



PROJECT DOCUMENTATION

Court Case Management System
For
Ceylinco Insurance PLC.

DON ARUNA LAHIRU MANCHANAYAKE
BCIS No: 305407

Acknowledgement

I would like to take this opportunity to express my grateful thanks and gratitude to the able lecturers of IDM for the help and guidance in the preparation of the Project Documentation.

In particular my project supervisor, Mrs. Avanthi Malalage who guided through the Project preparation deserves my special thanks. And I also like to thank my parents, who gave their fullest support to make this Project a success.

I also want to thank the Legal Department staff of Ceylinco Insurance PLC for the support given on analyzing the company's manual system.

Finally I want to thank all my friends who helped me in numerous ways to fulfil this Project.

Thank You.

Lahiru Manchanayake

Abstract

This Documentation is based on the Project carried out in order to fulfil the requirements of the Final Project of the B.Sc (Hons) In Computing & Information Systems. This project documentation is about the development project, Court Case Management System which was developed exclusively for Ceylinco Insurance PLC.

In this Documentation the methods of analysis, designing, coding & testing are documented. This project has used the Unified Software Development Process & Object Oriented techniques along with Unified Modeling Language. All the approaches taken, decisions made are documented.

This project is about the Court Case Management System which was developed to make the functional areas of Ceylinco Insurance more efficient & effective. One of the main intentions of this project was to provide a good service to the customers & to reduce the overheads in the divisions.

This project uses state of the art WAP communication technologies where the Lawyers and Legal Dept Staff can communicate via Intranet, WAP & Email. Mainly intranet has been used for the communication aspects.

The information system is developed in a database environment using MYSQL Server. The front-end is developed using PHP, WML and JavaScript. The system is web enable.

Table of Contents

	Pages
1. Introduction	
1.1. Company Profile	09
1.1.1. Main Functions of Ceylinco Insurance	10
1.1.2. Potential Benefits for using an Intranet Solution	11
1.2. Project Profile	12
2. Literature Survey	
2.1. The Role of a Court Case Management System	14
2.2. How Web Applications Work	14
2.3. What is WAP	15
2.4. What is WML	15
2.5. What is a WAP Gateway	15
3. System Modelling & Development Methodology	
3.1. System Modelling	
3.1.1. Importance of Modelling	17
3.1.2. Introduction to Object Oriented Modelling	18
3.1.3. Importance of OO Modelling in Large Scale Business Applications	19
3.1.4. Some Object Oriented Concepts in Brief	20
3.1.5. Software Modelling using Unified Modelling Language	21
3.1.6. Modelling the System Architecture	23
3.1.7. UML Diagrams in Brief	24
3.2. Development Methodology	
3.2.1. Importance of a Methodology	26
3.2.2. Unified Development Process	26
3.2.3. Why USDP was chosen for CCMS development	27
3.2.4. Unified Software Development Process Architecture	28
4. The Inception Phase	
4.1. Overview	30
4.2. Initial Project Review & Feasibility Study	
4.2.1. Current System Environment	31
4.2.2. Drawbacks of the Current System	31
4.2.3. Project Objectives	32
4.2.4. Advantages of CCMS	33
4.3. Discovery of Facts	34
4.4. Requirements Specification	35
4.5. Captured User Requirements	36
4.6. Project Scope	37
4.7. Sub Systems of CCMS	39
4.8. Proposed Project Schedule & Milestones	40
4.9. Project Prerequisites	41
4.10. Economical Feasibility	42

5. The Elaboration Phase

5.1. Overview	44
5.2. Business System Options & Technical System Options	45
5.3. Structural Modelling – Class Diagram	46
5.4. Behavioural Modelling	47
5.4.1. Use Case Diagram	49
5.4.2. Activity Diagram	50
5.4.3. Sequence Diagram	51
5.5. Development & Testing Strategy	52
5.6. Implementation Strategy	
5.7. Modelling the Database for CCMS	53
5.7.1. Database & Database Management System	53
5.7.2. Advantages of having a database	53
5.7.3. The Database management system for CCMS	53
5.7.4. Unique features of the MYSQL 5.0	53
5.7.5. The Database Design for CCMS	54
5.8. Modelling the Human Computer Interaction	55
5.8.1. Overview	55
5.8.2. The importance of HCI	55
5.8.3. Guidelines for developing an Effective HCI	56
5.8.4. Issues to be considered in HCI Designing	57
5.8.5. Characteristics of a Good HCI Design	57
5.8.6. Screen Layout for CCMS	58

6. The Construction Phase

6.1. Overview	60
6.2. Development Tools & Technologies	60
6.3. WAP Communication	
6.3.1. WAP Gateway Overview	62
6.3.2. WAP Architecture in CCMS	62
6.4. The Development of the CCMS	63
6.4.1. The Development Approach	64
6.5. Web Form Designs for CCMS	65
6.6. Components Used in Development	
6.7. Software Testing & Quality Assurance	66
6.7.1. Overview	66
6.7.2. Types of Testing	67
6.7.3. Testing Techniques	68
6.7.4. Software Validation & Verification	68
6.7.5. Testing Approach used in CCMS	68

7. The Transition Phase

7.1. Overview	70
7.2. Installing the CCMS	71
7.3. User Training & Support for CCMS	72
7.4. Method of Deployment	73
7.5. Deploying the CCMS	75
7.6. Security of the CCMS	76
7.6.1. System Security	77
7.6.2. Data Security	

8. Critical Appraisal

8.1. Overview	79
8.2. Challenges Faced	80
8.3. Lessons Learned	81
8.4. Strengths & Weaknesses of CCMS	82
8.5. Potential Enhancements	83

9. Conclusion	85
Reference List	86
 Appendices	
1. Appendix A - Rational Unified Process Architecture	88
2. Appendix B – Project Schedule	89
3. Appendix C – CCMS Diagram Notations	90
4. Appendix D – CCMS UML Diagrams	96
5. Appendix E – Database Design & Table Schemas	107
6. Appendix F – Coding Samples for System	111
7. Appendix G – Coding Samples for WAP Module	120
8. Appendix H – CCMS Web Form Design Samples	121
9. Appendix I – Test Case Design & Test Results	130
10. Appendix J – CCMS Physical System Architecture	131
 List of Figures	
Figure 1 - How Web Application Work	14
Figure 2 - Client Server Architecture on Web	14
Figure 3 - How WAP Gateway works	15
Figure 4 - System Architecture Model	23
Figure 5 - Class Diagram for the Case Registration	46
Figure 6 - Class Diagram for the Payment Module	46
Figure 7 - Use Case Diagram for the Case Registration	47
Figure 8 - Use Case Diagram for the Payment Management	47
Figure 9 - Activity Diagram for the Case Registration	49
Figure 10 - Sequence Diagram for Case Registration	50
Figure 11 – Screen Layout for CCMS	58
Figure 12 - WAP Architecture in CCMS	62
Figure 13 - System Development of the CCMS	63
Figure 14 - Web Form Design for Case Registration	64
Figure 15 - Component Diagram for the CCMS	65

Chapter 01

Introduction

1. Introduction

1.1 Company Profile

The Ceylinco Insurance commenced in 1939, and it has been the market leader since 1994. The division's unique packaged policies provide total protection against almost every possible hazard. The company was placed within the top four General Insurance Companies in Asia, competing with over 1,000 insurance companies and the company was also internationally recognized as an industry leader in product innovation, both in 1999 and 2001.¹

The Legal Department of Ceylinco Insurance plays a vital role to safeguard the cooperative image and the good will of the company from all kind of Legal threats that may arise against Ceylinco Insurance PLC. The co-responsibility of the Legal Department involves acting as the legal representative of the company in courts minimizing the losses that may arise as compensations against the company. The department consists of legal professionals. Administrative and supportive staffs to assist the routine and operational work within the department.

Company Name:	Ceylinco Insurance(General) PLC
Address:	69,Janadhipathi Mawatha,Colombo 01,Sri Lanka
Services We Offer:	All kind of Insurance Policies
Business Type:	Insurance
Geographic Markets:	Sri Lanka / Southeast Asia
No. of Employees:	Around 3000 People
Year Established:	1938 / 1939

¹ <http://www.ceylinco-insurance.com/index1.php>

1.1.1 Main Functions of Ceylinco Insurance:

- Big or Small Ceylinco Protects Them All.
- Ceylinco Insurance issue the all most all the insurance coverage in the insurance field.
- To provide protection and financial security of the highest quality to society, whilst adding to shareholders' wealth and recognising, rewarding and valuing the dignity of our staff.
- Ceylinco have been successful in positioning the Company as a regional player with operations in Nepal, Bangladesh, Mauritius and the Maldives. Together with partners ceylinco continue to improve their services to maximise their opportunities in the respective countries.
- VIP policy is the most comprehensive vehicle insurance policy available in Sri Lanka.
- Marine insurance policy targeted at the import/export community, providing a comprehensive cover for a vast variety of goods.
- Travel Classic policy offers great assurance for travelers and pays money, in case one falls ill far from home.

1.1.2 Potential benefits for Ceylinco Insurance by using an Intranet Solution

➤ Enhanced Productivity

With the support of CCMS, Ceylinco Insurance staff will be able to perform their tasks more effectively & efficiently due to the computerization. Previously they had to depend on manual documents which caused a lot of time wasting when it comes to searching for information.

➤ Enhanced Communication

With CCMS, Lawyers can communicate with the department staffs & system at anytime use of WAP Technology.

➤ Effective Planning & Monitoring

With the use of planning tools & reporting features available in the system Ceylinco Insurance management can take cost analysis reports anytime they want to.

➤ Enhanced Accessibility

Because of the use of a centralized database the data can be accessed within any location inside the corporate LAN. Also not like the manual system users can access information related to his/her work from anyplace.

➤ Enhanced System Maintenance

Because of the centralized client server architecture of the intranet solution system maintenance will be much easier & the system changes could be adopted rapidly.

1.2 Project Profile

The Court Case Management System (CCMS) is primarily engrossed on managing Case Registration, Court Types, Courts, Counsels, Fees, Counsel Fees, Case Positions, Damages, Documents & Cost Analysis Reporting Furthermore it will enhance the management of Case Appearances, Case Parties, Payment, Upload Scanning Documents & etc...

The Court Case Management System (CCMS) will be backed by a web server with Apache, PHP & MY SQL. CCMS is anticipated to be a highly user-friendly system in which the users will be guided throughout the whole process. The rationale of Court Case Management System is to "simplify the complications of Court Case Management by the use of Information & Communication Technologies".

The reporting features of Court Case Management System can be used to make & initiate decisions regarding the Court Cases by the management. This key rationale of CCMS will be used in the utmost effort to accomplish the corporate objectives & goals. CCMS will also provide a communication framework for the company with the convergence of mobile & internet technologies like Email & WAP 2.0 which can be used to provide a better solution for the communication problems within the company with more efficiency.

Chapter 02

Literature Survey

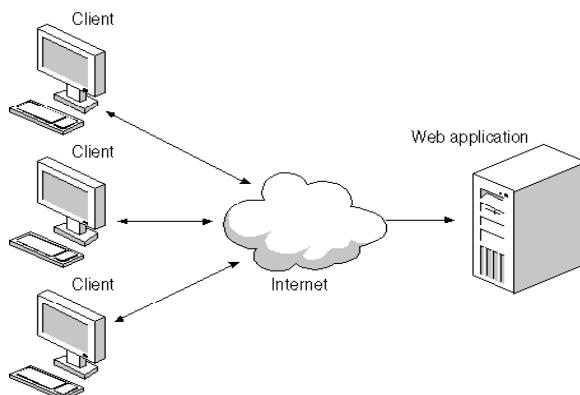
2. Literature Survey

2.1 The Role of a Court Case Management System

A Court Case Management System plays a major role in a Ceylinco Insurance. This is mainly because it takes care of most of the functions in Legal Department. The uniqueness of the CCMS is that it simplifies most of the communication aspects within a company and its branch network while concentrating on the main business functions.

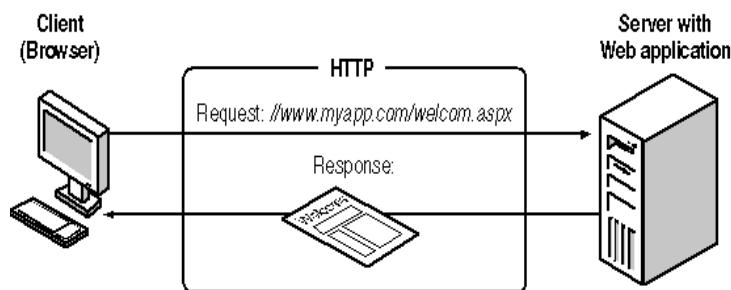
2.2 How Web Applications Work

Web applications use the client/server architecture. The Web application resides on a server and responds to requests from multiple clients over the Internet.



-Figure 1- (How Web Application Work)

On the client side, the Web application is hosted by a browser. The application's user interface takes the form of Hypertext Markup Language (HTML) pages that are interpreted and displayed by the client's browser. On the server side, the Web application runs under Apache Web Services. Apache manages the application, passes requests from clients to the application, and returns the application's responses to the client. These requests and responses are passed across the Internet using Hypertext Transport Protocol (HTTP).



-Figure 2- (Client Server Architecture on Web)

2.3 What is WAP

WAP is an open international standard for application layer network communications in a wireless communication environment. Its main use is to enable access to the Internet (HTTP) from a mobile phone or PDA.

A WAP browser provides all of the basic services of a computer based web browser but simplified to operate within the restrictions of a mobile phone, such as its smaller view screen. WAP sites are websites written in, or dynamically converted to, WML (Wireless Markup Language) and accessed via the WAP browser.

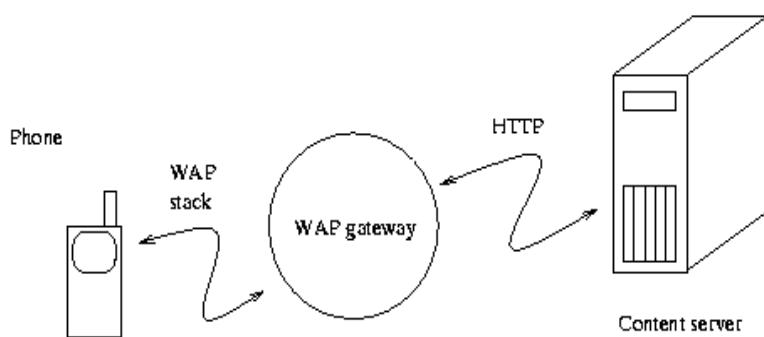
WAP 2.0 is a re-engineering of WAP using a cut-down version of XHTML with end-to-end HTTP. A WAP gateway can be used in conjunction with WAP 2.0 however, in this scenario; it is used as a standard proxy server. The WAP gateway's role would then shift from one of translation to adding additional information to each request.²

2.4 What is WML

Wireless Mark-up Language, based on XML, is a mark-up language intended for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones, and preceded the use of other mark-up languages now used with WAP, such as XHTML and even standard HTML.³

2.5 What is WAP Gateway

A WAP gateway sits between mobile devices using the WAP protocol and the World Wide Web, passing pages from one to the other much like a proxy. This translates pages into a form suitable for the mobiles, for instance using the WML. This process is hidden from the phone, so it may access the page in the same way as a browser accesses HTML, using a URL, provided the mobile phone operator has not specifically prevented this.



-Figure 3- (How WAP Gateway works)

² <http://kannel.org/>

³ <http://www.w3schools.com/wap/default.asp>

Chapter 03

System Modeling & Development Methodology

3. System Modelling & Development

Methodology

3.1 System Modelling

3.1.1 Importance of Modeling

A successful software development organization is the one which can satisfy their customers to the fullest by providing them with quality software which solves the main problem domain. This should be achieved by the effective & efficient use of resources both human and material. The main implication to this message is that the primary product of a development team should not be well documented documentations, award winning source coding or innovations but rather software it should be a software product that can satisfy the evolving customer needs. Every other thing should be considered as secondary. Unfortunately, many software organizations mystify secondary with irrelevant.

To set up software that satisfies its intended rationale, you have to arrange and organize the users in a disciplined fashion, to render the real requirements of your system. To develop software of lasting quality, you have to design a solid architectural foundation that's flexible to change. To develop software rapidly, efficiently, and effectively, with a minimum of software scrap and rework, you need to have the right people, the right tools, and the right focus. To do all this consistently and predictably, with an appreciation for the lifetime costs of the system, you must have a sound development process that can adapt to the changing needs of your business and technology.

Every process within a development project gathers around modeling & the models. Models are the main communication media to explain about the system also to visualize and control the system architecture. A model provides a better understanding of the system along with simplicity and promotes reusability through componentization.

3.1.2 Introduction to Object Oriented Modelling

As the name implies the whole modelling language is oriented around the objects. In an early book on Object-oriented analysis & design, Coad and Yourdon (1990) defined an object as follows:

“Object, an abstraction of something in a problem domain, reflecting the capabilities of the system to keep information about it, interacts with it, or both.”

This may not immediately appear to help very much, as parts of the definition are themselves a little obscure and raise further questions. Basically “abstraction” denotes the act of including only what is important or interested upon from a particular viewpoint. Also “reflecting capabilities” refers to including the possible capabilities of the object by keeping information about it, communicate with it, or to perform both.

Object-oriented analysis and design (OOAD) is often part of the development of large scale systems and programs often using the Unified Modelling Language (UML). OOAD applies object-modelling techniques to analyze the requirements for a context — for example, a system, a set of system modules, an organization, or a business unit — and to design a solution. Most modern object-oriented analysis and design methodologies are use case driven across requirements, design, implementation, testing, and deployment.

Object-Oriented Analysis (OOA) aims to model the problem domain, the problem we want to solve by developing an object-oriented (OO) system. The source of the analysis is a written requirement statements, and/or written use cases, UML diagrams can be used to illustrate the statements. An analysis model will not take into account implementation constraints, such as concurrency, distribution, persistence, or inheritance, nor how the system will be built. The model of a system can be divided into multiple domains each of which are separately analyzed, and represent separate business, technological, or conceptual areas of interest. The result of object-oriented analysis is a description of what is to be built, using concepts and relationships between concepts, often expressed as a conceptual model. Any other documentation that is needed to describe what is to be built is also included in the result of the analysis. That can include a detailed user interface mock-up document. The implementation constraints are decided during the object-oriented design (OOD) process.

Object-Oriented Design (OOD) is an activity where the designers are looking for logical solutions to solve a problem, using objects. Object-oriented design takes the conceptual model that is the result of object-oriented analysis, and adds implementation constraints imposed by the environment, the programming language and the chosen tools, as well as architectural assumptions chosen as basis of design. The concepts in the conceptual model are mapped to concrete classes, to abstract interfaces in APIs and to roles that the objects take in various situations. The interfaces and their implementations for stable concepts can be made available as reusable services. Concepts identified as unstable in object-oriented analysis will form basis for policy classes that make decisions, implement environment-specific or situation specific logic or algorithms. The result of the object-oriented design is a detailed description of how the system can be built, using objects.

3.1.3 Importance of Object Orientation in Large Scale Business Applications

"The Standish Group research shows a staggering 31.1% of projects will be cancelled before they ever get completed. Further results indicate 52.7% of projects will cost 189% of their original estimates. On an average, only 162 out of a thousand projects are completed on time and on budget. In the larger companies, the news is even worse: only 9% of their projects come in on time and on budget." ()

(Excerpt from a study by the Standish Group in 1995)

In the current technological advanced, competitive market, software systems are playing a major role in up keeping the performances & efficiency of a business entity. The following excerpts from IEEE Computing Practices¹ provide a very concise view of the challenges in the development of large business applications. Traditional business applications are characterized by some or all of the following:

- Large volumes of application data that are typically stored in a relational database
- Large amounts of user interface code to give a diverse set of users access to services and data in a variety of ways
- Well-defined business policies that must be enforced regardless of who works with the application and how
- The need to scale up to large numbers of users in a diverse and potentially distributed environment.
- Large teams of analysts, programmers and testers have to be effectively managed.

- The longer elapsed time, higher efforts and cost involved in the project require the execution of the project with schedule and cost controlled on an ongoing basis.

This is only a small sub set of the total problem inherent in developing large business applications. In fact the above problems can be solved by the usage of Object Orientation in the following way:

- Simplification of the real world & business entities
- Easy management of information & high CASE Tool support
- Objects & operations are easy to discuss with the user which leads to enhanced and precise requirements capture
- Rapid Application Development is more easily achieved and there is a flawless changeover from one phase to the other among the development phases such as analysis, design, implementation, testing & maintenance.
- Due to this seamless transition the management of the software becomes easier than ever
- With OO a library of objects can be developed which facilitates reusability, and conserves existing and consistent comprehension of the problem & business domain. Since many business domains are analogous to each other the pre-developed libraries can be used in latter project with or without customization
- Object orientation comprises of initiatives for aggregation, composition, inheritance and relationships along with the facilitations for hierarchical modeling, componentization, adaptability and evolution. Hence Object Orientation allows for management of complexity and the development of very large maintainable systems due to the above mentioned features.

3.1.4 Some Object Oriented Concepts in Brief⁴

Encapsulation

This is one of the basic concepts in Object Orientation. As the name itself implies Encapsulation, encapsulates attributes & behaviours to create objects and grouping of related items into one packaging called a Class. Implementation information of the object / class will be hidden from the external while providing only the interfaces to access. Encapsulation allows the developers of object to reuse already existing objects and if those have been well tested & documented those can be used in a much larger problem domain.

⁴ http://en.wikipedia.org/wiki/Object_oriented

Inheritance

Inheritance denotes the act of deriving a sub class by extending the feature set provided in its' super class. The subclass will inherit all the non-private attributes and behaviour of the super class. Also the subclass can override the behaviour of the super class by using Overriding. Inheritance can also be known as "IS-A" relationship. To get the very basic idea of reusability can be comprehended by implementing inheritance.

Polymorphism

The concept behind polymorphism is "Many (poly) forms (morphism)". This means a single method can have many different forms of behaviour implemented according to the context it is used within. This feature is very frequently used in a hierarchy of classes where many classes refer to the same parent class. This is achieved by specifying a general method on the super class & overriding the functionality of the same method in sub classes as different implementations.

Abstraction

Abstraction is the process of identifying software artefacts which are relevant to model the problem domain. When abstracting data each concept can be then converted into a class. Abstraction will be the first step in identifying the classes that will be used the application development.

Messages

Messages are simply a form of information passed from an object to another. Objects communicate by passing messages to each other. Actually this message is a method call, called by an object to carry out an activity on another. Messages can also pass parameters where objects can communicate with actual values.

3.1.5 Software Modelling using Unified Modelling Language

Overview

The Unified Modeling Language (UML) has now become a standard language for writing software blueprints. UML is a language for visualizing, specifying, constructing & documenting the artifacts of a software intensive development project. The UML is appropriate for modeling systems ranging from enterprise information systems to distributed Web-based applications and even to develop embedded & mission critical applications. UML is a very expressive language in which addresses all the viewpoints needed for development & deployment of such systems.

Even though it's expressive, it comprises of easy to learn & use symbolic language where special easy to remember symbols are used to denote the artifacts & units. The UML is only a language and it will be just another aspect of a software development project. The UML is process independent, although optimally it should be used in a process that is use case driven, architecture-centric, iterative, and incremental such as the Unified Software Development Process (USDP) to retrieve the maximum benefit out of it.

UML was developed by Grady Booch, James Rumbaugh & Ivar Jacobson. Started in 1994 and finished the version 1.0 by 1997. UML takes the best of each 3 of predecessor methods namely OOSE (Jacobson), OMT (Rumbaugh) & Booch.

UML is a Language for Visualizing – UML is a graphical language where a graphical notation is used to express the ideas rather using in a textual notation for modeling system. This facilitates ease of communication between the project stakeholders & the development team. This is achieved by a collection of easy to understand symbols attached with well-defined semantics.

UML is a Language for Specifying – The term specifying relates to the development of models which precise, unambiguous and complete. The UML addresses the specification of all the important phases & their decisions such as requirements specification, analysis, designing, implementation & deployment.

UML is a Language for Constructing – Due to the high support of the CASE tools which are incorporated with UML, UML has become not only visual modeling language but the artifacts developed in UML can be directly connected with popular programming languages such Sun Java, C++ or even the most popular Visual Basic. Not only the programming languages even the UML diagrams are totally convertible into a Relational or Object Relational Database schemas.

UML is a Language for Documenting – In a software-intensive system the documentation plays a major role when it comes to communication, measurements & controlling. UML addresses all these aspect by developing various artifacts relevant for each development phase in addition to the raw executable program coding. These artifacts include requirements, architecture, design, source code, project plans, test evidence, prototypes & releases. All of the above artifacts are very much important for project progress communication, measuring & estimations, controlling & monitoring activities.

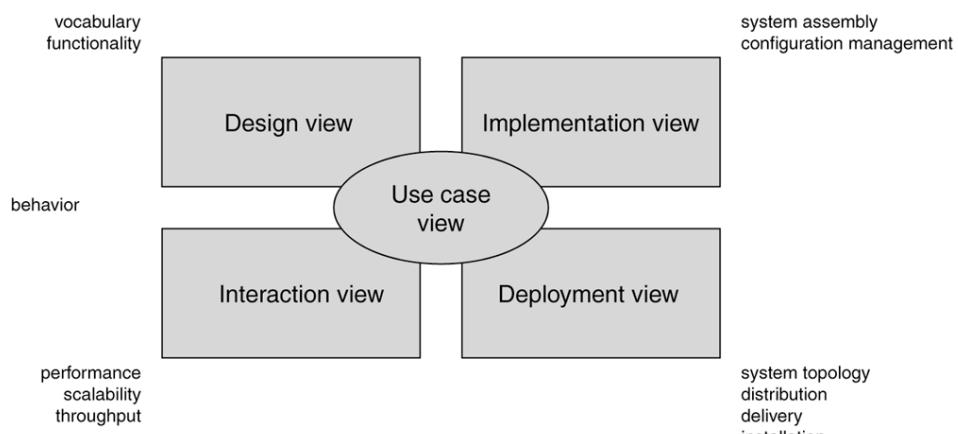
3.1.6 Modeling the System Architecture

Since there are many stakeholders, end users, development team members & managers gather around a project many different viewpoints are needed to satisfy different aspects of the same system. This may happen at different times in the project life cycle. Due to this reason the system architecture has become a vital aspect of system development hence control the iterative & incremental development throughout the project life cycle.

The system architecture helps to take the following crucial decisions:

- The software system organization
- The selection of the structural elements and their interfaces by which the system is composed
- Their behavior, as specified in the collaborations among those elements
- The composition of these structural and behavioral elements into progressively larger subsystems
- The architectural style that guides this organization: the static and dynamic elements and their interfaces, their collaborations, and their composition

Software architecture is also concerned about the functionality, flexibility, reusability, comprehensibility, feasibility, constraints, tradeoffs & performance.



-Figure 4- (System Architecture Model)

Use case view – encompasses the use cases that describes the behavior of the system as seen by the end users

Design view – encompasses the classes, interfaces and collaborations that form the vocabulary of the problem

Interaction view – shows the flow of control among its various parts

Implementation view – the various artifacts used to assemble to the system release

Deployment view – the nodes that make up the hardware topology on which it executes

3.1.7 UML Diagrams in Brief⁵

Following is a brief introduction the models used in UML to describe the various aspects of a system. For further information on the diagrams please review the appendices.

- Class Diagram

A class diagram is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, methods, and the relationships between the classes. Attributes identifies the characteristics of a class while methods identify the behaviour of it. Relationships are the logical links between classes and can be in different flavours. Namely Aggregation, Composition, Link, Association and Generalization.

