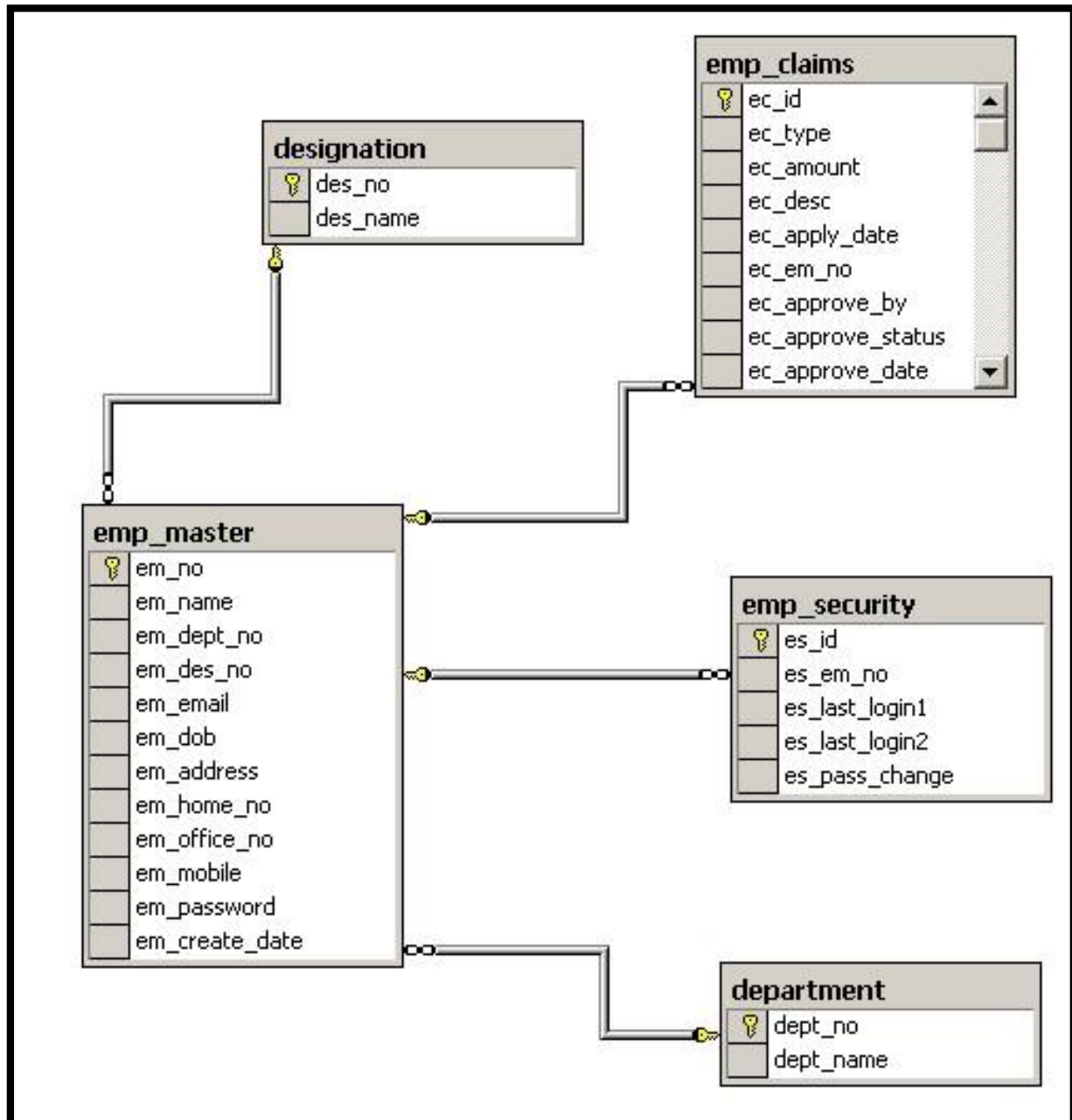
.4 Process Flow Chart



.5 ER Diagram 1



.6 ER Diagram 2



.7 Data Dictionary

8.7.1 Table: emp_master

Sr.	Field Name	Type	Primary Key	Foreign Key
1	em_no	int	Y	
2	em_name	varchar(50)		
3	em_dept_no	int		department(dept_no)
4	em_des_no	int		designation(des_no)
5	em_email	varchar(50)		
6	em_dob	datetime		
7	em_address	varchar(250)		
8	em_home_no	varchar(50)		
9	em_office_no	varchar(50)		
10	em_mobile	varchar(50)		
11	em_password	varchar(50)		
12	em_create_date	datetime		

[*emp_master* table is used to store employee details, *dept_no* and *des_no* is linked with department and designation master. *emp_no* is primary key and is identified as an employee in WCPS data dictionary]

8.7.2 Table: emp_claims

Sr.	Field Name	Type	Primary Key	Foreign Key
1	ec_id	int	Y	
2	ec_type	varchar(50)		
3	ec_amount	int		
4	ec_desc	varchar(150)		
5	ec_apply_date	datetime		
6	ec_em_no	int		emp_master(em_no)
7	ec_approve_by	int		
8	ec_approve_status	varchar(50)		
9	ec_approve_date	datetime		

10	ec_approve_remarks	varchar(150)		
----	--------------------	--------------	--	--

[*emp_claims* table is used to store claim details and claim status, *ec_em_no* is refers to employee master table]

.7.3. Table: emp_security

Sr.	Field Name	Type	Primary Key	Foreign Key
1	es_id	int	Y	
2	es_em_no	int		emp_master(em_no)
3	es_last_login1	datetime		
4	es_last_login2	datetime		
5	es_pass_change	datetime		

[*emp_security* table is used to login information of the employee, *es_em_no* is refers to employee master table]

8.7.4 Table: department

Sr.	Field Name	Type	Primary Key	Foreign Key
1	dept_no	Int	Y	
2	dept_name	Varchar(50)		

[Table *department* is used to store department details]

