Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technologies

CZECH
UNIVERSITY
OF LIFE SCIENCES PRAGUE

Diploma Thesis

Information System Design in UML

Author: Armend Qerimi

Supervisor: doc. RNDr. Dana Klimešová, CSc.

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Department of Information Engineering

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Armend Qerimi

Informatics

Thesis title

Information system design in UML

Objectives of thesis

In diploma thesis Information system design in UML is covered by a case study the design of Web Based Human Resources Management System. The main objectives of this thesis is to design a type of web application which will help small and medium organizations to manage human resources and spend less money using this type of software.

The first part of this thesis, contains research information and studies of UML and the requirements which will be used for the new web application design.

The second part of the objective contains the demonstration of the process and the final outcome of the application development using UML modeling. Regarding to the system, the web application is con-sidered as the common software (such as electronic office or social networking service) which could adaptive for everyday activities in the company or other organizations that needs to manage personnel resources. The core idea of the second objective part is to demonstrate the advantages by highlighting the particular features of the UML Language for the web application development.

In the last and the third part of the objective contains the comparison of the UML models of the web development system which is created in the previous section in this theses and the software applications which are created in other platforms (different from web). And also will be compared the technologies of the development of the previous application and web application. The comparison analyzes is made in terms of web application development having same purposes but built on various technologies by emphasizing their diversities.

Methodology

The methodology of the thesis is based on research of relevant information resources. Practical process will be derived from results which will be given by the research study. Finally, comparison of design and analysis of the Web Human Resources Management System with another similar system will be made. Based on the synthesis of the theoretical and the practical knowledge, final conclusions will be formulated. In the first part collected the necessary information about the UML Modeling and web application. The further step was to define requirements of the application in order to exactly characterize the system processes. This process is fulfilled based on analyzing the specifications concluded in the previous research and study. Finally formulation of the conclusions are expressly made based on theoretical and practical knowledge has done in diploma thesis.

The proposed extent of the thesis

60 – 80 pages

Recommended information sources

Addison-Wesley, Reading, MA.
Booch, G., Rumbaugh, J., and Jacobson, I., 1999, The Unified Modeling Language User
Conallen, J., 2000, Building Web Applications with UML (Object Technology Series),
Guide (Object Technology Series), Addison-Wesley, Reading, MA.
Jon Holt, 2004, UML for Systems Engineering: watching the wheels, Second Edition, The Institution of Engineering and Technology, London, United Kingdom
Vrana,I. Projecting of information systems with UML, CULS Prague, 2009, 150p.

Expected date of thesis defence

2015/06 (June)

The Diploma Thesis Supervisor

doc. RNDr. Dana Klimešová, CSc.

Electronic approval: 10. 11. 2014	Electronic approval: 10. 11. 2014
Ing. Martin Pelikán, Ph.D.	**Ing. Martin Pelikán, Ph.D.**
Head of department	Dean

Prague on 09. 03. 2015

Declaration

I declare that I have worked on my diploma thesis titled "System Information Design in UML" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any third person.

In Prague 2015

____

Armend Qerimi

Acknowledgement

I would like to thank my family for their continuous encouragement, support and love. Also, I would like to thank my friends. I could not have made it without all of you!

I wish to express my greatest gratitude to my supervisor, RNDr. Dana Klimešová, CSc. I am very thankful for her continuous guidance, patience and support throughout our discussions.

Special thanks to consulate of Czech Republic in Kosovo and Government of Czech Republic, especially Ministry of Education Youth and Sports for granting me a scholarship and giving me the opportunity to study at Czech University of Life Sciences, Prague.

I would also like to thank the academic staff of the Systems Engineering and Informatics of the Czech University of Life Sciences, Prague, for their help and academic guidance. It has been a great opportunity to learn new things and grow as a future professional.

Information System Design in UML

Návrh informačního systému v UML

Souhrn

Webový informační prostor roste rychle co do velikosti a rozmanitosti využívaných dat, tak i různorodostí a množství uživatelů. Důsledkem je, že webové informační systémy v mnoha případech nahrazují stávající, klasické informační systémy. Vzhledem k tomu, že webové informační systémy se liší svou povahou od tradičních informačních systémů, je velká poptávka po metodologiích specificky zaměřených na projektování webových informačních systémů. Jednou z těchto metod je UML-based web engineering (UWE), rozšíření UML, založené na webovém inženýrství, které je použito v tomto dokumentu pro návrh nového informačního systému. Tato metoda využívá UML notaci a diagramy pro vyjádření jednotlivých fází návrhu webového informačního systému, jako je analýza požadavků, koncepční model, navigační model a prezentace modelu. Bude také součástí realizační fáze, která zahrnuje vizuální pohled na celý systém. Vzhledem k vysoké poptávce a efektivnosti webových informační systémů, je v této práci navržen webový informační systém pro řízení lidských zdrojů. Vzhledem ke složitosti a velikosti systému, není možné prezentovat všechny pojmy a části návrhu v detailech. Cílem je vytvořit modely pro různé pohledy na analýzu a projektovou fázi návrhu. Zaměříme se a navrhneme tři základní modely pro webový systém řízení lidských zdrojů Personální management, Řízení odchodů a Vedení mzdové agendy, které jsou úzce propojeny a navazují na moduly na webu založeného informačního systému pro řízení lidských zdrojů (HRMS). Práce se rovněž věnuje srovnání systémů z hlediska použitelnosti navržených pro web, nebo vyvinutých tradičním způsobem.

Klíčová slova: Návrh projektu, Webová aplikace, Systém řízení lidských zdrojů, UML, UWE

Summary

The web information space is rapidly growing in size and the diversity of both its data and its audience. A consequence is that web information systems in many cases replace existing traditional information systems. Since the Web information systems differs from the nature of traditional information systems, there is a big demand for design methodologies specifically oriented towards Web Information Systems. One of these methods is UML-based web engineering, an extension of UML used in this document to design a proposed information system. The UWE method uses UML notation and diagrams to express phases of web information system design such as requirement analysis, conceptual model, navigation model and presentation model. This document introduces implementation phase which covers the visual view of the system. Due to high demand and effectiveness of Web Information system, it is covered a design of Web based Human Resources Management System. Regarding to the complexity and the size of the system, it is not possible to present all concepts in details. The focus will be to build models for different views of the analysis and design phase. It will contain the design of three main modules of Web based Human Resources Management System such as Personnel management, Leave management, and Payroll management, which are closely related to each other and other modules of the Web based Human Resources Management System. This document also covers the comparison in terms of usability and development of the Web based Human Resources Management System developed as SaaS and desktop based Human Resources Management System with the same functionality developed in desktop platform.

Keywords: Web Application, Human Resources Management System, Web design, UML, UWE, Project Proposal

Table of Contents

1. Introduction 1

2. Objectives and Methodology 2

3. Theoretical Background 3

3.1 The Unified Modelling Language 3

3.1.1 History of UML 4

3.2 The UML Version 2.0 7

3.2.1 UML diagram types 7

3.2.2 Structure Model 9

3.2.3 Dynamic Model 13

3.2.4 Interaction Model 16

3.3 UML-Based Web Engineering (UWE) 17

3.3.1 Model Driven Development and Separation of Concerns 20

3.3.2 Modeling with UML 22

3.4 Software as a service 23

3.4.1 SaaS characteristics 24

3.4.2 SaaS Advantages 25

4. Human Resources Management System 26

4.1 What is a Human Resources Management System? 26

4.2 Structure of Human Resources Management System 27

5. Design of Web Based Human Resources Management System 31

5.1 The Web Based Human Resources Management System Requirement Specification Analysis....34

5.1.1 The Web Based Human Resources Management System Use Case overview 34

5.1.2 Personnel Management Use Case 36

5.1.3 Leave Management Use Case 39

5.1.4 Payroll Management Use Case....42

5.2 The Web Based Human Resources Management System Conceptual Modelling....46

5.2.1 Personnel and Leave Management Conceptual Model 46

5.2.2 Payroll Management Conceptual Model....48

5.2.3 User Profile Conceptual Model....50

5.3 The Web based Human Resources Management System Navigational Model....51

5.3.1 Personnel and Leave Management Navigational Model....52

5.3.2 Payroll Management Navigational Model 53

5.4 The Web based Human Resources Management System Presentation Model....55

5.4.1 Personnel and Leave Management Presentation Model....56

5.4.2 Payroll Management Presentation Model 58

5.5 The Web based Human Resources Management System Conceptual Implementation Process 59

5.5.1 Personnel Management Implementation Model....60

5.5.2 Leave Management Implementation Model......61

5.5.3 Payroll Management Implementation Model......63

6. Comparison 64

6.1 Comparison of proposed system with an existing systems developed in different technologies......64

7. Evaluation......66

7.1 Evaluation of proposed information system 66

8. Conclusion......68

9. Bibliography......69

Table of Figures

Figure 1. From war methods to the UML 5

Figure 2. UML 2 models 8

Figure 3. Association between classes 10

Figure 4. Aggregation and Composition hierarchies 11

Figure 5. Generalization and Specialization hierarchies 12

Figure 6. Example of a state transition diagram 14

Figure 7. Notation of UML use case 15

Figure 8 Notation of UML sequence diagrams 17

Figure 9. Web modelling approaches 18

Figure 10. UWE models 19

Figure 11. Requirements of Web application modelling in UWE 21

Figure 12. Web Based Human Resources Management System Use Case 32

Figure 13. Main Actors in Web based HRMS 34

Figure 14. Personnel management use case diagram 36

Figure 15. “View employee profile” activity diagram 37

Figure 16. Leave management use case diagram 40

Figure 17. “Employee leave request” activity diagram 41

Figure 18. Payroll management use case diagram 43

Figure 19. “Calculate salary” activity diagram 45

Figure 20. Personnel and Leave management class diagram 47

Figure 21 Payroll class diagram..49

Figure 22. User model 50

Figure 23. HR employee user navigation model 52

Figure 24. Payroll management navigation model...54

Figure 25. Web based HRMS home page..56

Figure 26. Personnel management presentation model 57

Figure 27. Payroll management presentational model...58

Figure 28. Web based HRMS home page concept 60

Figure 29. Personnel management web design concept 61

Figure 30. Leave management web design concept 62

Figure 31 Payroll management web design concept 63

Table of Tables

Table 1. Multiplicity types..12

1. Introduction

As result of high cost and limited functionality of traditional software systems, it has been an increased demand of web based information systems. One of these systems is Human resources management system. Due to high demand of small and medium organizations to have a Human Resource Management system which covers all requirements and has low cost, this document covers the design of Web based Human Resources Management System as a replacement of traditional Human Resources Management System. The system has in overall ten modules, but due to complexity there are covered three of them. The difference between Web information systems and traditional information systems is that a huge number of information are organized in a web structure which is served to the users via web pages and hyperlinks. To make this process simpler, for the development and implementation phases, UML as a core modelling language of information systems is not enough to achieve this goal. As a result it is used UML-web based engineering (UWE) methodology to create a conceptual structure of information, navigation structure, and presentation structure of the system. UWE methodology is an extension of UML that covers different design phases of web information system analysis and design.

Nowadays, organizations are located in different places separated in branches, and the need for managing human resources separately is very high, in this case web human resources management system using SaaS technologies will allow HR employees from each branch to manage their own human resources. The main users of the system are Employees of Human Resources department. System contains user roles, where each role has different views in the system. The hierarchy of users is managed by an administrator which is also member of human resources department. This document covers different views of the system from the user perspective, starting from the requirement analysis, when it is described the system and the user interaction using use case diagrams and activity diagrams, and continue with conceptual model, navigation structure and presentation model. Each process uses UML notation and diagrams for presenting the main functionalities of the system as well as user interaction with a system.

2. Objectives and Methodology

In diploma thesis, Information system design in UML is covered by a case study, the design of Web based Human Resources Management System. There are a number of human resources management systems developed as desktop based applications which are mainly used from organizations who are located in one place and have a big number of employees, but this doesn’t cover the demand of the small and medium organizations which are located in different places and have a smaller number of employees. The main objective of this thesis is to analyse and design a type of web information system ready for development, that allows small and medium organizations to manage human resources and as well as cover organization requirements and spend less money using this type of information system.

The objective of the first part of this thesis, is to summarize theoretical background of analysis and design methodologies such as UML and UML-web based engineering used to model web information systems, while the objective of the second part is to describe human resources management system in details, and the demonstration of the processes and the final outcome of the application design using UML diagrams and UWE phases. The core idea of the second part is to demonstrate the advantages by highlighting the particular notation and extensions of the UML focused in web information system design.

The objective of the last and the third part of this document is to present the comparison of the proposed web based information system with the existing information systems created in other platforms (different from web). And also to compare them in terms of technologies. The comparison analysis is meant to be in terms of web information systems having the same purposes, but built on various technologies by emphasizing their diversities.

The methodology of the thesis is based on research of relevant information resources. Practical process will be derived from results which will be given by the research study. Based on the synthesis of the theoretical and the practical knowledge, final conclusions will be formulated. In the first part the necessary information about the UML and web information systems was collected. The further step is to define requirements of the application in order to exactly characterize the system processes. This process is fulfilled based on analysing the specifications concluded in the previous research and study.

3. Theoretical Background

3.1 The Unified Modelling Language

Nowadays, the Unified Modelling language (UML) is widely spread notation for analysis and design of web information systems. UML was created by a group which was called Object Management Group (OMG). The UML is a graphical language which is used for modelling, specifying, construction, documenting and description of complicated information systems. UML is being used independently in the application. It provides some notation elements for static and dynamic aspects of analysis, design and architecture. In particular, UML supports object-oriented paradigms. UML paradigm is a high-level architectural design. It is used for software modeling, hardware modeling, business process structures and systems engineering.

Unified modelling language is described as the successor of object oriented analysis and design. Objects consists of data which represent the state of an object and methods that control these data. Objects are described by a classes (section 3.2.2) which form a hierarchy to model real world system. Objects are the real world entities that exist, which uses UML to be represented. The basic concepts used by UML to represent objects are: abstraction, encapsulation, inheritance, and polymorphism. (Bezivin & Muller, 1998)

UML represents all concepts that exist in object oriented analysis and design. UML diagrams are used for representation of object oriented concepts. So, before going deep in UML, it is very important to understand object oriented concepts in details.

Below are some fundamental concepts of object oriented world (Vrana, 2009):

Abstraction: Abstraction represents the behaviour of a real world entity. Also abstraction can be defined as an act of concentrating the essential or general qualities of similar things.

Encapsulation: Encapsulation is the mechanism of binding the data together and hiding them from outside world.

Inheritance: Inheritance is the mechanism of making new classes from existing ones called superclass. So, the new class inherits attributes and operations from superclass, and also adds his own operation and attributes.

Polymorphism: It defines the mechanism to exist in different forms, so the same operations can behave differently in different classes. Operations are implemented by methods. In this way polymorphism allows us to implement more than one method.

3.1.1 History of UML

The history of Unified Language for object-oriented modelling started between the 80's and early 90's. Before this period, numerous methodologists published different ideas and approaches for object-oriented modelling. This period, by some authors was characterized as a period of "war methods".

Authors and theorists of programming languages bickered among themselves, figuring out which approach is better, and developers ignored all these methods and concluded that none of the methods has reached the level of industry standard. Later, three of the largest specialist in the field, authors of the most popular methods decided to join forces with the purpose to unify their (and not just their own) development in accordance with the social order (Booch, et al., 1999).

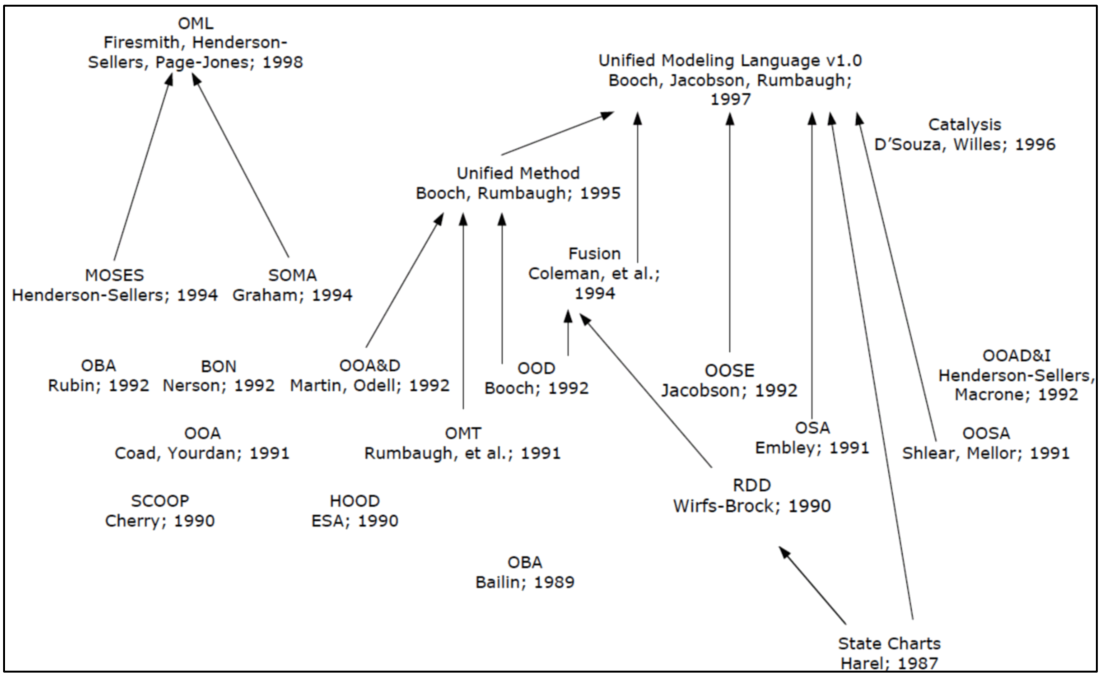


Figure 1. From war methods to the UML *(Source: (Werner, et al., 2006))*

Actually the creation of UML began in mid-1990s. At this time, Gardy Booch and James Rambaugh joined forces at Rational Software Corporation and began to combine several methods of object-oriented modelling and created the first version of the UML.

Booch, Rumbaugh and Jacobson adopted four goals (Bezivin & Muller, 1998):

- To represent complete systems (instead of only the software portion) using object-oriented concepts
- To establish an explicit coupling between concepts and executable code
- To take into account the scaling factors that are inherent to complex and critical systems
- To create a modeling language usable by both humans and machines

The authors of the Unified Method rapidly reached a consensus with respect to fundamental object-oriented concepts. However, convergence on the notation elements was more difficult to obtain, and the graphical representation used for the various model elements went through several modifications (Bezivin & Muller, 1998).

By the end of 1995, they created the first unified specification method, called the Unified Method, version 0.8. These specifications were widely distributed, when user community announced thousand detailed comments which were taken from the authors and in June 1996 version 0.9 was created.

However, it was version 0.91, released in October 1996, which represented a substantial evolution of the Unified Method. The main effort was a change in the direction of the unification effort, so that the first objective was the definition of a universal language for object-oriented modelling, and the standardization of the object-oriented development process would follow later. The Unified Method was transformed into UML (*the Unified Modelling Language for object-oriented development*) (Bezivin & Muller, 1998).