- Use Case Diagram

A use case diagram includes a set of use cases where each use case is a description of the functionality of the system from the user's perspective. Use case diagrams are used to show the functionality that the system will provide and to show which users will communicate with the system in some ways to use that functionality.

- Activity Diagram

An activity diagram represents the business and operational step-by-step workflows of components in a system. An Activity Diagram shows the overall flow of control.

- Sequence Diagram

A sequence diagram shows an interaction between objects arranged in a time sequence. Sequence diagrams can be drawn at different levels of details and to meet different purposes at several stages in the development life cycle. The commonest application of a sequence diagram is to represent the detailed object interaction that occurs for one use or for one operation.

- Collaboration Diagram

A Collaboration diagram shows the objects and their association with other objects in the system apart from how they interact with each other. The association between objects is not represented in a Sequence diagram. A Collaboration diagram is easily represented by modeling objects in a system and representing the associations between the objects as links.

⁵ http://en.wikipedia.org/wiki/Unified_Modeling_Language

- State Chart Diagram

State chart diagrams, also referred to as State diagrams, are used to document the various modes ("state") that a class can go through, and the events that cause a state transition. In comparison the other behavioural diagrams which model the interaction between multiple classes, State diagrams typically model the transitions within a single class.

- Component Diagram

A component diagram depicts how a software system is split up into physical components and shows the dependencies among these components. Physical components could be, for example, files, headers, link libraries, modules, executables, or packages. Component diagrams can be used to model and document any system's architecture.

- Package Diagram

A package diagram depicts how a system is split up into logical groupings by showing the dependencies among these groupings. As a package is typically thought of as a directory, package diagrams provide a logical hierarchical decomposition of a system. Using the packages as developers we can maximize internal cohesion while minimizing the coupling between units. With these guidelines in place, the packages are good management elements.

- Deployment Diagram

A deployment diagram serves to model the hardware used in system implementations, the components deployed on the hardware, and the associations between those components. The elements used in deployment diagrams are nodes (shown as a cube), components (shown as a rectangular box, with two rectangles protruding from the left side) and associations.

3.2 Development Methodology

3.2.1 Importance of a Methodology

Over many years information system development methodologies have been developed and introduced especially to conquer the problems which arise in software projects. However no methodology has been able to solve all the problems and to fulfil the requirements. This is due to the dynamic nature of the computing world & the variations in business domains. In a changing world it's unlikely that yesterday's solution will ever completely solve today's problems. Nevertheless, many advantages are there to be perceived from a methodology. Following is yet another set of advantages we can perceive.

- Helps to produce a better quality product, in terms of documentation standards, acceptability to the user, maintainability and consistency
- Helps to ensure the user requirements are completely satisfied
- Helps the project manager by giving a better control over the project, perform better estimations & reduce unessential overheads.
- Promotes communication between project members, & other stakeholders by defining the essential participants and their interactions in different phases.
- Encourages the transmission of knowledge throughout a project & the organization among the members.

3.2.2 Unified Development Process⁶

"The RUP is a well-defined and well-structured software engineering process. It clearly defines who is responsible for what, how things are done, and when to do them. The RUP also provides a well-defined structure for the lifecycle of a RUP project, clearly articulating essential milestones and decision points."

One of the software development processes currently in wide use is the Unified Software Development Process (USDP). The best known and extensively documented refinement of the USDP is the Rational Unified Process or RUP. The Unified Process is not simply a process but rather an extensible framework which can and should be customized for organizations & projects. It is also claimed that the RUP embodies much of the currently accepted best practices in information systems development. Below are the features of RUP.

⁶ http://en.wikipedia.org/wiki/Unified_Process

- Iterative & Incremental

The RUP uses an iterative & incremental approach where each project phase comprises of a series of iterations. Each iteration results in an increment, which is a release of a system that contains added or improved functionality compared with the previous releases.

- Use case driven

The RUP uses Use Cases to capture the functional requirements for the system and to define the iteration contents. Each iteration takes a set use cases all the way from requirements analysis to deployment.

- Architecture Centric

The RUP claims that the system architecture resides in the heart of the efforts taken by the project team which are needed to shape up the system. Since no single model is sufficient to get a better understanding of a system & to cover all the aspects, RUP supports for multiple architectural viewpoints & models.

- Risk Focused

In RUP at the beginning of each iteration a total risk analysis for that iteration is carried out to ensure that there will be no problem in the midst of the iteration. This allows the project team to focus on addressing the most critical risks early in the project life cycle.

3.2.3 Why USDP was chosen for CCMS development

USDP has been chosen as the development process for this project because it used the well defined best practices which have been proven the commercial software domain.

- Better Risk Management

In USDP Risk management is built in to all the phases with different extents. However in each phase the risks for that phase & the successor are identified

- Iterative & Incremental Process

In each iteration we can expect for a release which addresses the most critical sections of the system, hence the non critical aspects can be considered later

- Verifiable Software & High Quality

USDP bases software projects are always easier to verify & validate because the quality assessment activities are built into the process

- CASE Tool Support

USDP comes with a higher capability to be used CASE Tools. Modeling with UML & Managing USDP bases project has got easier like never before. Rational Rose (UML), Rational Software Architect (Development Support) & Rational ClearCase (Configuration management) are the most popular CASE Tools Available.

3.2.4 Unified Software Development Process Architecture

The USDP does not follow the traditional life cycle but adopts an iterative approach within four main phases. These phases reflect the different emphasis on tasks that are necessary as system development proceeds. These differences are captured in a series of workflows that run through the development process. Each workflow defines a series of activities that are to be carried out as part of the workflow and specified the roles of the people who will carry out those activities.

Main four Phases and their objectives in brief:

- Inception – Understand the business case & the scope of the project
- Elaboration – Mitigate major technical risks, build the baseline architecture, understand what it takes to build the system
- Construction – Build the operational system
- Transition – Build the final version & deploy at the customer premises.

The Nine Workflows

- Business Modelling
- Requirements
- Analysis & Designing
- Implementation
- Test
- Deployment
- Configuration & Change Management
- Project Management
- Environment

The important fact to bear in mind is that in the waterfall lifecycle activities and phases are one and the same, in iterative lifecycles like the RUP the activities are independent of the phases, and it is the mix of activities that changes as the project proceeds.

REFER APPENDIX - A FOR THE USDP ARCHITECTURE DIAGRAM

Chapter 04

Inception Phase

4. The Inception Phase

4.1 Overview

This is the first phase within the Unified Process. The main purpose of the inception phase is to get the total understanding of the system so the proceeding phases will not come to any confusing situations. Compared to the previous linear & iterative models the Inception phase resembles the Feasibility & Requirements capturing phases. Following are the main objective of the Inception Phase.

- To establish the main business case for the project
- To establish the project scope, project boundary & its conditions
- To assess the potential risks of the project in comparison with its potential benefits
- To assess the project viability technically, organizationally & economically
- To prepare a preliminary project schedule and cost estimate

4.2 Initial Project Review & Feasibility Study

4.2.1 Current System Environment

Ceylinco Insurance PLC, Legal Department currently practices a manual methodology in maintaining the information. Some of the data is kept in spreadsheets in a single computer rather than in a networked manner. Data is stored & recovered manually from flat files as and when required which consumes a great effort.

Only the Payment functions operate on a computerised system while the others in manual flat files & spreadsheets. Currently all the computers are interconnected using a Local Area Network which encloses all the computers in management areas.

4.2.2 Drawbacks of the Current System

This project is all about mitigating the complications of Court Case Management & there are some drawbacks in the current semi-automated process also.

- Management of the court cases consumes lot of time
 - Delays in finding case records when required
 - Delays in payments processing
 - Delays in finding court case files (not in digital media)
- Diminished security
 - The system is vulnerable to external threats
 - Users have equal and the highest privileges rather than distributed privileges
- Data duplication & redundancy
 - Data is kept in flat files which re-introduces duplication & redundancy
 - Diminished efficiency in recovering information from data files
- Standalone system
 - The users feel isolated due to the detached nature of the system from the network
 - Data is not accessible from remote locations
- Integrity checking is impossible
 - Error corrections are complex and consume a lot of time.

- Not user friendly
 - The manual system is not user friendly; it carries a lengthy procedure to maintain cases details
- No backup facilities
 - The system does not maintain backup procedures

4.2.3 Project Objectives

Due to the drawbacks of the current system mentioned in the previous section, the CCMS is intended to meet the following objectives & to satisfy the following requirements.

4.2.3.1 Requirements

- User-friendly system
 - More user-friendlier system
 - Effortless processing of information
 - Quicker responses from the system
- Network enabled system
 - Remotely accessible system
 - Central data repository
- More channels of communication
 - More communication methods for interaction between the system
 - Quicker methodology to communicate from anywhere either within the corporate premises or on the move (WAP)

4.2.3.2 Objectives

- To implement Up to date court cases data for effective management
- To provide a user-friendly computerised system which can cut down the overheads caused by the current procedures
- To provide the maximum accessibility level to the system by incorporating mobile communication technologies such as WAP & E-Mail
- To provide a system with a central repository of data this lessens down the redundancy & duplication

- To provide a online-real time system which can be accessed from anywhere in the corporate network

4.2.4 Advantages of CCMS

- Managerial tasks can be accomplished in a lesser amount of time with the introduction of latest web & data repository technologies
- To design a system to abstract data from the system via a mobile phone
- Reduce the overheads incurred by the use of manual flat files & other manual paper work
- Security can be elevated by the use of role based security model
- To implement an open and flexible paperless system
- To provide a online-real time system which can be accessed from anywhere in the corporate network

4.3 Discovery of Facts

The expectations of a new system are to fulfil the customer's requirements which engage them in a productive process. In order to achieve this as the analysts & designers having accurate information about the business domain is crucial. So it is important to assemble the information regarding what people do in that organization & the roles of their work. Most commonly when developing a new system confusions & problems arise regarding how the current system works & its inadequacies, so it is also crucial to capture the user's expectations from the new system and the things they could not do with the existing system. Following are some of the methods used to capture the existing system & new system requirements.⁷

➤ **Background Reading**

In this method the analysis team will be employed in the organization which is the subject of the fact gathering exercise. While working there the team could get a better understanding of the system by going through reports, charts & documents related to the organization.

➤ **Interviewing**

This is the most widely used method. In this method both the analysis team and the organization personnel will meet on an agreed appointment. The basic routine is to ask & clarify the problems regarding the domain from the organization personnel live & any suitable questionnaires may be used for further information gathering.

➤ **Observation**

In this method the analysis team can observe the work of the organization personnel in their natural setting by looking at them for a time period. Rather than an interview here the team can analyze the whole routine they follow to do a task rather than specifying it vocally and forgetting the exceptional situations which actually arise.

➤ **Document Sampling**

In this method the working & blank documents which the staff uses could be collected during the course of interviews and observation sessions. Finally the documents could be analyzed to determine the data they use.

➤ **Questionnaires**

In this method copies of written questions are distributed among the staff. Finally the team could analyze the answers to determine the requirements.

⁷ www.rocw.raifoundation.org/computing/BCA/systemanalysis/lecture-notes/lecture-21.pdf

4.4 Requirements Specification

The requirements specification is a high level document which describes about the function & non functional requirements of the end users. Also it specified the potential expectations about the proposed system & what it needs to achieve. The requirements specification will work as an agreement between the end users & the development team and will agree upon the specified requirements.

Following are some benefits we can gain from a requirements specification as specified in the IEEE 830 Standard for Software Requirements Specification.⁸

- Establish the foundation agreement between the end users and the development team and will contain everything regarding the requirements that the system should satisfy
- Reduce the development time & effort by clarifying all the misunderstandings & confusions regarding the requirements beginning at the initial phases. If these errors were not removed from the requirements specification can lead to extreme problems in latter phases
- Provides a foundation for cost & schedule estimation
- The Specification is a good basis for validation & verification in latter phases to check the deliverables of those phases with the initial requirements.

A good requirements specification should be:

- Accurate
- Unambiguous
- Complete & Consistent
- Verifiable & Traceable

⁸ http://en.wikipedia.org/wiki/Software_Requirements_Specification

4.5 Captured User Requirements

Below are some of the requirements captured by doing Interviews & observations with the help document sampling.

Functional Requirements

- More channels of communication
 - More communication methods for interaction
 - Quicker methodology to communicate from anywhere either within the corporate premises or on the move
- Better Planning & Controlling
 - More control over the data & the ability to view the information in different view points
 - Produce management reports efficiently
- Central Repository of Information
 - A central repository for information rather than holding the information in a distributed manner.
 - Higher data accessibility, data should be accessible from any location within the corporate LAN.

Non Functional Requirements

- User-friendly System
 - More user-friendlier system
 - Effortless processing of information
 - Quicker responses from the system
- Network Enabled system
 - Remotely accessible system via the LAN or internet
- Enhanced Security
 - Role based security to ensure only authorized personnel view different
 - Sections of the system
 - Easy user & role management via administration panel
- High Performance Reporting Generation & Mobile Communication

4.6 Project Scope

Court Case Management System (CCMS) is primarily targeted to be activated in areas where the legitimate user within Legal Department and Anyone just can view the data within corporate network. It is basically according to their authorization level. CCMS will be the key framework behind the scenes of these elements. Since this project is based on the client-server & online-real time system architectures, the users will not feel any isolation at all and the users will be able to get their hands-on CCMS from anywhere of the corporate network.

Functions of CCMS:

- **Case Registration Management**
 - Case Registration Module
 - Case Appearance Module
 - Case Parties Module
 - Document required Module
 - Settlement Information Module
 - Updating Module
 - Document Upload (Scanning Documents) Module
- **Counsel Management**
 - Counsel Module
 - Fee Module
 - Counsel Fee Module
 - Case Position Module
- **Court Management**
 - Court Type Module
 - Court Module
- **Report Management**
 - Search Module
 - Search Module (By GPRS enable mobile phones)
 - Cost Analysis Report Module
- **Peripheral Management**
 - Damage Module
 - Document Module

➤ **Payment Management**

- Payments Module
- Payments Progress Module (By GPRS enable mobile phones)

➤ **System Administration Management**

- User Module
- Database connectivity Module

4.7 Sub Systems of CCMS

According to the captured user requirements I have come to a conclusion about the sub systems required to build the complete system. For this I have used a UML – Package diagram.

The package diagram depicts how the system is split into logical groupings & the dependencies between these groupings. A package can be thought of as a logical directory and it provides the decomposition of the system in a logical & hierarchical way. Packages are good elements for managing the system where we can organize with maximized internal cohesion and reduced coupling between the packages.

REFER APPENDIX - D SECTION (G) FOR PACKAGE DIAGRAM OF THE CCMS

4.8 Proposed Project Schedule & Milestones

This project is expected to deliver the following deliverables and milestones which are set are estimates of the planned tasks.

Milestone	Starting Date	Ending Date
Business Feasibility Study	01/02/2008	12/02/2008
Submit Project Proposal	15/02/2008	15/02/2008
1 st Presentation	15/06/2008	15/06/2008
2 nd Presentation	03/09/2008	03/09/2008
3 rd Presentation	20/10/2008	20/10/2008
Submission of Project Documentation	02/11/2008	02/11/2008

REFER APPENDIX – B FOR THE PROJECT SCHEDULE (CHART)

4.9 Project Prerequisites

In order to fulfil the corporate requirements & project objectives the following prerequisites should be in place.

Software

- Development
 - Adobe Dreamweaver CS 3.0
 - Zend Studio 6.1
 - Apache Server
- Testing
 - VMWare Workstation 5.5
 - WAP Proof 2007
- UI Design
 - Adobe Photoshop CS 3.0
 - Adobe Flash CS 3.0
- Data Repository
 - MySQL Server
 - Toad for MySQL Freeware
 - PHP My Admin

Hardware

- ✧ Development
 - 2.4 GHz Dual Core or higher CPU Power
 - 1GB or higher physical memory
 - At least 10GB Hard disk space for the development tools
 - Removable storage for backups
 - Internet Access
 - Mobile Telephone for WAP base Communication

4.10 Economical Feasibility

Category	Component	Qty	Cost
Software	Adobe Dreamweaver CS 3.0	1	\$ 399.00
	Toad for MySQL - Freeware	1	\$ 0.00
	WAP Proof 2007	1	\$ 129.00
	VMWare Workstation 5.5	1	\$ 189.00
	UI Developing Software - Available	0	\$ 0.00
	PHP, MySQL, Apache - Freeware	1/1/1	\$ 0.00
Hardware	Web and Database Server	1	\$ 3,000.00
Connectivity	GPRS Connection	1	\$ 25.00
	TOTAL ESTIMATION		\$ 3742.00

Chapter 05

Elaboration Phase

5. The Elaboration Phase

5.1 Overview

During this phase the main concentration shifts to the design of the system from which costs can be reduced, performance could be gained & customer satisfaction is guaranteed. This is done theoretically by developing a set of designs for the system that demonstrates how it can be built within an agreed & acceptable timescale & budget. However, the primary goals of this phase are to deal with the known risks and to ascertain the system architecture. This architecture will be validated through the implementation of the final executable files & architecture.

In this phase, an executable prototype is built through one or more iterations, depending on the size, scope, risk, and uniqueness of the project. This effort should at least address the critical use cases identified in the previous phase, which in general depicts the major technical risks of the project.

Following are the main activities performed in this phase.

- The establishment of the requirements specification
- The development of the basic architecture of the system
- The main development of the analysis & design models
- The Final identification of risks

5.2 Business System Option & Technical System Option

Prior to start the development of the Court Case Management System after having a discussion the management of Ceylinco Insurance agreed for the below BSO & TSO.

Business System Options

- The CCMS to be developed as a Web Based application
- The CCMS to be a centralized system which can be utilized for managing the court cases all over the country.
- Anyone with a Legal Department & branches able to access the system
- To accommodate mobile technologies (WAP) as the major channel of external (lawyers) communication

Technical System Options

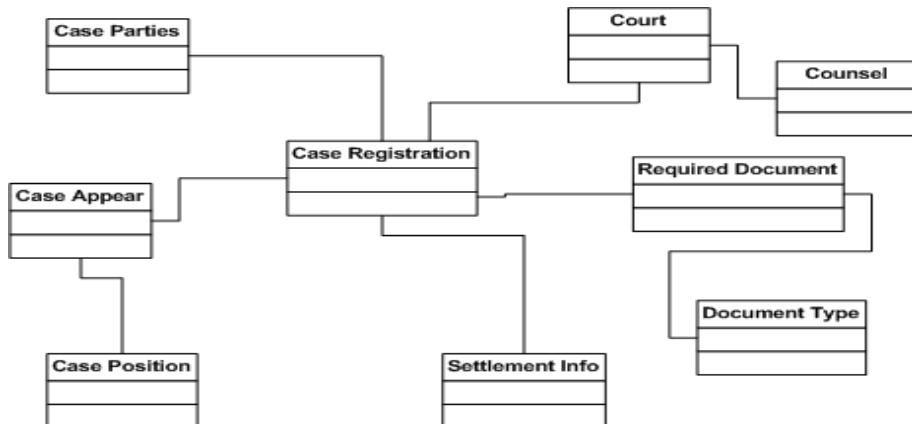
- The CCMS to be deployed in a Red Hat Enterprise Linux 5 server environment bounded by security essentials
- Create a new database while synchronizing with the existing database & flat files
- To implement the mobile communication channels via a Kannel WAP/SMS Gateway software.
- The ceylinco staffs and the lawyers will be authenticated by the security mechanisms within the CCMS

5.3 Structural Modelling

Class Diagram

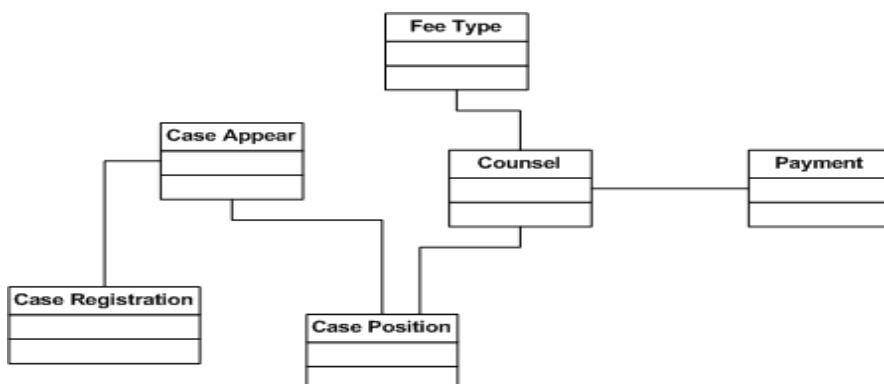
In an Object Oriented development project, a class plays a major role & is the most important building block. A class is a blueprint for set objects that share the same attributes, operations, semantics & relationships. An object is something that has a well defined boundary and a meaning within the problem domain. An object can be a person, organization, concept, thing or an abstraction. An object contains attributes & behaviors while the object identity distinguishes from similar objects even with the same structure. Classes can be used to understand & capture the terminology of the system we are developing. Within the classes there may include abstractions of the problem domain as well as implementation classes which are part of the actual development.⁹

Class Diagram for the Case Registration Management Sub System



-Figure 5- (Class Diagram for the Case Registration)

Class Diagram for the Payment Management Sub System



-Figure 6- (Class Diagram for the Payment Module)

REFER **APPENDIX – C SECTION (A)** FOR CLASS DIAGRAM NOTATION

REFER **APPENDIX – D SECTION (A)** FOR OTHER CLASS DIAGRAMS OF CCMS

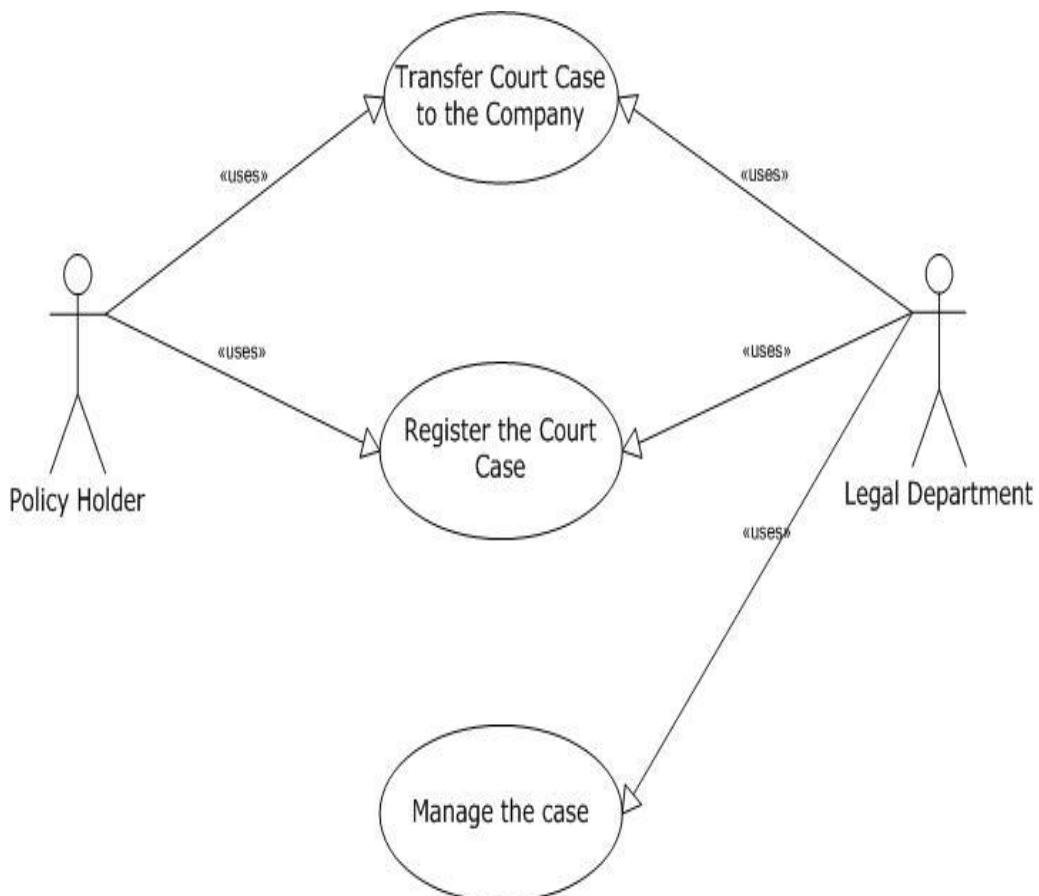
⁹ <http://www.agilemodeling.com/style/classDiagram.htm>

5.4 Behavioral Modeling

5.4.1 Use Case Diagram

No system executes in isolation. Every system needs to communicate with the outer environment, its users or with automated actors for some purpose. A use case diagram is a collection of use cases & their interactivity with the actors. We can apply the Use Case modelling to understand the behaviour of the system we are developing, without moving on with how that behaviour is implemented. Use cases provide a way to agree on a common agreement regarding the requirements among the developers, end users & domain experts. Since the Rational Unified Process is use case oriented these initial use cases we create will be cross referenced again in different later phases. The use case diagrams are one of the diagrams which model the dynamic aspects of a system.¹⁰

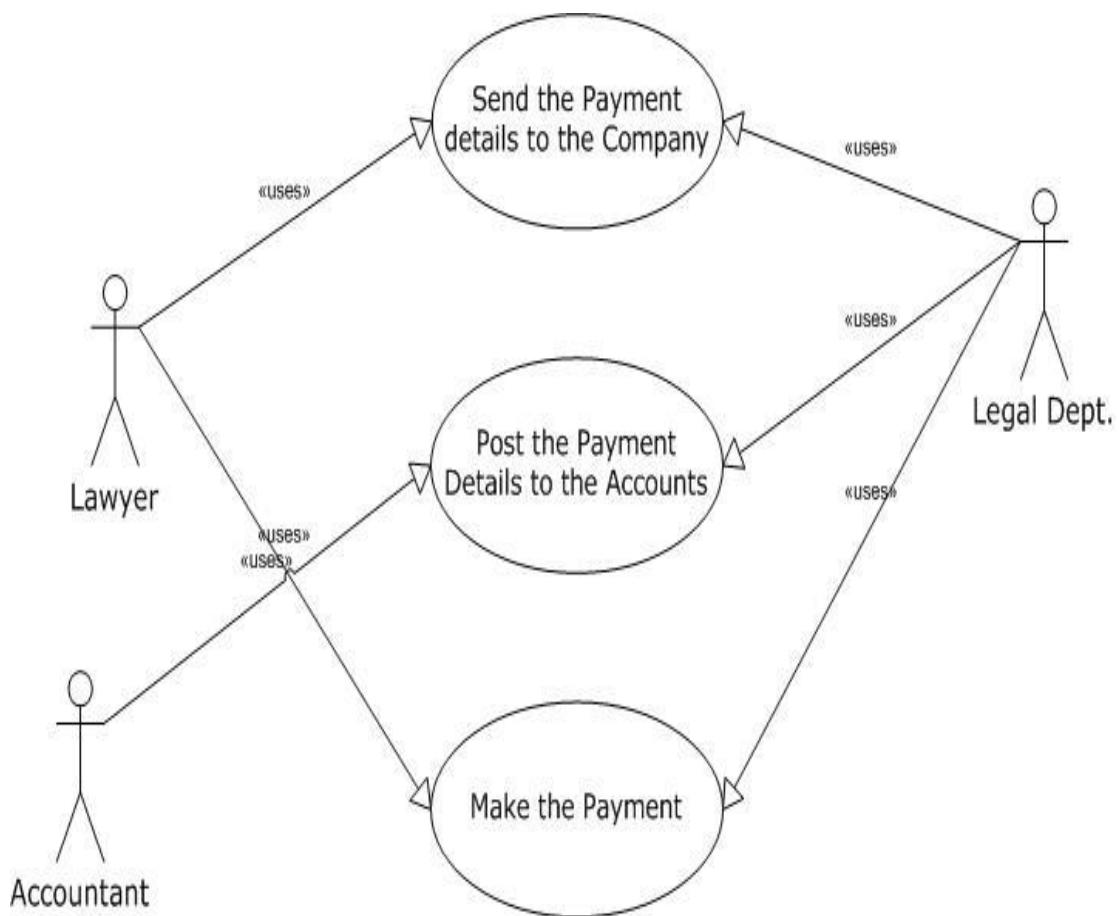
Use Case Diagram for the Case Registration Management Sub System