8.7.5. Table: designation

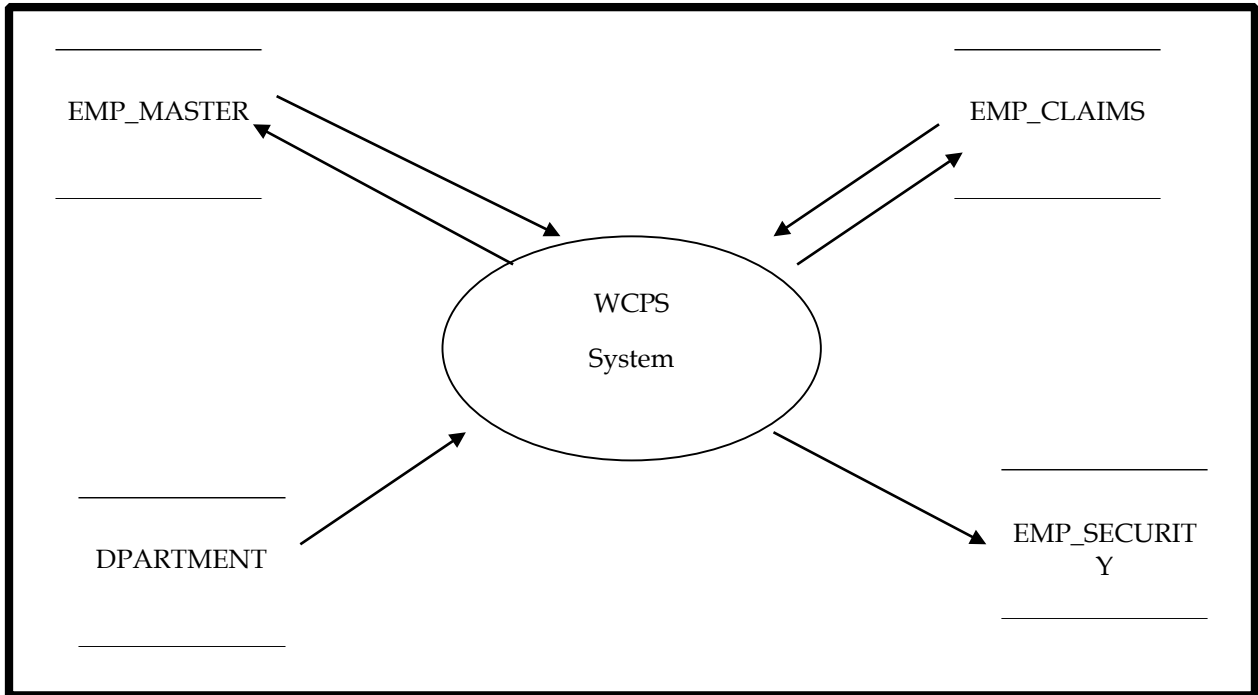
Sr.	Field Name	Type	Primary Key	Foreign Key
1	des_no	Int	Y	
2	des_name	Varchar(50)		

[Table *designation* is used to store designation details.]

.8 Data Flow Diagram DFD

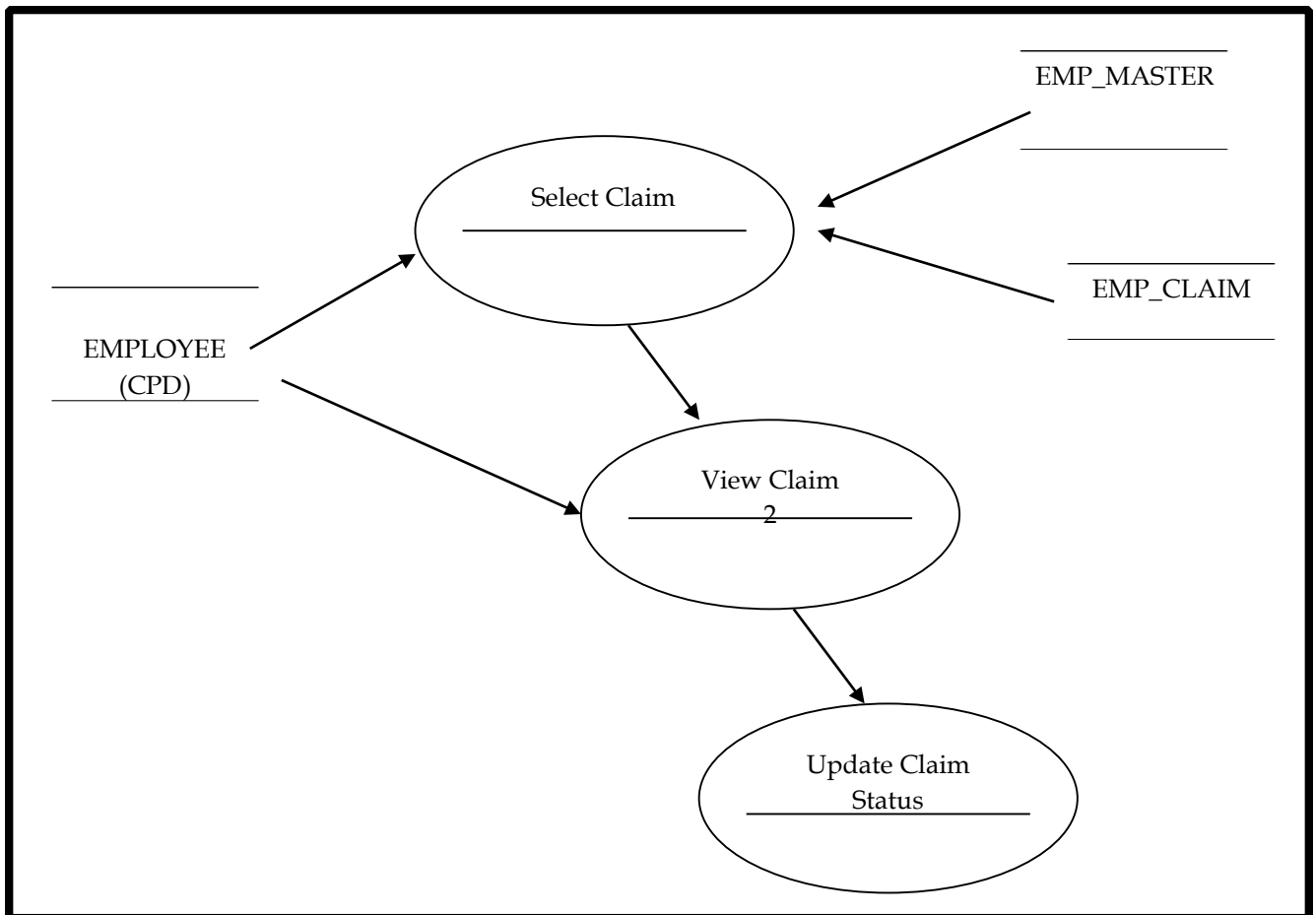
A data-flow diagram (DFD) is a graphical representation of the "flow" of data through an information system. DFDs can also be used for the visualization of data processing (structured design). On a DFD, data items flow from an external data source or an internal data store to an internal data store or an external data sink, via an internal process.

8.8.1 Context Level



This context-level data flow diagram first, which shows the interaction between the system and external agents which act as data sources and data sinks. On the context diagram (also known as the Level 0 DFD) the system's interactions with the outside world are modelled purely in terms of data flows across the system boundary. This context diagram shows the entire Web Based Claims Processing System (WCPS) as a single process,

.8.2 Level 1 (High Level Diagram)



This level (level 1) shows all processes at the first level of numbering, data stores, external entities and the data flows between them. The purpose of this level is to show the major highlevel processes of the Web Based Claims Processing System (WCPS) and their interrelation. A level-1 diagram must be balanced with its parent context level diagram, i.e. there must be the same external entities and the same data flows, these can be broken down to more detail in the level 1, e.g. the "Select Claim" data flow could be spilt into "View Details" and "View Claim" and still be valid.