The first version of UML was adopted by a consortium of OMG (Object Management Group) in January 1997. In that year, in September version UML 1.1 was approved which was adopted by major companies – software manufacturers such as Microsoft, IBM, Hewlett-Packard and also manufacturers of CASE-tools that supported UML implementation in their software products (Paradigm Plus, Microsoft Visual Modeller for Visual Basic, Delphi and others).

During the year 1999, OMG (OMG, 2011) decided to extend UML by integrating the XML Metadata Interchange Format - XMI. This standard based on Extensible Mark-up Language for exchanging metadata information whose metamodeling can be expressed in Meta-Object Facility. This extension enriched the UML.

In 1999, the UML integrated the XML Metadata Interchange Format - XMI. A standard based on the Extensible Mark-up Language. In this way, Unified Modeling Language has become the symbol of a single graphic in a textual notation in a notation with the same cardinality (Bray, et al., 2006).

The experience gained by the users continued to increase and much more users are using the UML through recent years, so that experience has a big influence in recent versions and finally led to the launch of the UML Version 2.0 in 2005 continuing with the current working revision 2.4 launched at the beginning of 2010 (Werner, et al., 2006).

3.2 The UML Version 2.0

After all the effort of OMG to create a usable version of UML, there were several reasons that have driven to the development of new releases in UML. The main reasons were complains about the complexity of existing versions. Technology was evolving very fast and new software engineering techniques introduced new requirements to description methods. OMG gathered all the information and the proposal a new version of UML, developed between 2000 to 2003, with the specification to UML Version 2.0 which was adopted by OMG shortly thereafter (OMG, 2010).

In general UML specification is organised into two volumes *UML 2: Infrastructure* (OMG, 2011) *and UML 2: Superstructure* (OMG, 2010), consistent with the breakdown of modelling language requirements into two RFPs (OMG, 2010). According to the OMG these two volumes cross-reference each other and the specifications are fully integrated, which could easily be combined into a single volume at a later time (OMG, 2010) (OMG, 2011). The Infrastructure document describes basic language constructs and the architectural foundations. Based on this, the Superstructure document describes the diagram notation and its semantics. These two documents were gradually evaluated and due to delays the superstructure document has been finalized in mid-2010 while the Infrastructure document is still under review. Both specifications have been improved due to the time and in mid-2010 UML 2.4 and UML 2.4.1 were already underway (OMG, 2010) (OMG, 2011).

3.2.1 UML diagram types

UML is compound of several diagrams that express static and dynamic aspects of an application. Static aspects are related to the structure of the system, being true all the time. The purpose is to describe the entities of a system and how they will always be related. On the other hand, dynamic aspects refer to the evolution of the application: the creation and destruction of objects and their connections over time; formally, the transformations in the global state (*the set of objects and relations*) of the system (Borges & Mota, June 2007).

The latest UML 2.4 offers fourteen diagram types. Six of them describe the static part of the system which are part of Structure diagrams. The other diagrams describe the dynamic part of the system which are part of behaviour diagrams. These diagrams are categorized into interaction diagrams. UML 2.4 has introduced some new diagrams which are not part of the earlier versions of UML (Werner, et al., 2006).

- The package diagram,
- The interaction overview diagram,
- The timing diagram and
- The composite structure diagram.

All types of UML 2.4 available diagrams for each model are shown in figure below.

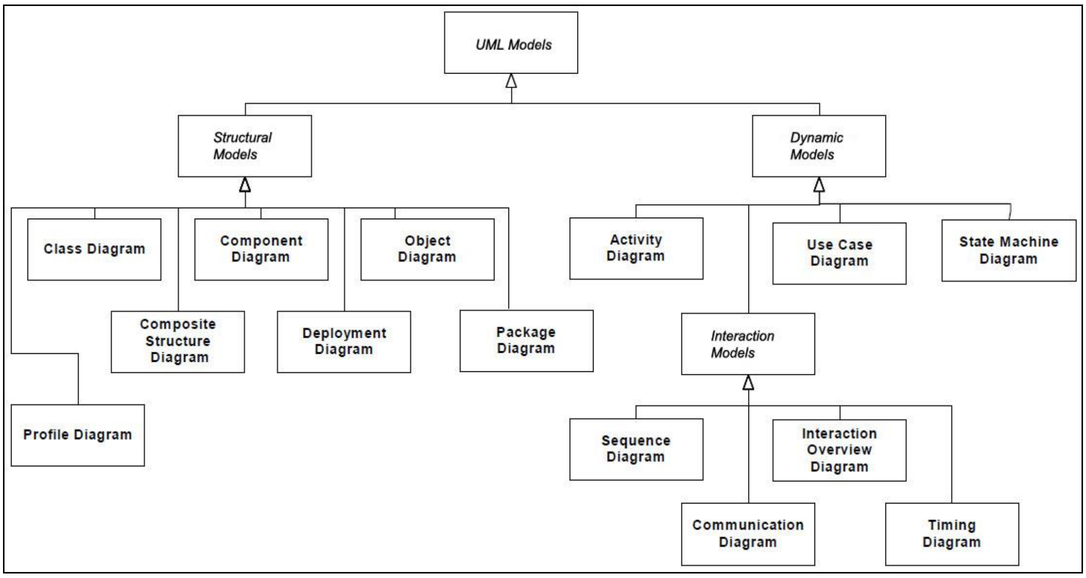


Figure 2. UML 2 models *(Source: (OMG, 2010))*

The UML notation standards are defined by many elements with different semantics and complexity for each aspect. All these elements in UML are grouped into these diagram types, which are many, and each of them expresses different part and behavior of the system. For instance, attributes, operations, associations, and relation between classes define the group consisting of classes, which are part of the structure diagram. Relation can be a different type, such as association, composition, aggregation, generalization etc. These represents the basic class diagram structure. Each UML diagram has its own representation and elements.

3.2.2 Structure Model

Structure diagrams define a static architecture of the model. They are used to model parts that make up a model – the classes, interface and physical components. Structure diagrams are used to model the relationship and dependencies between elements. These diagrams are a type of diagrams that shows the elements of a specification which are irrespective of time (Werner, et al., 2006). There are several types of diagrams that are part of structure diagrams. Diagrams that are part of structure diagrams are: Class Diagrams, Package Diagrams, Profile Diagrams, Composite Structure Diagrams, Deployment Diagrams, Object Diagrams, and Component Diagrams. In this document some of the relevant structure diagrams are described briefly.

Class Diagram

Class diagram is the most common diagram found in modelling object-oriented systems. A class diagram shows a set of classes, interfaces, and collaborations and their relationships. Class diagrams address the static design view of a system. Class diagrams that include active classes address the static process view of a system (Booch, et al., 1999).

While designing a system, a class diagram is used to show the existence of classes and their relationships in the logical view of a system. A single class diagram represents a view of the class structure of a system. During analysis, we use class diagrams to indicate the common roles and responsibilities of the entities that provide the system's behaviour. During design, we use class diagrams to capture the structure of the classes that form the system's architecture (Booch, et al., 2007).

Classes are the main entities of an object-oriented application and are modelled as rectangle divided into counterparts (Pilone & Pitman, 2005). A typical class is divided into three compartments such as: the name of the class (in bold), its attributes, and its operations (Fowler, 2003). We can find many definitions from various authors which in general describe classes. In this contests Tim Weilkiens in his book Systems Engineering with SysML/UML described class as:

> *"A class describes structure and behaviour of objects that have the same characteristics and semantics. The structure is described by attributes, while behaviour is described by operations."*

Attributes are stored information of an object while methods written describe what an object or class can do. An attribute defines a structural property of a class. The description is composed of visibility, name, type, and a multiplicity (Weilkiens, 2008). Methods are the object-oriented equivalent of functions and procedures. They define the behaviour of an Operation (Werner, et al., 2006).

Classes and Objects represent the core in object orientation. A class is the smallest building block of an object-oriented system. Classes are interconnected by associations to form larger structures. Objects are concrete elements that are created based on the building plans of classes, in this context object is an instance of a class (Weilkiens, 2008).

Association

An association is a structure relationship between two classes. It is described as a line connecting the two classes such as the connection between class A and class B in the figure 3.

The association is characterized by its name and optionally a black arrow directed from destination to source, otherwise or both sides (Gomaa, 2011).

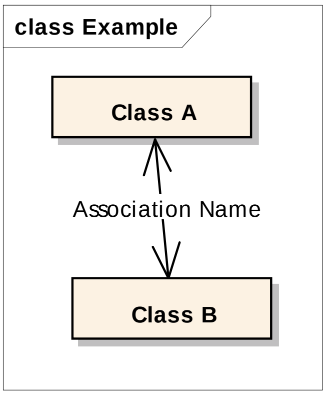


Figure 3. Association between classes *(Source: (Gomaa, 2011))*

Associations are closely related to web modelling when the arrow can show the navigation. There are different types of association which describe different behaviours of the system.

Aggregation and Composition Hierarchies

Aggregation and Composition are two types of Association. An aggregation shown in Figure 4a defines the whole-part relationship between classes. Aggregation is a stronger version of association. Unlike association, aggregation typically implies ownership and may imply a relationship between lifelines. Aggregations are usually read as "...owns a..."(Pilone & Pitman, 2005).

Whereas Composition represents a strong relationship between two classes. Composition is used to capture a whole-part relationship. The "part" piece of the relationship can be involved in only one composition relationship at any given time. A composition relationship is usually read as "...is part of..." which means you need to read the composition from the part to the whole (Pilone & Pitman, 2005).

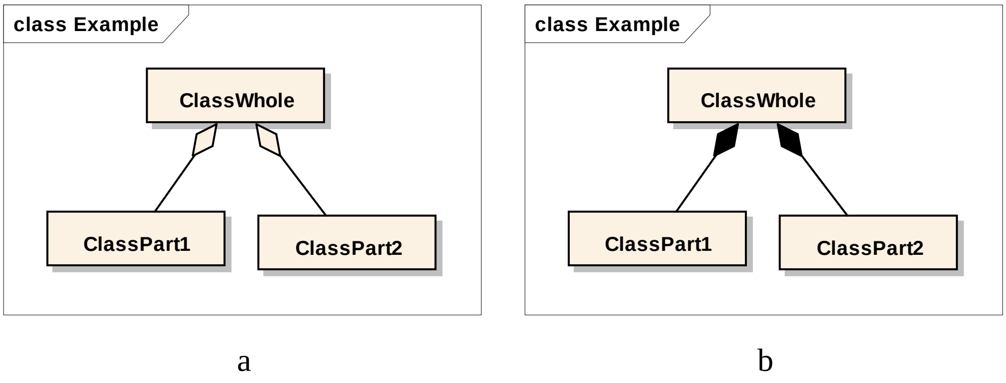


Figure 4. Aggregation and Composition hierarchies *(Source: (Gomaa, 2011))*

Generalization / Specialization Hierarchy

A special relationship between classes is Generalization or Inheritance and Specialization. Generalization is a relationship between special classes to a general class (superclass). Generalizations are usually read as "...is a...” starting from the more specific class and reading toward the general class. Generalizations are different from associations, because the relationships are not named and don't have any kind of multiplicity (Pilone & Pitman, 2005). Generalization is shown in Figure 5.a. On other hand, Specialization is the opposite of generalization, the direction is from superclass directed to subclasses as shown in figure 5b.

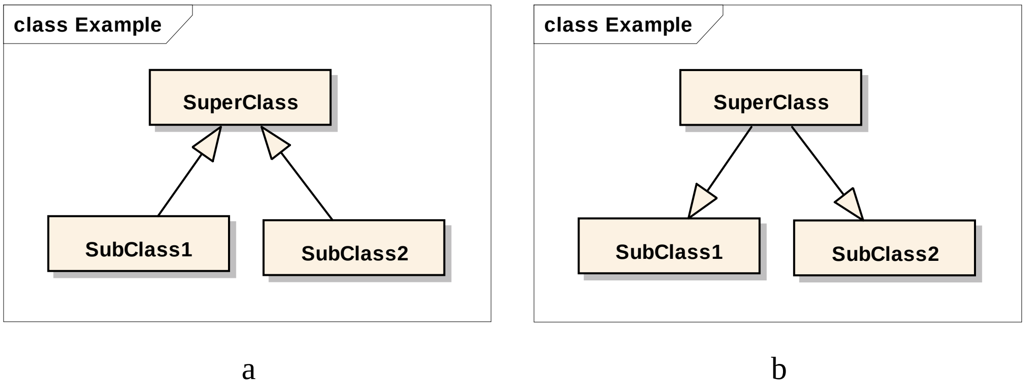


Figure 5. Generalization and Specialization hierarchies *(Source: (Gomaa, 2011))*

Multiplicity

Two other important features in classes is multiplicity. Multiplicity describes the number of objects that are involved in an association. In practice there are three common kinds of multiplicity across an association, but in general there are more options. In the table below we will see the all kinds of multiplicity (Grässle, et al., 2005).

Multiplicity	Option	Cardinality
0..0	0	Collection should be empty
0..1		There are no instances
1..1	1	Only one instance
0..*	*	Zero to more instances
1..*		At least one and more instances
5..5	5	5 instances
m..n		At least m but no more than n instances

Table 1. Multiplicity types *(Source: (Grässle, et al., 2005))*

Often the association between classes is complex. The relationship is not presented simply by an association line, but in the middle another class is introduced which is called an association class. Association attributes in are justified mainly in many-many associations.

Interface

A useful concept which can be used to support abstract specification of a system, and modular development, is the notion of an interface. An *interface* is essentially an abstract class which contains a list of operations, which clients of the interface will require, and which implementations of the interface must implement.

Interfaces are written as class rectangles with the stereotype <<interface>> above or beside the interface name. An interface cannot inherit from a class, but it can be the endpoint of an association from a class (Lano, 2005). Interface features must be public.

3.2.3 Dynamic Model

Dynamic diagrams define dynamic behaviour of the system. Dynamic models include Use Case diagrams, State Machine diagrams, and Activity diagrams, also Interaction diagrams which include Sequence diagram, Communication diagrams, Interaction overview diagrams and Timing diagrams. In general they emphasize what happens in the system or business process. They are used to describe the functionality of the system. (Wazlawick, 2014)

State transition diagram

State transition diagrams represents a set of states in which a system, actor, or entity could be at a given instant. The state machine diagram specifies events in its flows. An event triggers the flow that takes the entity from one state to another. (Wazlawick, 2014) It describes the possible timely sequence of states and actions through which an active object can go during its lifetime as a result of its reaction to discrete events. A state reflects a situation during the lifetime of an object in which this object performs some actions or waits for some event to occur. According to the UML metamodeling, states can belong to one of the following categories: simple states, composite states and submachine states (Werner, et al., 2006).

A state transition is a graph of states and transitions. State transition when it is constructed it is attached with class and describes the response of an instance of a class to events that it receives. It can also be attached to behaviours, use cases, and collaboration to describe their execution. A state transition defines a small number of named states. A state can be characterized in three complementary ways: as a set of object values that are qualitatively similar in some respect as a period of time during which an object waits for some event or events to occur: or as a period of time during which an object performs some ongoing do activity (Rumbaugh, et al., 2004). Example of a state machine diagram is shown below:

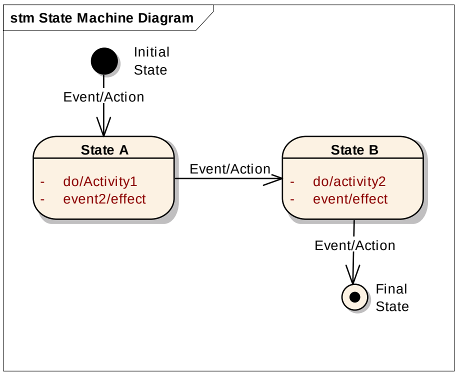


Figure 6. Example of a state transition diagram *(Source: (Gomaa, 2011))*

A Relationship between two states is called Transition. A transition may have a trigger, a guard and an effect. A trigger is the cause of the transition to execute. Guard is a condition which must be satisfied in order to enable the trigger to cause the transition. Effect is a set of actions that will be invoked on the object that owns the state transition resulting from the triggered transition (Werner, et al., 2006).

Use Case Diagrams

A use case is initiated by actors, which define a sequence of interactions between the actor and the system (Gomaa, 2011). A use case diagram is a list of actors and use cases, which shows actors participating in each use case. An actor is represented as a stick figure, and system is represented as a box, while a use case is represented as an ellipse inside the box (Rumbaugh, et al., 2004). The communication associations connect actors with the use cases in which they participate. Relationships between use cases are defined by using relationships such as include and extend and generalization. In figure 7 a UML use case diagram notation is shown.

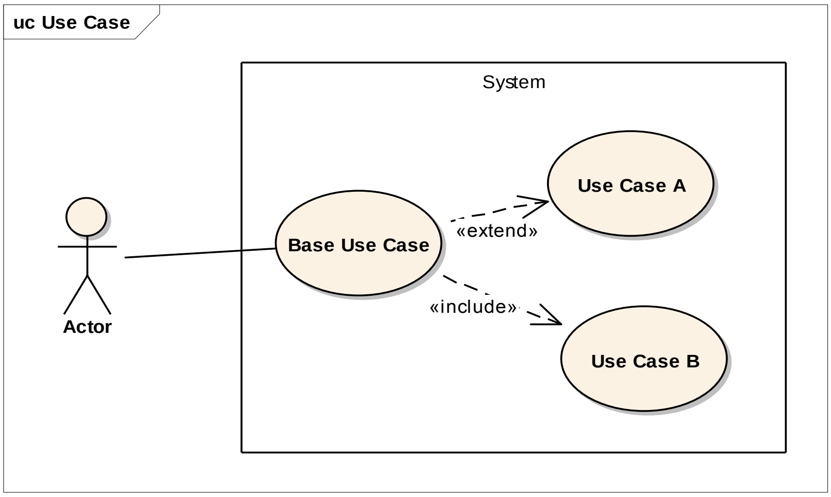


Figure 7. Notation of UML use case *(Source: (Gomaa, 2011))*

An include relationship between use cases means that one use case can include and use other use cases in an explicit defined place in its description. A notation of include relationship is represented with stereotype «include». While an extend relationship between use cases means that the base use case is extended with additional behaviour by the extending the use case. The extend relationship is represented by stereotype «extend». And the generalization between use cases is the same as the generalization between classes (Eriksson & Penker, 2000).

Activity diagram

Activity diagrams are used to describe the sequence of activities of internal processing as well as action object flow. UML Activity diagrams can also be used for modelling the dynamic aspects of systems and for constructing executable systems through forward and reverse engineering. Activity diagrams emphasize the flow of control from activity to activity. An activity is an ongoing non-atomic execution within a state machine (Gogolla & Kobryn, 2001).

According to OMG UML Specification (OMG, 2011) an activity diagram is a special form of state machine intended to model computations and workflows. The states of the activity diagram represent the states of executing the computation, not the states of an ordinary object. An activity diagram contains activity states which represent the execution of a statement in a procedure or the performance of an activity in a workflow (Gogolla & Kobryn, 2001). An Activity diagram notation is a box with rounded ends which contains the description of an activity.

There are two common control flow nodes in the activity diagram: decision nodes and parallelism nodes. The decision nodes (branch and merge nodes) are represented by diamonds, and parallelism nodes (fork and join nodes) are represented by bars (Wazlawick, 2014). Activity diagrams in web development methodology show different flows. In the web modelling methodology activity diagrams show the interaction between clients and server, also possible user interaction with clients' side, and business logic in the server side (Stevens, et al., 2003).