-Figure 7- (Use Case Diagram for the Case Registration)

¹⁰ http://en.wikipedia.org/wiki/Use_case_diagram

Use Case Diagram for the Payment Management Sub System



-Figure 8- (Use Case Diagram for the Payment Management)

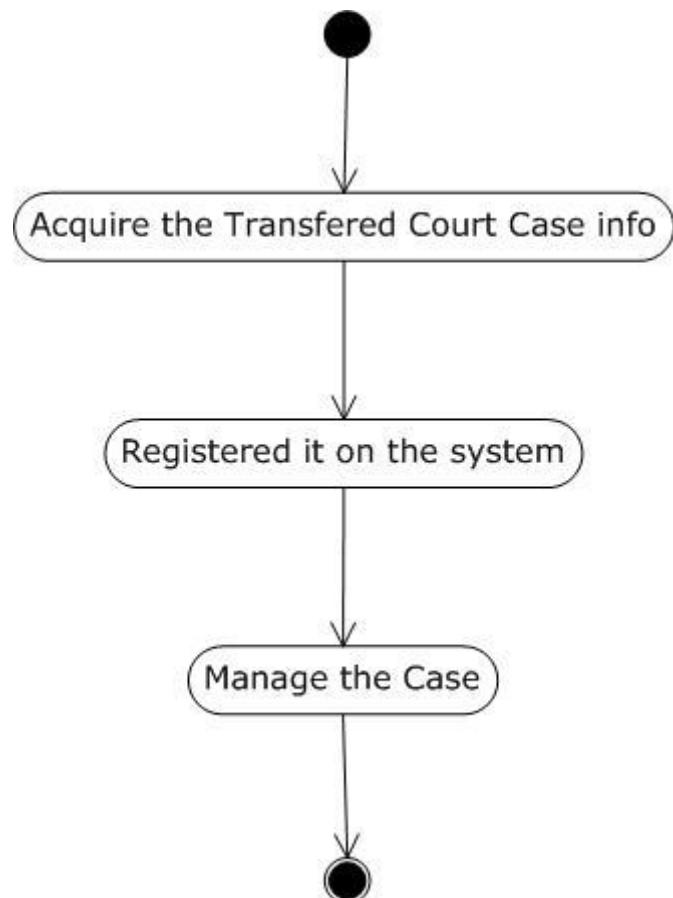
REFER APPENDIX – C SECTION (B) FOR USE CASE DIAGRAM NOTATION

REFER APPENDIX – D SECTION (B) FOR OTHER USE CASE DIAGRAMS OF CCMS

5.4.2 Activity Diagram

Activity diagrams are used for modeling the dynamic aspects of systems. An activity diagram is fundamentally a flowchart, showing control flows from activity to activity. Unlike a traditional flowchart, an activity diagram can show concurrency as well as branches of control within different activities. An activity always results in some action which is in turn leads to a change in the state of the system or return of some computational result. The actions depicted by the activity diagram may include operations, sending messages, creating & destroying objects or some algorithmic expression.¹¹

Activity Diagram for the Case Registration



-Figure 9- (Activity Diagram for the Case Registration)

REFER APPENDIX – C SECTION (C) FOR ACTIVITY DIAGRAM NOTATION

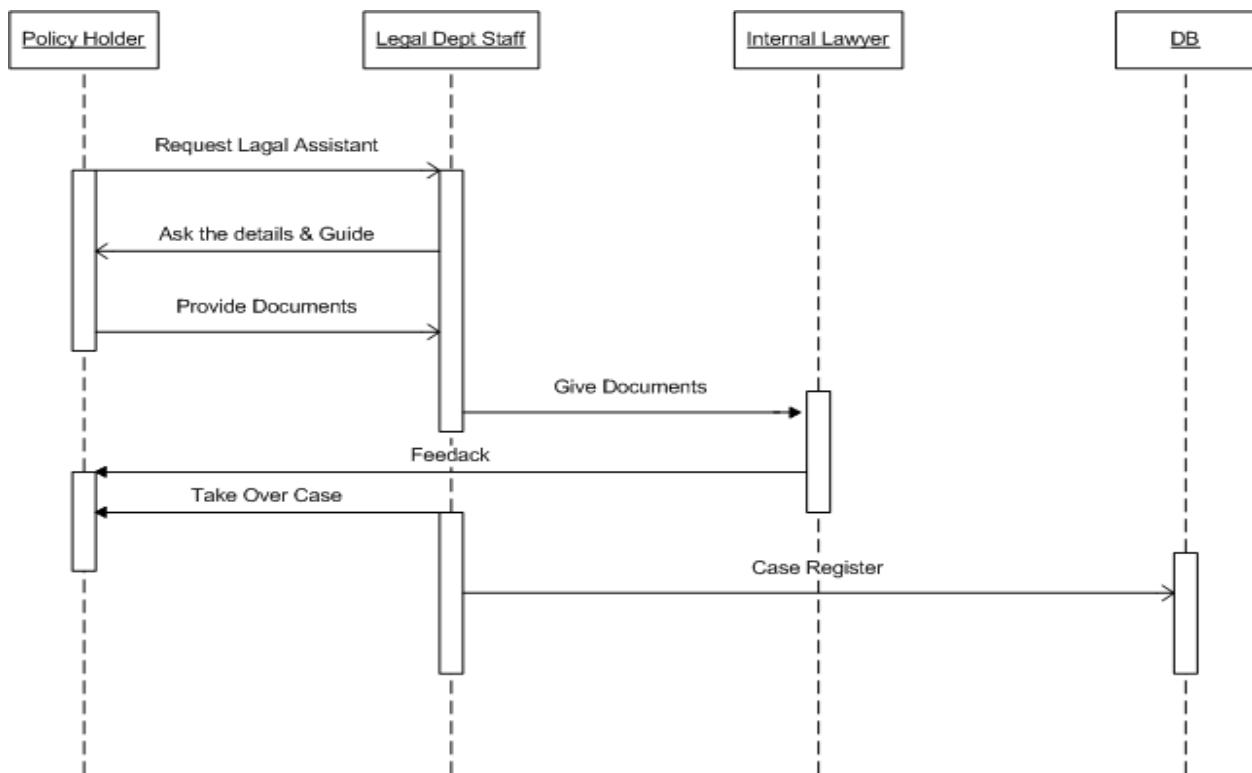
REFER APPENDIX – D SECTION (C) FOR OTHER ACTIVITY DIAGRAMS OF CCMS

¹¹ <http://www.agilemodeling.com/style/activityDiagram.htm>

5.4.3 Sequence Diagram

The purpose of the sequence diagram is to show the interactions that occur within a use case. These interactions will be ordered in a sequential manner where they get ordered according to the time. Unlike the class diagrams the sequence diagrams even can be used with the business managers to communicate the requirements & the interactions they have within the problem domain. Use cases are more often expressed in detail using sequence diagrams. Sequence diagrams fall into the same category as the Collaboration Diagrams where they both share the semantics of the "Interaction Diagrams" but sequence diagrams order the interactions in the order they occur.¹²

Sequence Diagram for Case Registration



-Figure 10- (Sequence Diagram for Case Registration)

REFER APPENDIX – C SECTION (D) FOR SEQUENCE DIAGRAM NOTATION

REFER APPENDIX – D SECTION (D) FOR OTHER SEQUENCE DIAGRAMS OF CCMS

¹² http://en.wikipedia.org/wiki/Sequence_diagram

5.5 Developments & Testing Strategy

Development

For the completion of the CCMS project, the Rapid Application Development (RAD) model will be chosen. Since this model comprises of both prototyping & iterations, it will develop the system module by module and after each module a prototype will be published for further refinement.

Each iteration will consist of the following phases

- Requirements specification
- Analysis
- Design
- Development
- Unit Test
- Integration Test
- Publish Prototype

System modelling will be done using the Unified Modelling Language (UML) with the help of Rational Rose 2003 modelling tool. System development will be done using the PHP, WML & AJAX & etc.

Testing

Since each iteration produces a prototype, each prototype will be compared & verified against requirements specification of each iteration. The prototypes will be tested as single units & then together with the previous prototype to verify the integration is successful. For testing purposes a special testing environment will be simulated by the use of VMWare Workstation 5.5, which is a special tool for creating virtual computers.

5.6 Implementation Strategy

System Implementation

The system will be deployed in a testing environment which simulates the live environment. The deployment will be approaching an incremental conduct where later versions of the prototypes are built upon the previous versions. The main application will reside in the corporate web server as a web application & the mobile communication handling software will be deployed in a separate or same server but as a service.

Mobile Communication Implementation

Mobile communication facilities will be facilitated by the use of GPRS. This will use the same SIM card which is used in ordinary mobile handsets and will operate in the current GPRS or 3G network. No Special agreement is needed for this kind of implementation with mobile operators.

5.7 Modelling the Database for CCMS

5.7.1 Database & the Database Management System

A database is a collection of information which are related to each other & controlled under the Database Management System. Typically there are several database types in use but some are obsolete. Namely Hierarchical, Network, Relational, Post Relational, Object Oriented Databases, Deductive Databases & so on. Relational & Object oriented databases are the most popular, network & hierarchical are obsolete while the others in use only in a limited application domains. The main tasks for the DBMS are to control the database's security, concurrency, backups & restoring, transaction management & recovery from failures.¹³

5.7.2 Advantages of having a database

Compared to the legacy data storing techniques, the Database systems are much more reliable than ever before with the following features

- Reliability & Scalability
- Security & Access Controls on different information for different users
- Data Integrity & Integration

5.7.3 The Database Management System for the CCMS

Since the project is going to be an intranet based application the application data needs to be accessed from anywhere within the corporate LAN. So the DBMS should be able to handle so many connections at once in a networked environment. Also the database for this system should be able to handle an enormous amount of information regarding cases, lawyers and etc. By considering these factors the MYSQL Server 5.0 database management system has been chosen.

5.7.4 Unique features of the MYSQL 5.0

The CCMS system uses MYSQL as the DBMS due to the following reasons:

MySQL is characterised as a free, fast, reliable open source relational database. It does lack some sophistication and facilities, but it has an active development team and, as it goes from release to release, more capabilities are added. At certain times there will be a trade off between speed and capabilities.

¹³ <http://www.mountainman.com.au/software/history/>

5.7.5 The Database Design for the CCMS

The Table Schemas in Detail

Following are the tables with their schemas & functionality used inside the database of the CCMS. The following list is not the complete table list. For the complete set please refer the appendices.

TABLES

- cou_doc_detail
- cou_doc_main
- cou_m_case_regis
- cou_m_counsel
- cou_m_counsel_fee
- cou_m_court
- cou_m_user
- cou_r_branch
- cou_r_court_type
- cou_r_cposition
- cou_r_damage
- cou_r_docs
- cou_r_fee
- cou_t_case_appear
- cou_t_case_parties
- cou_t_counsel_fee
- cou_t_doc_required
- cou_t_settle_info

REFER **APPENDIX – E** FOR DATABASE DESIGN & TABLE SCHEMAS

5.8 Modelling the Human Computer Interaction

5.8.1 Overview¹⁴

The term “Human Computer Interaction” resembles the interactions that take place between the computer system & its users. This is a highly crucial aspect which needs more attention. Basically the interface is what the users see & use. To the user the interface they & the way they interact with it is the system. The HCI design can totally control their attitudes towards the system & the contributions they make in order to make the system a success. HCI combines two sciences regarding human, Psychology and ergonomics with computer science.

5.8.2 The Importance of HCI

The HCI design should match with the user expectations and is the most important aspect of the system. It does not matter how well the system works behind the scenes. What does matter for the user is how they interact & what they see from the system. Following are some of the reasons for the HCI to be an important aspect of software designing.

- Users are busy in their work

Most of the time the users of a system will be busy with their activities. So the system should have a User interface which can minimize the work done by the user & providing user friendly operations.

- Majority not Computer Experts

The majority of the users are not experts or professionals in computer science. For them the software is not a wonder and if the HCI design is not up to the standard it can take the interest upon the system away from the user. So the designs should be pleasing to use & should help any type of user even a beginner.

- Sceptical about the advantages of the system

After a software implementation most users feel like the software is been thrown at them. That's because until they realize the value of the system they may have doubts about the functionality and the benefits they can gain from the system. Due to this the UI design should reflect ease of use, minimum number of operations, user friendly help & etc.

¹⁴ http://en.wikipedia.org/wiki/Human-computer_interaction

5.8.3 Guidelines for Developing an Effective HCI

According to the published literature when developing an effective and efficient Human Computer Interaction we must consider the following factors. These factors have been proven by experts in the commercial software development domain.

- Distinguished HCI & Application Logic

When developing the HCI, the components that deal with the HCI should always be separated from the actual business or the application logic behind the main functionality of the system. This separation will allow the developers to modify either aspect without affecting each other. This means that the changes done in the HCI will not affect the business logic or vice versa.

- HCI Development along with Analysis & Designing

It is recommended that the components regarding the HCI and their prototypes should be developed at the same time that the other analysis & designing models are developed. By doing this the changes in the analysis or design classes or problem domain can be made available in the HCI designs too.

- Developing the HCI using a Higher Level or CASE tool

It is always recommended to use a higher level programming language or a specialized CASE tool for developing the HCI components rather than using a low level language like C, C++ or Java. Changing HCI components developed using low level programming language will be a tedious task & time consuming. Also those HCI components will resist to inevitable changes in the future.

- Provisions for Future Trends

Always when developing the HCI components it is necessary to have the knowledge about the future trends in software development & interface designing. These future trends contain operating systems, interface rendering systems within the operating systems. Windows Presentation Foundation (UI technology) which comes with Windows Vista is a good example for future UI trends.

5.8.4 Issues to be considered in HCI Designing

According to the published literature following factors should be considered when designing & developing the HCI for a system.

- Characteristics of the Prospective Users

When designing the HCI the developer should first consider the characteristics of the intended users. These characteristics include the experience level of the users as well their thinking strategies. These factors are crucial because different users are with different levels of experience & have different ways of thinking.

- Dialogue types to be used

When communicating with the users having an effective dialogue type is always a must. Currently the Keyboard, Mouse, Trackball, GUI's is used for the dialogue. But future trends are for Voice recognition, image recognition & touch responsive based dialogue types

- Data Presentation Formats to be used

Another aspect is the data presentation format to be used when communicating the information with the users. Basically for data presentation tabular formats as well as graphical formats like Charts, Graphs and even animations can be used.

5.8.5 Characteristics of a Good HCI Design

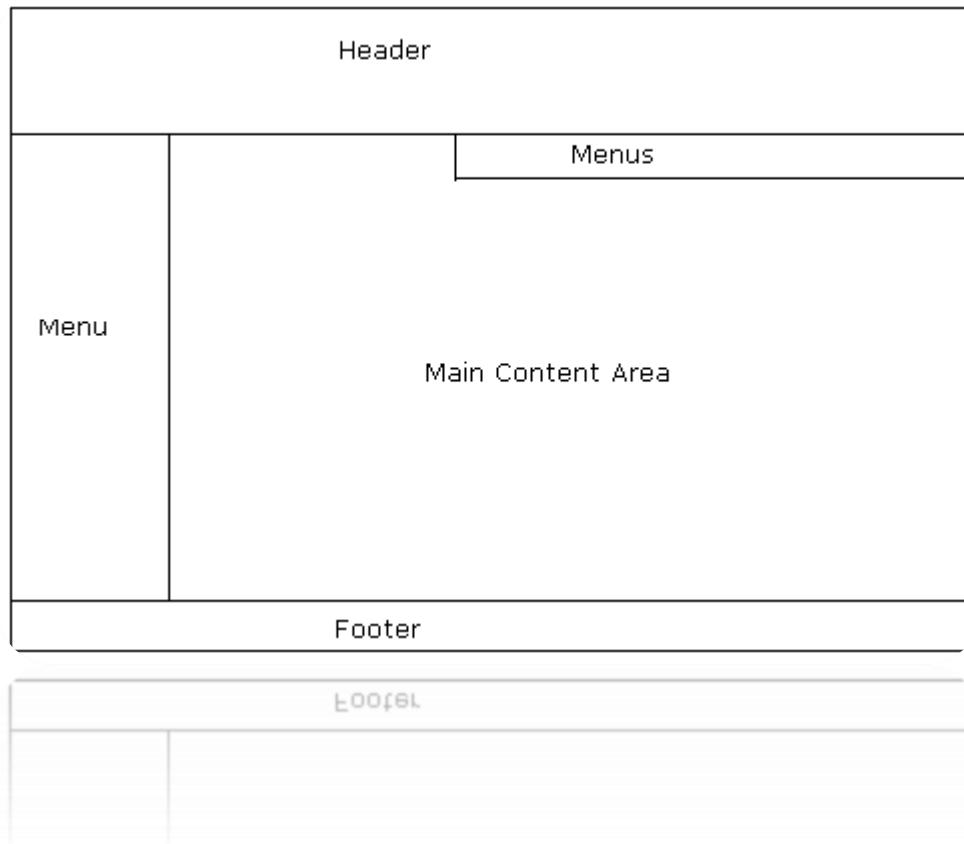
- Consistency – The UIs should be consistent throughout the application and due to that the user can learn the system very easily with less time. The menus used & the operation should be consistent so that the user can use the same instructions to perform the same activity in different sections.

- Appropriate User Support – When the user has got stuck with the operations or has made an error the system should prompt necessary help & error messages and suggestions. Here tooltips can be used to display small popup messages regarding the components displayed on the interfaces.

- Adequate Feedback from the system – Always it's recommended that the system should respond to user actions whenever they perform something. Feedback is essential because it will signify that the system is in contact with the user thus avoids unnecessary actions by user.
- Minimal User Input – Users are resistant to unnecessary key presses and mouse clicks. It is recommended to minimize the inputs given by the user hence it will minimize any potential error that can appear at run time. This can be done by:
 - Selecting a value from a list rather than typing the most obvious
 - Editing incorrect values rather than asking the user to re-enter
 - Using default values for fields where necessary
 - Deriving the information automatically rather than asking the user to enter them.

5.8.6 Screen Layout for CCMS

For the screen designing of the CCMS the following template has been used. The main menu will be on the left.



-Figure 11- (Screen Layout for CCMS)

Chapter 06

Construction Phase

6. The Construction Phase

6.1 Overview

The construction phase actually involves with the real development of the system based on the findings of the inception phase and the designs developed in the elaboration phase. The main concentration will be on the software development and this will be done through a series of iterations, each resulting a release of the system. Along with the development the construction phase also focuses on the testing & implementation rapidly as testing the release are crucial for the transition phase.

6.2 Development Tools & Technologies

Adobe Dreamweaver CS3 (Development)

Adobe Dreamweaver is a web design and development application now owned by Adobe Systems. Dreamweaver is available for Windows operating systems & others. Recent versions have incorporated support for web technologies such as CSS, JavaScript, and various server-side scripting languages and frameworks including PHP, Cold Fusion and Java Server Pages. The main advantage of Dreamweaver is within the framework we can be able to build the interface in graphical mode which simplifies the most tedious tasks. This is the one of key point I used this without much using Zend Studio.

PHP 5.2.6 (Language)

PHP is a scripting language, originally designed for producing dynamic web pages. It has evolved to include a command line interface capability and can be used in standalone graphical applications. PHP is a widely-used general-purpose scripting language that is especially suited for web development and can be embedded into HTML. It generally runs on a web server, taking PHP code as its input and creating web pages as output. It can be deployed on most web servers and on almost every operating system and platform free of charge (This is an Open-Source product).

MYSQL Server 5.0 (Database)

MySQL is an Open Source relational database management system (RDBMS). The MySQL database has become the world's most popular open source database because of its consistent fast performance, high reliability and ease of use.

AJAX

Ajax (asynchronous JavaScript and XML), or AJAX, is a group of interrelated web development techniques used for creating interactive web applications or rich Internet

applications. With Ajax, web applications can retrieve data from the server asynchronously in the background without interfering with the display and behaviour of the existing page. Data is retrieved using the XMLHttpRequest object or through the use of Remote Scripting in browsers that do not support it. Despite the name, the use of JavaScript, XML, or its asynchronous use is not required

WML

Wireless Mark-up Language, based on XML, is a mark-up language intended for devices that implement the Wireless Application Protocol (WAP) specification, such as mobile phones, and preceded the use of other mark-up languages now used with WAP, such as XHTML and even standard HTML

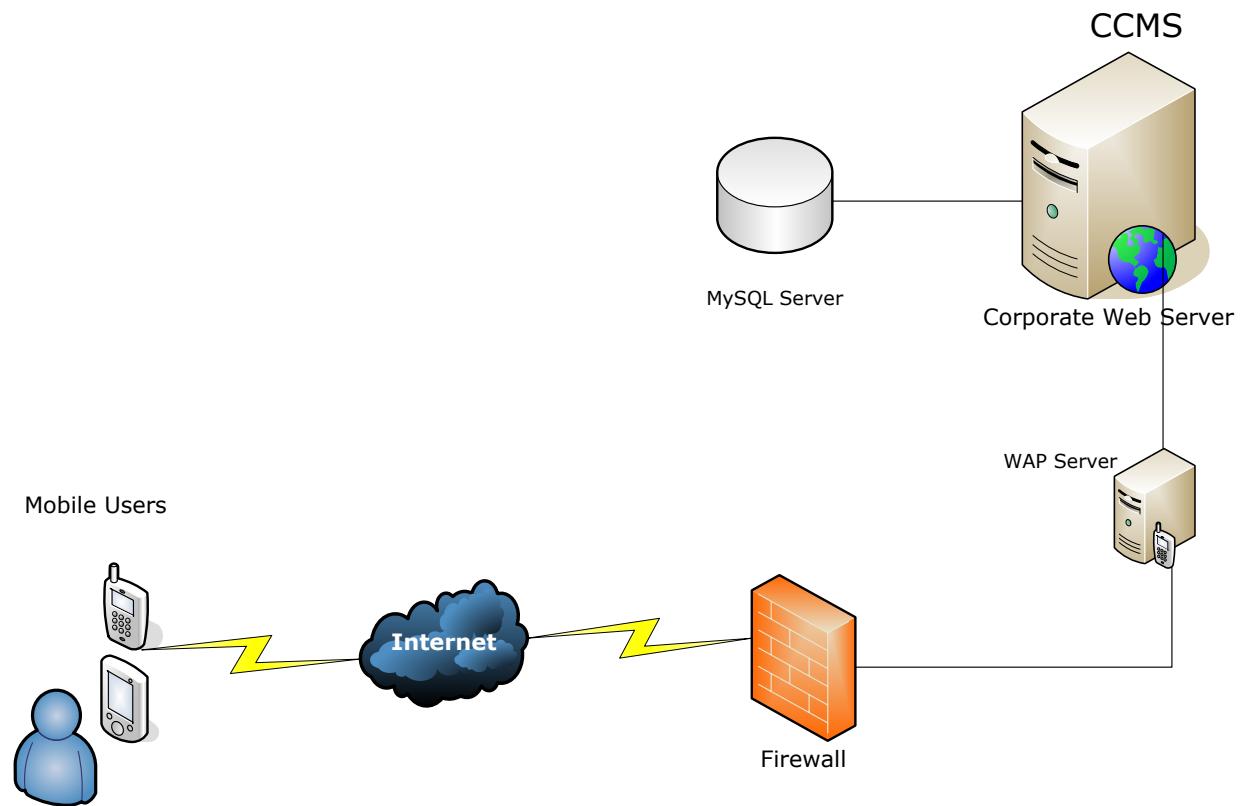
6.3 WAP Communication

6.3.1 WAP Gateway Overview

A WAP gateway sits between mobile devices using the WAP protocol and the World Wide Web, passing pages from one to the other much like a proxy. This translates pages into a form suitable for the mobiles, for instance using the WML. This process is hidden from the phone, so it may access the page in the same way as a browser accesses HTML, using a URL, provided the mobile phone operator has not specifically prevented this.

6.3.2 WAP Architecture in CCMS

The following architecture has been used for the Court Case Management System WAP communication aspects. The internal staffs and external lawyer can take the advantage of this facility and they can view reports and status while travelling. In CCMS WAP gateway implement through Kannel and which is the one of best open source product in market. As well as it provide built in SMS gateway facilities too.

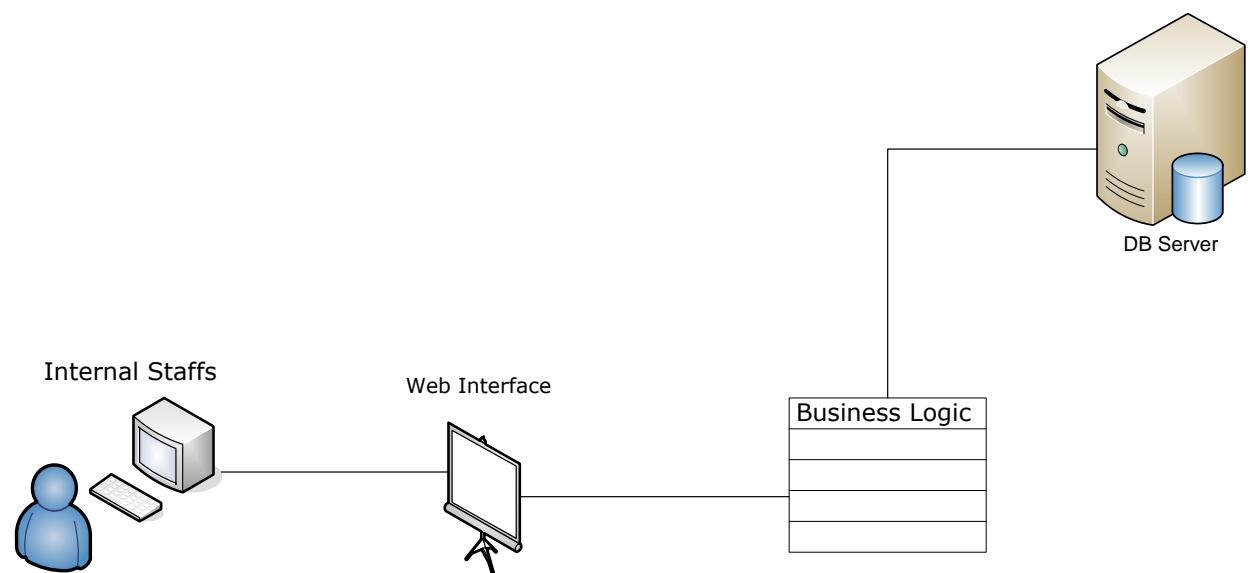


-Figure 12- (WAP Architecture in CCMS)

6.4 The Development of the CCMS

6.4.1 The Development Approach

In order to develop the Court Case Management System the Object Oriented approach has been used where all the business logic resides in a library of classes. The interface components are with the web pages where it constantly communicates with the class library to retrieve the required data & to perform the necessary operations. The diagram below shows the development architecture used for the development of the CCMS.