.9 Planning

Planning is very important in every aspect of development work. Software project plan indicated scope of the project, milestones and deliverables, project

estimates, resource allocation, risk management, scheduling techniques and quality control and standard. Software project plan can be viewed as the following:

8.10 User Interfaces

The use of Textbox and List box for accepting data. The input data are then stored in the database. These data can also be retrieved from the database and displayed in the form of tables.

8.11 Hardware Interfaces

Raw data inserted into this software are permanently stored in tables; hence processor having free space will be useful for faster storage and retrieval of data. Here we have used 3.6 GHz P4 processor, which will have an additional support for our software.

8.12 Software Interfaces

This software is operated in a WINDOWS XP environment. This will help for firewall security provisions if the software is used along with internet connection.

8.13 Cost of Implementation

This project can be implemented in the organization in one to two weeks. The cost of this project is derived from effort, hardware cost, travel expenses, training cost, telecommunication costs etc.

.14 Effort:

It includes the total number of manpower per months. As this project is completely computerized hence less number of manpower will be used to successfully run this project. At least 2-3 persons will be enough to maintain this project.

8.14.1. Hardware cost:

It includes 2 INTEL P4 Standalone Computers. Cost around 60,000.

8.14.2 Training Cost:

One Software personnel will be allotted for providing training to the manpower allotted

8.14.3 Project duration:

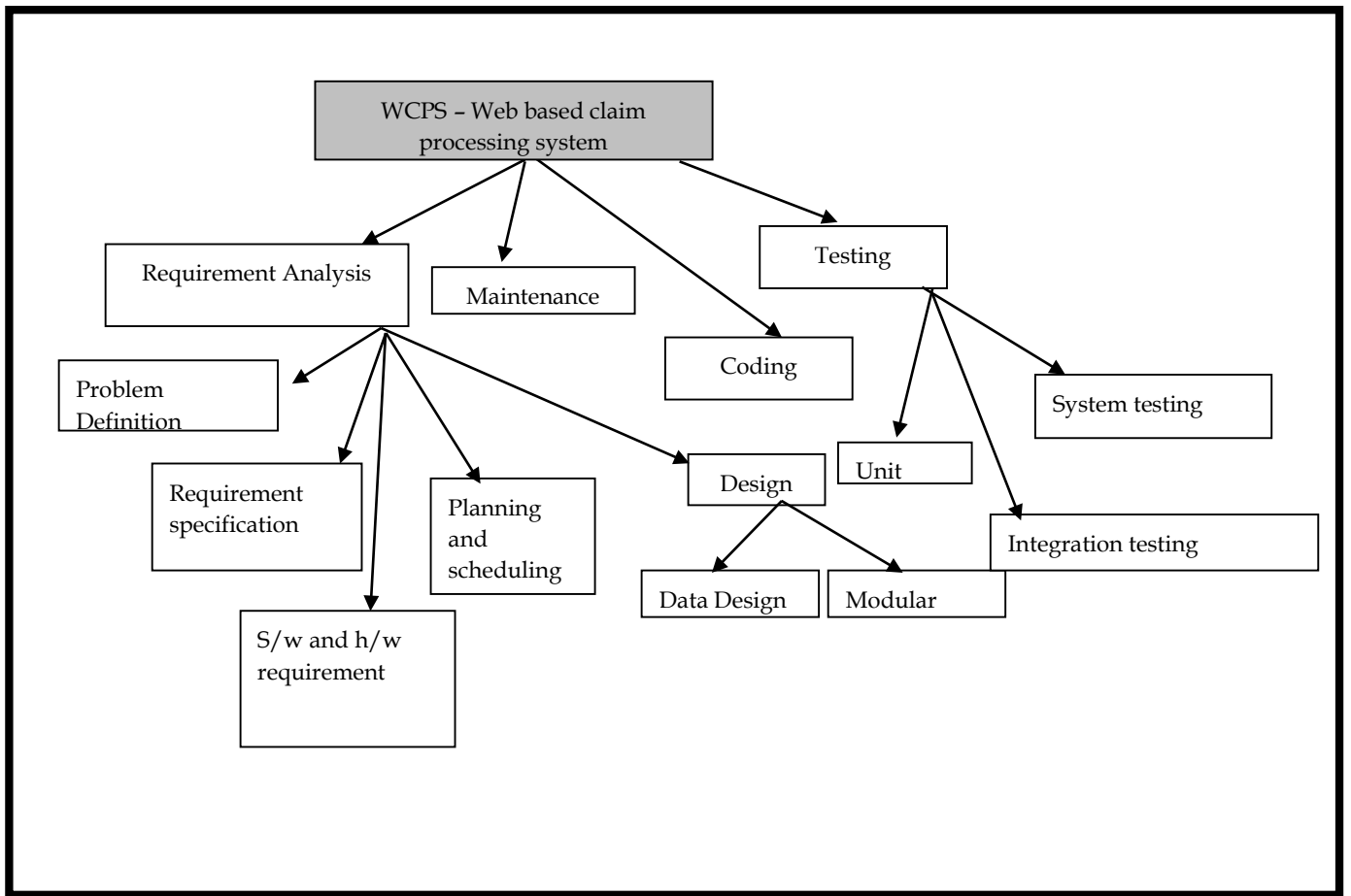
It will take complete 2 months for completion. After that it will take another 2-3 weeks for implementation and testing. Another 1-2 weeks is kept in hand for any inconvenience.