3.2.4 Interaction Model

Interaction models are part of behaviour UML diagrams (OMG, 2011). Interaction diagram depicts how objects interact. Interaction diagrams describe the sequence of messages exchanged between the objects, only in the context of the main loop (Muller, et al., 2004). Usually the most used interaction diagram is Sequence diagram. The other diagrams which are part of interaction models are Communication diagrams, Timing Diagrams and Interaction Overview diagrams. In general interaction diagrams objects are described in rectangular boxes (Gomaa, 2011).

Sequence Diagram

Sequence diagrams are used to show the interaction among objects, which depicts objects arranged in time sequence. A sequence diagram is a two dimensional diagram in which the participating objects in the interaction are shown horizontally, and in vertical position is shown the time which we refer as *lifeline* (Gomaa, 2011). As long as an object exists in time, the role is shown with dashed line, and as long as the activation of a procedure on the object is active, the lifeline is shown with double line. A message is shown with arrow from the source to a destination object (Rumbaugh, et al., 2004). In UML 2 has extended the notation of sequence diagrams to allow *for loops* and other conditionals (Gomaa, 2011). In figure below a typical sequence diagram is shown.

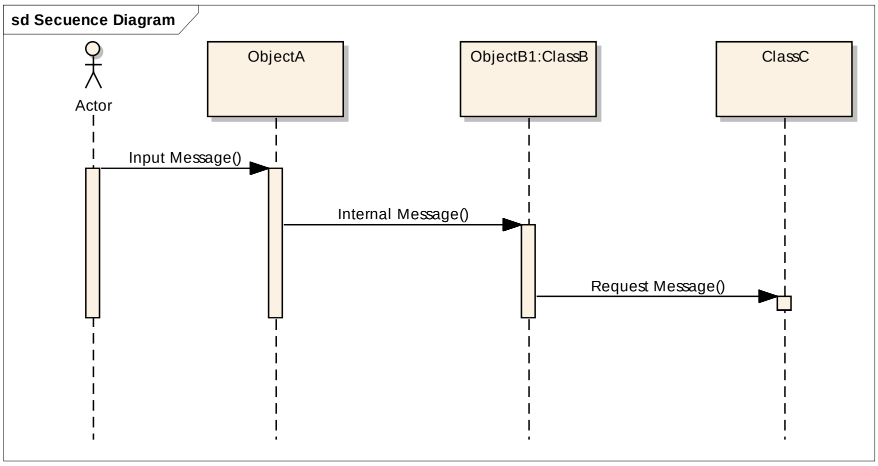


Figure 8 Notation of UML sequence diagrams *(Source: (Gomaa, 2011))*

3.3 UML-Based Web Engineering (UWE)

For years, the World Wide Web has become a platform for the execution of all applications that meet client server functions. From static pages, the Web has evolved into incorporating elements of security, optimization, concurrency and other requirements that are necessary to create solid solutions. However, the development of a Web application includes elements that are not common to a desktop application. This requires major changes in the way of performing and supervising the development process, i.e. move from Software Engineering to Web Engineering.

During the years, a big issue was the development of methods, which will cover the design, and development of web applications. UML is a modeling language with covers different parts of software development, but does not include methods and models for modeling web application aspects such as web navigation, presentation model (Kappel, et al., 2006).

Based on the need to create web modeling methods, as a result, different approaches have been introduced. The aim of model driven development is to provide a concept by modeling languages for the specification and design of web applications (Kappel, et al., 2006). There are now a large number of modeling languages, as is shown in Figure 4. Many approaches are based on a process model and provide tool support, thus, the generated model can generate a running prototype automatically.

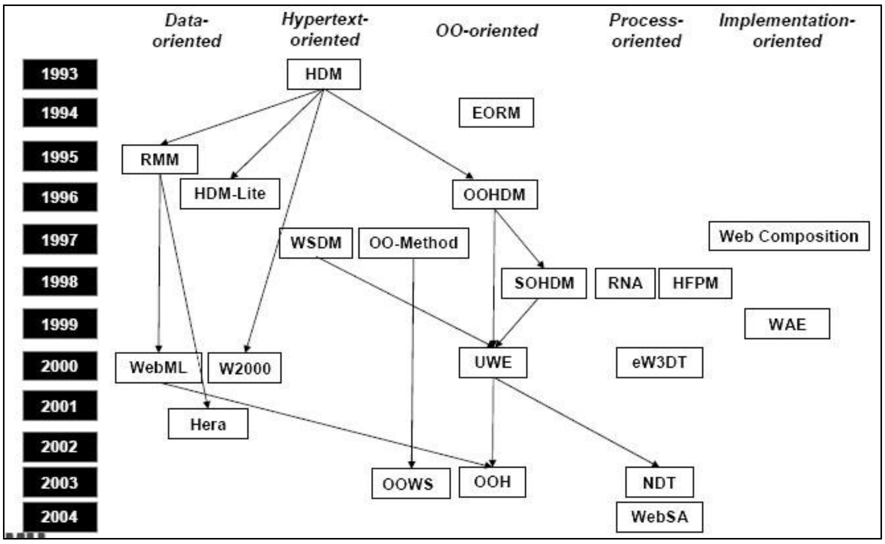


Figure 9. Web modelling approaches *(Source: (Koch, et al., 2001))*

One of these approaches which is Object Oriented is UML-Based Web Engineering UWE.

UML-based web engineering has been introduced during 1990 by Hubert Baumeister (Baumeister, et al., 1999), based on idea of finding a standard way for developing analysis, design and implementation of Web based applications, by combining pre developed methods such as OOHDM introduced by Schwabe and Roosin in their publication in 1999 (Schwabe & Rossi, 1995), and RMM introduced by Isakowitz, Edward and Papyrus (Isakowitz, et al., 1995) . The aim, which is still in development, was to create a common language or at least to define meta-model-based mappings among existing approaches (Koch, et al., 2008).

One of the first methods developed was the UML-based Web Engineering (Koch, 2006). UWE is a methodology to better specify a web application creation process (Koch, et al., 2008) maintains a standard notation based on the use of UML (OMG, 2011) to their models and methods, thus facilitating transition.

The methodology defines clearly the construction of each of the model elements. In its implementation, the following stages and models should be considered (Koch, et al., 2008):

- Requirements analysis. Express functional requirements of Web application through UML use case and activity diagrams.
- Conceptual model. Defined by a diagram of classes, concept details involved in the application.
- Navigation model. Represents the navigation of objects within the application and a set of structures as indexes, menus and consultations.
- Presentation model. Represents user interfaces through abstract views.

The Process models represent activity aspects, which are connected with different type of processes. As noted, UWE provides different models for describing a web applications several abstract points (Koch, et al., 2001). These models are related as shown in Figure 4. Each of these is represented by UML models (OMG, 2011) such as packages; those packages, which are related, may be refined in successive iterations that UWE processes during development (Koch, et al., 2001).

The requirements analysis in UWE is modeled using use cases. It is developed by the actor and use case elements. In this regard, actors are used to model users of the Web application.

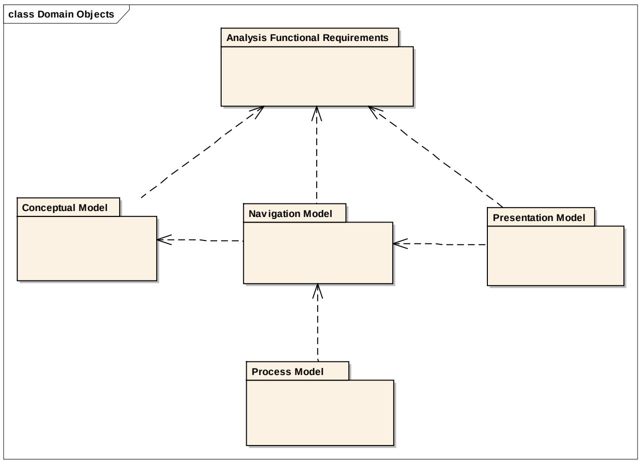


Figure 10. UWE models *(Source: (Gomaa, 2011))*

The content model is the conceptual model of the application domain taking into account the requirements specified in the use cases and activity diagrams (Koch, et al., 2001) which is represented by a class diagram. Based on the requirements analysis and the content model, the navigation model is obtained. Navigation model is represented by navigation classes that will be explained in our Web application designed in this document. Based on the navigation model and the aspects of user interface (requirements), the presentation model is obtained. This model describes the structure of user interaction with the Web application. The navigation model can be extended by process classes. The process model represents the way they look at a class action process.

3.3.1 Model Driven Development and Separation of Concerns

The evolution of Web systems development is a subject in continuous changes in user and technology requirements (Koch, et al., 2008). Web applications built in stages of the development process should be easily adaptable to these changes. So the scope of UWE methodology is to advocate a strict separation of concerns in the early phases of the development and implements a model driven development process. The ultimate challenge of UWE methodology is to support a development of a process that allows the automated generation of Web Applications (Koch, et al., 2001).

Same as other Web engineering methodologies the UWE method separates modeling of concerns, which describe WEB application design and development. Web models belong to the different phases of WEB application such as requirements engineering, analysis, design and implementation of the development processes, which represent different views of the web application corresponding different concerns such as content represented by conceptual model, navigation structure, presentation and implementation (Koch, et al., 2008).

UWE uses UML notation for representing different aspects of a web application. The structural aspects of different views of web application proposed by UWE use UML diagrams for visualization of each model. Mostly used diagrams for visualization of web application aspects are Use case diagrams, Activity diagrams and Class diagrams. However, UML interaction diagrams or State Machine diagrams, are used to represent the aspects of behaviour of the web applications (Koch, et al., 2008).

Figure 11 shows three levels of Web applications modelling, differently from two levels used in modelling of traditional applications. The three levels are content, hypertext and the presentation, which are represented in three orthogonal dimensions: development phases i.e. analysis, design and implementation, Views i.e. content, navigation structure and presentation, and Aspects i.e. behavior and structure, which are represented by UML diagrams. Many methods, which are used to model Web applications, follow this separation of levels (Schwinger, et al., 2006).

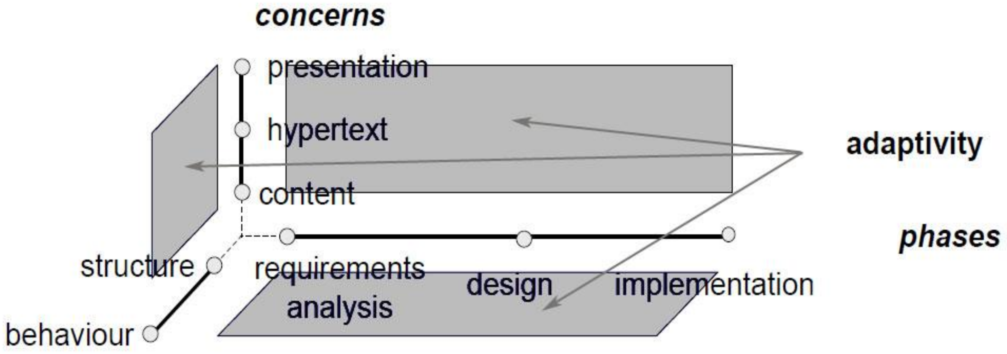


Figure 11. Requirements of Web application modelling in UWE *(Source: (Schwinger, et al., 2006))*

A separation of the web application parts such as content, hypertext and presentation allows better understanding of each concern and helps to reduce complexity. For example, the content of the application which is interpreted by conceptual model deals exclusively with a structure of the web application, whereas hypertext or navigation structure and presentation have to do with navigation and presentation of the web application. To design an efficient navigation, content has to be offered redundantly on several nodes in the navigation levels. Due to separation of concerns, content or conceptual model is modeled by the class diagrams, and the navigation structure model just references the corresponding content or conceptual diagrams. Therefore, in this way users can find the information over several access parts in the web application (Schwinger, et al., 2006).

In addition, when modelling the presentation levels, the main purpose is to uniform presentation structure for the pages to archive the highest level for the navigation of users. Although, the visual appearance of a web application is very important and for this reason UWE method has its mechanism to study separately (G. Kappel & Retschitz, 2004).

The model driven development is an approach that advocates the use of models for the web application development, and emphasizes the need of change in all phases of development from the requirements specification to implementation of web applications. The changes between models provide a way that make it possible to automate implementation of an application with success for different models (Koch, et al., 2008).

3.3.2 Modeling with UML

The distinctive feature of UWE is its UML compliance since the model elements of UWE are defined in terms of a UML profile and as an extension of the UML meta-model (Koch, et al., 2001).

UML modelling language expresses enough options to model web requirements, which arise in Web application development; it is not enough and does not offer specific elements for modelling web applications. To extend the modelling in order to have opportunity to concentrate in special aspects of web applications the UWE special views are defined which are considered as an extension mechanism of UML (Koch, et al., 2001).

UML modeling language techniques consist of constructing static and dynamic views of the systems using different diagrams such as object and class diagrams, deployment and component diagrams, use case, state, and activity diagrams, sequence and communication diagrams (Koch, et al., 2008). The mechanism of UML extensions is used to define stereotypes that can be used to represent specific parts of web construction. The tags and definitions written by Object Constraint Language can be used. UWE notation (Koch, et al., 2001) use pure UML tags, such as class diagrams, activity diagrams, associations, etc. this can explain better the concept of UML extensions, which inherit all properties and problems of UML such as lack of complete formal semantics covering all modeling elements (Koch, et al., 2008).

The UWE extension objective is to cover all aspects of modeling and development of web applications using UML diagrammatic techniques and its own notation. The life cycle of web modeling changes from standard modeling, web-modeling starts from requirements modeling, and continues through design models, including aspect and architecture models.

3.4 Software as a service

Nowadays cloud computing is very popular in the software technology industry which includes several categories of services, which are all offered on demand over the internet in a pay-as-you-go model. One of these categories is Software as a service (SaaS), the other categories include platform as a service (PaaS) and infrastructure as a service (IssS) (Spence, et al., 2009).

Software as a service (SaaS) had an effect which changed entirely the development and implementation of software applications. Software as a service (SaaS) has changed the way software applications are hosted, provided to the customers and how they have access to them. Traditionally the software applications would be installed in the machine from which they could be accessed only. However SaaS has changed this by hosting the software applications in the remote hosting, while the users can be access them remotely, from their personal computers via Internet (Ivanka Menken, 2009).

SaaS is considered as a software business model, where software is not offered as a product, which has happened with traditional software applications, but as a service. The web application runs on a web server which the end users use it through internet connection (Liu, et al., 2010). The advantage for users is that they don’t have to pay for the whole software application, they only pay for using it. Following todays advanced infrastructure and software technology, a new era of application development has started. These applications are characterized by robust web based applications designed as centralized, shared-instances, multi-tenant applications called software as a service (Liu, et al., 2010).

A new design, development and implementation aspects are driven by SaaS today. There are several aspects that should be considered while designing a system which is dedicated as software as a service. These considerations focus on SaaS and its properties. Some of the aspects that should be considered are (Hogg, et al., 2004):

- concurrent multi-tenant high-performance
- mass data storage and access
- unpredictable security threats from the Internet,
- a variety of individual needs
- customized and convenient user experience
- reliable trust from users
- how service providers sustain operations and
- maintenance management
- seamless opt-in upgrades
- open API for integration

3.4.1 SaaS characteristics

Software as a service (SaaS) is a platform software-delivery model which is named in many ways. Different researches define it as on-demand, on premise, subscription software or pay-as-you-go. The main characteristic of SaaS is that the software is delivered as a service to the clients.

Nowadays this methodology is used in apps and new web application development. SaaS offers a software in web-based form, so the clients are able to access in the system via web browsers to the application which is installed in a data centre somewhere connected to the internet (Anerousis & Mohindra, 2006). Another characteristic of SaaS platform is that it allows the software to create multiple instances, where each of them can be connected with its own database. Based on this feature, our concept of design of Web based human resources management has begun. Furthermore, SaaS use a multi-tenant architecture which allows multiple users to use the same application at the same time. In addition, SaaS is delivered through frameworks which make possible the changes and customization of a web application according to the user requirements without interrupting the other users who use the same software (Anerousis & Mohindra, 2006).

3.4.2 SaaS Advantages

Software as a Service offers a big advantage for both the provider of the application as well as the client. The primary advantage that SaaS offers for providers is that it reduces cost for maintenance of the application (Liao & Tao, 2008). So, since they are able handle all clients, they don’t need to go and visit personally their client, they can provide services online (Chou, 2007). Another advantage for providers is that SaaS helps them to get on faster in software updating in order to improve their competitiveness in the market (Liao & Tao, 2008). Due to the fact that application is offered online through internet, providers can update new versions faster (Waters, 2005).

The advantage of using SaaS also affects the customers or users. One of the main advantages is that SaaS helps the customer to save money in all aspects starting from hardware devices, network devices, and also software application (Anerousis & Mohindra, 2006). Also, SaaS helps clients to decrease the number of potential employees which maintain systems, when providers of the software take care of this issue (Liao & Tao, 2008). Another very important advantage is that SaaS reduces the risk resulting from software acquisition. Also, clients are able to increase the number of users due to the multi-tenant architecture which is a very important feature (Kaplan, 2007). There are many other advantages in both aspects, technical and usage aspects.

4. Human Resources Management System

4.1 What is a Human Resources Management System?

Organizations treat personnel (employees) personal data as any other asset or resource. They must be written, organised, and managed effectively in order to have quality information that can be found and used when it is needed. In the organisations, there can be many employees who are currently working, also there can be many others who will leave and come. There is a huge flow of personal data which can be organized only by development of new Management System called Human Resources Management System.

An effective HRMS, requires adequate updated personal data on actual employees within the organisation, as well as employees who leave and potential employees in the labour market. Information Technology has improved the way this data is gathered, and organised by storing them in the Human Resources Management System (Kavanagh, et al., 2011). HRMS includes processes and proper systems in order to connect the Human Resources Management with Information Technology. Organizations know the importance of this Management System, which is one of the first systems which are implemented in the organization right after ERP and CRM.

HRMS is defined as “An Integrated system used to gather, store and analyse information of human resources within the organizations, comprising of databases, computer applications, hardware and software necessary to gather, store, manage, present and manipulate human resources data (Hendrickson, 2003). Within the organization there is a separate department which controls the human resources. HRMS development follows the needs of the HR department itself (Ruël, et al., 2004), but the use of HRS does not provide benefits only to the HR department, it can also be used from other groups of the organizations such as management and employees.

HRMS touches the overall business performance by contributing with quick and effective services such as monitoring, reporting, statistics, and serving administrative support. HRMS has an important role for the organizations by its ability and effectiveness to manage human resources assets. Many organizations spend a lot of money to develop and implement HRMS to assist the HR department in their daily operations. HRMS must satisfy the needs of the organization and its users in order to be successful (Noor & Razali, 2011).

4.2 Structure of Human Resources Management System

HRMS applications are database oriented applications which allow users to gather, store and tack different type of data which have to do with human resources assets in the organizations. It’s important to say that organizations spend a lot of money buying HRMS software which they will not fully use: e.g. need only modules such as Personnel Module or payroll. The use of software solutions differ according to the needs of the company. A proper Human Resources management system includes many different modules which are connected together by using the same database. These are systems that are used to collect, analyse, store, and manipulate HR information.