-Figure 13- (System Development of the CCMS)

REFER **APPENDIX – F** FOR SYSTEM CODING

REFER **APPENDIX – G** FOR WAP MODULE CODING

6.5 Web Form Designs for CCMS

With reference to the HCI in the Elaboration phase following are some of the screen designs used in the development of the Court Case Management System.

Web Form Design for Case Registration

-Figure 14- (Web Form Design for Case Registration)

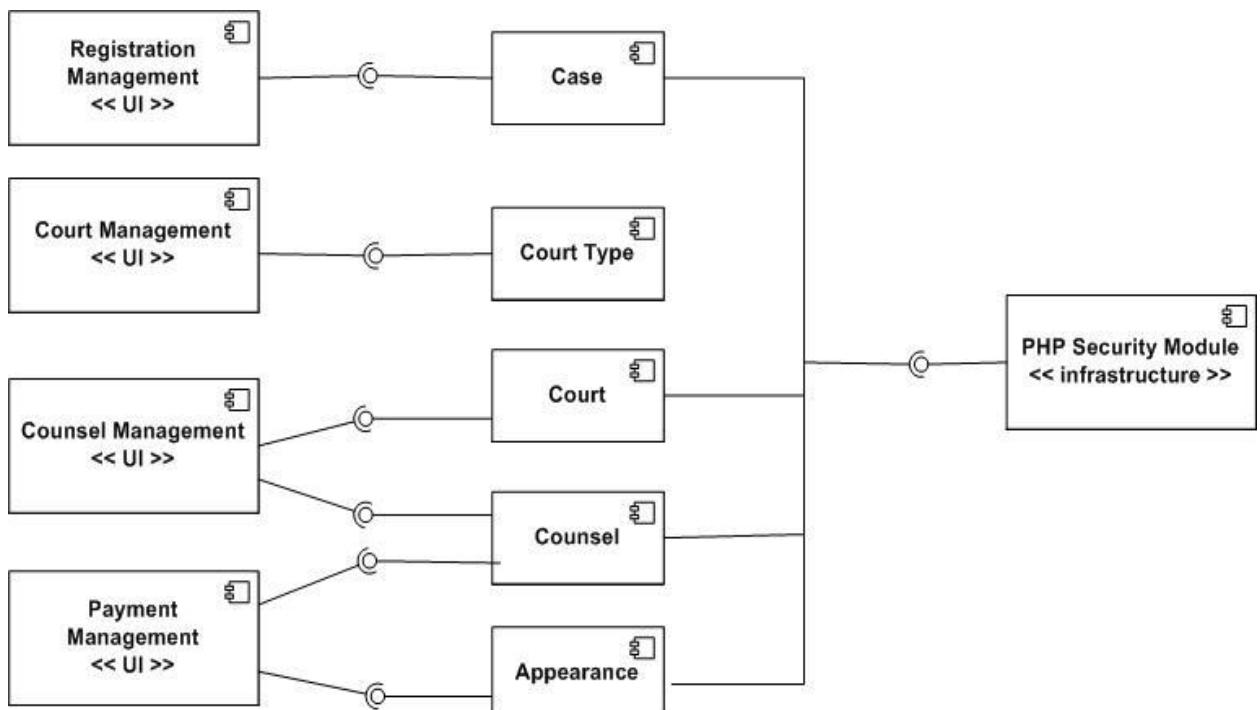
REFER APPENDIX – H FOR CCMS WEB PAGES DESIGNS

6.6 Components Used in Development

Component Diagram

In a large development project there can be a huge number of different physical developed files when the system is implemented. The component diagram is kind of an Implementation diagram where the physical components & their dependencies are shown. These components can be physical module files, other resources or reusable components.

Component Diagram for the Court Case Management System



-Figure 15- (Component Diagram for the CCMS)

REFER APPENDIX – C SECTION (E) FOR COMPONENT DIAGRAM NOTATION

6.7 Software Testing & Quality Assurance

6.7.1 Overview¹⁵

Software Testing is the process of ensuring the accuracy, completeness, quality & the security aspects of developed software. Testing is a technical investigation process which is intended to reveal the quality related information. Practically the testing process is carried out by physically executing the software and checking all the sections according to a plan. In this software execution process the tester will look for any errors that might appear. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design, and code generation.

Software Quality Assurance is the process of ensuring the quality of the software product. SQA covers the entire software development, code reviews, source code, configuration management, source code control, change management and software testing.

6.7.2 Types of Testing

Unit Testing – Unit testing is the process used to validate the accuracy of the individual units of source coding. A Unit is smallest testable part of the source coding. In a procedural language the unit will always be a procedure, function or a small program. A class or a method of a class can be taken as the unit in object oriented programming.

Integration Testing – Integration testing is the process of testing the combination of all the individual software modules and they are tested as a group. Integration takes the unit tested modules as inputs, group them into larger groups & then test them as a whole according to a test plan.

System Testing – The purpose of this test is to evaluate whether the system is compliant with its specified requirements. The system testing comes within the scope of black box testing and does not require the inner coding design or logic. System testing targets on testing the integrated units and how they work together as a group. Mainly in this type of testing the following aspects are checked.

- Performance & Compatibility Testing
- Volume & Stress Testing
- Reliability & Recovery Testing
- User Interface & Usability Testing
- Installation & Maintenance Testing

¹⁵ <http://www.aptest.com/resources.html>
http://en.wikipedia.org/wiki/Software_testing

System Integration Testing – The System Integration Testing involved in testing the combined source coded programs that has been developed to complete part of a system. The main aims are to fix any variance that occurs compared to the specification. Mainly the system integration process depends on a formal methodology.

Acceptance Testing – Acceptance testing refers to the process done prior to the accepting the transfer of ownership of the developed system by the customer. This is also a kind of black box testing where only the functional requirements are tested.

6.7.3 Testing Techniques

Black Box Testing – This considers the outside view of a program module. It checks whether the module is satisfying the specification of it. It is concerned with input values & output values but not the internal operations of the module.

White Box Testing – In White Box testing the internal operations of a module is studies. It considers the coverage of the module. That is almost all sections of the module it tried to be covered by the test cases. In covering all the internal details or the execution of a module each statement, iterations & selection are considered.

6.7.4 Software Validation & Verification

Software Verification and Validation (V&V) is the process of ensuring that software which was developed or changed satisfies the functional and other requirements (validation) and each phase in the development process of building the software yields the right artefacts (verification).

Validation & Verification is intended to be a technical and systematic evaluation of software and allied products of the development and maintenance processes. Code Reviews, Walkthroughs and tests are done at the end of each phase of the development process to ensure software requirements are complete and testable and that design, code, documentation, and data satisfy those requirements.

6.7.5 Testing Approach used in CCMS

In the process of ensuring the quality & accuracy of the Court Case Management System several components were tested. All the testing was performed in a controlled environment and used the VMWare Workstation software package for that. For the testing Unit testing was performed on the following Modules:

- Case Registration Module
- Document Upload (Scanning Documents) Module
- Cost Analysis Report Module
- Payments Module
- Payments Progress Module (By GPRS enable mobile phones)

REFER **APPENDIX – I** FOR TEST CASE DESIGNS & TEST RESULTS

Chapter 07

Transition Phase

7. The Transition Phase

7.1 Overview

This is the final phase in the USDP life cycle. In this phase the main intention is to actually deploy the System in the target user environment. Also in this phase before the transfer of the system there will be some checking done against the original requirements specifications & use cases and further refinements may occur. Following are the objectives for this phase.

- Deployment of the system in the target user environment
- User Training
- Preparing the documentations & user guides
- Preparing Installation Guides

7.2 Installing the Court Case Management System

Installing the court case management system is a very easy process & takes only a less time. When installing there are prerequisite software components to be installed in order to deploy the CCMS successfully. Following are the prerequisites required.

- Apache Web Services 2.2.0 or above

The Apache is the web server component which is used to host the CCMS. Therefore to execute the CCMS the Apache should be installed before the installation.

- PHP 5.1.2 or above

This is the main application development language used in the CCMS to cater the required facilities.

- MYSQL Server 5.0 or above

This is the main database server used in the CCMS as the repository for business information.

So when we are going to deploy CCMS with Red Hat Enterprise Linux (RHEL) 5 those above requirements are already comes with the operating system. So we have to just configure it according to our requirements. These are the advantage of installing the system with open source environment.

Or else we can install CCMS in windows base environment. If so we have to install the above software separately and configure or we can use the “apache2triad” or “XAMPP” package on windows base environment.

7.3 User Training & Support for CCMS

Since the CCMS is totally new system user training and continuous support should be provided to the users. The users will be trained division by division on different time scales according to the sub system they belong to. Mainly there are six divisions affected by the training.

- Legal Department Staffs
- Lawyers
- Branches Staffs
- Legal Administration Division
- Management & other Staffs
- Network & System Administration

The first four divisions will be trained on how to use the CCMS. The Legal Staffs & the Lawyers will be trained in parallel to the others. For the Network & System Administration, a person who is already fluent in that area will be assigned & chosen from the existing staff. The users will be trained according to the following training schedule.

The Training program will be started as soon as the System Prototype is in place & implemented so that after publishing the final system the users will be fluent in their working areas.

For Lawyers special permissions will be granted to use the WAP system of the CCMS and also department's staffs can use these facilities when ever they want.

7.4 Method of Deployment

There are four main strategies for switching over to the new system.

- Direct Changeover
- Parallel Running
- Phased Changeover
- Pilot Project

Direct Changeover

In this method on an agreed date the users of the system will stop using the current system and will start using the new system. This direct changeover is usually scheduled to happen on weekends by keeping some time for data conversion and implementation of the new system. Some preparatory work will be carried out in advance & this does not mean that everything is going to happen in a couple of days. This method is very much suitable for small scale system where there is low risk of failures in implementation. The main advantages will be;

- Organization gets immediate benefits because the system is implemented
- Users will get no time for undermining the new system by using the old system
- This is the simplest method which requires minimal planning

But the main disadvantages would be;

- No fallback procedures to cope up with problems
- Contingency plans are required to address exceptional situations

Parallel Running

In this method the users will continue to work with the current system along with the new system. Parallel running should be used in situations where there is a high level of risk associated with the project the system is crucial for the main business activities.

The main advantages would be;

- Always there is a fallback if problems occur in the new system
- Outputs & performances of the two systems could be compared

The disadvantages would be;

- Staffing problems can occur because now the staffs need to work with two separate systems in parallel.
- There will be a higher cost associated with separate activities to compare the outputs & performances
- Users will feel reluctant to pay attention to the new system as it is very much easier for them to work with the more familiar system

Phased Changeover

In this method the system will be introduced to the users in stages. The stages will be decided according the sub systems & their importance to the business. This method is suitable for large systems where the sub systems are not heavily depending on each other. The stages could be implemented on departmental or geographical basis.

The advantages of this method would be;

- Each sub system introduced can get more attention
- Can gain a fast return on investment if the correct sub systems are introduced in the initial stages
- Thorough testing & performance comparisons can be performed as each sub system is introduced.

The main disadvantages would be;

- If there are problems in the earlier stages, dissatisfaction & rumours can spread among the users regarding the later stages
- The business will have to wait for a long time to get the full benefit out of the system as it will take some time to implement all the stages

Pilot Project

This is a variation of the previous method, the Phased Changeover. Usually in this method the complete system will be installed in a particular department or on a particular site. Then the stakeholders & project team can decide on the extensions of the other departments. The first implementation is known as the pilot project. When deciding to extend the other parts mainly the success of the pilot project is taken into account. The pilot project can be used to cover the learning curve & necessary on the fly modifications can be done. This is more suitable for smaller systems which contain packaged sub systems.

7.5 Deploying the CCMS

For this method the Parallel Running has been used. Therefore after the release of the System Prototype the system was implemented on the customer premises & the users were advised to work with the new system while they are working on their current system. Training was provided along with the parallel run according to the previously mentioned training schedule & users were not abandoned with the new system.

Deployment Diagram

Deployment diagrams are the other type of implementation diagram where the physical implementation hardware is shown with their dependencies. These hardware elements represent the real environment component where the system is getting implemented. The components can be hardware devices, locations and can contain one or more sub components too. The hardware elements are known as Nodes.

REFER **APPENDIX – C SECTION (F)** FOR DEPLOYMENT DIAGRAM NOTATION

REFER **APPENDIX – D SECTION (F)** FOR CCMS DEPLOYMENT DIAGRAM

REFER **APPENDIX - J** FOR CCMS PHYSICAL SYSTEM ARCHITECTURE

7.6 Security of the Court Case Management System

7.6.1 System Security

The term System security refers to the actions taken to overcome the security related problem regarding the system. This will basically cover the aspects on how to secure the system from unauthorized access. There are many where we can implement this.

- Physical Security – Security Checkpoints, Security Guard, Bio Scanners
- Network Security – Authenticating the users from the Computer Network
- In-System Security – Providing Role based security within the system

7.6.1.1 Security Measures of CCMS

When developing the CCMS the following security measures were done.

- The users will be identified by a particular user name & a password.
- All users will get assigned a particular Job role.
- These roles will then be attached with the necessary sub systems so that to access a particular sub system a user should be with the specified role for that sub system.
- If the users are inactive after they have logged on to the system for 15 minutes their session will get automatically disconnected. This will prevent from other unauthorized personnel from accessing the system while the legitimate person is away while he/she is logged on.
- According to the role of the logged on user the left main system menu will Active & Inactive the content. That means if the user has less privileges then the sections which the user cannot access will not be shown to that user, hence avoids the curiosity to explore inactive areas.
- The security information will be kept in a separate database other than the database which holds the business data. This reduces the probability of having a SQL Injection attack to gain control over the user accounts.

To provide the above mentioned some features the Apache Web Services security architecture has been used.

7.6.2 Data Security

The term data security refers to the process of ensuring the integrity & accuracy of the business data held. Mainly this particular type of security will be implemented from the database management system along with the specialized tools available within it.

The following countermeasures have been taken on CCMS to protect the data.

- Automated backups – A backup is taken once a day in an incremental manner while taking a full copy on starting of each week
- Integrity Constraints – The tables are bounded by a set of relationships where the data are within the constraints hence reduces the inaccuracies.

Chapter 08

Critical Appraisal

8. Critical Appraisal

8.1 Overview

This project documentation is the final conclusion of a very challenging project which was started some time back and done exclusively for the final project of the B.Sc (Hons) In Computing & Information System held by the London Metropolitan University. This is the final section of this documentation where I as the author of this documentation explain about the actual development work I had to perform and the challenges I faced.

As the Chapter 01 of this documentation explains the total development project was done based on the operations carried out by the legal department staff of Ceylinco Insurance PLC.

While the sub systems reflect the tasks done by the staff at Ceylinco Insurance PLC, some of the features were imaginary at the beginning. But after some researching I performed all my imaginations final became real & essential to the system. The main objective of writing this chapter is to tell the reality behind the scenes of the documentation.

8.2 Challenges Faced

The challenges had in the development of the CCMS are enormous. The challenges I had spread across different aspects of software development. I have listed the challenges I had & how I overcame them.

Basically it was a major challenge how to finish the development of the project during the given time period. Due to lack of experience, it was difficult to estimate the time duration and to meet deadlines and requirements at the same time. There was a trade off between the development and the learning of the tools and programming languages. At the start of the project I was not aware of server side programming languages (PHP, WML), so it was indeed a challenge to learn these languages and at the same time to develop the system. But as time passed by it was a challenge which was achieved quite smoothly.

When you consider the integration of mobile technology to the system, as mentioned earlier it was the biggest challenge out of all. From the beginning, I always wanted to do something new and different on the project. So this feeling ended up by me trying to introduce the mobile technology to the system. At the beginning I knew this would be a quite a challenge where sometimes I thought of leaving out the mobile section out of the project. But I had inner feeling that I could achieve this. So I did a research and some studying on how to access the system (Integration of PHP & WML) from a mobile GPRS enabled phone device.

For the mobile integration to the system, I even got help from a friend who knew how to deal with mobile content. But it was not easy to do so. To demonstrate and to work out WML coding you could download the software, Nokia Mobile Internet Toolkit from the Nokia.com site. But it did not recognize the WML content with PHP integrated. So it always ended up with producing unknown errors. It was indeed frustrating to develop the Mobile section of the project due to fact that it always ended up with errors which I could never figure out the reasons for those errors. I even consulted the nokia.com site for why it was difficult to develop dynamic sites using WML. I also searched programming for CCMS and even published my problem on those for CCMS.

After months of research, on the Integration of PHP & WML, I got to know that Mobile Internet Toolkit software does not support WML with PHP. But the internet browser "Opera" Or WAP Proof 2007 was ideal for the execution of WML coding. So after months of research I finally achieved the integration of mobile technology in the system. It was indeed a challenge on making the integration of mobile technology a reality.

8.3 Lessons Learned

After getting so many challenges & after experimenting on the technologies I have learnt many lessons regarding the aspects of software development. These lessons will be helpful in my future endeavours as a Software Engineer or System Architect. Following is a list of the lessons I have learnt.

The first lesson learnt is applying theories learnt from books in real practice, in real world. There is a big difference between learning something, and applying that particular thing. Coping with such situation is the major thing learnt from this Project.

Project Management is another lesson learnt from this project. Numerous unexpected obstacles were faced during the work. No matter what it was needed to achieve the Deadlines. Managing the work in such situations is another important experience gained from this project.

It was not possible to implement everything as planned with 100% accuracy. In such occasions it was needed to change the plan such a way that would not eliminate the main objectives of the project. This was learnt and applied through out the project.

As a whole how to develop a System in a professional way was learnt from this project.

8.4 Strengths & Weaknesses of CCMS

After completing the development of the CCMS I have found some of the weaknesses & strengths of it. These weaknesses will be much important for the CCMS to be extended to include other business functions as well & the strengths it encompasses have surpassed the weaknesses.

Weaknesses in brief:

➤ Lack of support for Graphical Charts

The CCMS lacks the support for graphical charts & the ability for viewing data in different viewpoints. This can be done using a data warehouse & with some specialized programming.

➤ Potential problems in deployment

To deploy the CCMS we need to have a good server with processing power & primary memory capacity. Because all the branches within the company VPN may access the system in any time. So the load balancing and concurrency control is important.

➤ Lack of on the fly help & support

Since the CCMS is a web based system it has some lacking features for providing comprehensive user help when needed. Compared to a normal system working on windows, web based systems are much more prone to errors & all the things we perform on a windows system cannot be achieved because the web application is executed using a web browser. Eventually the web browser will decide the performance of the web application when it comes to high end web applications.

Strengths in brief:

➤ WAP base communication

One of the greatest strengths I see on this system is the ability communicate with the outside world using the mobile technology. The use WAP has contributed towards the success of the system as it helps to communicate with the system while the users are on the move. So wherever they are they can keep in touch with all the latest information.

➤ Tabbed working environment with a rich user experience

When working with the system the users will get a tabbed environment where all the necessary information is displayed in tab format on one page. This basically reduces the time need to traverse through different pages to get the simplest thing done. Also the users will be working in an AJAX enabled applications where the pages are not refreshed and the users can keep in touch with the system without any delays for waiting.

➤ Provision of latest security mechanisms

The security measures of the CCMS are using the latest technology where it is quite impossible for unauthorized personnel to get in to the system. Since the CCMS information is crucial and confidential I had to concentrate more on the security. The CCMS security is based on roles where each user gets one or more roles which enable them to traverse through the application.

➤ On the fly Reports

Another major strength of CCMS is the provision for generating cost analysis reports. In the manual system to perform this task the staff has to look for all the information on manual documents & then compile a final report. But now everything can be done in just a click of a button.

8.5 Potential Enhancements

I would like to mention about some of the potential improvements that can be done to CCMS in order to maximize the effectiveness to the business

- Payment module can be coded in on line manner so lawyer can get their payment directly to the bank.
- Give the facilities to lawyer to update the case information in real time through WAP communication.
- Add BI module to the system. If so management can able to get on the fly reports.
- Provide the chat base communication module to users when they need any help.

Chapter 09

Conclusion

9. Conclusion

Finally I would like to emphasize more on the advantages we have in web applications when compared to the normal systems. Detailed information on this can be found on the Literature Survey.

The use of Web Application technologies is widespread in all over the world. Now most of the tasks have gone to the internet. Examples would be E-Government, E-Business, E-Education & so on. The use of Mobile telecommunication techniques has contributed towards the success of major business systems. With the emerging new trends now the system has gone wireless where the users can access the system from anywhere they are.

With this development of the CCMS now the efficiency & effectiveness of the work at Ceylinco Insurance PLC have been escalated and it will continue. Since CCMS supports a wide variety of business functions it's easier to maintain one system rather than having to maintain different application for different functions. With the use of web technology the CCMS system will be initially launched as an intranet application and it is fully compatible with the internet too.

Talking about the development for this Project I followed the Concepts behind the Rational Unified Process & used UML as the Modelling Language. PHP was used to develop system and backed by a MYSQL Server database and hosted with Apache web service.

Finally I would like to conclude this appraisal of my system which has been a life time experience compared with other previous projects I have had undertaken.

Reference List :

Books Referred

- Cutts G; Structured Systems Analysis and Design Methodology; Paradigm Publishing, London; 1988
- Date CJ; An Introduction to Database Systems, 7th Edn; Addison Wesley, Wokingham; 2000
- Flynn DJ; Information System Requirements: Determination and Analysis; McGraw-Hill; 1992
- Martyn Mallick; Mobile and Wireless Design Essentials; John Wiley & Sons 2003
- Bill Routt; Wireless Markup Language (WML): Scripting and Programming using WML, cHTML, and xHTML; ALTHOS 2004
- Stephen H. Kan, "Metrics and Models in Software Quality Engineering", Pearson Education, 1995
- Rasmus Lerdorf & Peter MacIntyre, Kevin Tatroe; Programming PHP, 2nd Edition ; O'Reilly April 2006
- Michael Kofler; The Definitive Guide to MySQL 5; Apress 2005
- Peter Lavin; OBJECT-ORIENTED PHP. ; William Pollock 2006
- PHP Fast and Easy development by Julie C. Meloni.
- Professional PHP5 by ED Lecky-Thompson, Heow Eide-Goodman, Steven D. Nowicki, Alec Cove.
- Sams Teach your Self PHP in 24 Hours
- PHP 4.0 Bible by Tim Converse & Joyce Park
- Information Systems Development by Beynon Davies
- Booch, Grady, Rumbaugh, James, Jacobson, Ivar. 1999. The Unified Modeling Language User Guide. Addison Wesley Longman, Inc

URLs Referred

- www.mysql.com
- www.php.com
- www.phpbulder.com
- www.weberdev.com
- www.phpbuilder.com
- www.devshed.com
- www.php.faqs.com
- www.sothink.com
- www.w3schools.com/wap
- www.developer.com
- www.tutorialized.com/tutorials/PHP/WAP-and-WML/1

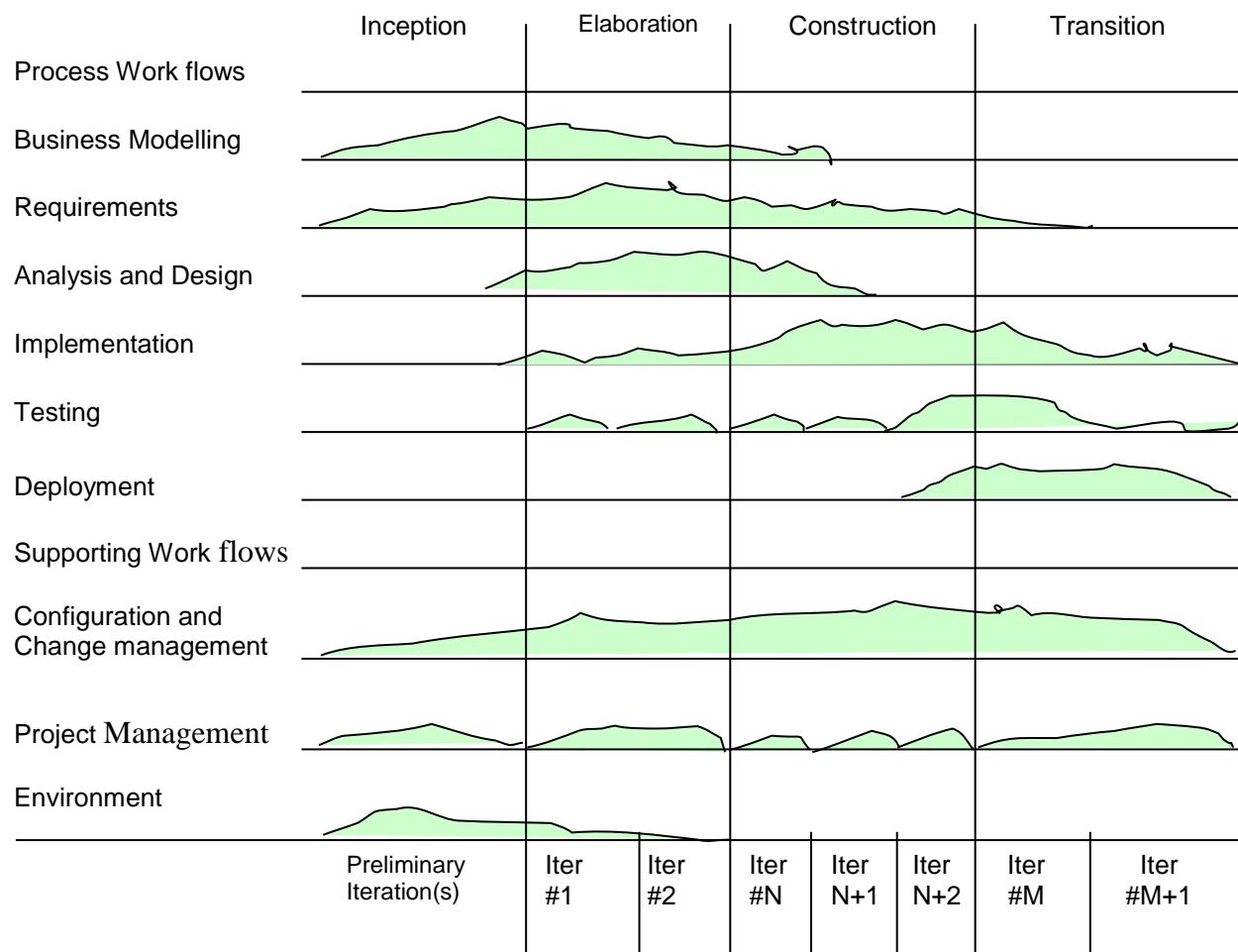
Other Materials Referred

- Management Support Systems Course Material
- Object Oriented Analysis & Designing Course Material
- Advance Database Management Course Material

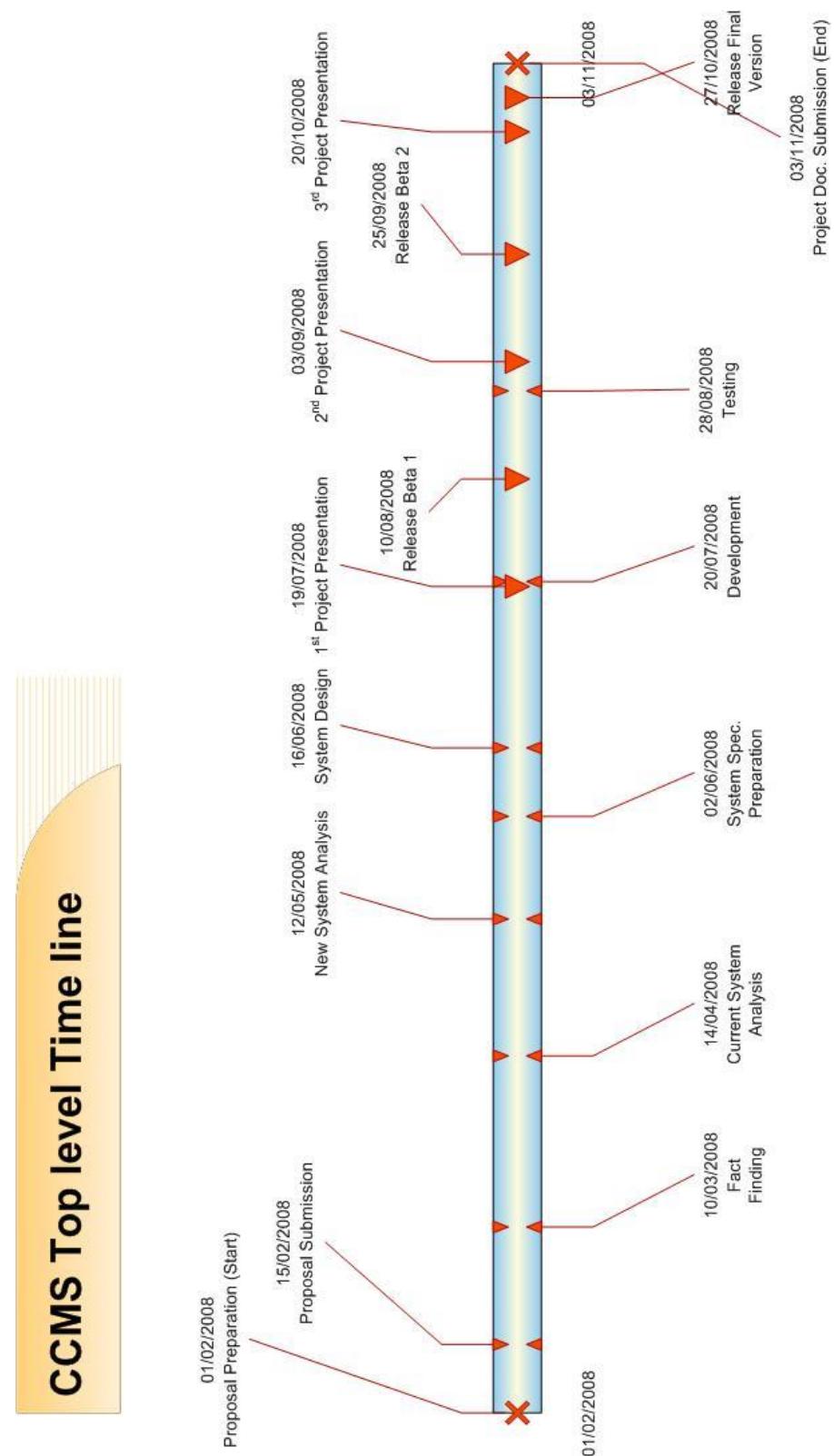
Appendices

	Page
Appendix A – Rational Unified Process Architecture	88
Appendix B – Project Schedule	89
Appendix C – CCMS Diagram Notations	90
Appendix D – CCMS UML Diagrams	96
Appendix E – Database Design & Table Schemas	107
Appendix F – Coding Samples for System	111
Appendix G – Coding Samples for WAP Module	120
Appendix H – CCMS Web Form Design Samples	121
Appendix I – Test Case Design & Test Results	130
Appendix J – CCMS Physical System Architecture	131

Appendix A – Rational Unified Process Architecture



Appendix B – Project Schedule



Appendix C – UML Diagram Notations

Section A – Class Diagram Notation

A class is represented a rectangle with three partitions on it. The three partitions are for the Class name, the list of attributes with their visibilities & data types and finally the list of operations with their return types, arguments & visibilities.