8.14.4 With respect to the customer:

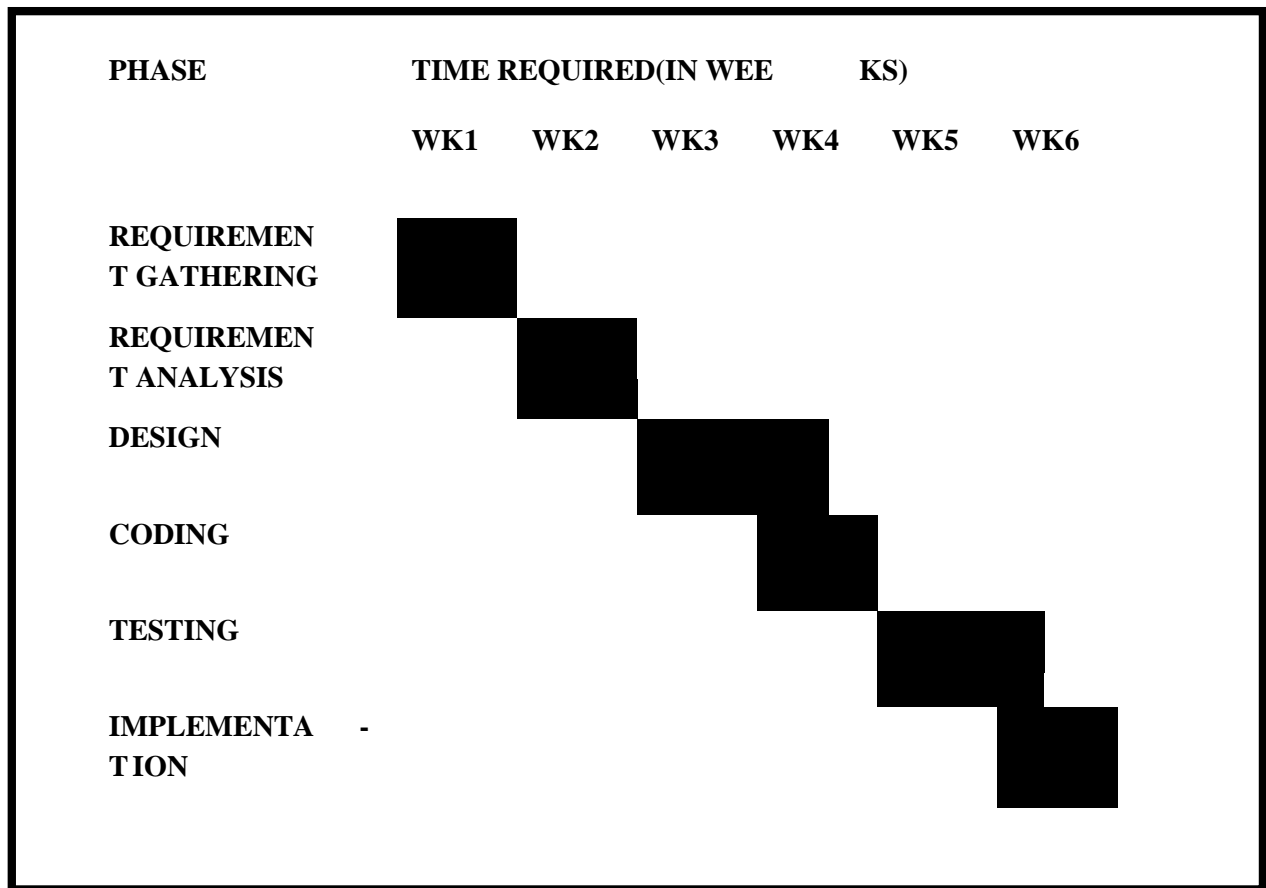
Weekly or timely meetings will be scheduled with the customers for getting time to time feedback. These meetings will be accompanied with presentation reports. After getting feedback further modifications and developments will be done.

.15 Scheduling

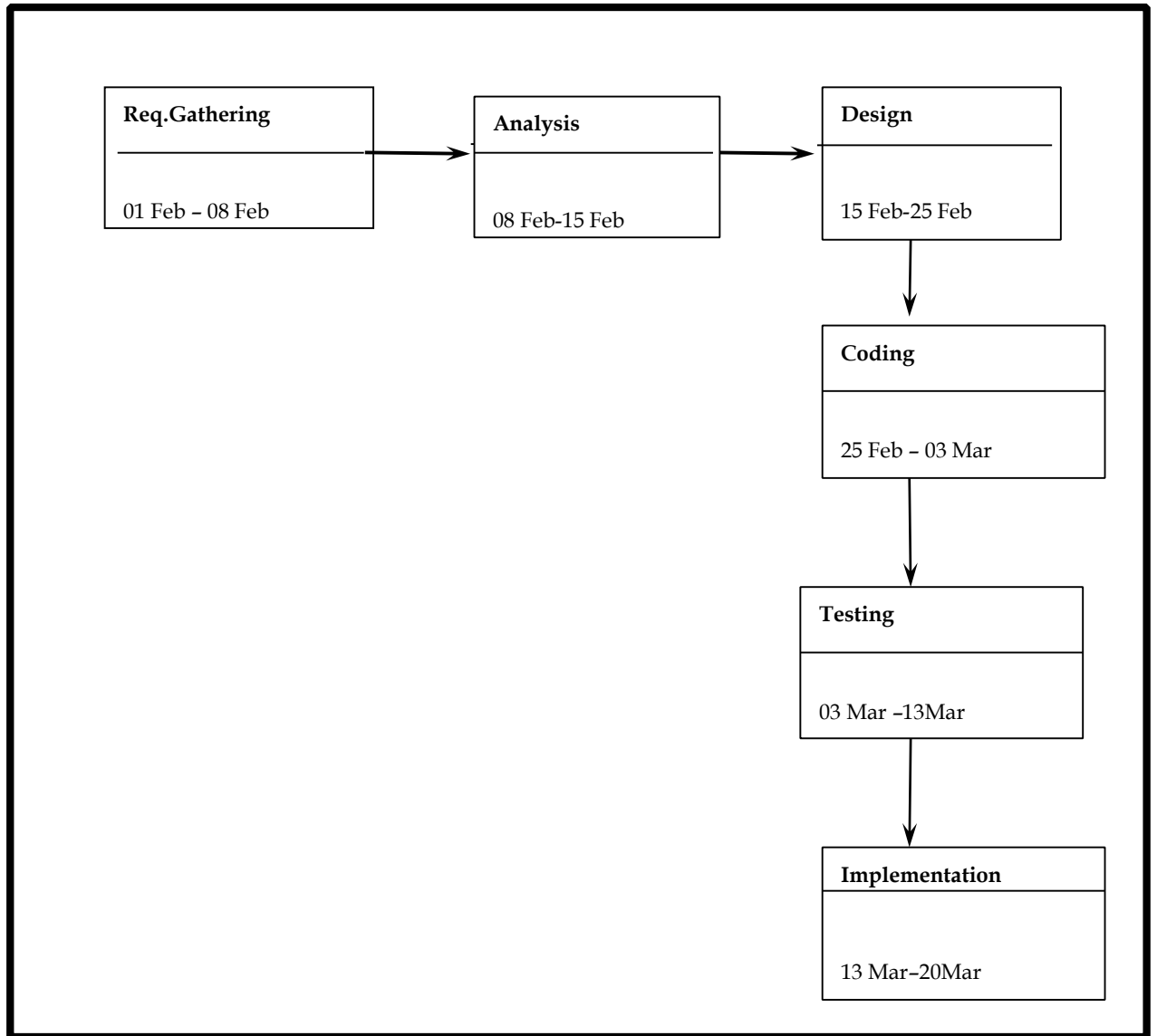
Scheduling of a project can be correlated to prioritizing various jobs with respect to their cost, time and duration. Scheduling can be done with resource constraint or time constraint in mind.