HRMS applications contain the following modules:

- **Personnel Module** is the main module of HRMS which keeps all personal data of all employees. Employees’ information are standardized and there are plenty of data which can be saved by the company and are very important for each organization to have them. For example, if an organization which has 1000 employees needs some information whether the employee has a certificate for a particular task, the only way to find it, is a search in the standardized records.
  This module usually has personal data of the employee such as full name, date of birth, marital status, contact information, emergency contacts, family members, data of pervious jobs (job history), characteristics of employees, information of disciplinary actions, work injuries, trainings and certificates and other information regarding the employees.
- **Employee Self Service** is a module which allows all employees of the organization to have access and together with HR department professionals to manage the employees’ database. Employees access this module with their personal username and passwords created by the HR, with the limited permit to

access their data such as read-only or edit status. Employee self-service module allow access to the employee's personal information and payroll details. This module allows users to change their information such as contact details, family members, benefits and banking information.

- **Notification Module** helps the organizations and particular departments to speed up the tasks for the particular work. For example: if there is a new hiring there should be many departments informed, such as security department for the security issues, management of the organization, card department, and many other departments while each of these departments has their separated tasks. In order to speed up the tasks there is notification module which will inform them immediately when the new employee is added in the system by HR department.
- **Payroll Management** is designed to keep track and process payment of employees according to employee attendance. Calculations include various deductions and taxes. Each periodic calculation is followed by a pay check and employee tax report. Calculation is different for each organization and differs from Country to Country. This varies from internal regulations of the organization and laws of the particular state. Payroll module most of the time is connected with financial management systems which directly affects the work of the financial department. This HRMS system is designed in that way that it is not possible to directly connect with the financial management systems of the company while special access to these systems is needed. Special reports are needed in order to make this module effective. These are special reports for the financial department which are possible to import into the financial management systems.
- **Performance Management** is one of the most important modules in organizations development and effectiveness because it monitors and evaluates the performance and work of their employees. In the beginning of a year the company creates its objectives for the future development and where the objectives are met and progress is made during the year, this module will have all this data. The overview of this process is not only about criticism of the employees and the managers but also monitor the performance of human resources assets by checking the fulfilment of their objectives in the end of the year.

- **Training Management** is a very important module which keeps track of the employee certification, training, and development held inside the organization as well as outside the organization. This module will help HR department to keep track of the qualification of their employees by updating their personal profile separately for each employee, as well as managing internal courses organized by the organization.
- **Vacation / Leave Management** contains the holiday days of the employees. Also keeps track whether the employee is on holiday leave, or sick leave or missing without a reason. This module has two parts, employee part and HR part. Employee part has to do with the requests of employees for holiday, the HR part has to do with controlling and decisions whether the employee should go on holiday.
- **Recruiting Management** is a module which allows HR department to open a new work positions and allows candidates to apply for a certain open position in the organization. The information's required from the system which will be added from the applicant and collected by HR department, in this way they will track all the applicants who will apply for a particular position. After the collection of all applications, the system allows the HR department to follow the procedure of the hiring till the end. This includes selection of the appropriate candidates, monitoring of the interviews and the score assigned by the staff responsible for the evaluation. After the process finishes and the selection of candidates is finished, the information will automatically be sent to the employee data and will be saved in the HRMS.
- **Internal Recruiting Management** is a recruiting module inside the organization. There are plenty of open position inside the organization which are opened for the current employees. This application allows HR department to monitor this positions and keep track of the employees and their move inside the company.
- **Compliance Management** allows employees of the organization to write their compliance for employees, managers and other staff inside of the company. This module is directly connected with HRMS and allow higher management to make decisions according to the employee suggestions. Also this is an opportunity for

employees to say everything they agree or disagree with organization structure and employee behaviour inside the organization.

- **Time Attendance Management** is a collection of the attendance of employees in the work. The relation with HRMS is very important because it shows the performance of employees, and also affects the payment of employees for those whose base payment are the hours. This module collects all the employee time that they have been in their job and directly communicates with Payroll module.

In addition, due to the size of the web application there will be a few modules which our designed system will have. All of this modules are separate modules where each of them covers important parts of the organization structure. It depends from the organizations needs what module they will use, and also it depends how many users will access it.

5. Design of Web Based Human Resources Management System

Web based Human Resources Management System (Web based HRMS) is an online web application for managing human resources of the organization. Web based HRMS consists of a number of modules, which are described in section 4 in this document. These modules are designed in that way that organizations can use them depending on their needs. The idea of designing Web based HRMS can be better understood by presenting the following questions:

- Organizations have different branches in different places? How can each branch manage their own employees?
- Small and medium companies nowadays are incredibly growing in all fields? How are they managing their employees?

By evaluating organisations which have different branches distributed in different places, while their main human resources department is located in one place, it is obvious that these companies need a Web information system to have control of all their employees wherever their location is, and whatever department they work in. Web based HRMS can be accurate in IT industry, Banking Industry, Mining Industry, Agriculture organizations and many others. The other reason that Web based HRMS is needed to be analysed and designed is that it fulfills the high demand of small and medium organizations. Many of these organizations nowadays do not have an appropriate software for managing their employees, they use table-based software like Excel to register and keep track of their employees. Web based HRMS is an appropriate solution to fulfill their needs while they do not have to spend a lot of money in HRMS applications, which will take time for implementation.

Web based HRMS will help these organizations financially and by allowing them to manage their employees professionally while they only pay for services of the information system. The payment will be according to the modules they need and the number of users they have.

In general, analysis and design of Web based HRMS increase the productivity of both organizations, customers (users of information system) and the organization who analyze, design and develop the information system.

Web based HRMS has ten modules, described in details in Section 4, and each of them designed and developed for different purposes. In the figure below the use case diagram for the whole information system and its modules is presented.

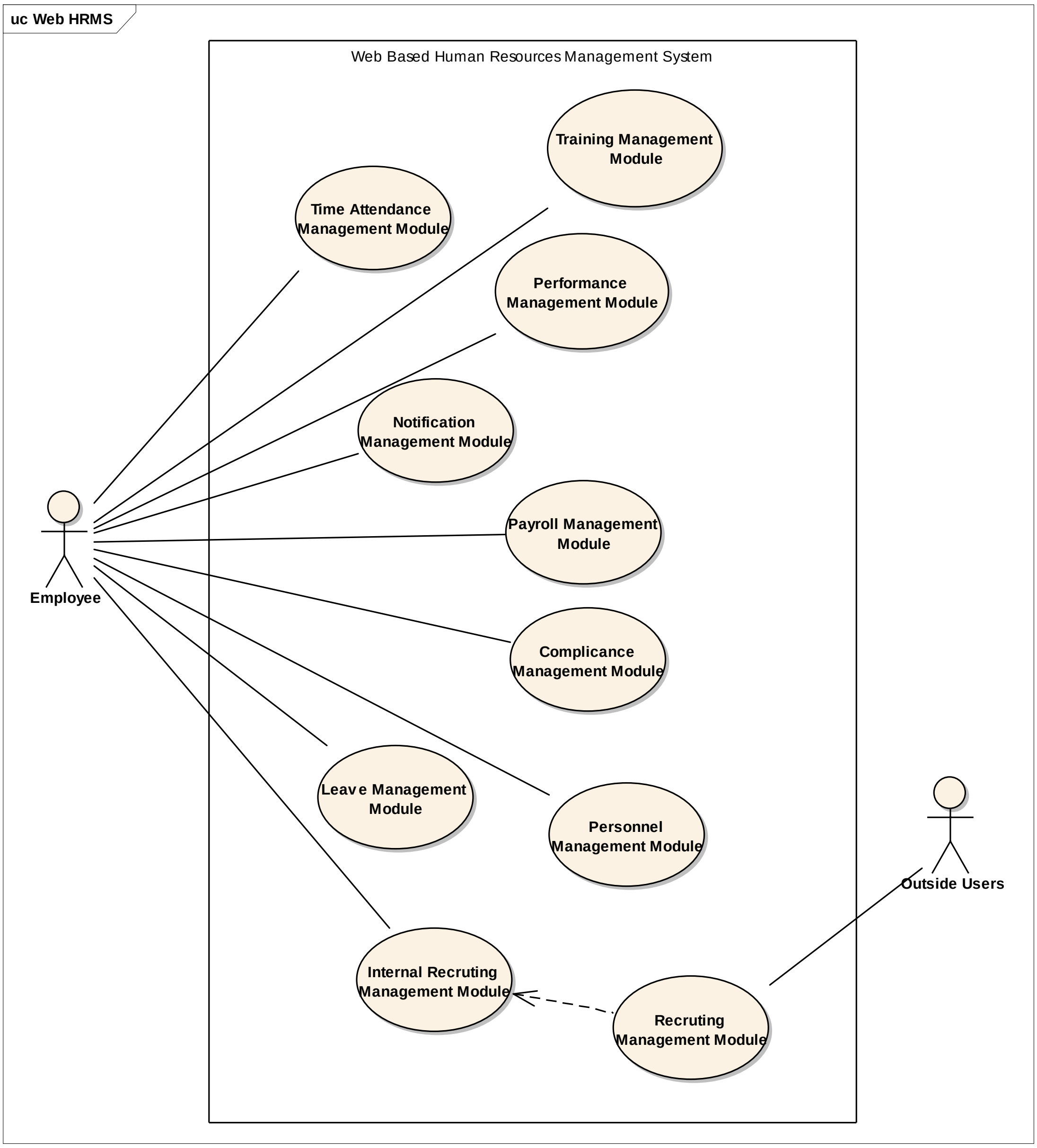

uc Web HRMS
Web Based Human Resources Management System
Training Management Module
Time Attendance Management Module
Performance Management Module
Notification Management Module
Payroll Management Module
Complicance Management Module
Leave Management Module
Personnel Management Module
Internal Recruting Management Module
Recruting Management Module
Employee
Outside Users

Figure 12. Web Based Human Resources Management System Use Case *(Source: Own)*

Modules are connected with each other by using the same database that has many different tables related to each module. The main table is Employee table, which is the base of the use of the system and all of its modules. Web based HRMS is dedicated to HR department, which helps them keep track of the employees by allowing them to register, edit, and delete information about employees of the appropriate company. These are different kind of employee data such as personal information, job history, trainings, leave history, salary, and other information related to employees, while the access to the modules is regulated with roles. Therefore, each role is configured to have access to specified parts of the system. Roles are designed by the administrator of the system, depending on the needs of the organization. The administrator of the information system in many cases is part of human resources department of an organization.

The purpose of this section is to analyze requirement specifications of Web based HRMS, which are described using UML use case and activity diagrams followed by development of a conceptual model of the system that defines the structural, and behavior of the parts of the information system, which leads to the final development, and implementation of the Web application. Following the UML-Based Web Engineering UWE method. Section 5.3 describes the applications navigational model. The navigation structure defines how the navigation is achieved. Building those models, different types of UWE stereotypes are defined and used.

Due to the size of the Web based HRMS presented in Figure 12, it is not possible to present all concepts in details. Our focus will be to build models for different views of the analysis and design phase. We will design the Personnel Management Module, Leave Management Module, and Payroll Management Module, which are closely related to each other and other modules of the Web based HRMS. In this way, we will select the most representative parts of the application and represent them in depth in details.

5.1 The Web Based Human Resources Management System Requirement Specification Analysis

Requirements specification is the first step toward developing a Web information system, which are specified in UWE with a separated model, called requirements analysis model. Following the UWE method, there are two levels of web application modeling. First, is a description of the functionalities what the system offers, modeled by UML use cases, and the second step is description of these processes in more details, in this case by using UML activity diagrams, which describe more precisely responsibilities and actions of the actors.

5.1.1 The Web Based Human Resources Management System Use Case overview

Use case diagrams are part of UML elements, which are built using Actors and Use Cases. Actors are used to specify the users of the Web information system. In this proposed information system, there are different users. The main user is *Employee*, which is separated by inheritance in other categories, such as *HR Employees, HR Managers,* and *Managers*. The diagram of users is presented in figure below.

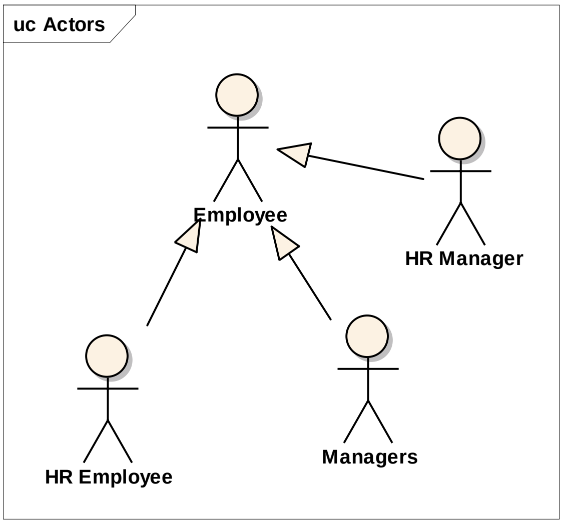


Figure 13. Main Actors in Web based HRMS *(Source: Own)*

Use case diagrams include use cases, actors, and association among them, which describe in visual way the role of the actors that play in interaction with the system.

The method which is used to model the proposed information system, categorizes three types of use cases: navigation, business process, and personalized use cases. Navigation use cases are used for modelling user behaviour in case of interaction with the Web information system, such as browse through Web information system or search with keywords.

Below in this chapter there are created some use cases which depict the main functionality of our system. The use case model presented in Figure 14, describes the main functionality of the information system module Personnel Module which contains information about employees of the organization. In this use case <<navigation>> stereotype is included, such as *View Employee Profile, View Employee Services, and View Employee Compliances*.

The second group of use cases which is Process use cases describes business tasks that end user performs in the system; these processes are modelled using the same methodology as it is done in the traditional software applications. The business tasks describe transactional actions on the database. To represent business tasks a typical UML notation is used, which are use cases such as *Change Employee Salary, Add/Edit/Delete Employee Skill, Add/Edit/Delete Employee Experience* and other use cases presented in this use case diagram. And third group of use cases are the ones that describe the personalization of the Web application, such as *View the Imminent Expiry of Contracts*. This group is notated with stereotype *<<personalized>>*. The personalization is triggered by user into system settings.

Also we have modelled use cases of other modules which follow the same method as Personnel Module Use Case diagram. All elements of UML for modelling use case diagrams are used while constructing Web information system use case diagrams, starting from **package**, **system boundary box**, **generalization**, and stereotypes such as **<<extend>>** and **<<include>>**. Below we will describe in detail use cases followed by activity diagrams for the modules of Web based HRMS that are chosen to design in this document.

5.1.2 Personnel Management Use Case

While representing the use case diagrams for our modules it is important to mention that the level of details for requirement specification depends on the size of the project risk and the complexity of the Web Application which is going to be designed. There are used use case diagrams to specify the main needs of our information system. Below in Figure 14, the use case diagram for the Personnel management module is shown, which contains specified Actors and Use Cases, and also the interaction between each other. The main actor of a Personnel Management module is HR Employee which as presented in figure 13, inherits all properties of Employee actor, and has its own additional options. HR Employee is a special case of actor Employee which is mostly interacting with the system.

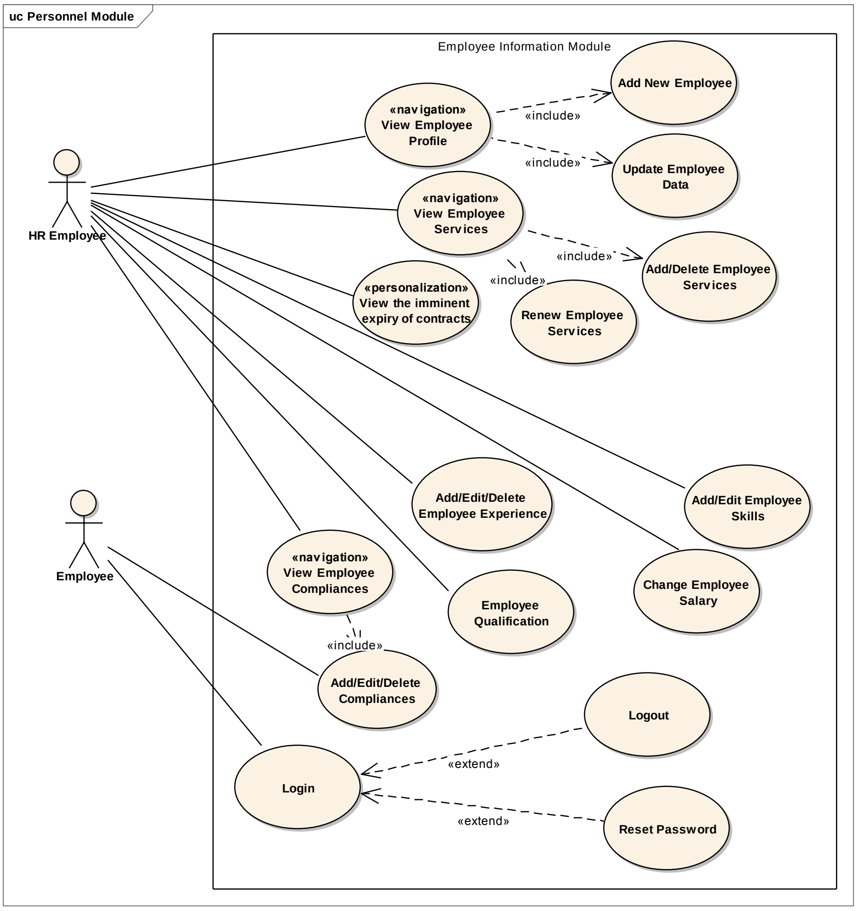

uc Personnel Module
Employee Information Module
HR Employee
Employee
«navigation» View Employee Profile
Add New Employee
Update Employee Data
«include»
«include»
«navigation» View Employee Services
«include»
«include»
Add/Delete Employee Services
Renew Employee Services
«personalization» View the imminent expiry of contracts
Add/Edit/Delete Employee Experience
Add/Edit Employee Skills
«navigation» View Employee Compliances
Employee Qualification
Change Employee Salary
«include»
Add/Edit/Delete Compliances
Logout
«extend»
Login
«extend»
Reset Password

Figure 14. Personnel management use case diagram *(Source: Own)*

Following UWE method for modelling web applications, in this use case three types of stereotypes are introduced. The notation of stereotype is shown with “<<name >>” above the name of use case. The main functionality of this Personnel Management module is depicted by this use case diagram, and the main interaction of HR Employee user is shown by the connection with appropriate use cases. The HR Employee role is a super user in the information system.

To understand better the functionality of the system, there are followed the principles of usage of UML, to refine and specify the requirement of the web application. For this purpose, it is important to go deeper in this system. In this case, there are used UML activity diagrams for close overview of the system. For each business use case, there are created at least one activity diagram in order to have an entire overview of the system. Activity diagrams consist of activities, control flows elements, description of steps for each element.