Types of Classes

Below are the different classes that can be included in the Class Diagram.

- Boundary Classes – “These model the interaction between the system and its actors” (Jacobson et al, 1999).
- Entity Classes – “used to model information and associated behaviour of some phenomenon or concept such as an individual, a real life object, or a real life event” (Jacobson et al, 1999)
- Control Classes – “represent coordination , sequencing, transactions and control of other objects” (Jacobson et al, 1999)

Relationships between Classes

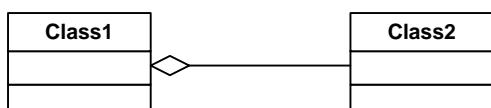
A “relationship” is a term used to denote the specific types of logical connections between classes & their instances

Instance Level Relationships:

- Link – The link relationship is the most basic relationship among all. It can be represented as a line connecting two or more classes. A link is a logical connection between two or more objects and depicts an instance of an ‘Association’.



- Association – An association is the logical connectivity between two or more classes. An association can be named and both ends & can be represented with the multiplicity, visibility, role names & ownership indicators.
- Aggregation – Aggregation describe about “the whole and its parts”. Also known as the “HAS-A” relationship the aggregation depicts the classes are the used to built another in a collective way. The basic representation is done by using a diamond shaped head attached to a line.



- Composition – Composition is another flavour of “HAS-A” relationship where if the container class is destroyed all the contents of it are also destroyed.

Class Level Relationships:

- Generalization – This relationship depicts the “IS-A” relationship where one or more classes are derived from the structure of another class. The derived classes are known as the Sub classes and the class from which the sub classes were derived is known as the Super Class. The super class is known to be a “Generalization” of the sub classes where it contains all the common class features of the sub classes. Generalization is shown as a hollow triangle attached to a line & the triangle will point at the super class.

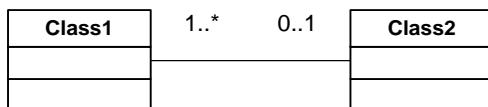


- Realization – Realization is a relationship between two elements of a model where one element realizes the dynamic aspects (behaviour) of the other.

General Relationships

- Dependency – This depicts that a change done on element affects the other and shown by a dashed line.
- Multiplicity – Shows the number of instances from each class which participates in an association or link. Following are the several multiplicities available.

0..1	Zero or maximum of one	1	At least one element
0..*	Zero or many instances	1..*	At least one element or many instances



Class & Member Visibility

The visibility of a class, an attribute or method specified the scope where it is active & where it can be referenced. By using the visibility feature of a class, it can be restricted or open by/for other referencing parties. There are four levels of visibilities.

- Public – Any outside class may access
- Private – Only within the class itself may access
- Protected – Any descendant or the class itself may access
- Package – Any class within the same package may access

Class
-Private
+Public
#Protected
~Package
+Public()
-Private()

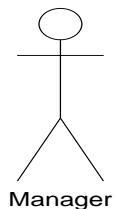
Section B – Use Case Diagram Notation

Use Case



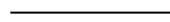
"Use cases describe a set of sequences or actions, including variants that yield an observable result of value to the actor"(Booch et al.,1999).

Actor



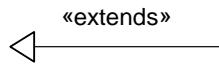
An actor depicts a consistent set of tasks that users of use cases perform when interacting with these use cases. Usually, an actor denotes a task that a person, hardware device, or even another system performs on the system.

Link



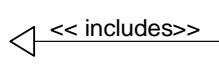
Depicts a link between the actors & the use cases

Extends



Shows the use case of which particular use cases are a specialization.

Includes



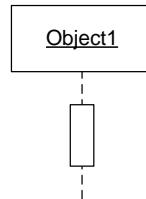
Shows the use cases referenced by a use case.

Section C – Activity Diagram Notation

Initial State		Depicts the initial state of the activity or basically the start of the activity
Final State		Depicts the final state of the activity or the ending of an activity
Activity		Denotes an action been carried out in the diagram. It could be a message call of another object, an expression or a task
Control Flow		Denotes the control flow from one activity to another activity when the previous activity finishes its work
Decision		Shows a decision point where a decision is made in order to choose the execution path.
Concurrent Activities		Activities aren't always sequential. When we need to have concurrent processing the Forking (left) & Joining (right) symbols can be used.

Section D – Sequence Diagram Notation

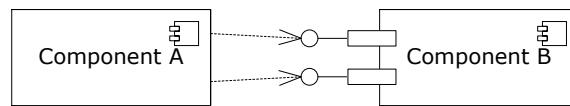
- Objects – The interacting objects are laid out near the top of the diagram and they are arranged in way which simplifies reading
- Lifeline – A lifeline is drawn from the objects that interact and represents that particular object in the interaction
- Activation – Along the lifeline Activation boxes are used to denote the execution of an operation which is carried by the object. The length of the activation box determines the duration of the activation or operation.



- Messages – Messages are used to show the interaction between objects. Basically these messages will be used to invoke another operation from one object to the other. Following are the types of messages available
 - Simple – Transfer the control from one object to the other →
 - Synchronous – Sends a message to another and wait for feedback to continue →
 - Asynchronous – Sends a message to another but doesn't wait to continue →

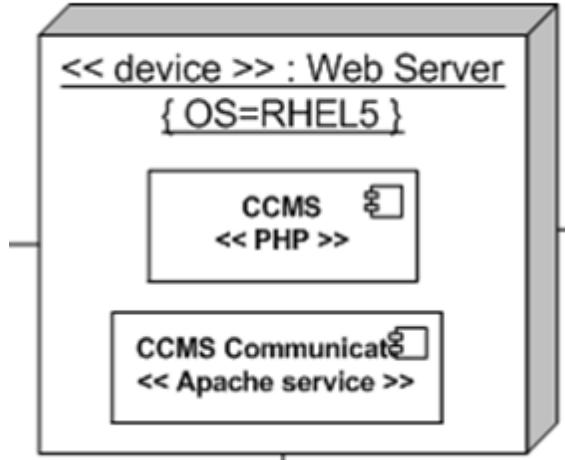
Section E – Component Diagram Notation

- Component – A component is represented by using a rectangle with two more rectangles which stick out of the main rectangle
- Dependency – Showing the relationship between different components a dependency is represented by using a dotted arrow pointing from the dependent to the depending component
- Interfaces – An interface is a collection of methods & attributes provided by a component for other to interact with. This represented by using the lollipop notation



Section F – Deployment Diagram Notation

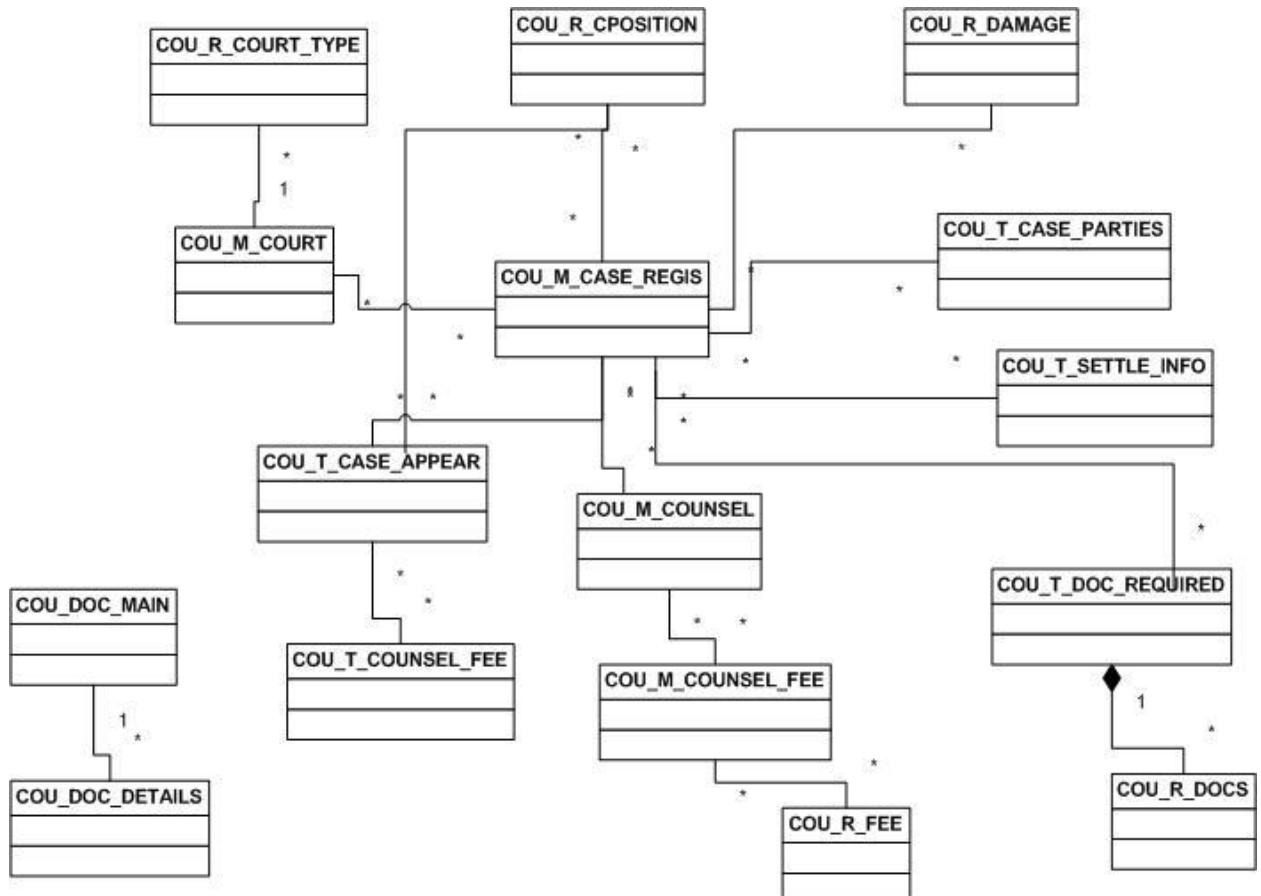
- Component – The components that belongs to a hardware node are shown in the same format as in the Component Diagram
- Node – Represents a piece of hardware in the physically implemented system