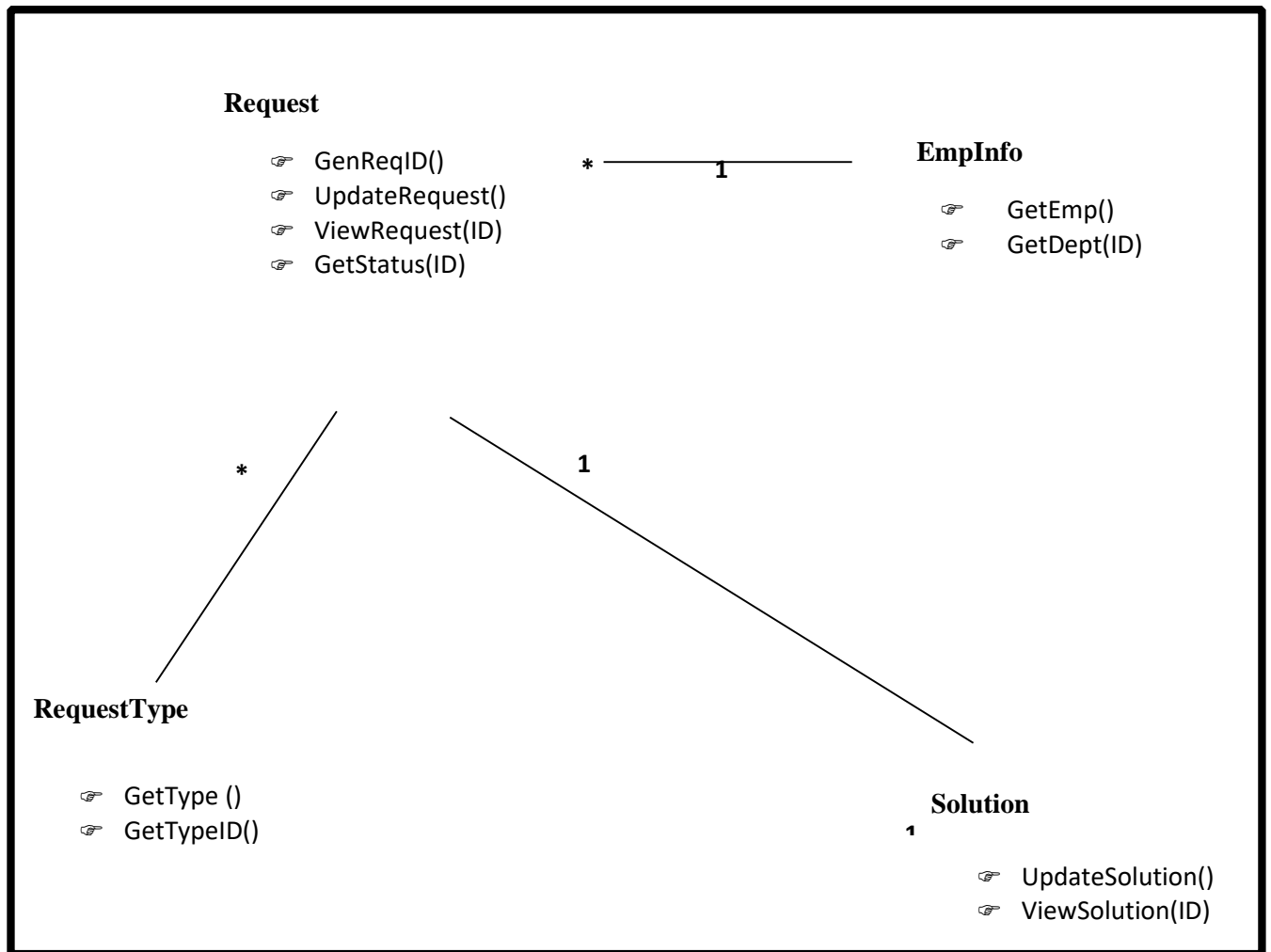
.16 Gantt chart:



.17 PERT chart:



.18 Class Diagram



```

graph TD
    Root[WCPS - Web based claim processing system] --> 1[1 Database & test data creation]
    Root --> 3[3 Login & Change Password]
    Root --> 4[4 Employee]
    Root --> 5[5 New claim]
    Root --> 6[6 View claim]
    Root --> 7[7 Approve claim]
    Root --> 8[8 Reject Request]
    Root --> 9[9 Reports]
    Root --> 10[10 Integration & Testing]

    1 --> 2.1[2.1 Database creation]
    1 --> 2.2[2.2 Test data creation]

    3 --> 3.1[3.1 User Interface Coding]
    3 --> 3.2[3.2 Handler Class Coding]
    3 --> 3.3[3.3 Testing]

    4 --> 4.1[4.1 User Interface Coding]
    4 --> 4.2[4.2 Handler Class Coding]
    4 --> 4.3[4.3 Testing]

    5 --> 5.1[5.1 User Interface Coding]
    5 --> 5.2[5.2 Handler Class Coding]
    5 --> 5.3[5.3 Testing]

    6 --> 6.1[6.1 User Interface Coding]
    6 --> 6.2[6.2 Handler Class Coding]
    6 --> 6.3[6.3 Testing]

    7 --> 7.1[7.1 User Interface Coding]
    7 --> 7.2[7.2 Handler Class Coding]
    7 --> 7.3[7.3 Testing]

    8 --> 8.1[8.1 User Interface Coding]
    8 --> 8.2[8.2 Handler Class Coding]
    8 --> 8.3[8.3 Testing]

    9 --> 9.1[9.1 Design]
    9 --> 9.2[9.2 Testing]

    10 --> 10.1[10.1 System Integration]
    10 --> 10.2[10.2 Integration Testing]
  