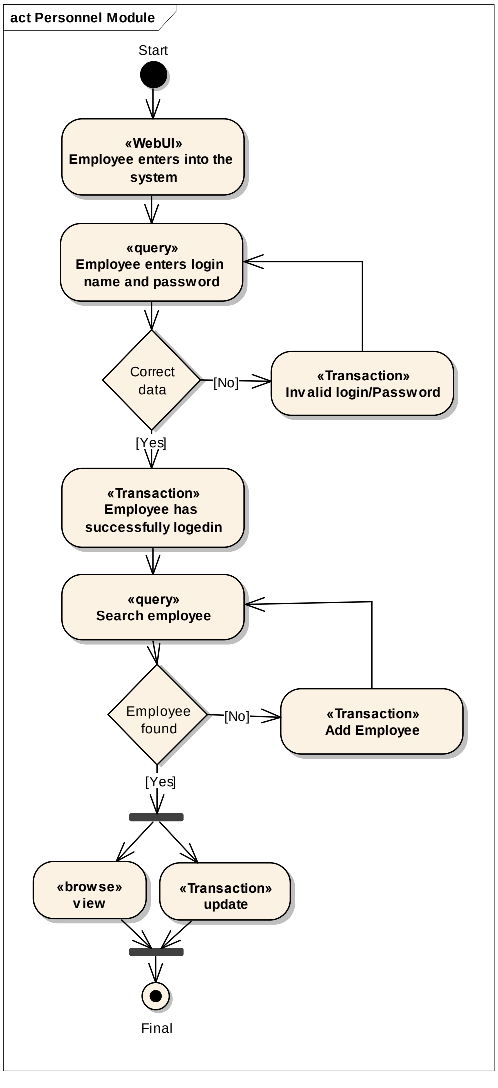

act Personnel Module
Start
«WebUI» Employee enters into the system
«query» Employee enters login name and password
Correct data
[No]
«Transaction» Invalid login/Password
[Yes]
«Transaction» Employee has successfully logedin
«query» Search employee
Employee found
[No]
«Transaction» Add Employee
[Yes]
«browse» view
«Transaction» update
Final

Figure 15. “View employee profile” activity diagram *(Source: Own)*

In Figure 15, the activity diagram for the use case *View Employee Profile* of Personnel Management module is presented, which is the main module of the Web based HRMS. Following UWE method for modelling the web information system, in activity diagrams a set of stereotypes are used, which are added in specific web part of UML object and activity nodes. Notation of stereotypes is done using Enterprise Architect default notation style followed by <<name>>. In this model there are used different types of activities separated in two parts, stereotype of visualization and stereotype of activities. For visualization stereotype Web user interface notated by <<WebUI>> and Content stereotype notated by <<content>> are used, while for activities stereotype, different possible actions of users in the Web information system, such as search, browse and different transactional activities that cause changes in the Web information system database are used. The notation of these stereotypes is done using keywords same as stereotype name so that stereotype search is notated by <<search>>, stereotype browse is notated by <<browse>>, and stereotype transactional activities is notated by <<transaction>>.

The module described above is almost useless without relation with other modules such as Leave module which records employee Leave, and also Payroll management which is directly connected with leave management module and Personnel Management module and also other modules which are part of Web based HRMS, mentioned in the previous section. Following in this chapter are described use case diagrams for two other modules such as Leave Management module and Payroll Management module. The logic of development of these use case diagrams is same as the development of the use case diagram shown in Figure 14.

5.1.3 Leave Management Use Case

In Figure 16 the use case diagram that describes the main actors and their interaction with the Leave management module is presented. From the diagram it is seen that there are used two types of stereotype which are navigation and personalization. In this model there are also the same actors as they are before in the previous use case diagram shown in Figure 14. This use case shows the main tasks that an actor can do in the information system. The main functionality of Leave management is to add new employee requests for leave (holiday), and also keep the history of employee leaves during a certain period. This period can be annually, monthly or weekly according to the organization requirements.

In order to complete a whole process, all actors should participate. The process begins when employee applies for leave via employee leave interface. In order to apply for leave day, employee must have unused free days of leave, which are calculated automatically from the system. The automatic calculation of the system in this use case is presented as <<*personalization*>> stereotype. After the employee has applied, the request will go to manager of the employee which has the option to accept or reject the employees leave days request. After the managers' response, the request is followed to HR department where they can check other administrative rules, when they can accept or reject the request. This process is valid when employee requests leave days. But does not include the process when employee is sick or other similar situations. In this case the process begins from HR department and everything is finished inside this department. For this reason the connection is made between HR employee use cases *Manage Employee Leave* which is specified with <<navigation>> stereotype.

Another use case which is considered as <<navigation>> stereotype is *View Holidays* which allows HR department to manage holidays, which does not count as a leave day if any employee leave day falls on holiday.

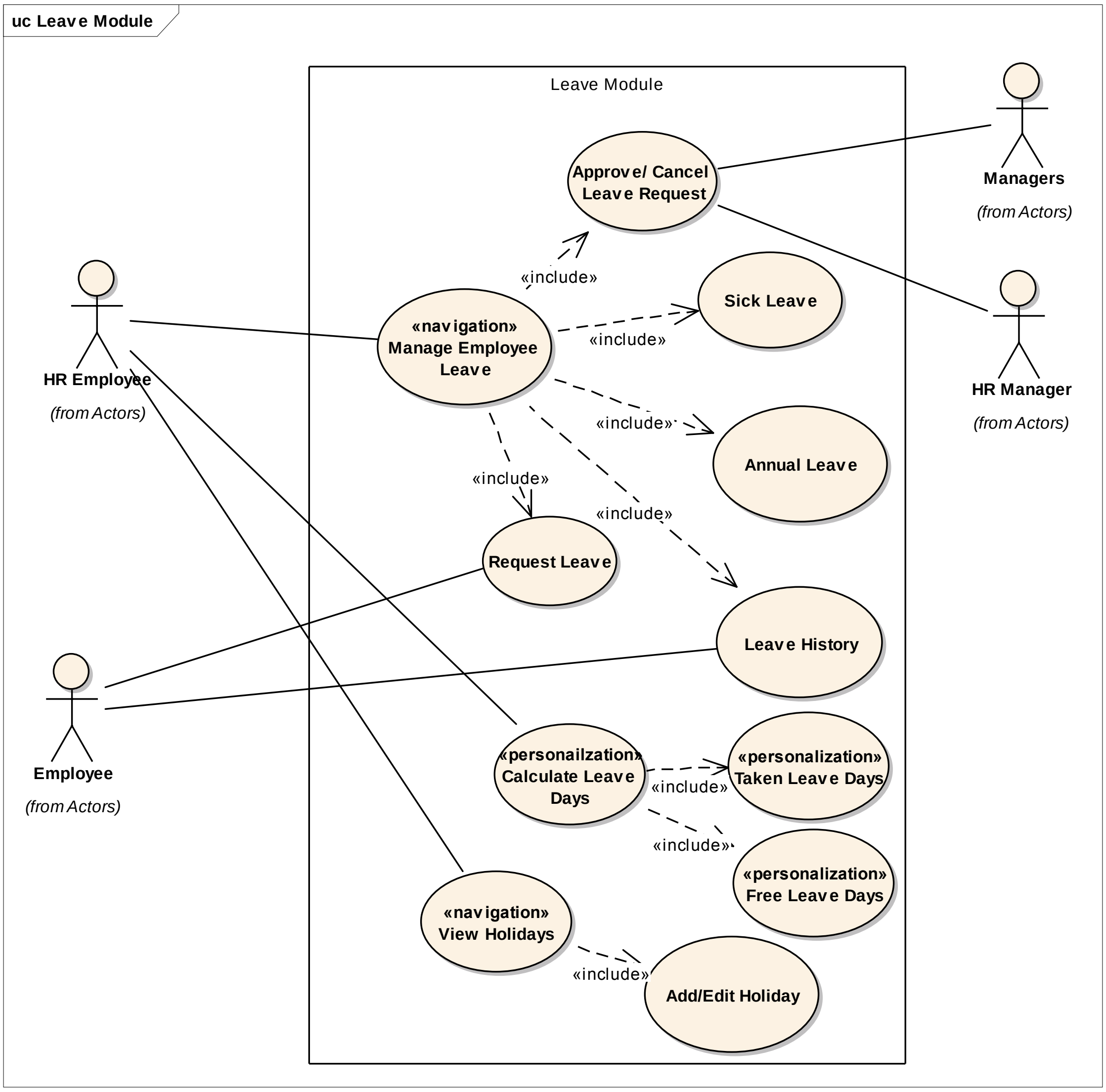


Figure 16. Leave management use case diagram *(Source: Own)*

Another way to understand each process of use cases is using activity diagram for each use case. In Figure 17 is introduced the activity diagram which is related to process when employee requests for leave days. This activity diagram elaborates in details steps of the information system behaviour from the beginning of the process until the end, when request is done.

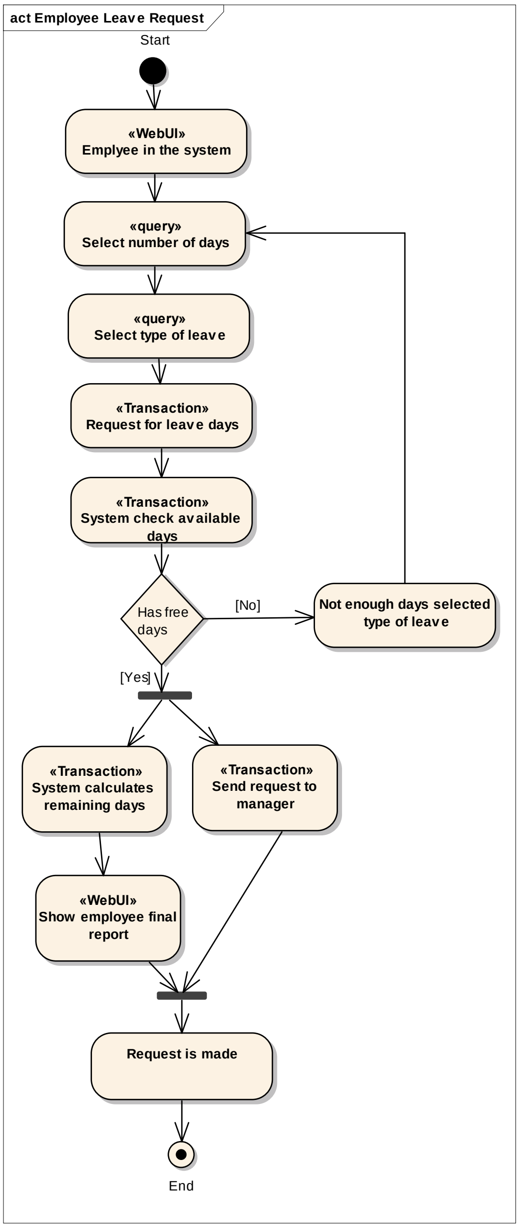


Figure 17. "Employee leave request" activity diagram *(Source: Own)*

In this use case, as in previous one created for the Personnel management module, there are used UWE stereotypes which describe behavior of the information system interacting with user. The stereotypes used in this use case diagram describe three types of behavior of the system. Transaction, Query and WebUI. Transaction stereotype is triggered when information system makes a transaction inside the database. In this case, this has happened when employee request for leave days, also transaction happens when information system calculates remaining leave days. All this so-called transactions make changes in the database. The other type of stereotype, which is used in this activity diagram, is query notated by <<query>>, which happens in two cases, when employee selects number of leave days, and select type of leave.

Third stereotype used in this activity diagram is Web user interface notated by <<webUI>>, which describes the behavior of the system when it interacts with the user. This stereotype happens when users' request a login in the information system, and when information system shows the final report of used and remaining leave days for particular type of leave. As this activity diagram describes the procedure of user request for leave days through web information system, can be created other activity diagrams for particular use case in the Figure 16, which will describe other processes. This module is quite simple to understand and the processes are quite the same in all of the times. Therefore, this activity diagram explains only the way that employee can request new leave day.

5.1.4 Payroll Management Use Case

Another module that is elaborated in this document, which has close relation with modules described above is Payroll management. This is a very sensitive and complicated module. During my experience, most of the time, each organization uses different ways of calculating salary for their employees. This depends on the type of organization, place where the organization is located, and department which employee belongs to. Another thing that is important to mention, is that this module has a strong relation with other software applications, which are not part of this web application such as time attendance software, finance software. For this reason, this software has its difficulties working outside of the organization. This means that the systems, which are related with each other, should be in the same network, when connection between them will not be a problem.

In this model, is presented the most used way of salary calculation, which can create special reports, which are needed for communication with other software applications inside and outside the organization.

In the Figure 18 a use case diagram which represents all information, which flows when HR employees interacts with web information system is presented. This use case diagram describes Payroll management system content, which contains actors and use cases, and also the interaction between each other. In this use case are introduced two more actors which are not part of use case diagrams of other modules created in this document. Except default actors such as *Employee, HR Employee, Manager and HR Manager*, two very important actors are introduced, which are related directly with Payroll system which are *Time Attendance System* and *Finance Department System*.

The main actor is HR Employee which inherits all properties of actor Employee. HR Employee is a special case of employee actor which is mostly interacted with the information system. Other systems such as Time attendance contains all information about employee working hours in the company. The system helps payroll by automatic transfer from time attendance database into Payroll database. In this case the help of HR Employee is needed because these data has to be verified before the process goes further.

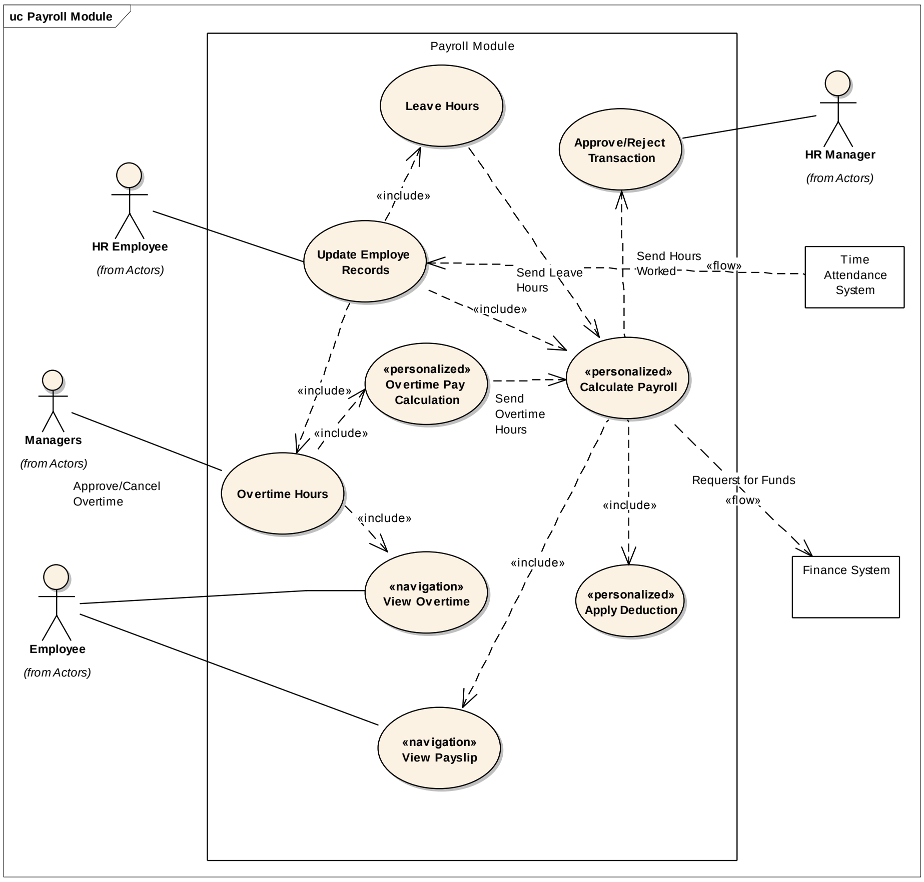


Figure 18. Payroll management use case diagram *(Source: Own)*

Following UWE method for modelling web information systems, like in other use case diagrams created above, in this use case diagram different type of stereotypes are introduced. First type used is navigation stereotype which represents the part of the system that is navigation and it is notated by <<navigation>>. Use cases such as View Payslip, View Overtime are considered as navigation stereotype. The other stereotype used in this use case diagram is called personalization which is notated by <<personalized>>. Use cases which are considered as personalization are *Calculate Payroll, Calculate Deduction, Calculate Allowances,* and *Calculate Overtime Pay*. The other use cases created in this model are pure UML use cases.

In this module most of the work is done automatically from the information system itself. Human assistance is necessary for calculating the salary, but their job is just to verify the data entered from other systems and modules, such as working hours, leave hours, and employee overtime.

To understand better the process, the modelling of activity diagram is needed. Below it is created an activity diagram which corresponds to use case Calculate Payroll shown in Figure 18. There is presented the process of payroll calculation by the information system, with initiation of HR employees.

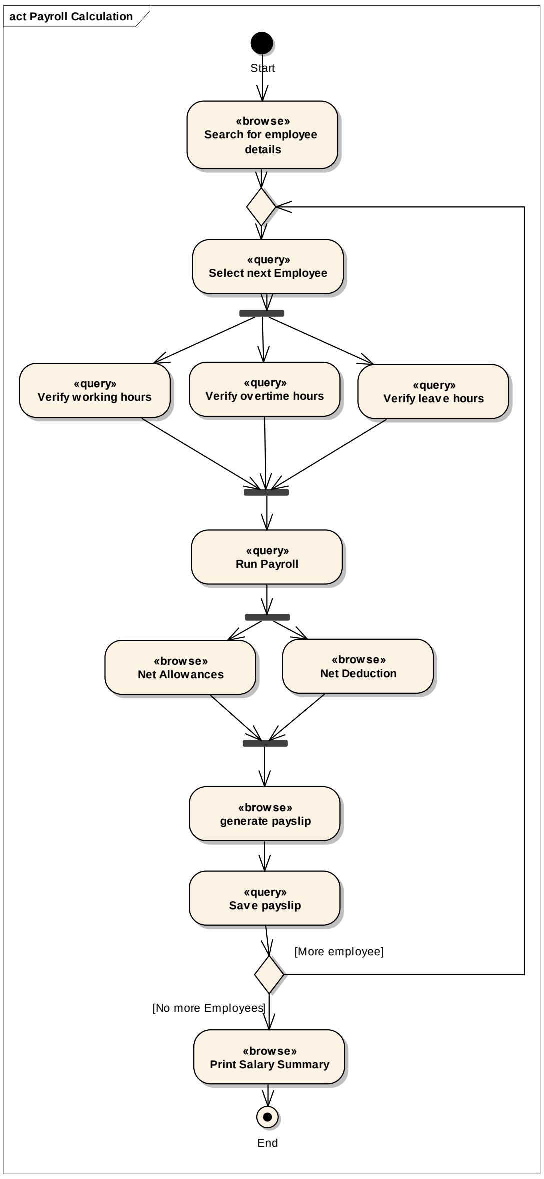


Figure 19. "Calculate salary" activity diagram *(Source: Own)*

In this diagram two type of stereotypes are used: browse and query. Browse stereotype is notated by <<browse>> and query stereotype is notated by <<query>>. This activity diagram describes the process of salary calculation. The process start from employee when the first step is to find a specific employee. If a searched employee is not found than the system starts from the beginning, otherwise if the employee is found then is verified overtime, working hours and leave hours. In case that everything is verified correctly the payroll is run and specific deductions and net salary is calculated. This process is made for each employee separately.

5.2 The Web Based Human Resources Management System Conceptual Modelling

Following the UML-web based engineering the first step to create a conceptual method is to describe the structural and behavioural aspects of the Web application. This information is gathered by the development of class diagram of the application, state-transition diagrams and functional models which are presented in the following section.