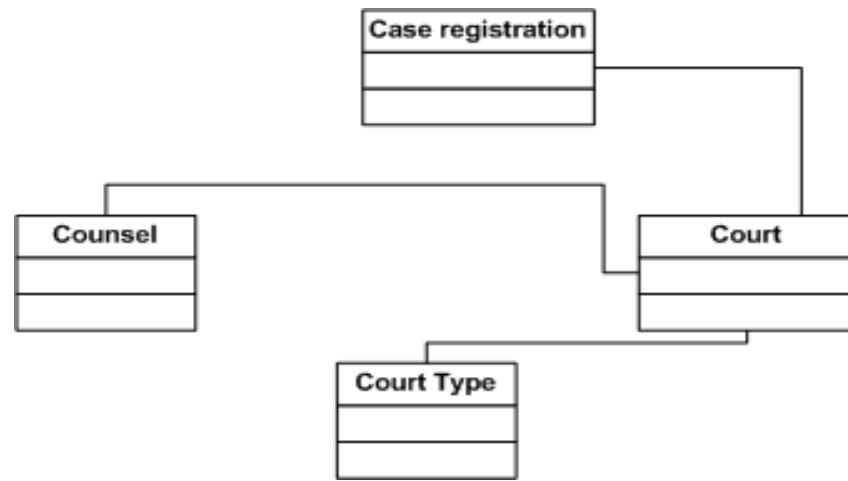
Appendix D – CCMS UML Diagrams

Section A – Class Diagrams

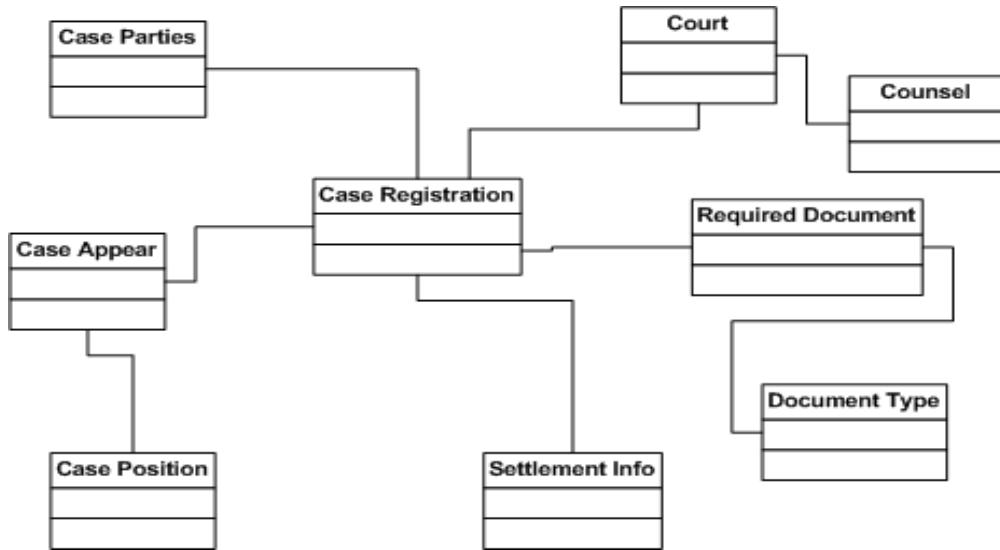
Court Case Management System



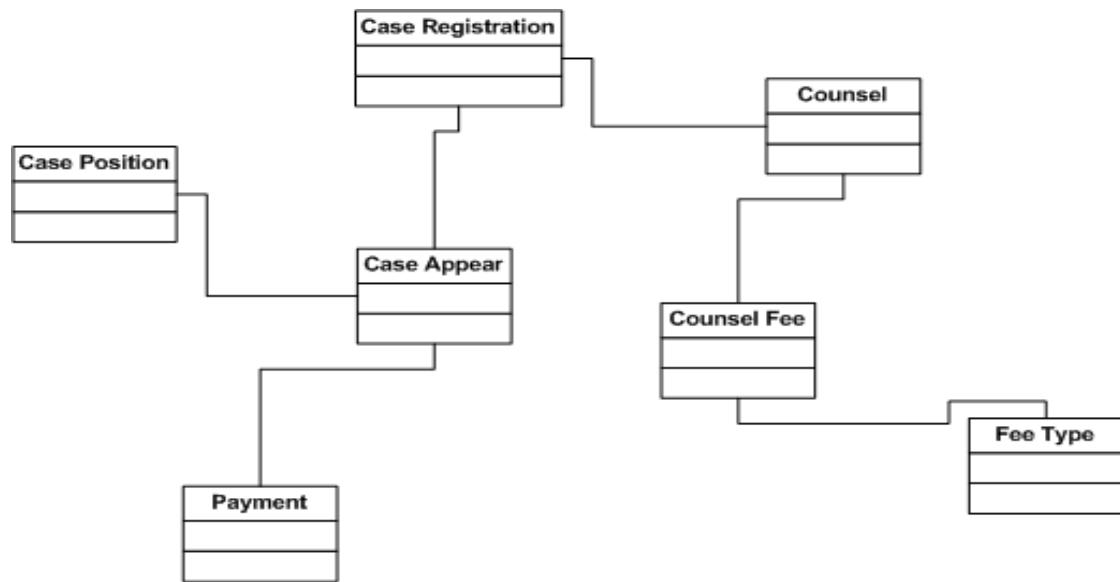
Court Management



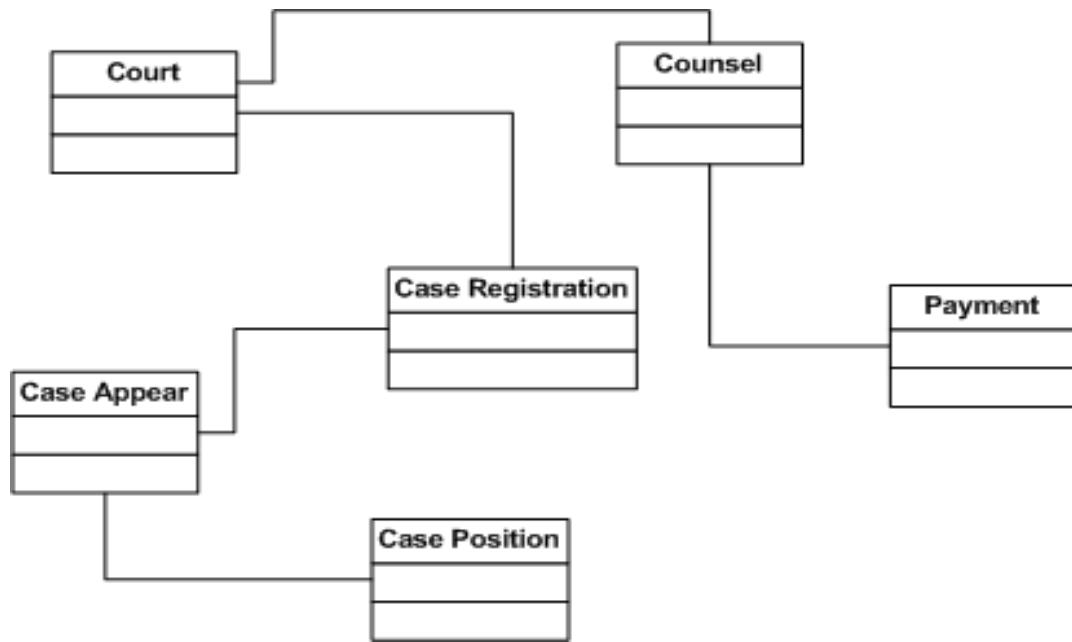
Case Registration Management



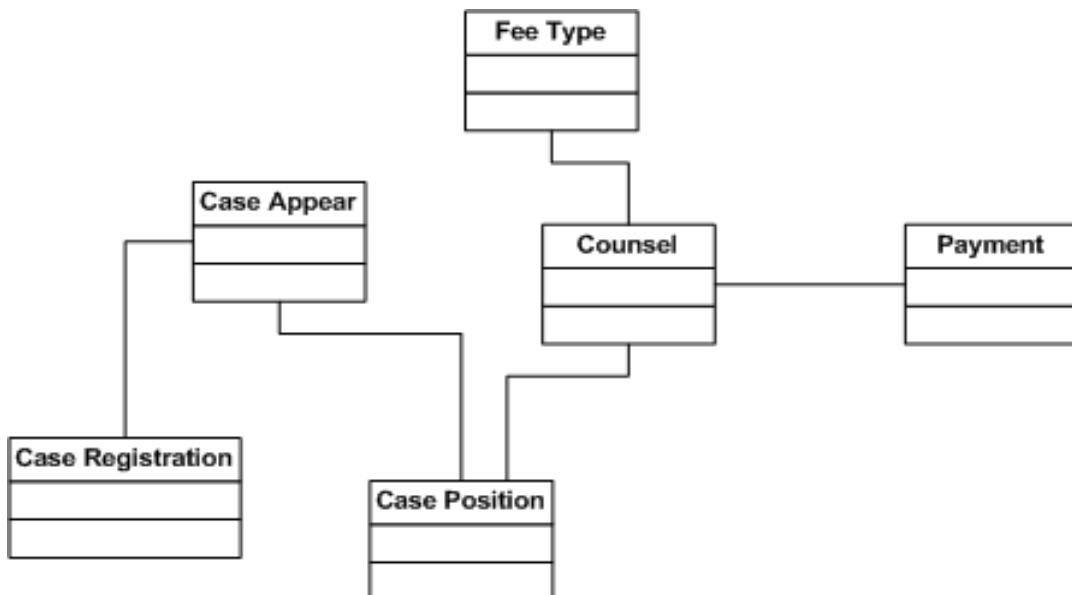
Counsel Management



Report Management

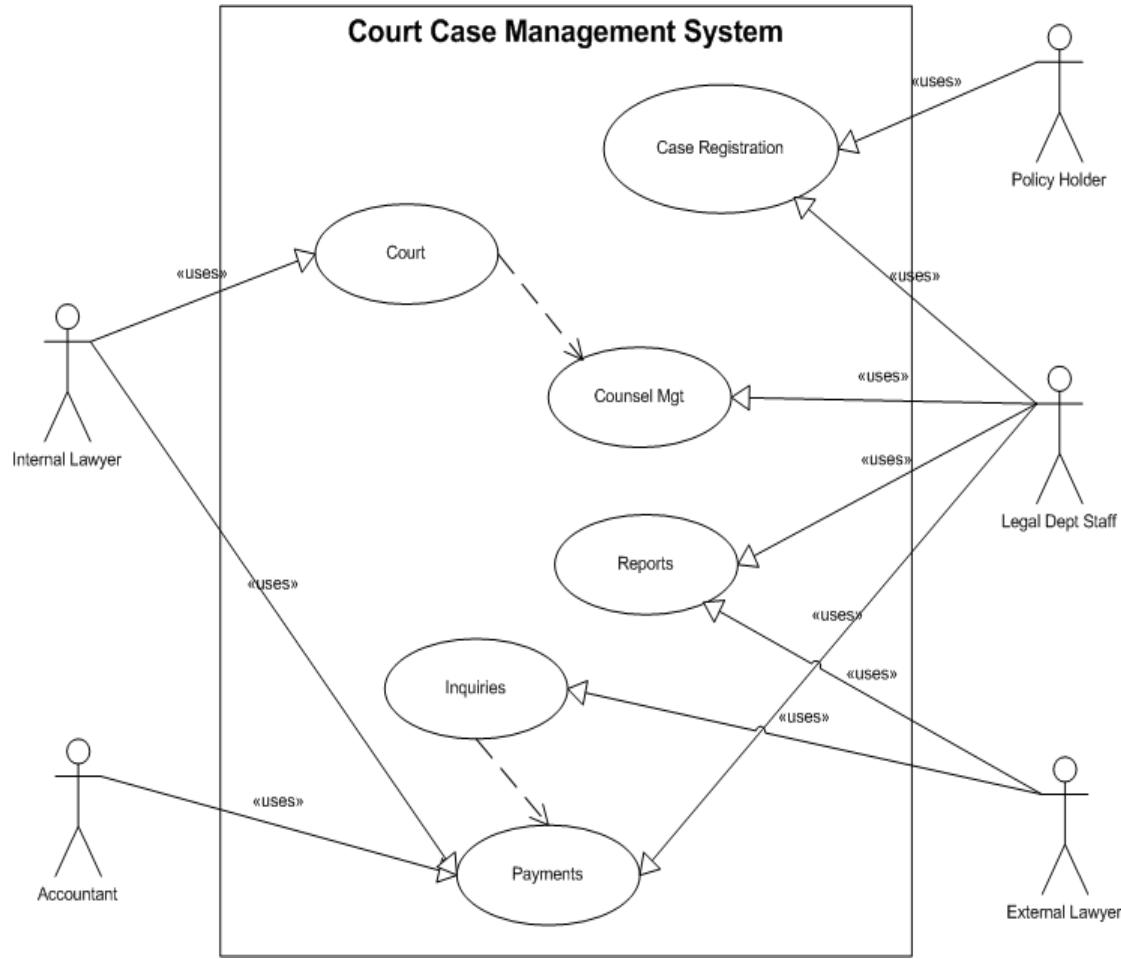


Payment Management

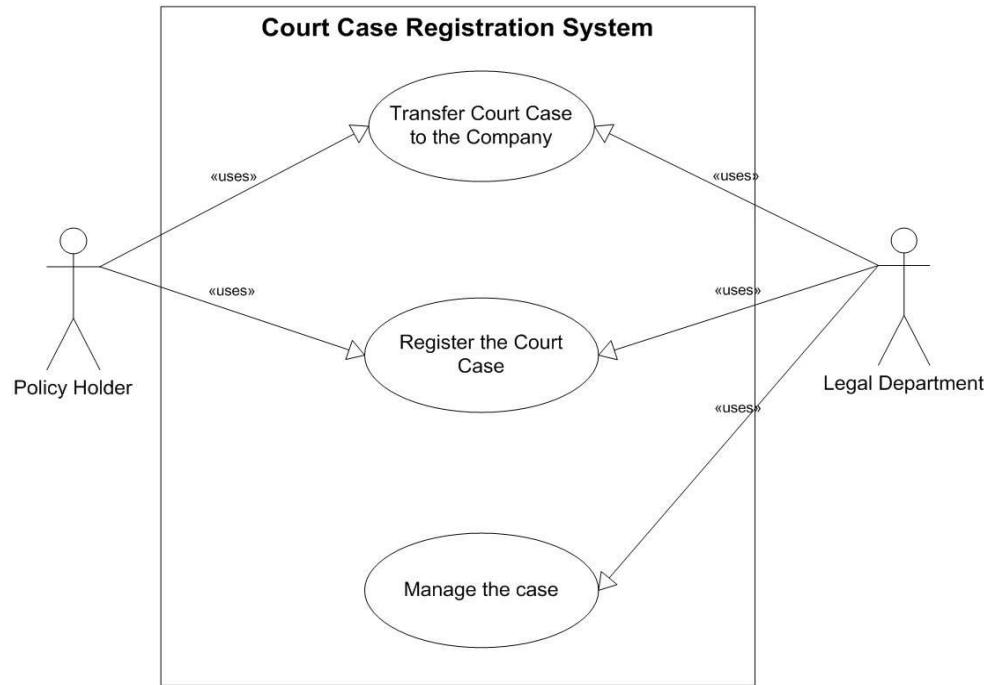


Section B – Use Case Diagrams

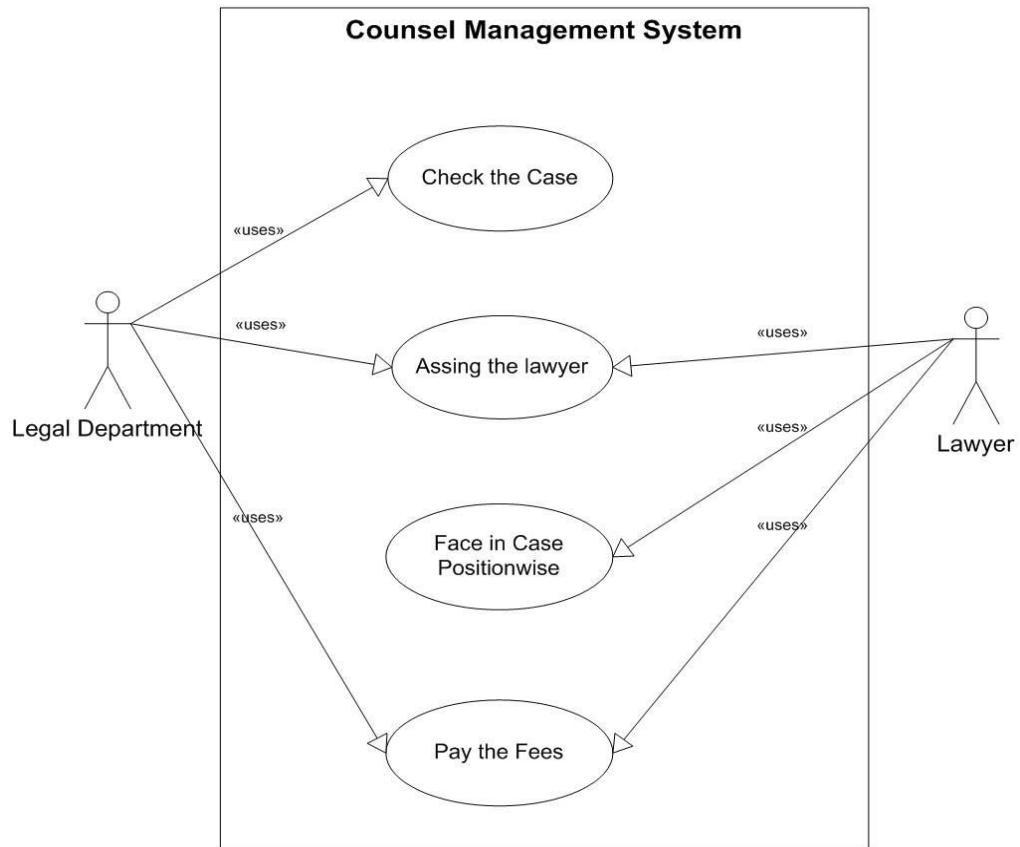
Court Case Management System



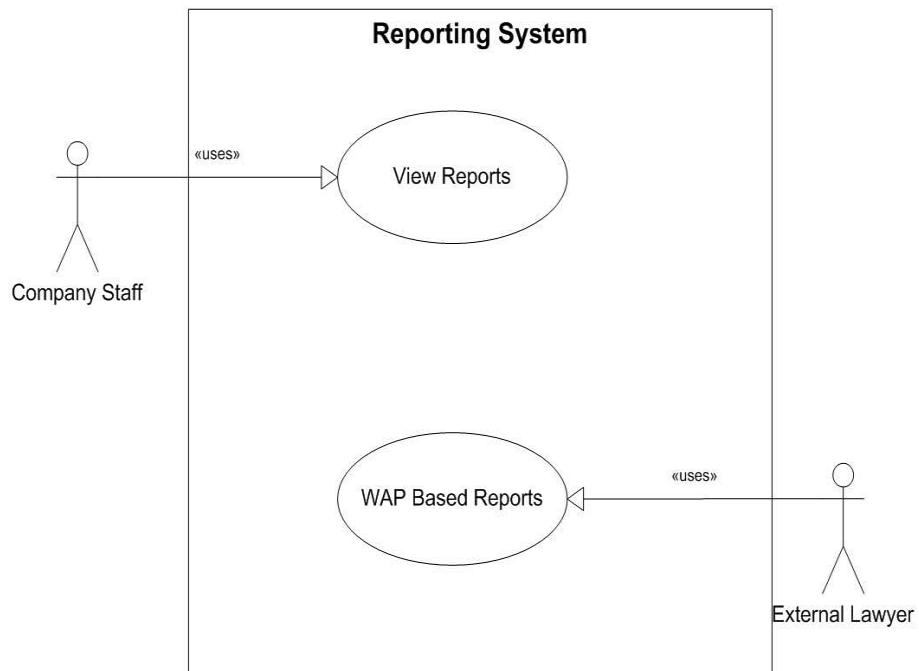
Case Registration Management



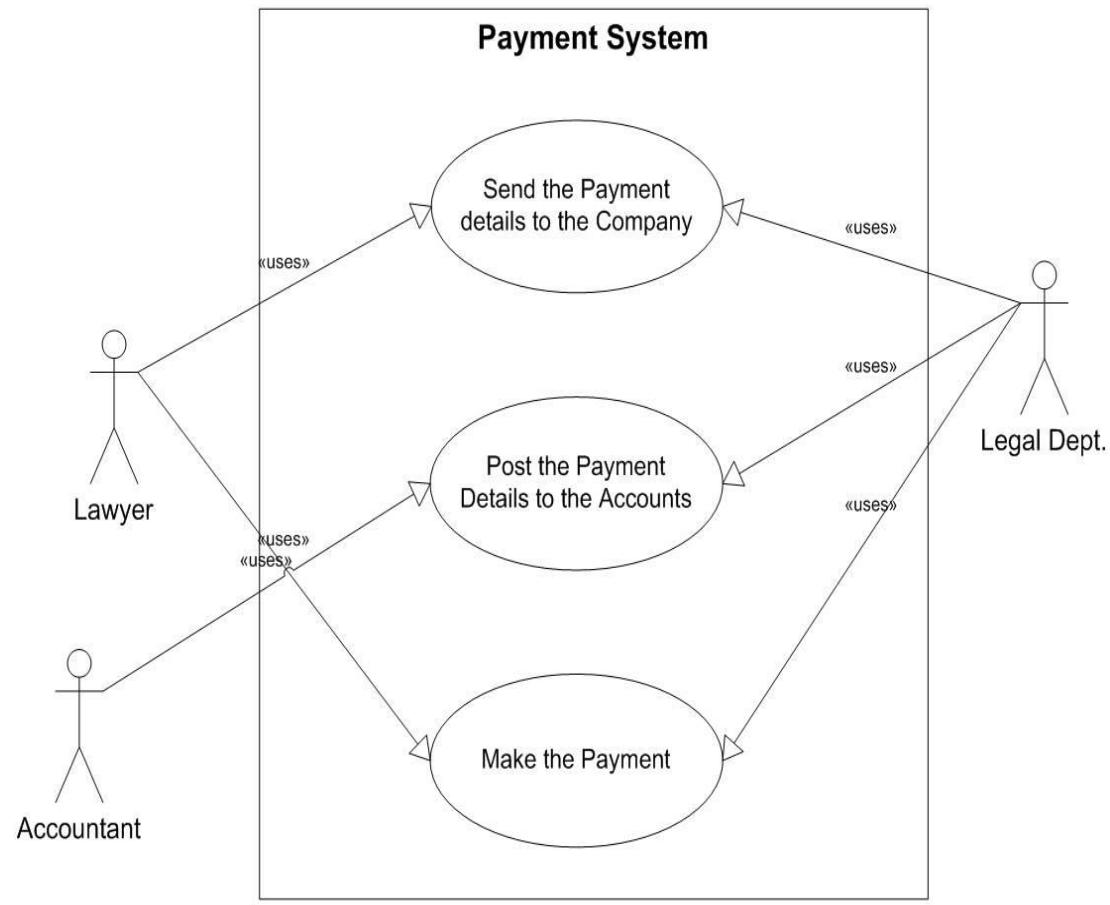
Counsel Management



Report Management

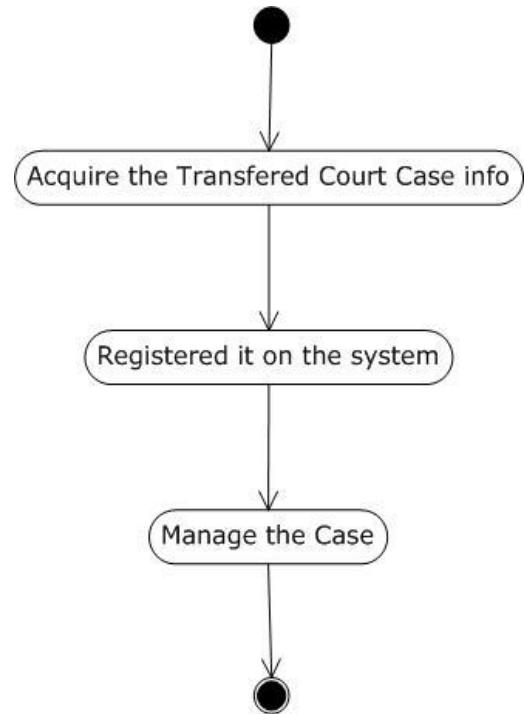


Payment Management

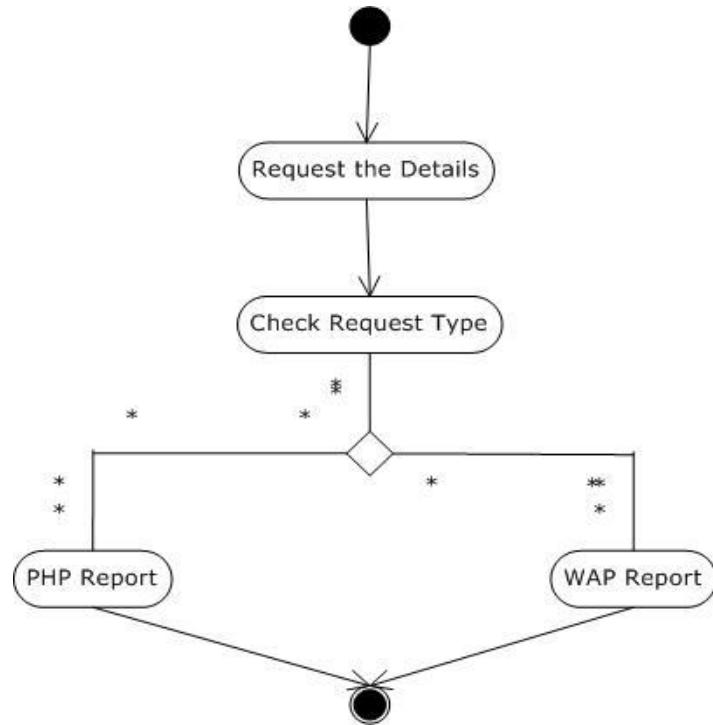


Section C – Activity Diagrams

Case Registration

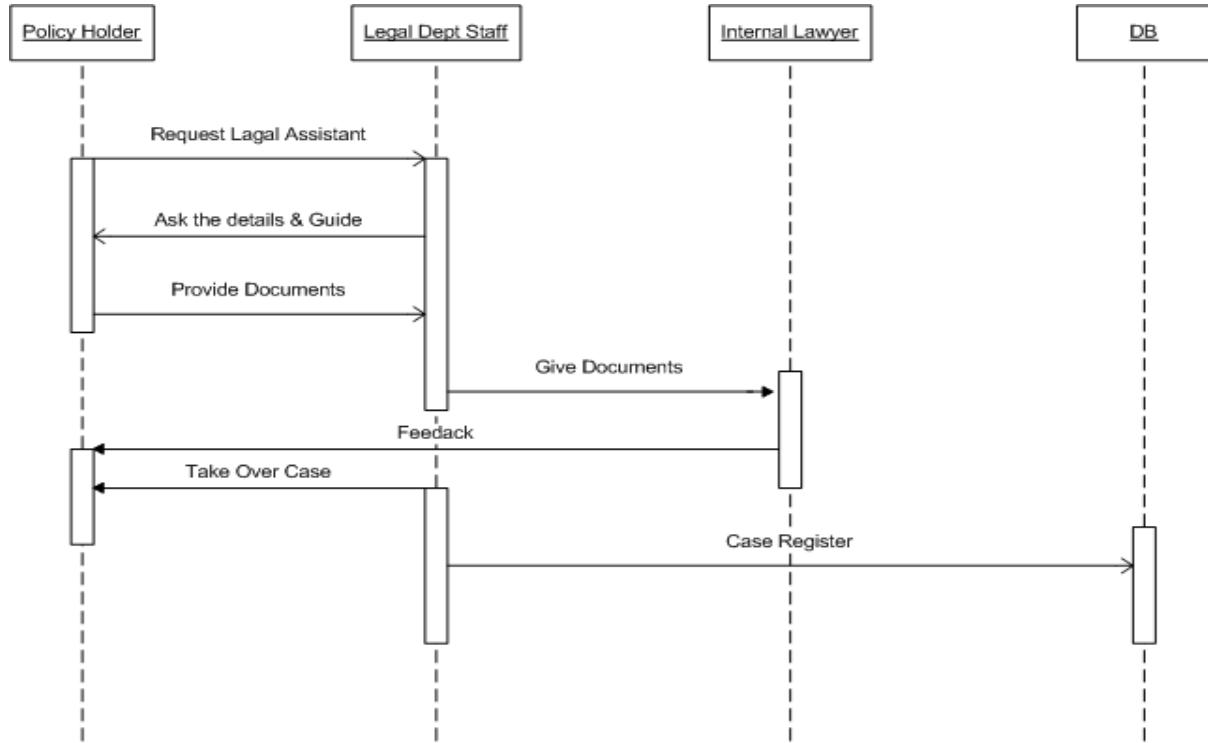


View Reports

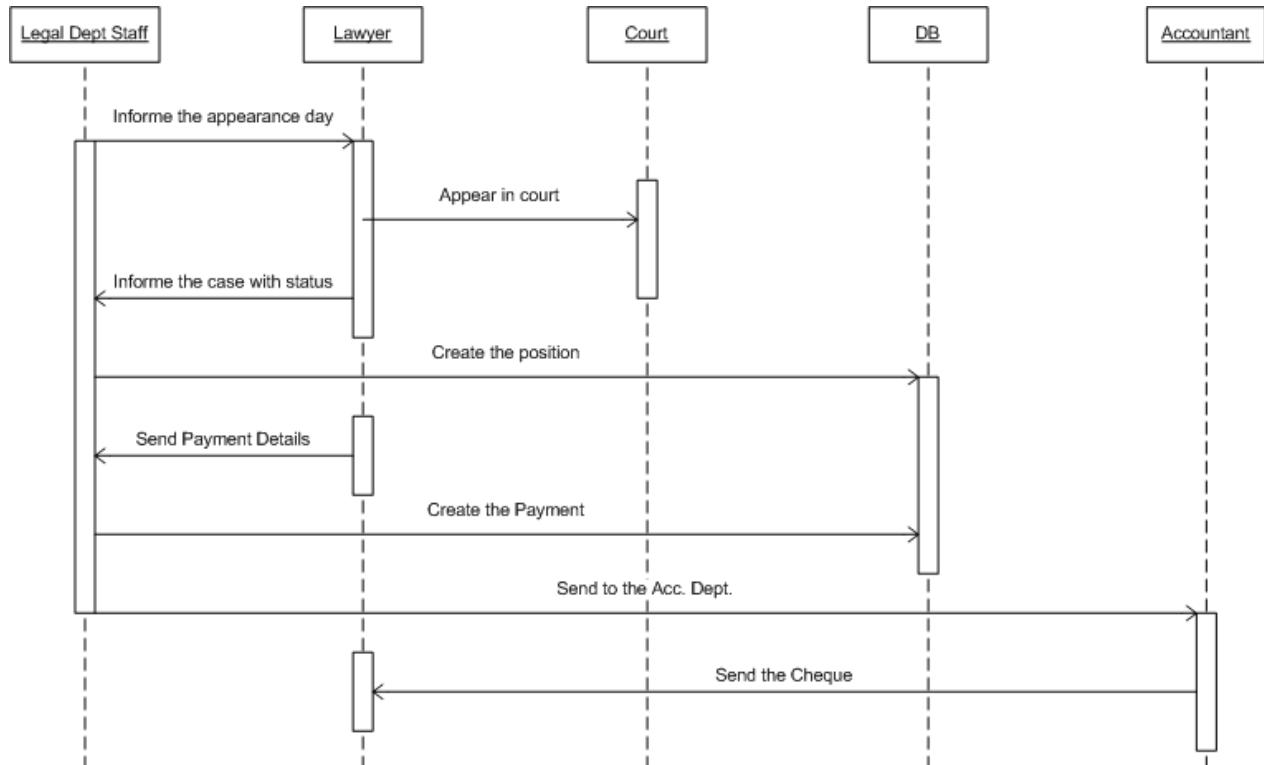


Section D – Sequence Diagrams

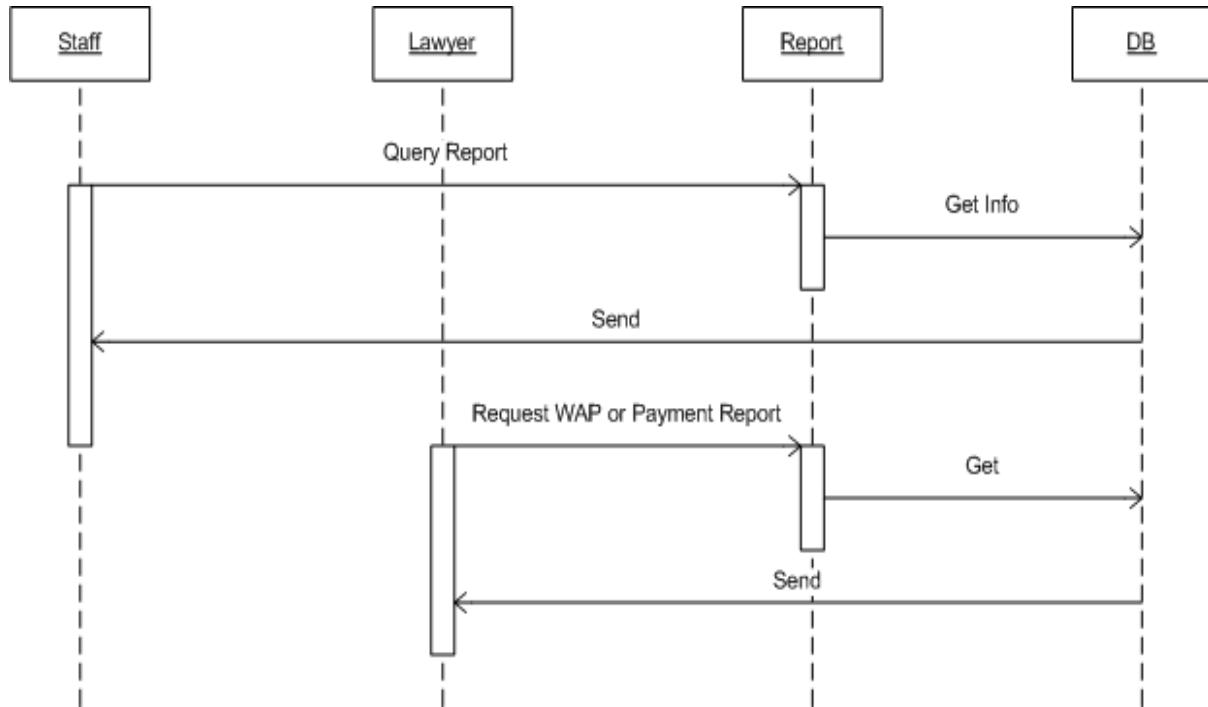
Case Registration



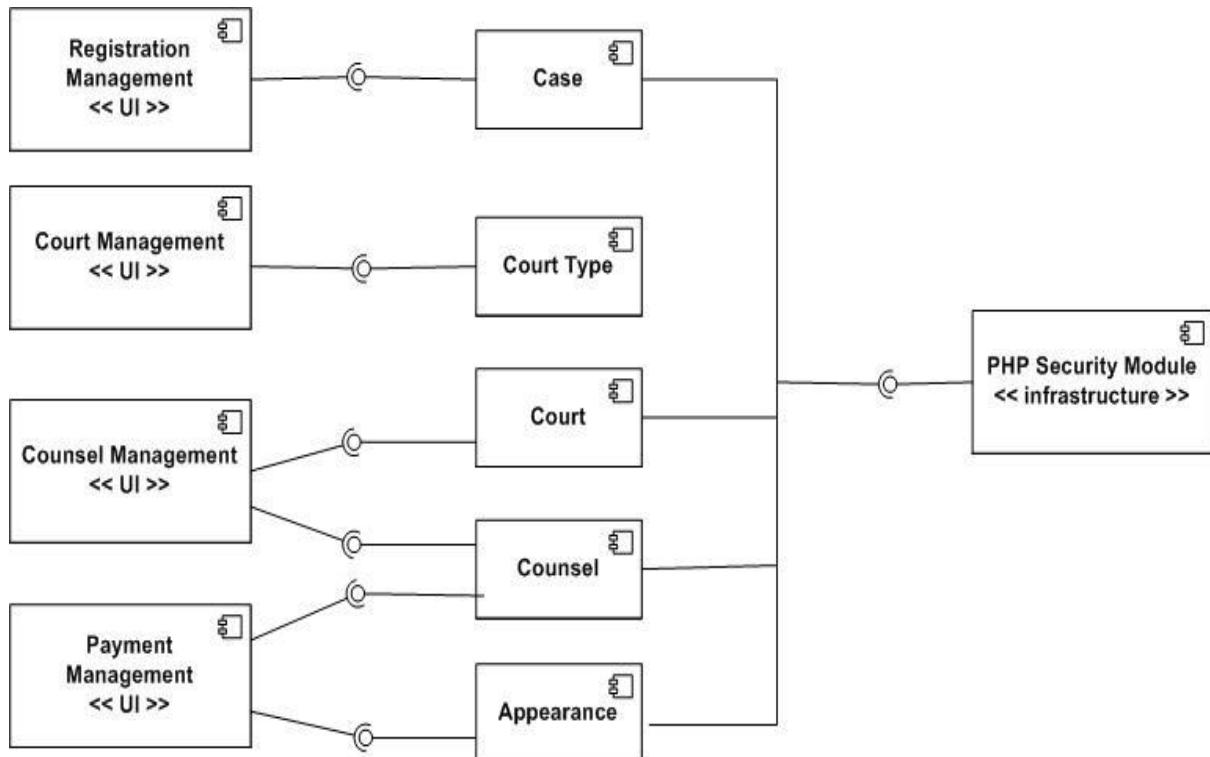
Payment



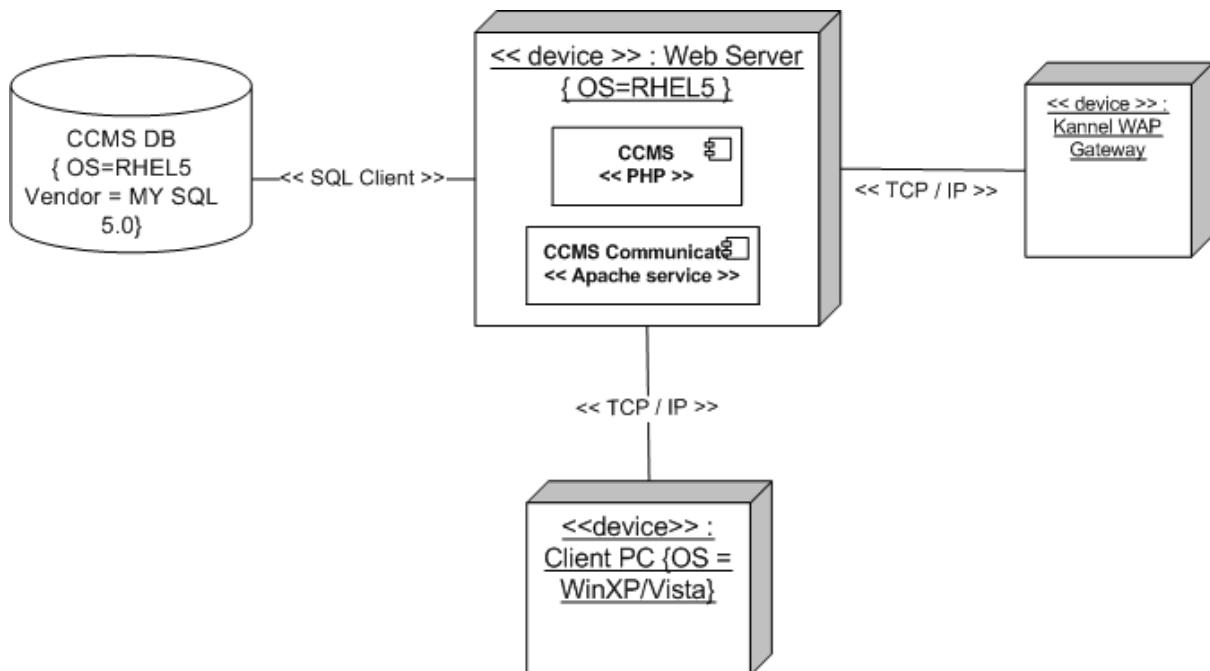
Reports View



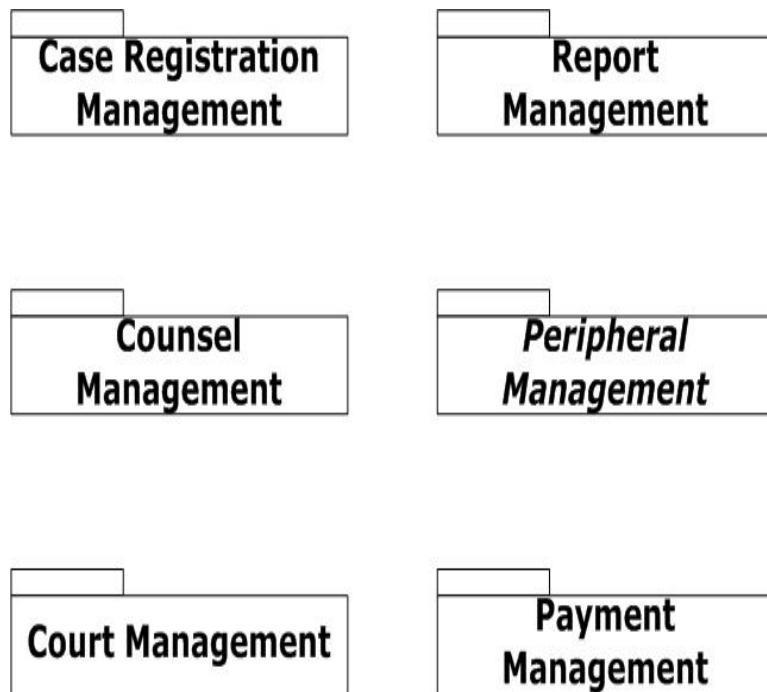
Section E – Component Diagram



Section F – Deployment Diagram



Section G – Package Diagram



Appendix E – Database Design & Table Schemas

Database Design

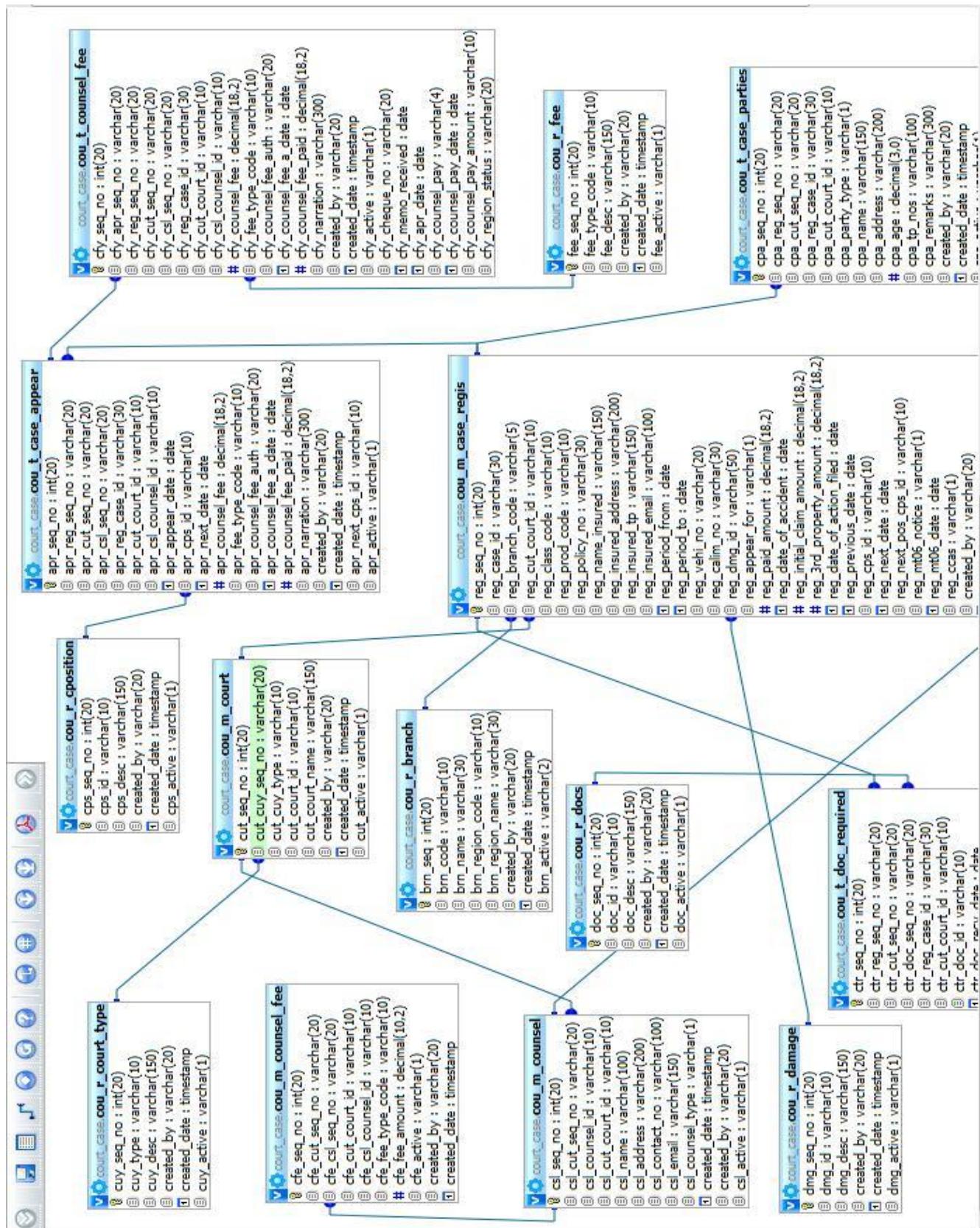


Table Schemas

➤ cou_doc_detail

court_case.cou_doc_detail	
acc_i_doc_id	varchar(15)
acc_i_doc_page_id	int(20)
acc_i_doc_path	varchar(300)
acc_i_page_num	varchar(3)
created_on	timestamp
created_by	varchar(20)

➤ cou_m_counsel

court_case.cou_m_counsel	
csl_seq_no	int(20)
csl_cut_seq_no	varchar(20)
csl_counsel_id	varchar(10)
csl_cut_court_id	varchar(10)
csl_name	varchar(100)
csl_address	varchar(200)
csl_contact_no	varchar(100)
csl_email	varchar(150)
csl_counsel_type	varchar(1)
created_date	timestamp
created_by	varchar(20)
csl_active	varchar(1)

➤ cou_doc_main

court_case.cou_doc_main	
acci_id	varchar(30)
acci_case	varchar(20)
acci_doc_seq	int(20)
acci_doc_type	varchar(200)
acci_doc_pages	varchar(10)
acci_doc_comment	varchar(300)
acci_doc_group	varchar(10)
acci_inserted_on	timestamp
acci_user	varchar(30)

➤ cou_m_case_regis

court_case.cou_m_case_regis	
reg_seq_no	int(20)
reg_case_id	varchar(30)
reg_branch_code	varchar(5)
reg_cut_court_id	varchar(10)
reg_class_code	varchar(10)
reg_prod_code	varchar(10)
reg_policy_no	varchar(30)
reg_name_insured	varchar(150)
reg_insured_address	varchar(200)
reg_insured_tp	varchar(150)
reg_insured_email	varchar(100)
reg_period_from	date
reg_period_to	date
reg_vehi_no	varchar(20)
reg_calim_no	varchar(30)
reg_dmg_id	varchar(50)
reg_appear_for	varchar(1)
reg_paid_amount	decimal(18,2)
reg_date_of_accident	date
reg_initial_claim_amount	decimal(18,2)
reg_3rd_property_amount	decimal(18,2)
reg_date_of_action_filed	date
reg_previous_date	date
reg_cps_id	varchar(10)
reg_next_date	date
reg_next_pos_cps_id	varchar(10)
reg_mt06_notice	varchar(1)
reg_mt06_date	date
reg_ccas	varchar(1)
created_by	varchar(20)
created_date	timestamp
reg_csl_counsel_id	varchar(10)
reg_previous_case_id	varchar(30)
rec_receive_date	date
reg_remarks	varchar(300)
reg_cut_seq_no	varchar(20)

➤ cou_m_counsel_fee

court_case.cou_m_counsel_fee	
cfe_seq_no	int(20)
cfe_cut_seq_no	varchar(20)
cfe_csl_seq_no	varchar(20)
cfe_cut_court_id	varchar(10)
cfe_csl_counsel_id	varchar(10)
cfe_fee_type_code	varchar(10)
cfe_fee_amount	decimal(10,2)
cfe_active	varchar(1)
created_by	varchar(20)
created_date	timestamp

➤ cou_m_court

court_case.cou_m_court	
cut_seq_no	int(20)
cut_cuy_seq_no	varchar(20)
cut_cuy_type	varchar(10)
cut_court_id	varchar(10)
cut_court_name	varchar(150)
created_by	varchar(20)
created_date	timestamp
cut_active	varchar(1)

➤ cou_m_user

court_case.cou_m_user	
user_seq	int(10)
user_full_name	varchar(100)
user_username	varchar(20)
user_password	varchar(20)
user_access_level	varchar(5)

➤ cou_r_branch

court_case.cou_r_branch	
brn_seq	int(20)
brn_code	varchar(10)
brn_name	varchar(30)
brn_region_code	varchar(10)
brn_region_name	varchar(30)
created_by	varchar(20)
created_date	timestamp
brn_active	varchar(2)

Project Documentation – Court Case Management System

➤  cou_r_court_type

court_case.cou_r_court_type	
cou_seq_no	int(20)
cou_type	varchar(10)
cou_desc	varchar(150)
created_by	varchar(20)
created_date	timestamp
cou_active	varchar(1)

➤  cou_r_cposition

court_case.cou_r_cposition	
cps_seq_no	int(20)
cps_id	varchar(10)
cps_desc	varchar(150)
created_by	varchar(20)
created_date	timestamp
cps_active	varchar(1)

➤  cou_r_damage

court_case.cou_r_damage	
dmg_seq_no	int(20)
dmg_id	varchar(10)
dmg_desc	varchar(150)
created_by	varchar(20)
created_date	timestamp
dmg_active	varchar(1)

➤  cou_r_docs

court_case.cou_r_docs	
doc_seq_no	int(20)
doc_id	varchar(10)
doc_desc	varchar(150)
created_by	varchar(20)
created_date	timestamp
doc_active	varchar(1)

➤  cou_r_fee

court_case.cou_r_fee	
fee_seq_no	int(20)
fee_type_code	varchar(10)
fee_desc	varchar(150)
created_by	varchar(20)
created_date	timestamp
fee_active	varchar(1)

➤  cou_t_case_appear

court_case.cou_t_case_appear	
apr_seq_no	int(20)
apr_reg_seq_no	varchar(20)
apr_cut_seq_no	varchar(20)
apr_csl_seq_no	varchar(20)
apr_reg_case_id	varchar(30)
apr_cut_court_id	varchar(10)
apr_csl_counsel_id	varchar(10)
apr_appear_date	date
apr_cps_id	varchar(10)
apr_next_date	date
# apr_counsel_fee	decimal(18,2)
apr_fee_type_code	varchar(10)
apr_counsel_fee_auth	varchar(20)
apr_counsel_fee_a_date	date
# apr_counsel_fee_paid	decimal(18,2)
apr_narration	varchar(300)
created_by	varchar(20)
created_date	timestamp
apr_next_cps_id	varchar(10)
apr_active	varchar(1)

➤  cou_t_case_parties

court_case.cou_t_case_parties	
cpa_seq_no	int(20)
cpa_reg_seq_no	varchar(20)
cpa_cut_seq_no	varchar(20)
cpa_reg_case_id	varchar(30)
cpa_cut_court_id	varchar(10)
cpa_party_type	varchar(1)
cpa_name	varchar(150)
cpa_address	varchar(200)
# cpa_age	decimal(3,0)
cpa_tp_nos	varchar(100)
cpa_remarks	varchar(300)
created_by	varchar(20)
created_date	timestamp
cpa_active	varchar(1)

➤  cou_t_counsel_fee

court_case.cou_t_counsel_fee	
cfy_seq_no	int(20)
cfy_apr_seq_no	varchar(20)
cfy_reg_seq_no	varchar(20)
cfy_cut_seq_no	varchar(20)
cfy_csl_seq_no	varchar(20)
cfy_reg_case_id	varchar(30)
cfy_cut_court_id	varchar(10)
cfy_csl_counsel_id	varchar(10)
# cfy_counsel_fee	decimal(18,2)
cfy_fee_type_code	varchar(10)
cfy_counsel_fee_auth	varchar(20)
cfy_counsel_fee_a_date	date
# cfy_counsel_fee_paid	decimal(18,2)
cfy_narration	varchar(300)
created_by	varchar(20)
created_date	timestamp
cfy_active	varchar(1)
cfy_cheque_no	varchar(20)
cfy_memo_received	date
cfy_apr_date	date
cfy_counsel_pay	varchar(4)
cfy_counsel_pay_date	date
cfy_counsel_pay_amount	varchar(10)
cfy_region_status	varchar(20)

➤ □ cou_t_doc_required

court_case.cou_t_doc_required	
ctr_seq_no :	int(20)
ctr_reg_seq_no :	varchar(20)
ctr_cut_seq_no :	varchar(20)
ctr_doc_seq_no :	varchar(20)
ctr_reg_case_id :	varchar(30)
ctr_cut_court_id :	varchar(10)
ctr_doc_id :	varchar(10)
ctr_doc_recv_date :	date
ctr_doc_inform_date :	date
created_date :	timestamp
created_by :	varchar(20)
ctr_active :	varchar(1)

➤ □ cou_t_settle_info

court_case.cou_t_settle_info	
set_seq_no :	int(20)
set_reg_seq_no :	varchar(20)
set_cut_seq_no :	varchar(20)
set_reg_case_id :	varchar(30)
set_cut_court_id :	varchar(10)
set_dmg_id :	varchar(10)
set_agrv_aprty_relation :	varchar(150)
set_nature_of_emplmnt :	varchar(150)
set_earnings :	varchar(150)
set_medi_rep :	varchar(300)
set_case_type :	varchar(1)
# set_estimate :	decimal(18,2)
# set_acr :	decimal(18,2)
set_insured_vehi_no :	varchar(30)
set_insured_insu_cover :	varchar(100)
created_date :	timestamp
created_by :	varchar(20)
set_active :	varchar(1)

Appendix F – Coding Samples for System

Code Sample for the Search Module

```

<?PHP
// Design & Develop by : Lahiru Manchanayake

session_start();

?>
<script language="JavaScript" type="text/JavaScript">
<!--
function MM_openBrWindow(theURL,winName,features) { //v2.0
  window.open(theURL,winName,features);
}
//-->
</script>

<?PHP

require("Connection.php");

if($_POST["action"]=="send"){

$Case_ID = strtoupper($_POST["Case_ID"]);
$Court_ID = strtoupper($_POST["Court_ID"]);
$Period_From = $_POST["Period_From"];
$Period_To = $_POST["Period_To"];
$Claim_No = strtoupper($_POST["Claim_No"]);
$Policy_No = strtoupper($_POST["Policy_No"]);
$Vehical_No = strtoupper($_POST["Vehical_No"]);
>Name_Insured = strtoupper($_POST["Name_Insured"]);
$Next_Date1 = $_POST["Next_Date1"];
$Next_Date2 = $_POST["Next_Date2"];
$Branch_Code = $_POST["Branch_Code"];
$Case_Position = $_POST["Case_Position"];
$Region = $_POST["Region"];

$_SESSION["Case_ID"] = $Case_ID;
$_SESSION["Court_ID"] = $Court_ID;
$_SESSION["Period_From"] = $Period_From;
$_SESSION["Period_To"] = $Period_To;
$_SESSION["Claim_No"] = $Claim_No;
$_SESSION["Policy_No"] = $Policy_No;