```

WCPS - Web based claim processing system

1 Database & test data creation

2.1 Database creation

2.2 Test data creation

3 Login & Change Password

3.1 User Interface Coding

3.2 Handler Class Coding

3.3 Testing

4 Employee

4.1 User Interface Coding

4.2 Handler Class Coding

4.3 Testing

5 New claim

5.1 User Interface Coding

5.2 Handler Class Coding

5.3 Testing

6 View claim

6.1 User Interface Coding

6.2 Handler Class Coding

6.3 Testing

7 Approve claim

7.1 User Interface Coding

7.2 Handler Class Coding

7.3 Testing

8 Reject Request

8.1 User Interface Coding

8.2 Handler Class Coding

8.3 Testing

9 Reports

9.1 Design

9.2 Testing

10 Integration & Testing

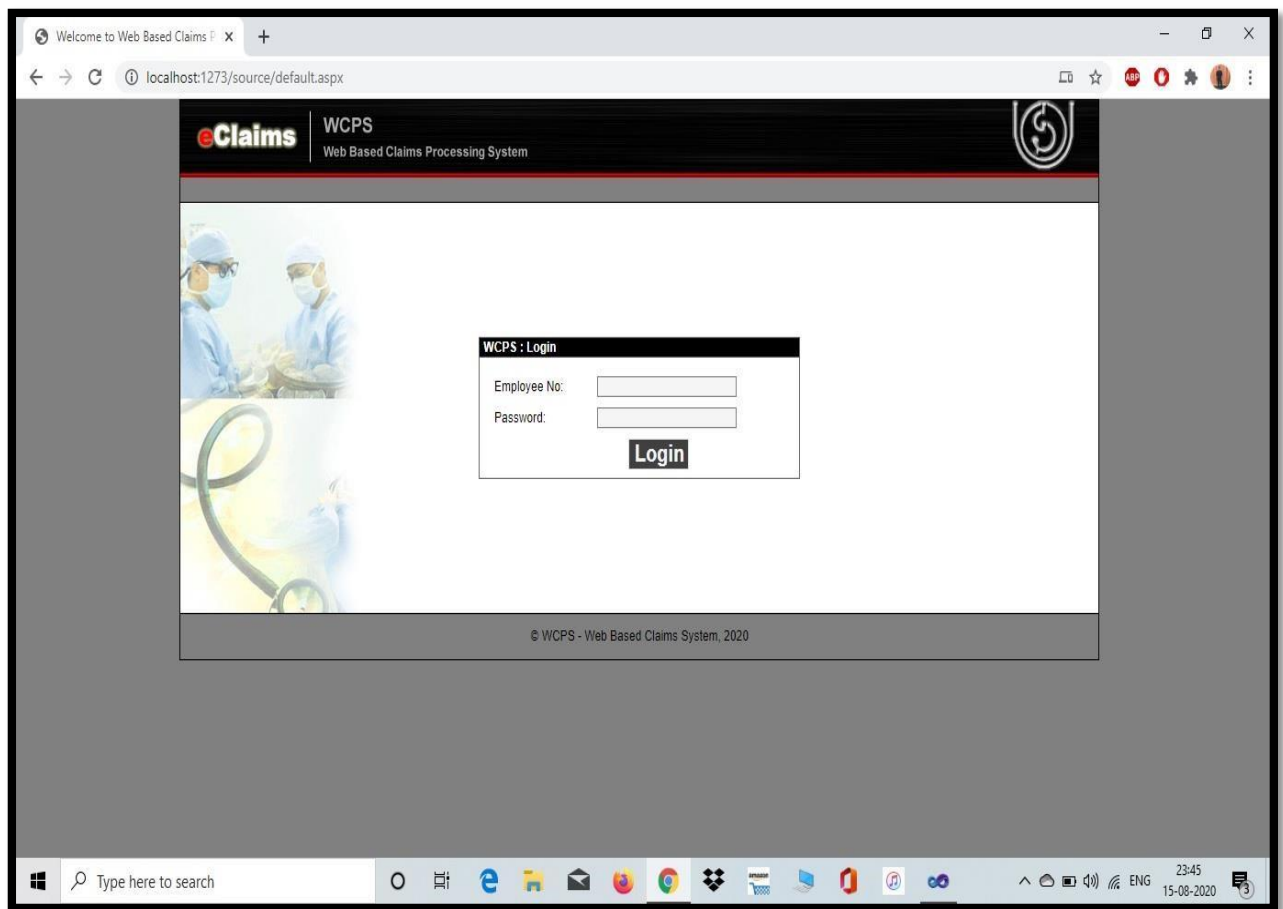
10.1 System Integration

10.2 Integration Testing

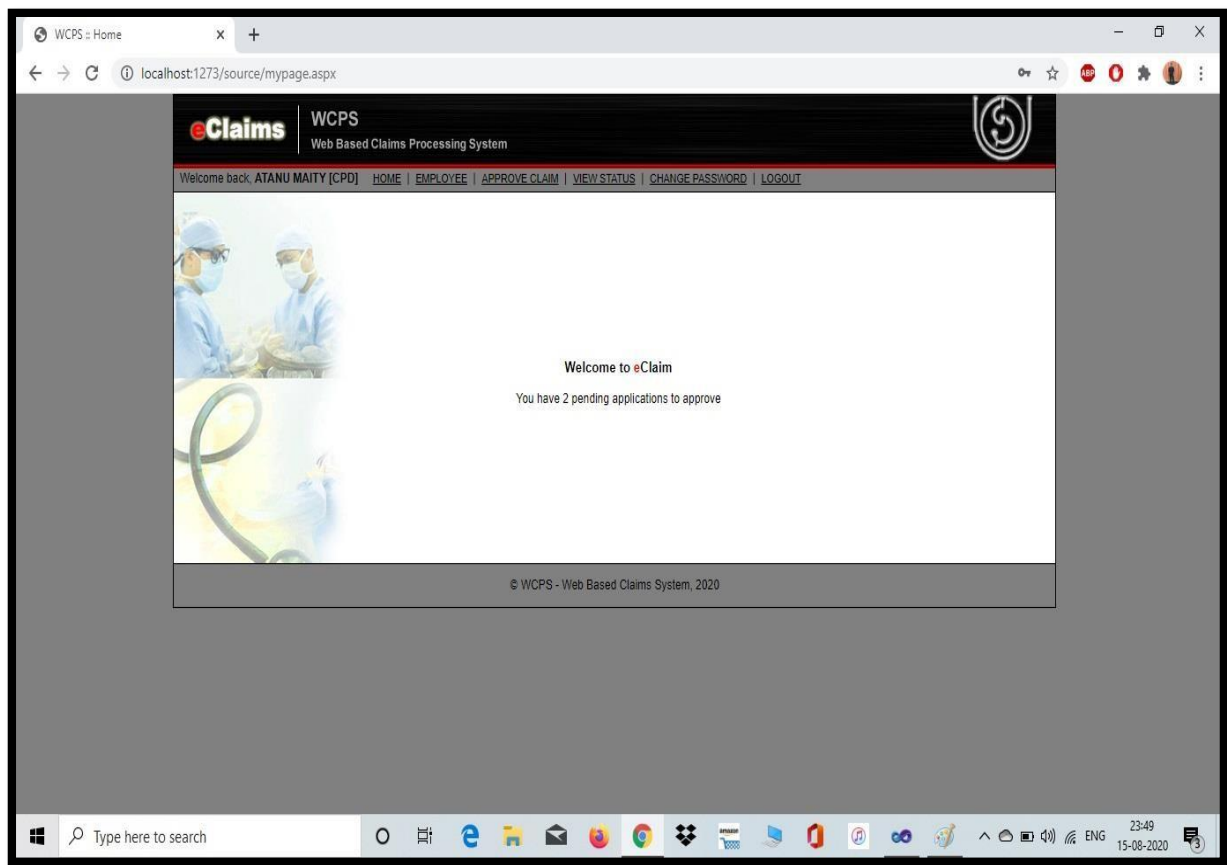
Work Breakdown Structure (WBS)