In this section specifically, the detailed class diagrams for the modules selected for this case study are presented. Due to complexity of Web application, class diagrams are created separately for each module in order to understand better the relationship between modules and easier identification of each respective class and its properties.

Furthermore, for better understanding the behaviour of the system is the development of Navigational Model of the information system, which can be created based on the development of user diagrams, which describe the different type of users who can use the system. The Navigational Model is also described and developed in this section, which is related to each user and describes the navigation and accessibility to the system. And finally, the last section shows some abstract presentation development models based on navigational and class diagrams of the system, which is a concept of how the system could be developed by development team. The presentation model specifically shows the navigational model which completes the Web Interface specification.

5.2.1 Personnel and Leave Management Conceptual Model

In order to create an UWE conceptual model, primarily there should be described the structural model which is shown in the class diagram of the application and its behaviour model by using functional and dynamic models. As specified before the content of the structural model of WB-HRMS should contain personal information of employees and other information related human resources such as trainings, holidays, compliances, payroll etc. In this phase is presented the development of class diagrams for Personnel Module, and Payroll Module which are part of Web based HRMS. The aim of conceptual model presented by class diagram is to better understand the content of web application by visual specification of relevant information, which eventually define the content of web application.

In Figure 20 is presented a class diagram which describes content of Personnel Module. In the class diagram there are identified classes, their attributes and operations, methods which define their behaviour and relationship between them using association, aggregation, multiplicity and other UML class diagram notation.

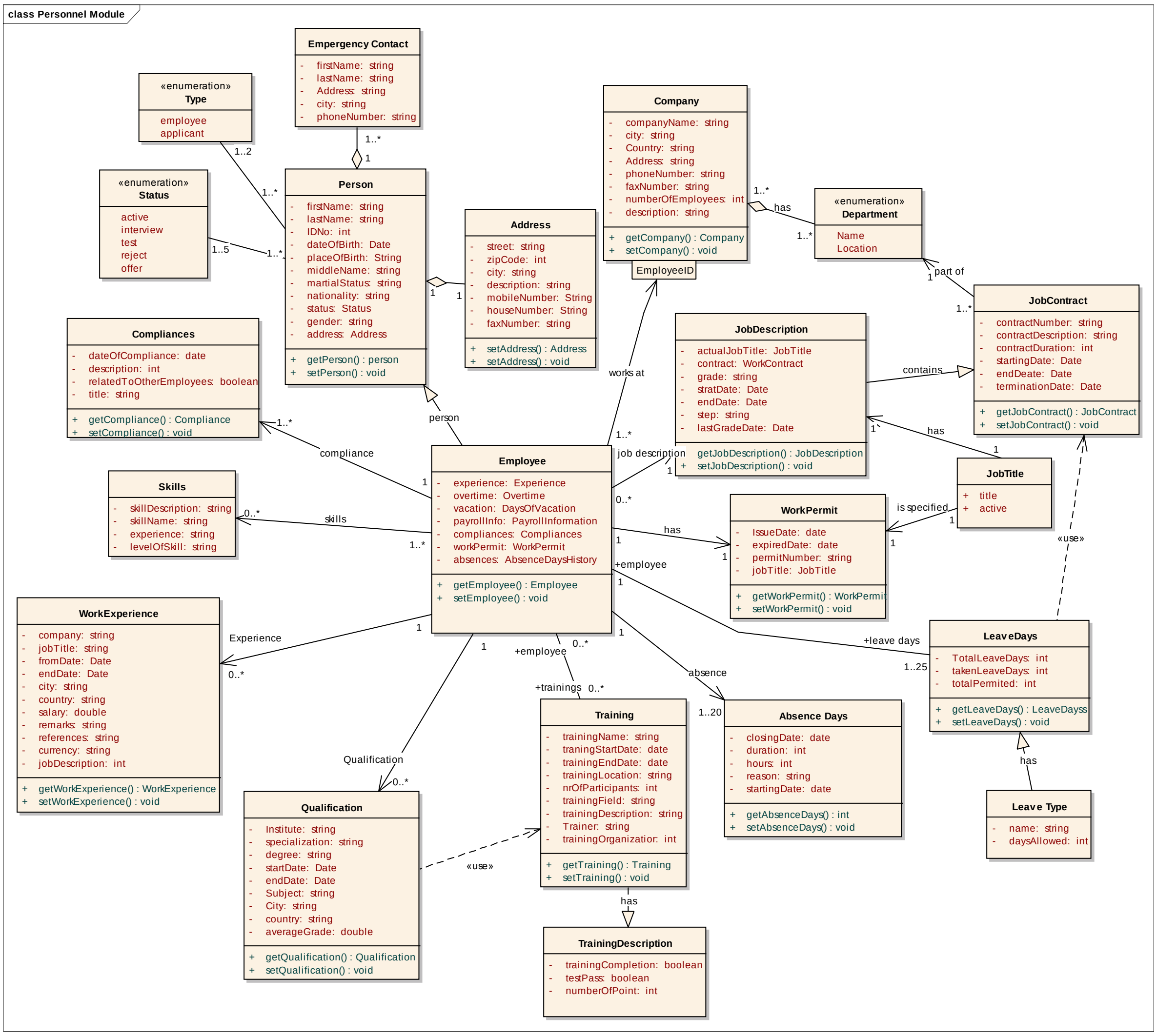


Figure 20. Personnel and Leave management class diagram *(Source: Own)*

Class diagram represents relation and the dependency of classes between each other. As it is presented in the figure the main or central class is *Employee* class, with the set of properties such as work permission, experience, absences, which corresponds to other classes in the model. From the connection that employee class has with other classes which all together define data related to employee, we can define that these classes provide a lot of information about the employee.

Personal information and addresses are presented by classes *Person* and *Address.* Class *Person* represents all information about people who has in one way interacted with company, they can be job applicants, current employees, and people who are registered in the system. In this way this class allows the company to manage people by using predefined information from the employee.

Other information that are presented in class diagrams are qualification, work experience, skills, absence days, all this information is presented by classes with same name. Another important class presented in class diagram is the *Company* class which represents direct relation between employees and company which they belong. Training module is also incorporated in this class diagram. This module is presented by two classes *Training* and *Training Description.* Class *Training* shows the general information about individual trainings, while *Training Description* class shows the specified information about employee attendance, their performance and other information.

5.2.2 Payroll Management Conceptual Model

Web application has different users. Internal and external. Internal users are the main users of the system. They are separated by roles. The most important user is HR Employee which inherits properties of Employee user. Other users such as HR Managers, and other Managers also inherit all properties of Employee user. External users are all users who apply for job, and who are registered directly in recruitment module which is part of Web Based Human Resource Management System. The topic related to the user will be described in detail in Section 5.4.

Except the class diagram presented for Personnel Module, in this document the class diagram for module Payroll will be presented. Figure 21 represents class diagram, which depicts the main classes, relation and the dependency between each other. As it is presented in figure the main or central class is *Employee* class which is inherited from class diagram created for Personnel Module. This class contains all information which are needed to perform an employee salary calculation. The most important class which relates Personnel Module and Payroll Module is class Employee and class JobContract.

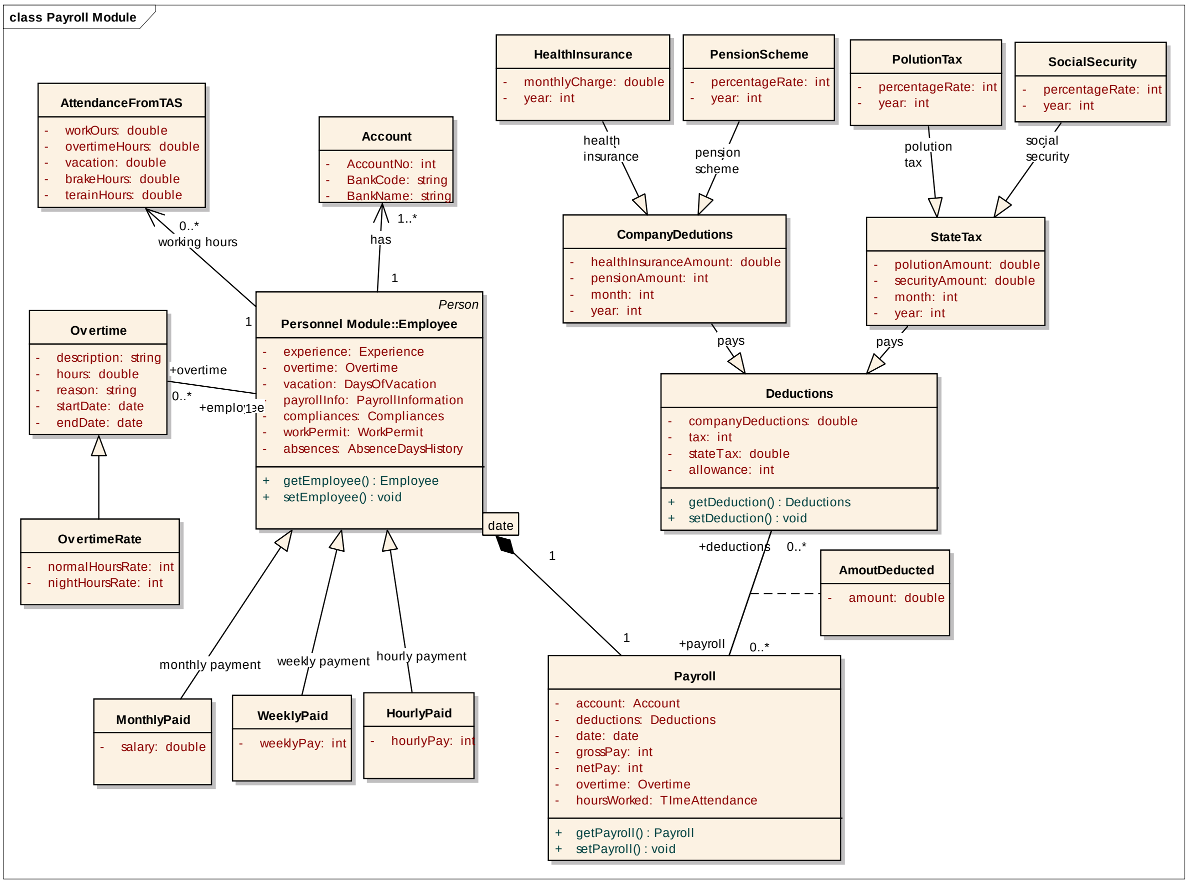


Figure 21. Payroll class diagram *(Source: Own)*

In this class diagram the content of Payroll Module is described. According to this class diagram, Salary is calculated periodically monthly, weekly or hourly depending on salary type of the employee. This is presented by classes MonthlyPaid, WeeklyPaid and HourlyPaid which use generalization to connect with class employee. The hours worked by employees is determined by another software application which will be imported directly to each employee. In class diagram are presented different types of Deductions which will be calculated when Payroll is triggered.

5.2.3 User Profile Conceptual Model

As described above, the aim of conceptual model is to provide a visual information for web system which depicts the content of the Web application. However, often it includes user entities of domain which are required for customization of web application. This entities usually are called as user models or user profile. Customization has to do not only with properties of users, but also with feature adaptation for web environment. For this reason in this case is created a so-called context profile. In our case we have created a user profile, shown in Figure 22. The model is created for one type of users, *Employees*, which have interaction with both modules described above. Entities representing user context are modelled by classes. UML associations are used to represent relationship.

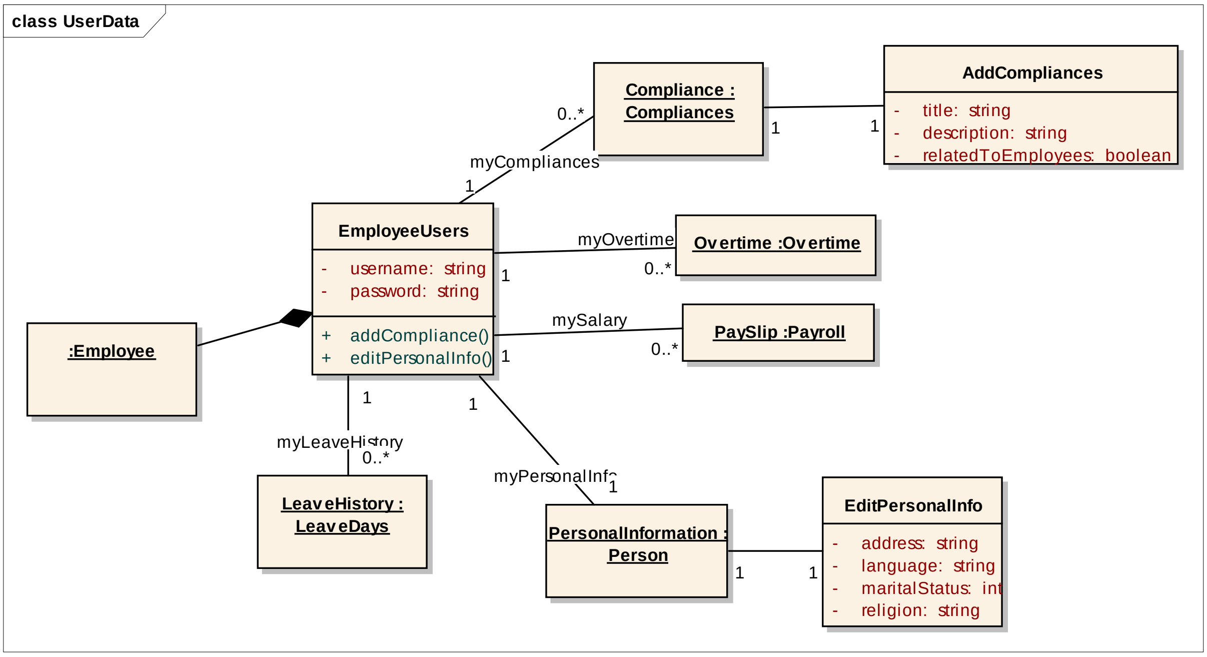


Figure 22. User model *(Source: Own)*

The user model contains user information which is needed for user to login in the system, and view his information and add compliances. All this data is pre-registered by user and HR department. In addition, system allows user to view history of his data such as Leave History, Overtime, and Payslip which are generated by respective modules such as Leave Module, Payroll Module etc.

In addition, there are presented two class diagrams which represent two different modules, and user model which shows relation with these two modules. The next section will continue with navigation model which represents the visual view of requirements analysis and conceptual model of the application.

5.3 The Web based Human Resources Management System Navigational Model

Navigation structure is based on the requirement analysis and conceptual modelling of Web application. Navigation model consists of navigation classes, menu-s, query and links. Navigation classes notated with stereotype <<*navigationClass*>> represent relation between nodes in the hypertext structure, and navigation links represent connection between navigation classes.

Conceptual model classes are relevant for definition of content of web application, and the same classes and their associations can be put into creation of navigation model. Navigation classes consists directly from classes represented in particular class diagram. Each navigation class is a direct representation of classes. Classes use associations to represent dependency with other classes. This linkage represents possible steps followed by a user, which has its importance. If a class has dependency to more than one class, the linkage is done separately with all of them. In navigation model this issue is chosen by introducing stereotype called menu notated by <<menu>>. Every navigation class which has more than one outgoing association is connected first with a menu. Query stereotype notated as <<query>> represents navigation model access primitives. Access primitives are used to select a single information represented in navigation classes using special options such as Search.

In general web application contains business logic. In this system business processes are included in many cases, such as calculation of leave days, calculation of services (contract), entry of compliances, etc. This should be included while creating navigation model. Business processes in this model are identified as process class, notated by

<<processClass>>, and also the linkage between process classes and navigation classes is created using process links.

5.3.1 Personnel and Leave Management Navigational Model

In Figure 23, navigation model for Personnel Module which describes the behaviour of HR Employee user when interact with system is presented. Eventually, in this navigation model both modules are included, Personnel Module and Leave Module, with possibility of connection of other modules that belong to this Web Based Human Resources Management System.

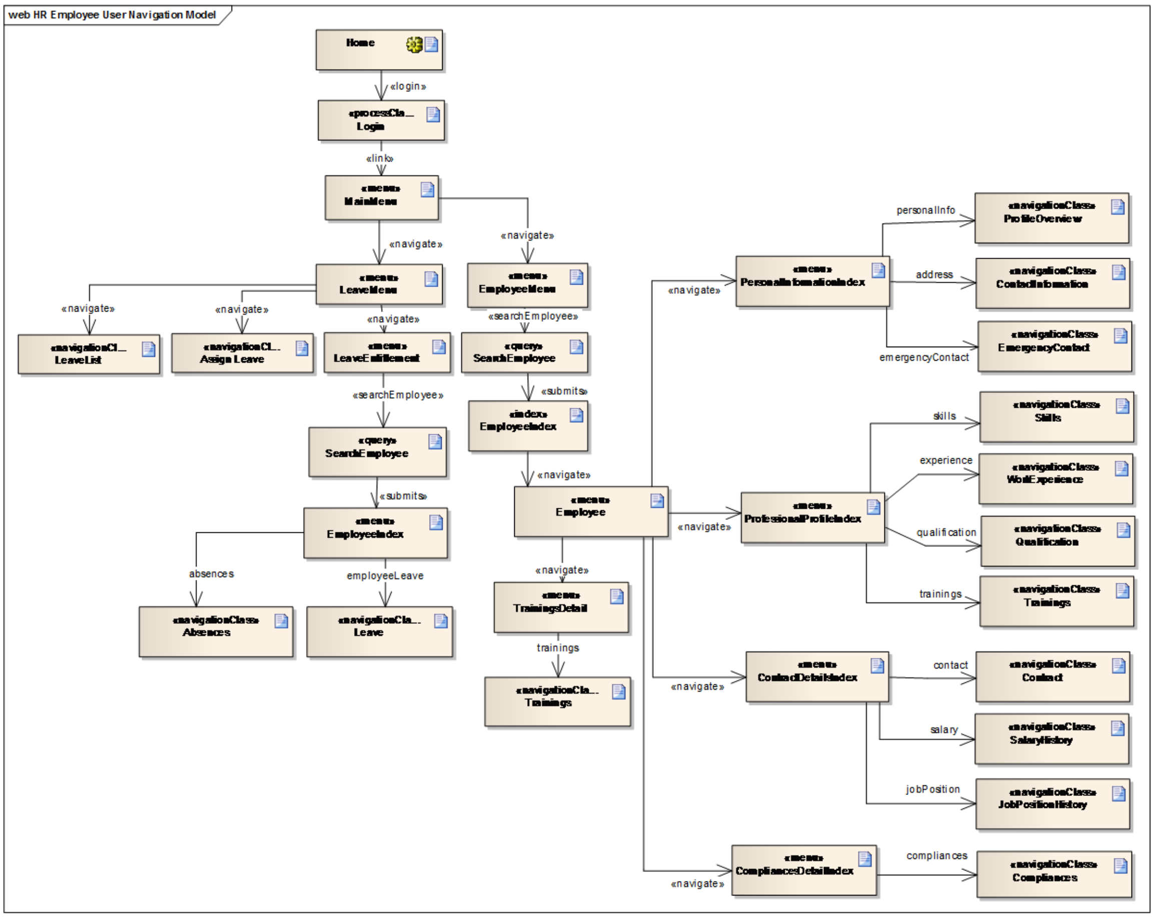


Figure 23. HR employee user navigation model *(Source: Own)*

First web page notated as *Home* represent the state of Web application when user enters in the system. In this case the only visible option for the user is Login Form. This stage is identified as *Login*, and notation of the stereotype <<query>>. Since this is considered as query, user should give to the system his credentials (username and password) to be able to view other options of the system. The credentials are compared automatically from the system, if the user exists in database. In case that user enters valid credentials, the application shows other options such as content of Personnel Module, and Leave Module.