$_SESSION["Vehical_No"] = $Vehical_No;
$_SESSION["Name_Insured"] = $Name_Insured;
$_SESSION["Next_Date1"] = $Next_Date1;
$_SESSION["Next_Date2"] = $Next_Date2;
$_SESSION["Branch_Code"] = $Branch_Code;
$_SESSION["Case_Position"] = $Case_Position;
$_SESSION["Region"] = $Region;

}

if($_POST["action"]!="send"){

$Case_ID = $_SESSION["Case_ID"];
$Court_ID = $_SESSION["Court_ID"];
$Period_From = $_SESSION["Period_From"];
$Period_To = $_SESSION["Period_To"];
$Claim_No = $_SESSION["Claim_No"];

```

```

$Policy_No = $_SESSION["Policy_No"];
$Vehical_No = $_SESSION["Vehical_No"];
$Name_Insured = $_SESSION["Name_Insured"];
$Next_Date1 = $_SESSION["Next_Date1"];
$Next_Date2 = $_SESSION["Next_Date2"];
$Branch_Code = $_SESSION["Branch_Code"];
$Case_Position = $_SESSION["Case_Position"];
$Region = $_SESSION["Region"];
}

if($Region != ""){
?>

<style type="text/css">
<!--
.style1 {
    color: #FF0000;
    font-weight: bold;
}
-->
</style>
<link href="style.css" rel="stylesheet" type="text/css">
<style type="text/css">
<!--
.style2 {color: #FFFFFF}
-->
</style>
<body bgcolor="#2577CD">

<table width="866" border="0" align="center" >
<?PHP
$sqlre="SELECT * FROM cou_r_branch WHERE brn_region_code = '$Region' ";
$resultre = mysql_query($sqlre);
if($rowre = mysql_fetch_assoc($resultre)){
?>
    <tr>
        <td align="center">
            <?= strtoupper($rowre['brn_region_name'])." ".REGION ?></td>
    </tr>
<?PHP
}
?>
</table>
<?PHP
?>
?>

<table width="866" border="0" align="center" bgcolor="#2577CD" style="background-repeat:no-repeat; background-image: url(..pic/law.jpg); background-position:center">
    <tr>
        <td width="87" height="23" bgcolor="#1E62A6"><a href="Search_View_Code.php?field=REG_CASE_ID" class="style1" style="color:#FFFFFF">Case ID</a></td>
        <td width="120" bgcolor="#1E62A6"><a href="Search_View_Code.php?field=REG_CUT_COURT_ID" style="color:#FFFFFF">Court</a></td>
        <td width="203" bgcolor="#1E62A6"><a href="Search_View_Code.php?field=REG_NAME_INSURED" style="color:#FFFFFF">Name</a></td>
        <td width="128" bgcolor="#1E62A6"><a href="Search_View_Code.php?field=REG_BRANCH_CODE" style="color:#FFFFFF">Branch</a></td>

```

```

<td width="95" href="Search_View_Code.php?field=REG_NEXT_DATE" style="color:#FFFFFF">Next Date </a></td>
<td width="123" href="Search_View_Code.php?field=REG_NEXT_POS_CPS_ID" style="color:#FFFFFF">Next Position</a></td>
<td width="80" bgcolor="#1E62A6"><a href="Search_View_Code.php?field=REG_VEHICULAR_NO" class="style1" style="color:#FFFFFF">Vehical No</a></td>
</tr>
<tr>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
<td>&nbsp;</td>
</tr>
<?PHP
$field = $_GET["field"];

$sql="SELECT * FROM cou_m_case_regis WHERE 1=1 ";
$sqlcount = "SELECT count(*) AA FROM cou_m_case_regis WHERE 1=1 ";

if($Case_ID != ""){
$sql = $sql."AND REG_CASE_ID LIKE '$Case_ID' ";
$sqlcount = $sqlcount."AND REG_CASE_ID LIKE '$Case_ID' ";
}

if($Court_ID != ""){
$sql = $sql."AND REG_CUT_COURT_ID = '$Court_ID' ";
$sqlcount = $sqlcount."AND REG_CUT_COURT_ID = '$Court_ID' ";
}

if($Period_From != ""){
$sql = $sql."AND REG_PERIOD_FROM = '$Period_From' ";
$sqlcount = $sqlcount."AND REG_PERIOD_FROM = '$Period_From' ";
}

if($Period_To != ""){
$sql = $sql."AND REG_PERIOD_TO = '$Period_To' ";
$sqlcount = $sqlcount."AND REG_PERIOD_TO = '$Period_To' ";
}

if($Branch_Code != ""){
$sql = $sql."AND REG_BRANCH_CODE = '$Branch_Code' ";
$sqlcount = $sqlcount."AND REG_BRANCH_CODE = '$Branch_Code' ";
}

if($Claim_No != ""){
$sql = $sql."AND REG_CALIM_NO LIKE '$Claim_No' ";
$sqlcount = $sqlcount."AND REG_CALIM_NO LIKE '$Claim_No' ";
}

if($Policy_No != ""){
$sql = $sql."AND REG_POLICY_NO LIKE '$Policy_No' ";
$sqlcount = $sqlcount."AND REG_POLICY_NO LIKE '$Policy_No' ";
}

if($Vehical_No != ""){
$sql = $sql."AND REG_VEHICULAR_NO LIKE '$Vehical_No' ";
$sqlcount = $sqlcount."AND REG_VEHICULAR_NO LIKE '$Vehical_No' ";
}

```

```

}

if($Name_Insured != ""){
$sql = $sql."AND REG_NAME_INSURED LIKE '$Name_Insured' ";
$sqlcount = $sqlcount."AND REG_NAME_INSURED LIKE '$Name_Insured' ";
}

if($Case_Position != ""){
$sql = $sql."AND REG_NEXT_POS_CPS_ID = '$Case_Position' ";
$sqlcount = $sqlcount."AND REG_NEXT_POS_CPS_ID = '$Case_Position' ";
}

if($Next_Date1 != "" && $Next_Date2 != ""){
$sql = $sql."AND REG_NEXT_DATE BETWEEN '$Next_Date1' AND '$Next_Date2' ";
$sqlcount = $sqlcount."AND REG_NEXT_DATE BETWEEN '$Next_Date1' AND '$Next_Date2' ";
}

if( $field != "") {
$sql = $sql."ORDER BY " . $field ;
}

//echo $sql;
$resultcount = mysql_query($sqlcount);
if($rowcount = mysql_fetch_assoc($resultcount)){
$totcount = strtoupper($rowcount['AA']);
}

$result = mysql_query($sql);

while($row = mysql_fetch_assoc($result)){

$courtnname = strtoupper($row['reg_cut_court_id']);
$pcps = strtoupper($row['reg_cps_id']);
$ncps = strtoupper($row['reg_next_pos_cps_id']);
$branch = strtoupper($row['reg_branch_code']);

$sqlbra = "SELECT *
FROM cou_r_branch
WHERE brn_code = '$branch' ";

$resultbra = mysql_query($sqlbra);
$branchdesc = "";
if($rowbra = mysql_fetch_assoc($resultbra)){
$branchdesc = strtoupper($rowbra['brn_name']);
}

$sql6="SELECT * FROM cou_m_court WHERE CUT_COURT_ID = '$courtnname'";
$result6 = mysql_query($sql6);
$courtdesc = "";
if($row6 = mysql_fetch_assoc($result6)){
$courtdesc = strtoupper($row6['cut_court_name']);
}

$sql8="SELECT * FROM cou_r_cposition WHERE CPS_ID = '$ncps'";
$result8 = mysql_query($sql8);
$ncpsdesc = "";
if($row8 = mysql_fetch_assoc($result8)){
$ncpsdesc = strtoupper($row8['cps_desc']);
}
}

```

```

?>
<tr <? if($ncps=='SET'){ ?> bgcolor="#666666" <? } ?> >

    <td> <A href="javascript:;" class="style2"
onClick="MM_openBrWindow('Search_View_Case_Register.php?id=<?=strtoupper($row['reg_seq_no']); ?>','scrollbars=yes,width=850,height=600')>
        <font color="#FFFFFF"><?= strtoupper($row['reg_case_id']); ?></font>
    </A> </td>

    <td><font color="#FFFFFF">
        <?= $courtdesc ?>
    </font></td>

    <td><font color="#FFFFFF">
        <?= strtoupper($row['reg_name_insured']); ?>
    </font></td>

    <td><font color="#FFFFFF">
        <?= $branchdesc ?>
    </font></td>

    <td> <font color="#00CC00"> <?= strtoupper($row['reg_next_date']); ?> </font> </td>

    <td> <font color="#FF0000"> <?= $ncpsdesc ?> </font> </td>

    <td><font color="#FFFFFF">
        <?= strtoupper($row['reg_vehi_no']); ?>
    </font></td>
<!--   <td>&nbsp;</td> -->
</tr>
<?PHP
}
?>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
</tr>
<tr>
    <td><font color="#FFFFFF">Total Cases :</font></td>
    <td><font color="#FFFFFF" style="font-weight:bold"><?= $totcount ?></font></td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td><div align="right"><a href="main_search.php" style="color:#FFFFFF">&lt;&lt; Back</a></div></td>
</tr>
</table>

</body>

```

Code Sample for the Payment Module

```

<script language="JavaScript" type="text/JavaScript">
<!--
function MM_openBrWindow(theURL,winName,features) { //v2.0
  window.open(theURL,winName,features);
}
//-->
</script>

<?PHP

require("Connection.php");

$Case_ID = strtoupper($_POST["Case_ID"]);
$Court_ID = strtoupper($_POST["Court_ID"]);
$Case_Position = strtoupper($_POST["Case_Position"]);
$Counsel_ID = strtoupper($_POST["Counsel_ID"]);
$Counsel_Fee = strtoupper($_POST["Counsel_Fee"]);
$Fee_Paid = strtoupper($_POST["Fee_Paid"]);
//Cheque_No = strtoupper($_POST["Cheque_No"]);
$Authorised_By = strtoupper($_POST["Authorised_By"]);
$Paid_Date = strtoupper($_POST["Paid_Date"]);
$Memo_Received = strtoupper($_POST["Memo_Received"]);
$Narration = strtoupper($_POST["Narration"]);
$App_Date = strtoupper($_POST["App_Date"]);
$region_status = strtoupper($_POST["region_status"]);

if ($Authorised_By == "Y") {

if ($Fee_Paid == ""){
    $Fee_Paid = $Counsel_Fee;
}

if ( $Case_ID != "" && $Court_ID != "" && $Counsel_ID != "" && $Case_Position != "" ) {

$sql = " UPDATE cou_t_counsel_fee
        SET
            CFY_COUNSEL_FEE_PAID = '$Fee_Paid',
            CFY_COUNSEL_FEE_A_DATE = '$Paid_Date',
            CFY_CHEQUE_NO = '$Cheque_No',
            CFY_MEMO_RECEIVED = '$Memo_Received',
            CFY_NARRATION = '$Narration',
            CFY_COUNSEL_FEE_AUTH = '$Authorised_By',
            CFY_APP_DATE = '$App_Date'
        WHERE CFY_REG_SEQ_NO = '$Case_ID'
            AND CFY_CUT_SEQ_NO = '$Court_ID'
            AND CFY_CSL_SEQ_NO = '$Counsel_ID'
            AND CFY_APP_SEQ_NO = '$Case_Position' ";

if (!mysql_query($sql,$con)){
    die('Error: ' . mysql_error());
}

***** TO GET COURT ID *****
$sql1="SELECT * FROM cou_m_court WHERE CUT_SEQ_NO = '$Court_ID'";
$result1 = mysql_query($sql1);
while($row1 = mysql_fetch_assoc($result1)){
    $CourtName = strtoupper($row1['cut_court_name']);
}

***** TO GET COUNSEL ID *****
$sql2="SELECT * FROM cou_m_counsel WHERE CSL_SEQ_NO = '$Counsel_ID'";
$result2 = mysql_query($sql2);
while($row2 = mysql_fetch_assoc($result2)){
    $CounselName = strtoupper($row2['csl_name']);
}
}