9.Web Pages

9.1 WCPS Home Page



9.2 WCPS – My Page



9.3 WCPS – Employee

WCPS : Employee

localhost:1273/source/employee.aspx

eClaims | WCPS
Web Based Claims Processing System

Welcome back, ATANU MAITY [CPD] | [HOME](#) | [EMPLOYEE](#) | [APPROVE CLAIM](#) | [VIEW STATUS](#) | [CHANGE PASSWORD](#) | [LOGOUT](#)

WCPS : Employees

3 Records Found.

Emp No	Emp Name	DOB	Department	Designation	Mobile	Email ID	Create Date
1001	ATANU MAITY	01-01-1975 00:00:00	CPD	MANAGER	9999898998	atanu.maity@rediffmail.com	Edit
1002	RAJ KUMAR	01-02-1960 00:00:00	IT	EXECUTIVE	9985623333	raj.kumar@rediffmail.com	Edit
1003	Sunil	01-02-1980 00:00:00	IT	EXECUTIVE	98992333	sunil@abc.com	Edit

[Add Emplo](#) [Back](#)

© WCPS - Web Based Claims System, 2020

Type here to search

11:42
16-08-2020

.4 WCPS Add / Edit Employee

The screenshot shows a web browser window displaying the 'WCPS : Employee Details' form. The browser's address bar shows the URL 'localhost:1273/source/employeeedetails.aspx?id=0'. The page header includes the 'eClaims' logo, the title 'WCPS Web Based Claims Processing System', and a navigation menu with links: 'Welcome back, ATANU MAITY [CPD]', 'HOME', 'EMPLOYEE', 'APPROVE CLAIM', 'VIEW STATUS', 'CHANGE PASSWORD', and 'LOGOUT'. The form itself is titled 'WCPS : Employee Details' and contains the following fields:

Employee No:	<input type="text"/>
Name	<input type="text" value="rohit chandrakant sulakhe"/>
Department:	<input type="text" value="IT"/>
Designation	<input type="text" value="MANAGER"/>
Email ID:	<input type="text" value="rohit.1524sulakhe@gmail.com"/>
Date of Birth:	<input type="text"/>
Address:	<input type="text" value="1504, chavandai residency, phase-2, parsik nagar, kalwa"/>
Tele (Resi):	<input type="text"/>
Tele (Off):	<input type="text"/>
Mobile:	<input type="text" value="09594284618"/>

At the bottom of the form are two buttons: 'Subm' and 'Back'. The browser's taskbar at the bottom shows the Windows search bar, several application icons, and the system clock indicating 23:55 on 15-08-2020.

WCPS : View Status

localhost:1273/source/approveclaim.aspx

eClaims | WCPS
Web Based Claims Processing System

Welcome back, ATANU MAITY [CPD] | HOME | EMPLOYEE | APPROVE CLAIM | VIEW STATUS | CHANGE PASSWORD | LOGOUT

WCPS : Approve / Reject Claim

2 Application pending for approve/reject.

Ref. No.	Emp. Name	Emp. No.	Type	Amount	Description	Apply Date	Approve/Reject
7	RAJ KUMAR	1002	Medicines	2500	I want to claim	17-03-2020 13:41:44	Edit
5	RAJ KUMAR	1002	Laboratory Tests	500	Test	29-07-2009 00:31:42	Edit

© WCPS - Web Based Claims System, 2020

Type here to search

23:58
15-08-2020

.5 WCPS Approve Claim – View Status
.6 WCPS Approve / Reject Claims

WCPS : Approve/Reject Claim

localhost:1273/source/approvereject.aspx?id=7

eClaims | WCPS
Web Based Claims Processing System

Welcome back: ATANU MAITY [CPD] | HOME | EMPLOYEE | APPROVE CLAIM | VIEW STATUS | CHANGE PASSWORD | LOGOUT

WCPS : Approve / Reject Claim

Employee: RAJ KUMAR [1002]
Apply Date: 17-03-2020 13:41:44
Claim Type: Medicines
Claim Amount: 2500
Details: I want to claim

Action: ☒ Approve ☐ Reject
Remarks:

Subm **Back**

© WCPS - Web Based Claims System, 2020

.7 WCPS View Status

WCPS : View Claim Status

Claim Status: **Subm**

7 Records found.

Ref. No.	Emp. Name	Emp. No.	Type	Amount	Description	Apply Date	Status	Approve/Reject Date
7	RAJ KUMAR	1002	Medicines	2500	I want to claim	17-03-2020 13:41:44	Pending	
6	RAJ KUMAR	1002	Surgery	20000	Leg Fracture	29-07-2009 23:11:24	Approved	29-07-2009 23:13:05
5	RAJ KUMAR	1002	Laboratory Tests	500	Test	29-07-2009 00:31:42	Pending	
4	RAJ KUMAR	1002	Laboratory Tests	4000	For Blood test	29-07-2009 00:28:53	Rejected	29-07-2009 00:30:00
3	RAJ KUMAR	1002	Surgery	5000	Operation Exp	28-07-2009 23:58:05	Approved	29-07-2009 00:26:24

1 2

© WCPS - Web Based Claims System, 2020

9

.8 WCPS Change Password

The screenshot shows a web browser window displaying the 'WCPS : Change Password' page. The browser's address bar shows the URL 'localhost:1273/source/changepassword.aspx'. The page header includes the 'eClaims' logo, the text 'WCPS Web Based Claims Processing System', and a navigation menu with links: 'Welcome back, ATANU MAITY [CPD]', 'HOME', 'EMPLOYEE', 'APPROVE CLAIM', 'VIEW STATUS', 'CHANGE PASSWORD', and 'LOGOUT'. The main content area features a 'WCPS : Change Password' form with three input fields labeled 'Old Password:', 'New Password:', and 'Confirm:'. Below these fields are two buttons: 'Subm' and 'Back'. To the left of the form is a decorative image of two medical professionals in blue scrubs and masks. The footer of the page contains the copyright notice '© WCPS - Web Based Claims System, 2020'. The Windows taskbar at the bottom shows the search bar, several application icons, and the system clock indicating '00:01' on '16-08-2020'.

WCPS : Change Password

WCPS
Web Based Claims Processing System

Welcome back, ATANU MAITY [CPD] | HOME | EMPLOYEE | APPROVE CLAIM | VIEW STATUS | CHANGE PASSWORD | LOGOUT

WCPS : Change Password

Old Password:

New Password:

Confirm:

Subm Back

© WCPS - Web Based Claims System, 2020

. Test Plan

Introduction

This document describes the user acceptance test plan for the Web Based Claims Processing System (WCPS). The complete test strategy for the Web Based Claims Processing System (WCPS) is to perform the following kinds of tests, in sequence:

1. **Component testing** of each component that makes up the Web Based Claims Processing System (WCPS)
2. **Integration testing** of the Web Based Claims Processing System (WCPS) to ensure the correct interworking of its components
3. **Validation testing** of the Web Based Claims Processing System (WCPS), to ensure that it works correctly in a pseudo-live environment
4. **User acceptance testing** of the Web Based Claims Processing System (WCPS), to ensure that its function is acceptable to its users

Acceptance testing is the last set of tests to be performed before the application goes officially live.