As it is described above, HR Employee user is also administrator of the system. He has every right in system configuration, and can view all modules included in the system. After login in the system, the user has option to view Leave Module or Personnel Module. Leave module contains different links which represent the different type of Leave information and options such as Leave List, Assign Leave, and search for particular employee. While, if employee selects Personnel Module, he will have option to search for particular employee and view all of his information such as personal information, professional profile etc.

Besides Personnel Module and Leave Module included in first navigation model, this document represent also navigation model of Payroll Module.

5.3.2 Payroll Management Navigational Model

In Figure 24 the navigation model for Payroll module is represented. As in the first navigation model created for other modules, also in this model UWE stereotypes are used to identify each step in our navigation model. Each class which is represented in class diagram is included in this navigation model.

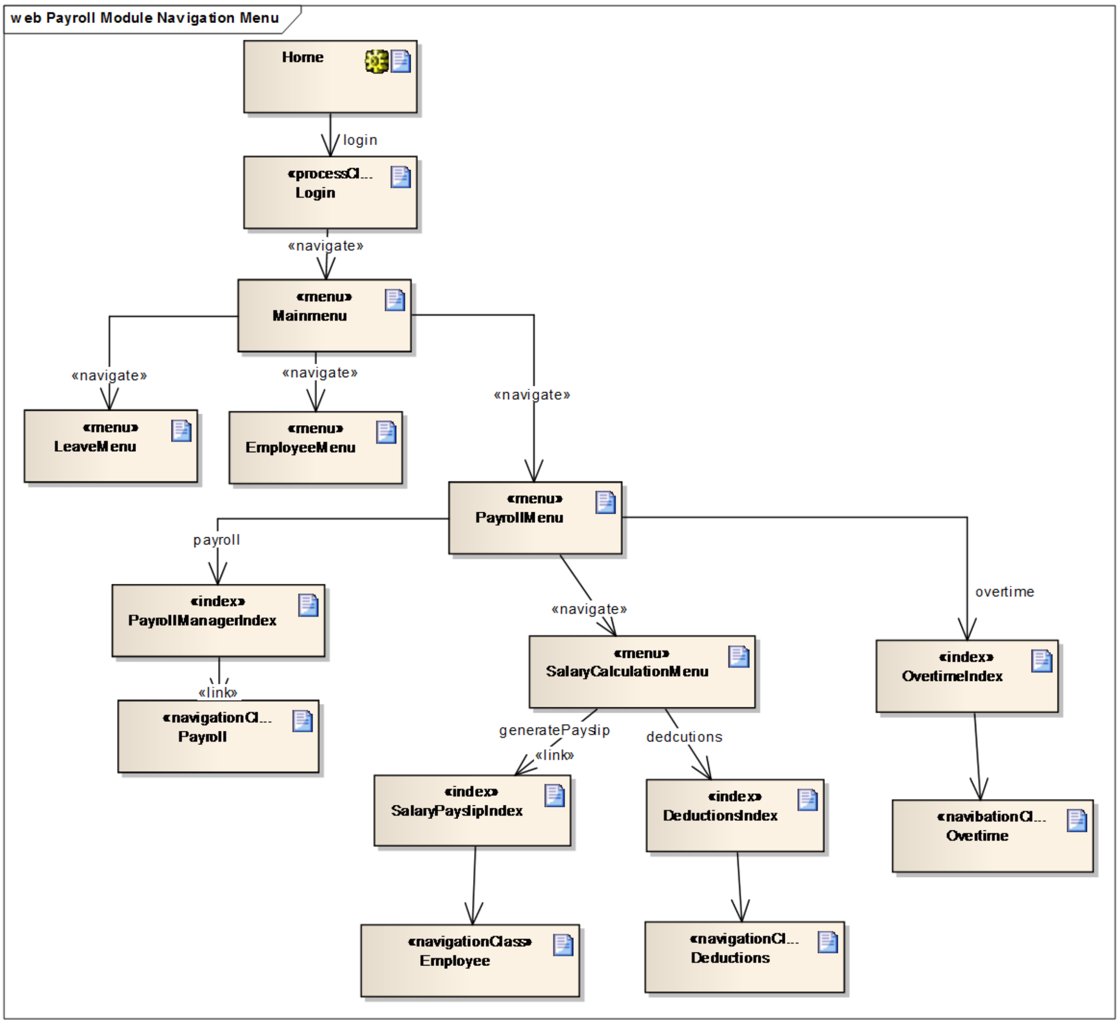


Figure 24. Payroll management navigation model *(Source: Own)*

Starting from Login of the user in the system, in main menu of web application will appear the link called Payroll, identified as PayrollMenu. This menu allows users to navigate through classes of payroll, such as Overtime, Individual Employees, and Payroll. Each class has its role in payroll calculation. As it was described above in this document, calculation of salary is not same in all organizations. This depends on place, internal regulations and other factors such as department, job position and type of contract. In this model a simple model of Payroll system is described. This module is directly connected with other modules such as Personnel Module, Time Attendance System and Finance System, many information needed to perform a calculation are inserted automatically from other systems.

5.4 The Web based Human Resources Management System Presentation Model

An abstract view of user interface of the web application is provided by presentation model. The development of this model is based on navigation model and concrete aspects of user interface (UI), such as fonts, colours, and place of navigation on the web page. Presentation model present the structure of user interface, when the concrete aspects of user interface such as colour, images, anchors, forms etc., are used to present the navigation model (Koch, et al., 2008).

The main focus in this step is model of the presentation or concrete view of the web application where the navigation objects and structure of elements will be presented to the user i.e. in our concrete web design this explains the solution that, which web form will be used for the presentation of employees, which links should be activated, and which employee information should be presented to the particular employee. In general, before the development and implementation of presentation model, designers discus the appropriateness of the presentation with customers and stakeholders. This is a big advantage of the presentation model, because of its independence from other models.

Presentation model has its own representation notation, and the basic concept of this model are the presentation classes. Each class represent one node in the navigation model such as classes, menus, process classes, and access primitives. A presentation class usually is represented by UI element, such as *text* notated by stereotype <<text>>, *button* notated by stereotype <<button>>, *anchor* notated by stereotype <<anchor>>, *images* by <<images>>, and *form* by <<form>> (Koch, et al., 2008). In figure 25 the main homepage of the web based human resources management system is shown, which has navigational menu with links to the modules described in this document, and the main page which will present different reports with important information.

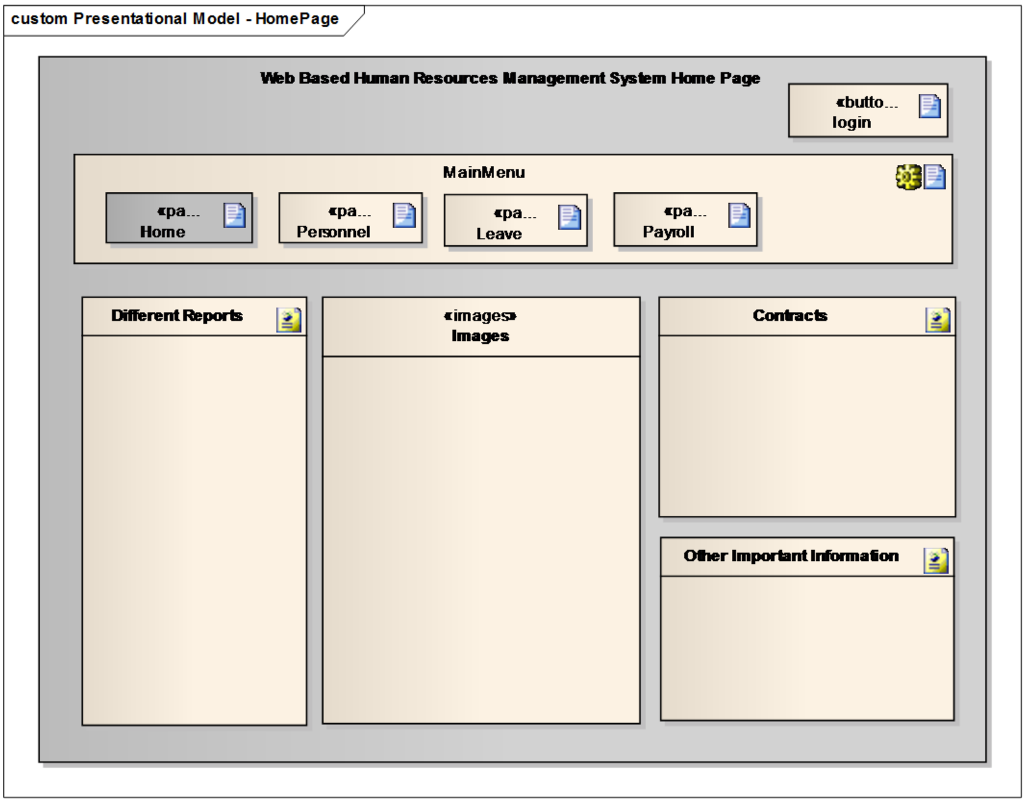

custom Presentational Model - HomePage
Web Based Human Resources Management System Home Page
«butto... login
MainMenu
«pa... Home
«pa... Personnel
«pa... Leave
«pa... Payroll
Different Reports
«images» Images
Contracts
Other Important Information

Figure 25. Web based HRMS home page *(Source: Own)*

Below in this document the presentation model for this Web based human resources management system is described, especially for three modules such as personnel management, leave management and payroll management.

5.4.1 Personnel and Leave Management Presentation Model

Figure 26 presents the real example of a presentation class for navigation model Personnel Management. For better identification of which navigation class is presented by the particular presentation class, there are used the names which corresponds to the name of navigation node. Also, all attributes that are used in navigation class are presented with appropriate stereotype of user interface elements. Usually, in practice of development and implementation of web applications the information's of several navigation nodes are presented on the main web page. The web page in UWE is modelled by *pages* <<pages>>.

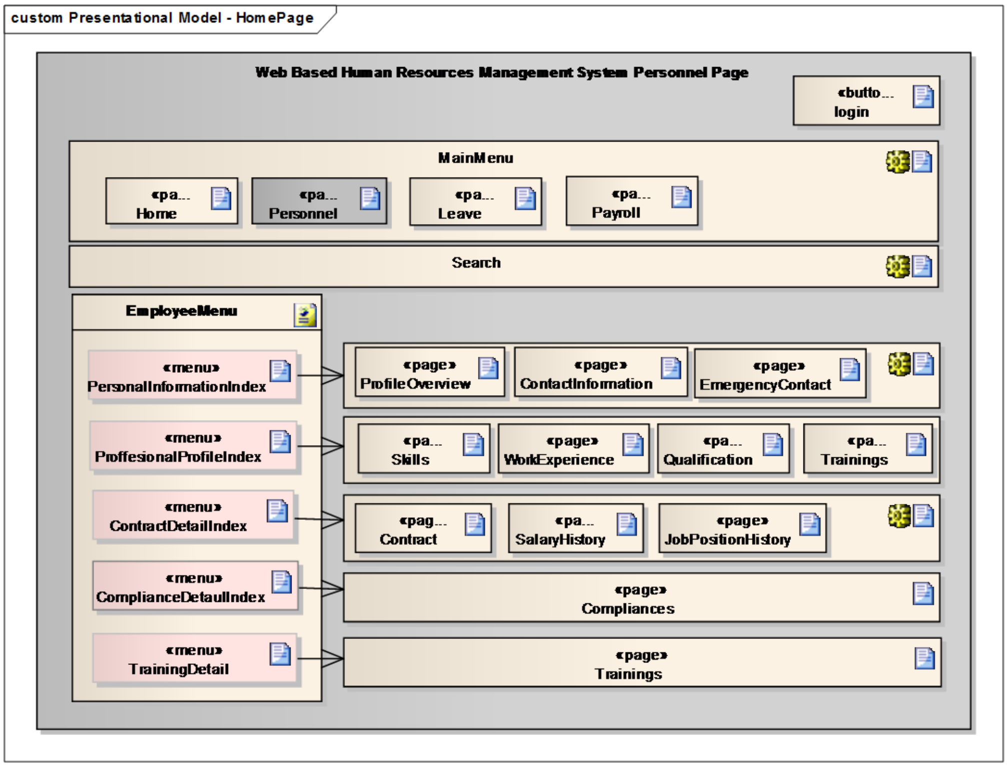


Figure 26. Personnel management presentation model *(Source: Own)*

Personnel Management presentational model shows all information that can be found in this module. This model is created according to the navigation model of the same module. Web Pages which present the information to the users as characterised by the stereotype <<page>>, while navigational menus have stereotype <<menu>>. This model is the base of design of the final web application which can be implemented to the client. This is only a concept how information can be organised in the system, but does not filter the information that can be shown to the users.

The separation of the views through different users is part of implementation process, which will pre-discussed with a costumer. Personnel menu contains five submenus which help users to navigate through the information that they offer. The first submenu is Personal information of employee. This submenu contains basic information related to particular employee. Submenu Professional profile contains information such as Skills, Work Experience Qualification and trainings. Also the most important submenu is Contract details which contains information about services of employee in the company such as contract, salary history and job position history. In this module there is information which is directly connected with two other modules which are not studied in this document. These modules are Compliances and Trainings. Compliance submenu listed in this presentational model contains information about employee compliances for each employee separately, also Trainings submenu shows information about employee actual trainings, which are registered or currently attending them. The other presentational model will be created for payroll Management, according to the navigation model for a same module.

5.4.2 Payroll Management Presentation Model

In Figure 27 the abstract form of presentational model for payroll management is presented. The concept of presentational module is based on navigational model and also the conceptual model. Note that all the presentational classes correspond to each node in conceptual model, and the relationship between conceptual classes is created with directed association.

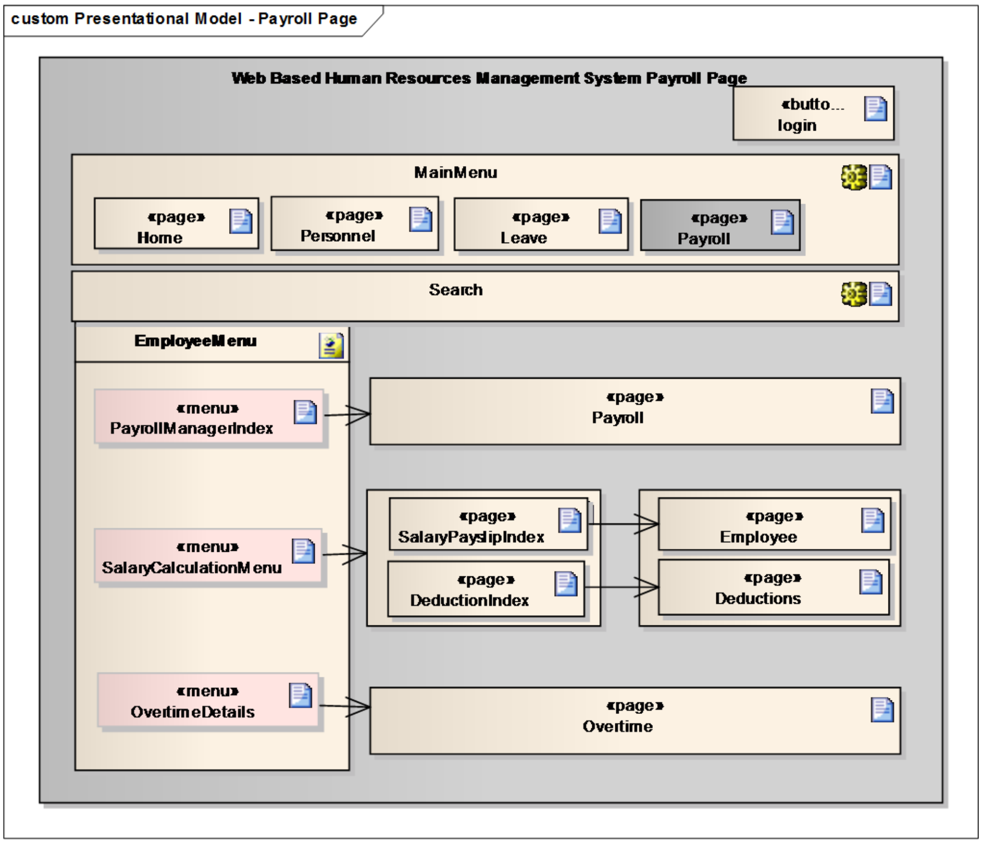


Figure 27. Payroll management presentational model *(Source: Own)*

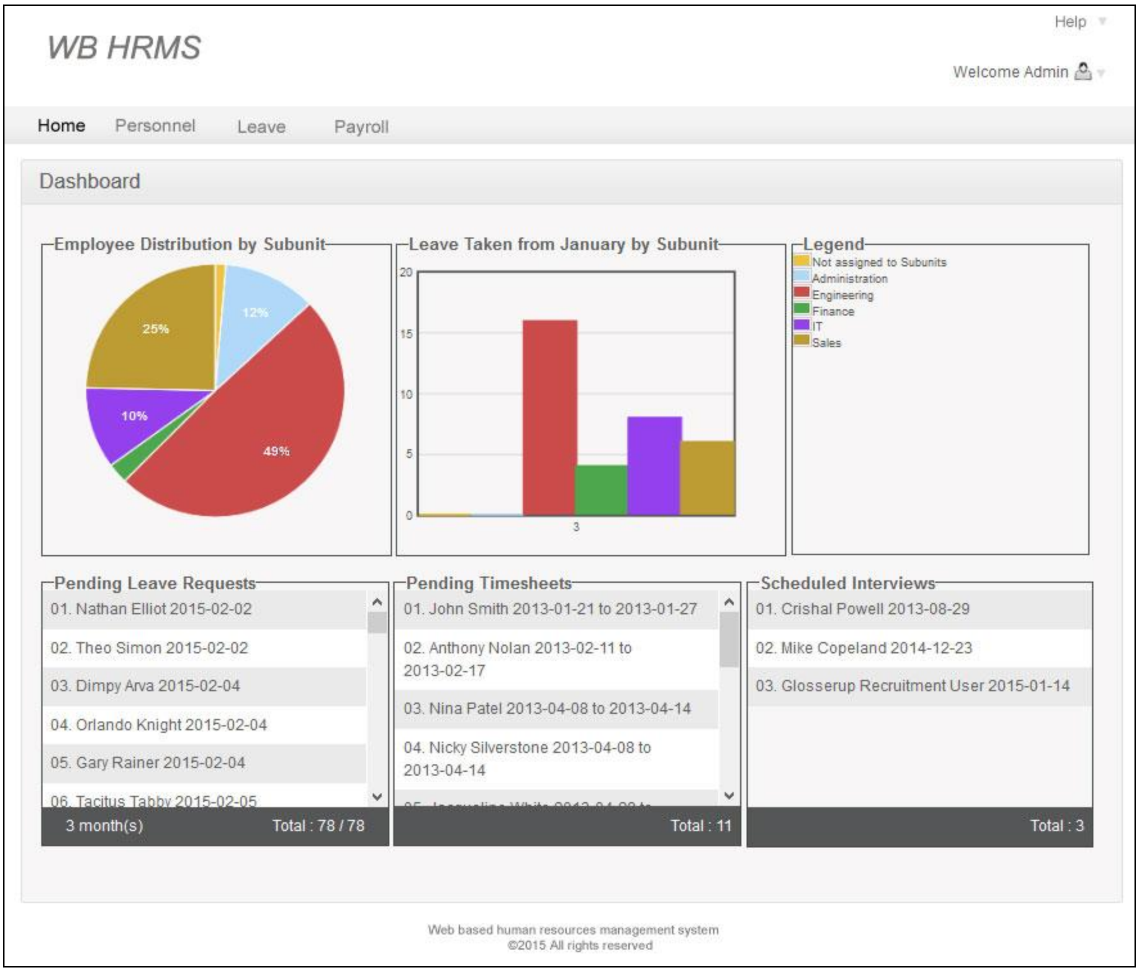
Usually, in the view of web applications pages such as payroll page several navigation nodes are presented. Except navigation classes, in the model navigation groups, such as Main Menu, Search, and Employee Menu are also presented. Each group contains one or more presentational classes. Each class present a particular information. From the model, we can see three submenus such as Payroll manager, which is the main page on the payroll management, salary calculation submenu contains two submenus connected directly to the salary calculation, and the last submenu is overtime details which contains the information page for approved overtime for employees. This module is used exclusively from HR department and focuses employees who are responsible for payroll calculations and monitoring. System has a special role dedicated to these employees. The administrator of the system monitors roles created by development and implementation teams.