```

```

?>
<link href="style.css" rel="stylesheet" type="text/css">
<body bgcolor="#2577CD" style="background-repeat:no-repeat; background-image: url(..pic/law.jpg); background-position:center">

<table width="751" height="285" border="0" align="center">
<tr>
<td width="93" rowspan="3"></td>
<td width="343" height="53"><strong>Ceylinco Insurance PLC</strong></td>
<td width="211" rowspan="3">&ampnbsp</td>
<td width="86" rowspan="3">&ampnbsp</td>
</tr>
<tr>
<td height="22"><strong>Legal Department ( Ceylinco House ) </strong></td>
</tr>
<tr>
<td height="18"><strong>Court Case</strong></td>
</tr>
<tr>
<td height="16" colspan="4" >&ampnbsp</td>
</tr>
<tr>
<td height="76" colspan="4"><table align="center" width="718" border="0">

<?PHP
$sqlx = "SELECT * from cou_m_case_regis WHERE REG_SEQ_NO = '$Case_ID' ";
$resultx = mysql_query($sqlx);
if($rowx = mysql_fetch_assoc($resultx)){
    $caseid = strtoupper($rowx['reg_case_id']);
    $court = strtoupper($rowx['reg_cut_court_id']);
    $policy = strtoupper($rowx['reg_policy_no']);
    $initialclaimamount = strtoupper($rowx['reg_initial_claim_amount']);
    $branch = strtoupper($rowx['reg_branch_code']);
    $dateofloss = strtoupper($rowx['reg_date_of_accident']);
    $vehical = strtoupper($rowx['reg_vehi_no']);
    $nextdate = strtoupper($rowx['reg_next_date']);

?>
<tr>
<td width="111">Case No </td>
<td width="10">:</td>
<td width="222"><?= $caseid ?></td>
<td width="114">Policy No </td>
<td width="10">:</td>
<td width="225"><?= $policy ?></td>
</tr>
<tr>
<td>Court</td>
<td>:</td>
<td><?= $CourtName ?></td>
<td>Amount Claimed </td>
<td>:</td>
<td><?= $initialclaimamount ?></td>
</tr>
<tr>
<td>Branch</td>
<td>:</td>
<td><?= $branch ?></td>
<td>Date of Loss </td>
<td>:</td>
<td><?= $dateofloss ?></td>
</tr>
<tr>
<td>Vehicle No </td>
<td>:</td>
<td><?= $vehical ?></td>
<td>Counsel </td>
<td>:</td>
<td><?= $CounselName ?></td>
</tr>

```

```

<?PHP
}
?>
    </table></td>
</tr>
<tr>
    <td height="16" colspan="4">&nbsp;</td>
</tr>
<tr>
    <td colspan="4"><table align="center" width="721" border="0">
<tr>
    <td width="113">Date Of Appear</td>
    <td width="10">:</td>
    <td width="223"><?= $App_Date ?></td>
    <td width="117">Next Date </td>
    <td width="10">:</td>
    <td><?= $nextdate ?></td>
</tr>
<tr>
    <td>Paid On </td>
    <td>:</td>
    <td><?= $Paid_Date ?></td>
    <td>Narration</td>
    <td>:</td>
    <td><?= $Narration ?></td>
</tr>
<tr>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td>&nbsp;</td>
    <td><strong>Legal Fee </strong></td>
    <td><strong>:</strong></td>
    <td><strong>Rs :<br/>
        <?= sprintf("%.2f", $Fee_Paid) ?>
        </strong></td>
</tr>
</table></td>
</tr>
<tr>
    <td colspan="4">&nbsp;</td>
</tr>
<tr>
    <td colspan="4" bgcolor="#1E62A6"><div align="center">~ Big or Small - Ceylinco protects them all ~
</div></td>
</tr>
<tr>
    <td colspan="4">&nbsp;</td>
</tr>
<tr>
    <td colspan="4">&nbsp;</td>
</tr>
<tr>
    <td align="center" colspan="2"><div align="center">      <a href="javascript:;">
        onClick="MM_openBrWindow('Print_Payment.php?Case_ID=<?= $Case_ID ?>&Court_ID=<?= $Court_ID ?>&Case_Position=<?= $Case_Position ?>&Counsel_ID=<?= $Counsel_ID ?>&Counsel_Fee=<?= $Counsel_Fee ?>&Fee_Paid=<?= $Fee_Paid ?>&Cheque_No=<?= $Cheque_No ?>&Authorised_By=<?= $Authorised_By ?>&Paid_Date=<?= $Paid_Date ?>&Memo_Received=<?= $Memo_Received ?>&Narration=<?= $Narration ?>&App_Date=<?= $App_Date ?>','scrollbars=yes,width=800,height=350')"> P r i n t </a> </div></td>
    <td><a href="Main_Payment.php"><font color="#0000FF">&lt;&lt; B a c k </font></a></td>
</tr>
</table>
<?PHP
}
?

?>
</body>

```

Code Sample for the Ajax Module

```

var xmlhttp;

function showCounselFeeType(str1, str2)
{
xmlhttp=GetXmlHttpObject();
if (xmlhttp==null)
{
alert ("Your browser does not support AJAX!");
return;
}
var url="CounselFeeType1.php";
url=url+"?Court_ID="+str1+"&Counsel_ID="+str2;
xmlhttp.onreadystatechange=stateChangedFeeType;
xmlhttp.open("GET",url,true);
xmlhttp.send(null);
}

function stateChangedFeeType()
{
if (xmlhttp.readyState==4)
{
document.getElementById("txtCounselFeeType").innerHTML=xmlhttp.responseText;
}
}

function GetXmlHttpObject()
{
var xmlhttp=null;
try
{
// Firefox, Opera 8.0+, Safari
xmlhttp=new XMLHttpRequest();
}
catch (e)
{
// Internet Explorer
try
{
xmlhttp=new ActiveXObject("Msxml2.XMLHTTP");
}
catch (e)
{
xmlhttp=new ActiveXObject("Microsoft.XMLHTTP");
}
}
return xmlhttp;
}

```

Appendix G – Coding Samples for WAP Module

Code Sample for the Search Module

Page : 1

```
<?php
// Design & Develop by : Lahiru Manchanayake
// send wml headers
header("Content-type: text/vnd.wap.wml");
echo "<?xml version='1.0'?>";
echo "<!DOCTYPE wml PUBLIC '-//WAPFORUM//DTD WML 1.1//EN\'"
. "\http://www.wapforum.org/DTD/wml_1.1.xml\">";
    require("Connection.php");
?>

<wml>
<card id="card3" title="Search Case">
<p>
    <do type="accept" label="Search">
        <go href="search_case_code.wml" method="post">
            <postfield name="case" value="$(case)"/>
        </go>
    </do>
    Case No :<input name="case" size="7"/>
</p>
<p>.. Lahiru Manchanayake .:</p>
</card>
</wml>
```

Page : 2

```
<?php
// send wml headers
header("Content-type: text/vnd.wap.wml");
echo "<?xml version='1.0'?>";
echo "<!DOCTYPE wml PUBLIC '-//WAPFORUM//DTD WML 1.1//EN\'"
. "\http://www.wapforum.org/DTD/wml_1.1.xml\">";
    require("Connection.php");
?>

<wml>
<card id="card4" title= "Case Details">
<p>
    Branch List : <br/>
<?php
    $case = strtoupper($_REQUEST["case"]);
    $sql = "SELECT * FROM cou_m_case_regis WHERE reg_case_id LIKE '%$case%'";
    $result = mysql_query($sql);
    while($row = mysql_fetch_assoc($result)){
?>
        <anchor> <?= strtoupper($row['reg_case_id'])." - ".strtoupper($row['reg_court_id']) ?>, <go href="search_case_details.wml?seq=<?= strtoupper($row['reg_seq_no'])?>"/> </anchor> <br/>
<?PHP
    }
?>
</p>
<p>.. Lahiru Manchanayake .:</p>
</card>
</wml>
```

Appendix H – CCMS Web Form Design Samples

Loading Page



Login Page



Home Page



Case Registration Page

Case Register	
Case ID :	2008-10-31
Date Action Filed :	2008-10-31
MT106 Notice :	Notice Given
Claim Amount :	
Damage :	Select Damage ID
Policy No :	
Period From :	2008-10-31
Name Insured :	
E-Mail :	
3rd Party Amount :	
Previous Case ID :	
Handled by :	Ceylinco
Branch Code :	Select Branch
Remarks :	
Case Received Date :	2008-10-31
Court ID :	Select Court ID
MT106 Date :	
Date Of Accident :	2008-10-31
Vehical No :	
Claim No :	
Period To :	2008-10-31
Insured Address :	
Appear For :	Defendant
Paid Amount :	
Class Code :	
Counsel ID :	Select Counsel ID
Product Code :	
Telephone No :	

Project Documentation – Court Case Management System

Counsel Fee Maintenances Page

Screenshot of the Counsel Fee Maintenance page:

The page title is "Court Case Management System" and the subtitle is "For Ceylinco Insurance PLC".

Left sidebar menu (Case Registration section):

- Case Registration
- Court Type
- Court
- Counsel
- Fee
- Counsel Fee
- Case Position
- Damage
- Document
- Search
- Update
- Reports
- Document Upload
- Payment

Main content area:

Create Counsel Fee

Court ID	: Select Court ID
Counsel ID	: Select Counsel ID
Fee Type Code	: Select Fee Type ID
Fee Amount	: <input type="text"/>

Court Name	Counsel Name	Fee Type Code	Fee Amount
ANURADHAPURA	MR. A. WICKRAMARACHCHI	CALLING	1000.00
ANURADHAPURA	MR. A. WICKRAMARACHCHI	DOCUMENTATION	2500.00
ANURADHAPURA	MR. A. WICKRAMARACHCHI	DOCUMENTATION	1000.00
ANURADHAPURA	MR. A. WICKRAMARACHCHI	TRAIL	2000.00
AMPARA DC	MR. KULARATNE R.G	CALLING	1000.00
AMPARA DC	MR. KULARATNE R.G	DOCUMENTATION	2500.00
AMPARA DC	MR. KULARATNE R.G	TRAIL	2500.00
AVISSAWELLA	MR. MANGALA SENEVIRATHNA	CALLING	4500.00
BADULLA	MR. KULARATNE R.G	DOCUMENTATION	4000.00
BADULLA	MR. RUWAN BASNAYAKE	CALLING	1500.00
BADULLA	MR. RUWAN BASNAYAKE	TRAIL	2750.00
COLOMBO DC	MR. UDUGODA ANURA	CALLING	1250.00
COLOMBO DC	MR. UDUGODA ANURA	DOCUMENTATION	2500.00

Search Page

Screenshot of the Search page:

The page title is "Court Case Management System" and the subtitle is "For Ceylinco Insurance PLC".

Left sidebar menu (Case Registration section):

- Case Registration
- Court Type
- Court
- Counsel
- Fee
- Counsel Fee
- Case Position
- Damage
- Document
- Search
- Update
- Reports
- Document Upload
- Payment

Main content area:

Search

Case ID	: <input type="text"/>	Court ID	: Select Court ID
Period From	: <input type="text"/>	Period To	: <input type="text"/>
Claim No	: <input type="text"/>	Policy No	: <input type="text"/>
Vehical No	: <input type="text"/>	Name Insured	: <input type="text"/>
Case Position	: Select Case Position	Branch Code	: Select Branch
Next Date (Period)	: <input type="text"/> to <input type="text"/>	Region	: Select Region

Page footer: © 2008 CeyinS PLC. Solution by Lahiru Manchanayake. All rights reserved.

Project Documentation – Court Case Management System

Search Result Page

Screenshot of the Search Result Page (http://localhost/court_case/Pages/Search_View_Code.php) showing a list of court cases.

Case ID	Court	Name	Branch	Next Date	Next Position	Vehical No
TEST	ANURADHAPURA	TEST	ANURADHAPURA BRANCH	2008-10-16	ANSWER FINAL	TEST123
TEST1	AVISSAWELLA	SFAFA				SFAFS
TEST1	COLOMBO HC	TEST		2008-10-18	SETTLED	TEST123
1763/M	AMPARA DC	S.A. WEERASINGHE	ANURADHAPURA BRANCH	2008-07-20	ANSWER	63-5563
2525/M	BADULLA	R. P. P. JAYARATHNE	MOUNT LAVINIA	2008-09-27	FILING PROXY	WPGA4528
21755/M	ANURADHAPURA	CHANDANA STORES	ANURADHAPURA BRANCH	2008-11-25	ANSWER	.
24661/M	AVISSAWELLA	P.S.M.JEWAHIR	GALLE BRANCH	2009-01-19	SUMMONS RETURNABLE	SPGS7777
55251/M	COLOMBO DC	K.A SARATH GUNASEKARA	MATARA BRANCH	2008-09-20	TRIAL OR SETTLEMENT	256-4578
4854/M	MT. LAVINIA - DC	W.K.P.P.K DAIS	NUGEGODA BRANCH	2009-02-08	SUMMONS RETURNABLE	WPKA4545
488/M	AMPARA DC	M.P.M. PERIMPARASA	JAFFNA BRANCH	2009-01-27	SUMMONS RETURNABLE	UPGA3232
25488/M	AVISSAWELLA	JAGATH PIERIS	MAHARAGAMA BRANCH	2008-03-12	SUMMONS RETURNABLE	WPGG8316
12/M	ANURADHAPURA	SGS	MAHARAGAMA BRANCH	2008-10-12	TRIAL	DFFWR

Total Cases : 12 [<< Back](#)

View Details Page

Screenshot of the View Case Register Details page (http://localhost/court_case/Pages/View_Case_Register2.php).

View Case Register Details			
Case ID	: 1763/M	Previous Case ID	:
Court ID	: DC - AMPARA DC	Branch Code	: AA00
Date Action Filed	: 2007-01-03	Product Code	: 1B
MT106 Notice	: N	MT106 Date	: 2007-03-27
Claim Amount	: 1000000.00	3rd Party Amount	: 0.00
Date Of Accident	: 2006-05-30	Paid Amount	: 0.00
Damage	: DEATH	Claim No	:
Vehical No	: 63-5563	Class Code	: MC
Policy No	: 02AA1B000274	Counsel ID	: MR. KULARATNE R.G
Period From	: 2006-01-01	Period To	: 2006-12-31
Name Insured	: S.A. WEERASINGHE	Telephone No	: 025-2222152
Insured Address	: NO. 25, NEW TOWN, ANURADHAPURA	E-Mail	:
Appear For	: D	Handled by	: Ceylinco
Case Position	:	Case Received Date	:
Previous Date	:	Remark	:
Next Date	: 2008-07-20		
Next Position	: ANSWER		
<u>Case Appear</u>		<u>Document Required</u>	
<u>Case Parties</u>		<u>Settle Info</u>	
<u>Edit Case</u>			
Done			

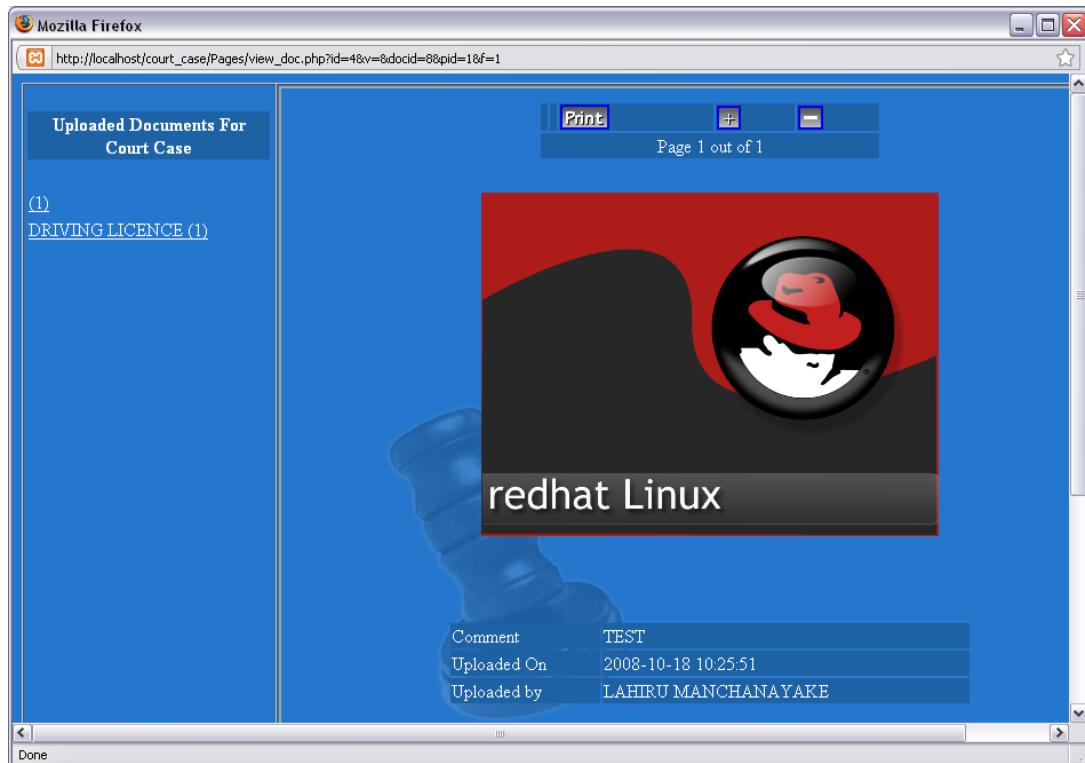
Document Upload Page



Scan File Uploading Page

The screenshot shows a Firefox browser window titled 'Upload Document - Mozilla Firefox' with the URL 'http://localhost/court_case/Pages/upload_tt_doc.php?id=4&case=1763/M'. The form fields include: 'Document Type : MAGISTRATE COURT CASE PROCEEDINGS' (dropdown menu), 'If Other Specify : test' (text input), 'Comment : ' (text input), 'Number Of Pages : 3' (text input), 'Upload Page1' (file input), 'Upload Page2' (file input), 'Upload Page3' (file input), and an 'Upload' button. A 'Done' button is at the bottom right.

Scan File Viewing Page



Payment Creation Page

Case ID	Court ID	Position	App. Date	Counsel	Amount Paid	Narration	Paid On	Cheque No
1763/M	AMPARA DC	ANS	2008-07-20	KULARATNE R.G.	1000.00		2007-08-15	622530
1763/M	AMPARA DC	SR	2007-02-16	KULARATNE R.G.	2000.00	STAMP 1000/-	2007-03-05	602255

[Update Voucher](#) [Print Report](#)

Single Payment Invoice

The screenshot shows a Mozilla Firefox browser window with the URL http://localhost/court_case/Pages/Print_Payment1.php?Case_ID=4&Court_ID=66&Counsel_ID=APDCC1 &Seq=4. The page header includes the Ceylinco Insurance PLC logo and the text "Ceylinco Insurance PLC", "Legal Department (Ceylinco House)", and "Court Case". The main content displays case details and payment information:

Case No	:	1763/M	Policy No	:	02AA1B000274
Court	:	AMPARA DC	Amount Claimed	:	1000000.00
Branch	:	AA00	Date of Loss	:	2006-05-30
Vehicle No	:	63-5563	Counsel	:	MR. KULARATNE R.G
 			Date Of Appear	:	2008-07-20
			Memo Received	:	2007-07-30
			Paid On	:	2007-08-15
			Next Date	:	2008-07-20
			Cheque No	:	622530
			Narration	:	
			Paid Amount	:	Rs : 1000.00

~ Big or Small - Ceylinco protects them all ~

Done

Bulk Payment Invoice

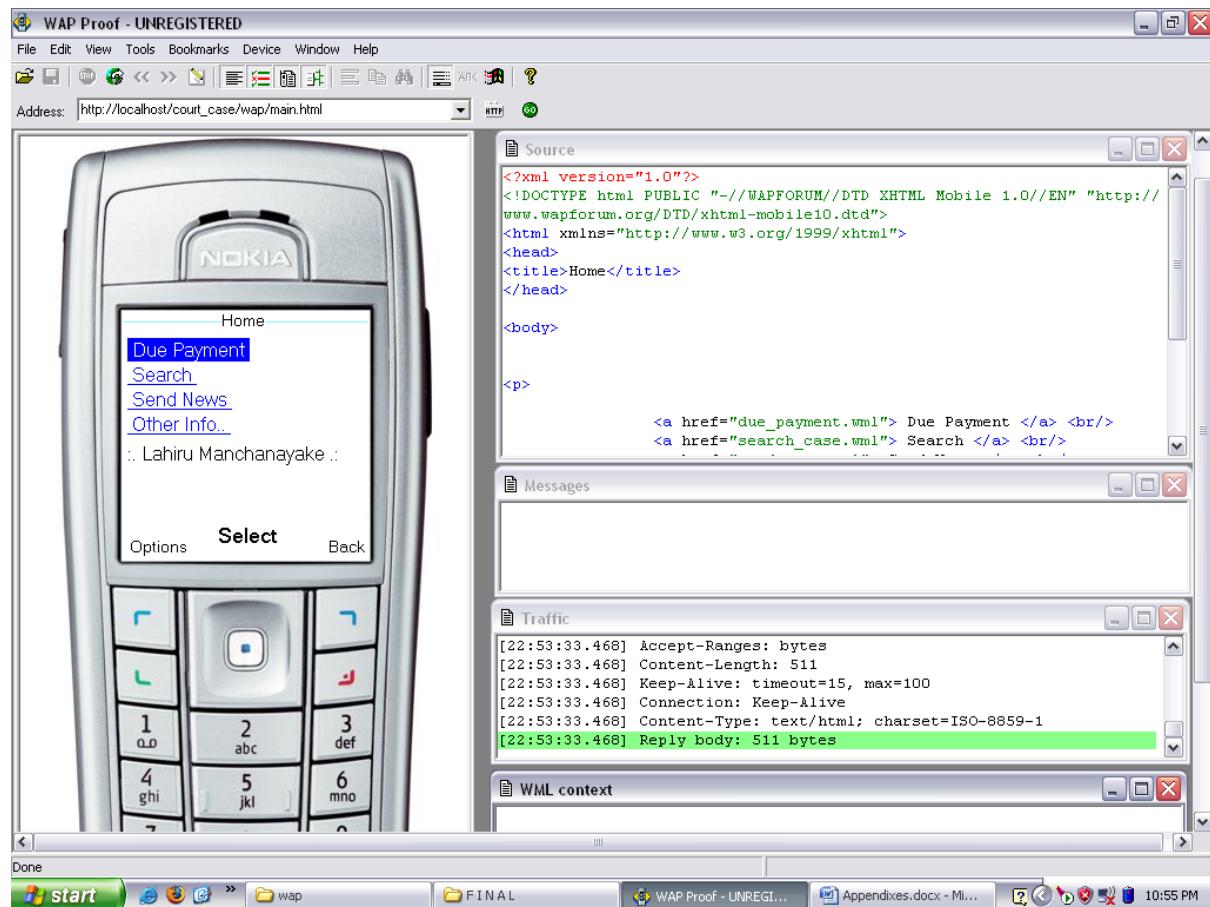
The screenshot shows a Mozilla Firefox browser window with the URL http://localhost/court_case/Pages/Print_Report.php?id=1763/M&courtseq=66. The page header includes the Ceylinco Insurance PLC logo and the text "Ceylinco Insurance PLC", "Legal Department (Ceylinco House)", and "Court Case". The main content displays case details and a table of payment transactions:

Case ID	:	1763/M	Amount Claim	:	1000000.00	
Court	:	AMPARA DC	Policy No	:	02AA1B000274	
			Branch	:	AA00	
Position	Appear Date	Counsel	Narration	Paid On	Cheque No	Amount Paid
ANSWER	2008-07-20	MR. KULARATNE R.G		2007-08-15	622530	1000.00
SUMMONS	2007-02-16	MR. KULARATNE R.G	STAMP 1000/-	2007-03-05	602255	2000.00
RETURNABLE						3000.00

~ Big or Small - Ceylinco protects them all ~

Done

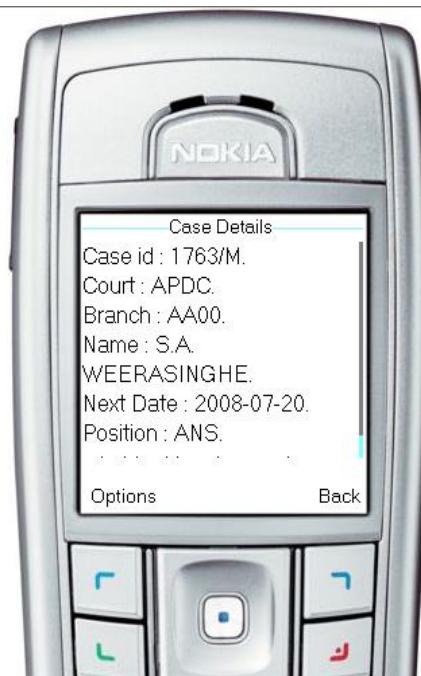
Mobile Home Page with Simulator



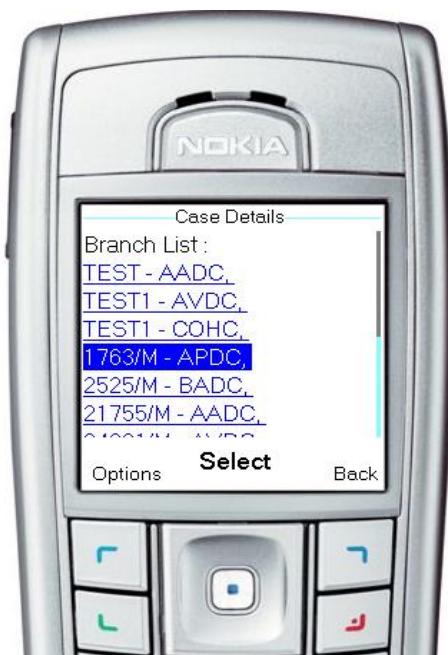
Case Search Page



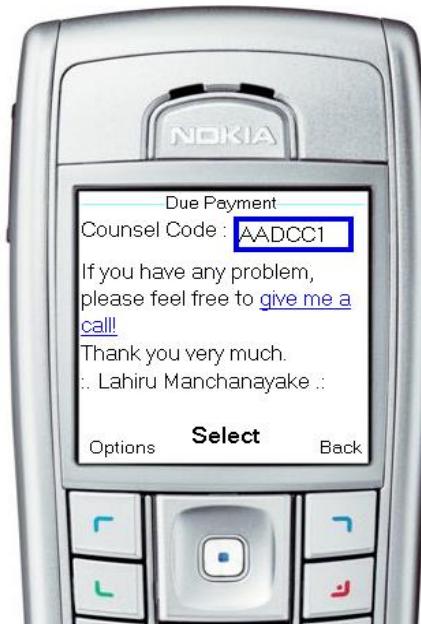
Case Search Result Details Page



Case Search Result Page



Search Due Payment Page



Appendix I – Test Case Design & Test Results

Test Case Design for User Creation

Test No	Input Data	Expected Outcome	Actual Outcome
1	UserName = lahiru	OK	OK
2	UserName = lahiru	ERR - Already Exists	ERR – Already Exists
3	Password = NULL	ERR – Required	ERR – Required
4	Database Connection Lost	ERROR	ERROR

Test Case Design for Adding a Case

Test No	Input Data	Expected Outcome	Actual Outcome
1	CaseID = 145/M	OK	OK
2	CaseID = 145/M	ERR - Already Exists	ERR – Already Exists
3	CaseID = %8473M	ERR – Invalid ID	ERR – Invalid ID
4	Name = "MR.KAMAL"	OK	OK
5	Court = NULL	ERR – Required	ERR – Required
6	Counsel = NULL	ERR – Required	ERR – Required

Appendix J – CCMS Physical System Architecture