10.1 Test Scope

The scope of the user acceptance testing covers:

- Version 1 of the Web Based Claims Processing System (WCPS)
- User-facing functionality defined by a set of use cases
- Administrator-facing functionality defined by a set of use cases

The aim of the testing is to determine how well the application meets its functional requirements from the perspective of the user, and to identify any issues so they can be resolved. Also, the testing serves to compile a set of test data and results that can be used during subsequent test cycles, to test for non-regression of the software in later releases or after the application is in maintenance.

Working practices might vary from user to user and are considered outside the scope of the testing.

.2 Test Strategy

The basis of user acceptance testing is that other tests were completed successfully, so the application and its required infrastructure are considered to be stable and reliable. Acceptance testing concentrates on the application from the user's perspective, that is, how the application is used and whether it meets the necessary quality criteria.

Change requests will be sent to the development team as the actionable documentation. Change criteria will be determined by the Test team and the Development team prior to the beginning of testing. For instance, criteria may include *impact to desired functionality*, *amount of code impacted by proposed change*, and *design required by proposed change*. The tester will evaluate the criteria. The test lead will determine Change Required or not. Once a bug has been determined as Change Required, the bug report will be translated into a Change Request and passed on to development.

The member of the acceptance testing is the System Users, Supervisor, Manager and MIS Executive for Web Based Claims Processing System (WCPS). The progress of the acceptance testing will be reported to the customer, together with any issues that are discovered and their planned resolutions. Sign-off of the tests, and therefore the acceptance of the application, will be performed by the customer or a selected representative.

10.3 Preconditions

The following items are required before testing can take place:

- A complete and coherent functional specification of the Web Based Claims Processing System (WCPS) expressed as use cases and usage scenarios
- A complete and validation-tested release of Web Based Claims Processing System (WCPS), delivered according to the delivery plan
- An agreed-upon procedure for dealing with any anomalies that are discovered during the testing process
- A set of test specifications describing how each functional area of the Web Based Claims Processing System (WCPS) is to be acceptance tested
- An implemented test environment for the testing
- Sufficient, suitable resources to carry out the testing
- Available standards for the acceptance testing

.4 Test Priorities

During testing of the Web Based Claims Processing System (WCPS), the following qualities will be tested in order of priority:

- Functionality—whether the required functions are available and working as expected
- Usability—how user-friendly and intuitive the Web Based Claims Processing System (WCPS) is
- Security—how well-protected and guaranteed corporate and user data is
- Performance—whether the response times are within acceptable limits

- Customization—how straightforward it is to use the application in new, unpredicted ways

10.5 Test Techniques

The following techniques will be applied:

- Scripted tests—sequences of user interactions (based on the use case and usage scenarios) using predefined data sets against predicted results
- Unscripted tests—based on scripted tests, the tester tries to modify the scenarios to explore what-if possibilities
- Penetration tests—scripted tests to attempt unauthorized entry into the system
- Usability checklists—tests to determine the complexity of interactions
- Performance statistics—generation of performance information to check against desired performance criteria

10.6 Test Organization

Roles and Responsibilities

The following roles are defined:

- QA lead/test manager—responsible for planning and ensuring the smooth running of the test process
- Tester—carries out the tests according to the test plan, and then reports the results

- Product manager—ensures that the tests are carried out successfully from a user perspective
- Project sponsor/client—acts as main stakeholder, and ensures that the needs of the customer community as a whole are considered
- Test support—provides technical assistance, such as test environment configuration, and non-technical assistance, such as methodological support

Weekly team meetings will be held involving the test manager, testers, and product managers. At these meetings, the progress of the testing process will be reported, any issues will be discussed, and actions will be agreed upon.

10.7 Deliverables

The following deliverables will be expected from the user acceptance testing process:

- Test plan—this document, together with any updates that have occurred during the testing process
- Change requests—any bugs, defects, or other changes required to the Web Based Claims Processing System (WCPS) as a result of the testing process
- Weekly reports—progress reports to enable the status of the testing process to be determined
- Completion report—a report to be signed off by the customer, to signify the successful completion of the user acceptance testing

10.8 Test Environment

Hardware and Software

The test environment will consist of:

Server

A single Intel-based computer running:

- Microsoft Windows
- Web Based Claims Processing System (WCPS) components

Client Workstations

Two Intel-based client laptop computers, each running:

- Microsoft Windows XP Professional
- Microsoft Office
- Internet Explorer 6 or greater

The following additional hardware will be required:

- One laser printer to print reports
- One color printer (laser or inkjet) to print screen dumps
- One CD-ROM drive to enable clean installation of the Web Based Claims Processing System (WCPS)
- Networking connectivity to permit interconnection of the server, clients.

10.9 Testing Automation Software

No testing automation software packages are selected at present.

10.10 Application Configuration

The following user accounts will be configured on the server:

- System Administrator
- CPD 1
- CPD 2
- Employee 1
- Employee 2

10.11 Test Management

Tests shall be managed according to the corporate test management standards, which cover:

- Conduct of tests
- Reporting of test results
- Defect tracking and resolution
- Configuration management of the test environment □ Configuration control of test deliverables.

10.12 Testing Schedules

The user acceptance testing schedules are shown in the project structure document and resulting Gantt charts.

10.13 Threats to Testing

Potential threats to the testing process are as follows:

- Insufficient resources available for testing. Testing resources have been seconded from the development departments, whose time is at a premium. Mitigation: ensure department heads apply a high priority to the testing of the Web Based Claims Processing System (WCPS).
- Availability of sales personnel for testing. The test team should be overseen by at least one administrator.

11. Future Enhancement

This project was developed to fulfill user requirement; however there are lots of scope to improve the performance of the Web Based Claims Processing System (WCPS) in the area of user interface, database performance, and query processing time. Etc.

So there are many things for future enhancement of this project. The future enhancements that are possible in the project are as follows.

- Linking and integration of other online web sites.
- Integration with other legacy accounting database through Web Services
- Connection to third-party OLAP applications □ In the area of data security and system security.

- Provide more online tips and help.

To optimize the query which is embedded in the system

12. Bibliography

12.1 Websites

Following websites are referring to create this project reports.

- <http://www.google.com>
- <http://www.microsoft.com>
- <http://www.programmer2programmer.net>
- <http://www.codeproject.com>
- <http://www.asp.net>
- <http://www.asp123.com>
- <http://www.wikipedia.org>