5.5 The Web based Human Resources Management System Conceptual Implementation Process

In this section the conceptual implementation of web based human resources management system is presented. In Figure 28 a conceptual design of home page of the web based human resources management system, based on presentation model created above is present. This form has basic options that one home page should have. It is separated in 3 parts, first part is header which contains the logo of the system and link which leads to the helpdesk, and the basic form of login. In the content part, the home page is presented by a Dashboard which contains different reports generated by the system that can be seen quickly from the HR administrator.

The reports are generated from different data, taken from different tables of the database, which belong to different modules of the system. From the figure it is seen that the first three reports are taken from two different modules, Personnel Management and Leave management which present basic information about employee distribution in the organization and the number of Leaves taken by employees in the current month for a given department.

The second part of the reports present the number of Leave requests done by employees of the organization, which are still in procedure for approve or disapprove. Another quick report presented in the home page is information about employee contracts which will be ending in the near future.



Pending Leave Requests

01. Nathan Elliot 2015-02-02
02. Theo Simon 2015-02-02
03. Dimpy Arva 2015-02-04
04. Orlando Knight 2015-02-04
05. Gary Rainer 2015-02-04
06. Tacitus Tabby 2015-02-05

3 month(s) Total : 78 / 78

Pending Timesheets

01. John Smith 2013-01-21 to 2013-01-27
02. Anthony Nolan 2013-02-11 to 2013-02-17
03. Nina Patel 2013-04-08 to 2013-04-14
04. Nicky Silverstone 2013-04-08 to 2013-04-14

Total : 11

Scheduled Interviews

01. Crishal Powell 2013-08-29
02. Mike Copeland 2014-12-23
03. Glosserup Recruitment User 2015-01-14

Total : 3

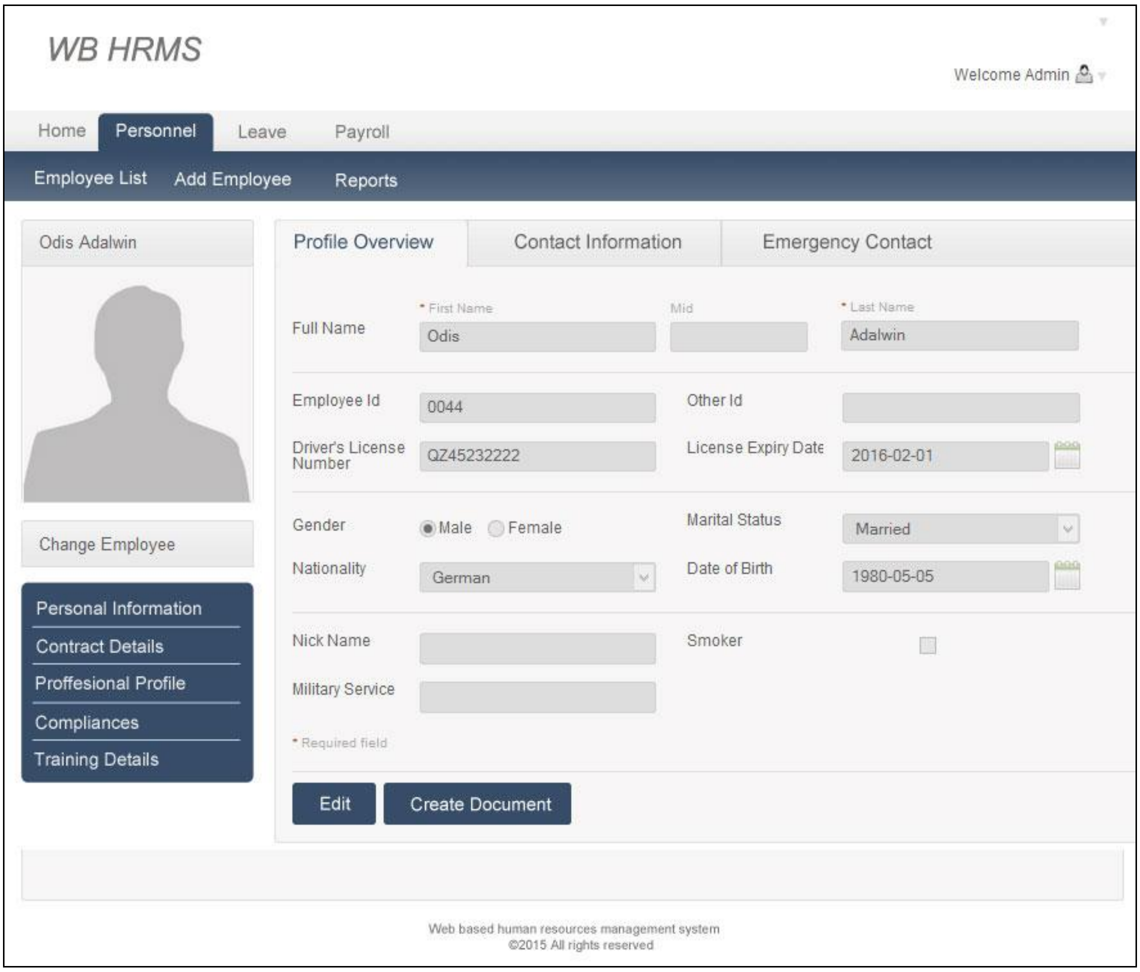
Web based human resources management system
©2015 All rights reserved

Figure 28. Web based HRMS home page concept *(Source: Own)*

5.5.1 Personnel Management Implementation Model

Personnel management web page design is presented in Figure 29. This module contains list of employees with all of their personal information. This web page is based on personnel management navigation model created in navigation structure in section 5.4.2 where it will have three levels of navigation.

First navigation level contains links of the overall modules related to the system, the second navigation contains subparts of the module that users can navigate. In this case there are three sub links which will help the users to have a better navigation in the page. The content of the page contains all information about the employee which can be found in the left menu in the page. Regarding the links that users navigate, the content of the page will change.



Full Name: * First Name Odis; Mid; * Last Name Adalwin

Employee Id: 0044
Other Id:

Driver's License Number: QZ45232222
License Expiry Date: 2016-02-01

Gender: ● Male ○ Female
Marital Status: Married

Nationality: German
Date of Birth: 1980-05-05

Nick Name:
Smoker: □

Military Service:

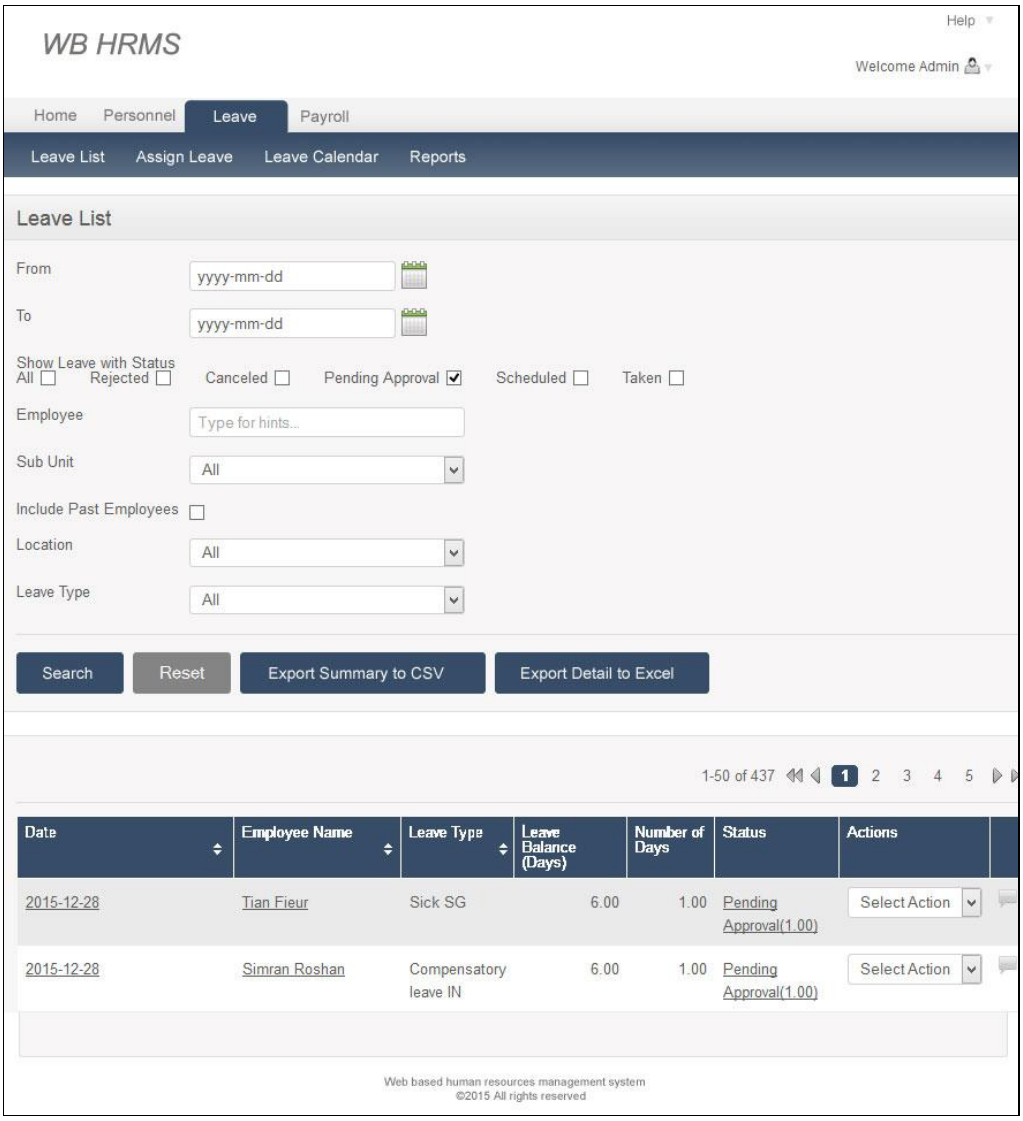
* Required field

Figure 29. Personnel management web design concept *(Source: Own)*

5.5.2 Leave Management Implementation Model

In figure 30 the design concept of Leave management web page is presented. This page contains the list of employees who have used holiday days or other type of leaves in the organization. The submenu contains the main links to operate with leave management.

Link Leave List contain the list of all employees who have at least one leave assigned in the system. Followed by link Assign Leave which is used by the HR employees to manually assign leave to the employees in case that they are not able to do it themselves. Another link is Leave Calendar which contains the calendar which allows users to have a better view of Leaves. And also there is a link Reports when users can generate different reports.



Leave List

From: yyyy-mm-dd

To: yyyy-mm-dd

Show Leave with Status: All □ Rejected □ Canceled □ Pending Approval ☑ Scheduled □ Taken □

Employee: Type for hints...

Sub Unit: All

Include Past Employees □

Location: All

Leave Type: All

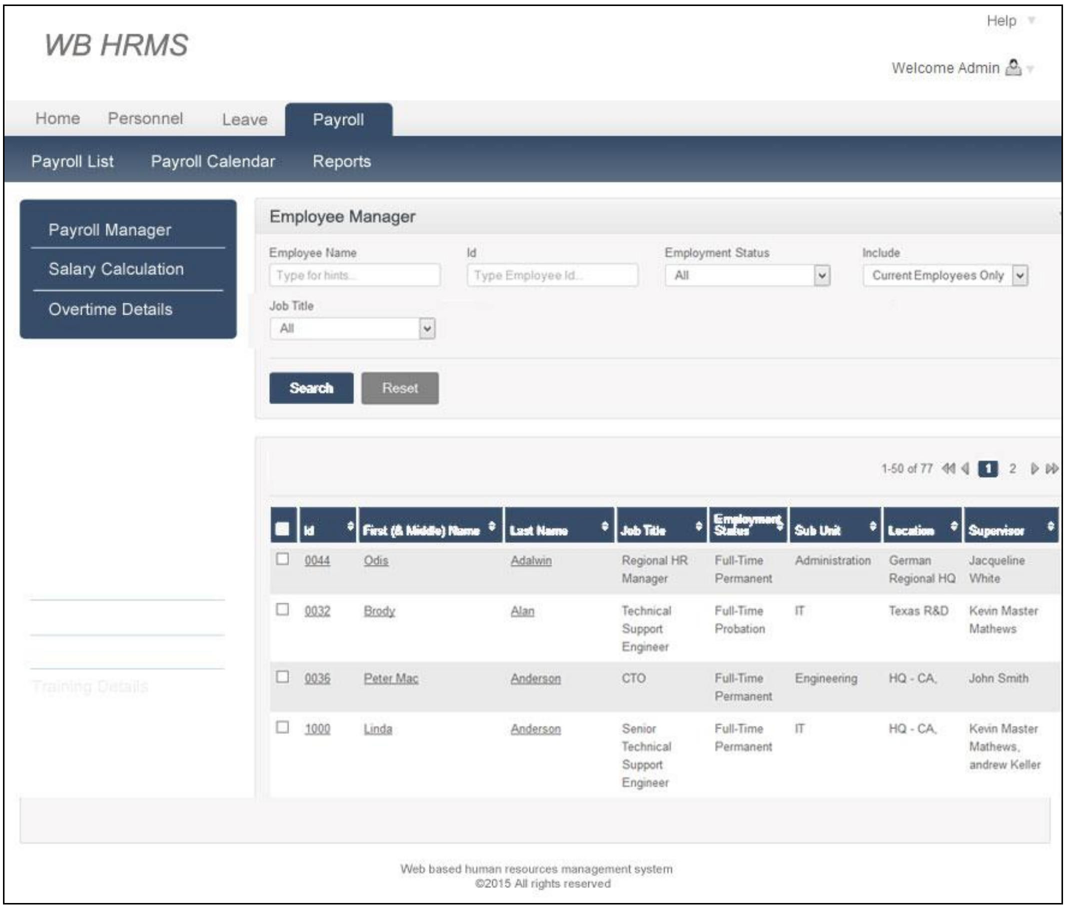
Date	Employee Name	Leave Type	Leave Balance (Days)	Number of Days	Status	Actions
2015-12-28	Tian Fieur	Sick SG	6.00	1.00	Pending Approval(1.00)	Select Action
2015-12-28	Simran Roshan	Compensatory leave IN	6.00	1.00	Pending Approval(1.00)	Select Action

Web based human resources management system
©2015 All rights reserved

Figure 30. Leave management web design concept *(Source: Own)*

5.5.3 Payroll Management Implementation Model

In Figure 31 a concept design for Payroll management is presented, created based on navigation and presentation model created before in this document. Same as other design concepts from other modules, this module contains three navigation menus. First navigation menu contains links which lead to other modules of the web system. The second navigation contains some quick view of the reports and monthly lists of payroll menu, and the third contains the main links to the salary calculation and employee salary management. Each link has different information connected in to the salary calculation.



□	Id	First (& Middle) Name	Last Name	Job Title	Employment Status	Sub Unit	Location	Supervisor
□	0044	Odis	Adalwin	Regional HR Manager	Full-Time Permanent	Administration	German Regional HQ	Jacqueline White
□	0032	Brody	Alan	Technical Support Engineer	Full-Time Probation	IT	Texas R&D	Kevin Master Mathews
□	0036	Peter Mac	Anderson	CTO	Full-Time Permanent	Engineering	HQ - CA,	John Smith
□	1000	Linda	Anderson	Senior Technical Support Engineer	Full-Time Permanent	IT	HQ - CA,	Kevin Master Mathews, andrew Keller

Web based human resources management system
©2015 All rights reserved

Figure 31 Payroll management web design concept *(Source: Own)*

Link Payroll manager found in left menu contains the list of employees which are needed to calculate salary. Also the link Salary Calculation is the main link when the overall salary is calculated. In the link Overtime Details approved overtime for employees is listed, which will be integrated as an extra work during salary calculation.

6. Comparison

6.1 Comparison of proposed system with an existing systems developed in different technologies

In this section is covered the comparison between web based human resources management system referred as Web based HRMS and traditional human resources management system referred as desktop based HRMS. Both development technologies have their advantages and disadvantages. Desktop based HRMS developed as a desktop information system is a standalone information system installed in a standalone machine. In the meantime they have been limited by hardware on which they are run.

The main disadvantage of desktop based HRMS is that they must be developed and deployed on a particular operating system, which they can require strict hardware requirements to be sure that they will function correctly. Another disadvantage is that updates or future changes of the application must be applied directly by the user to the machines of the clients where the application has been installed before. This dependency has typically a limited level of complexity in user interface. While with the invention of web information systems, the software development has changed the way (Conallen, 2002). This also changed the course of HRMS.

Web based HRMS developed in web technology using web services has started to replace desktop based HRMS. Web based HRMS use different architecture and usually are made in client-server architecture, where as a client interface a web browser is used, which is one of the reasons why web information systems are widely getting popular, not only in HRMS but in all types of software development. Following there is presented a comparison of desktop based HRMS and web based HRMS based on some predetermined parameters:

- **Maintenance** – Web based HRMS need to be installed only once in the server, while users from different organization can be connected through web browsers using internet. Traditional HRMS need to be installed separately on each client computer and can be used only by that user, also future updates of the traditional HRMS needs to be done in every single client computer which is not a case of web based HRMS, who have to be updated in only one computer and all the users can have them instantly.

- **Security** – Web based HRMS are exposed more to security risks than desktop applications. While for traditional HRMS we can have a total control over a standalone application and protect from various vulnerabilities, this may not be a case of web based HRMS as it uses internet when a large number of internet users wide the scale of threat, even though there are security policies used to prevent it.
- **Cost factor** – Web based HRMS development and its maintenance has higher costs but is more suitable for small and medium organizations as they don’t have to buy a whole system, in other hand traditional HRMS is purchased one time and the updates are harder to be done, also maintenance has a fee, which is not suitable for companies who have small number of employees and are located in different locations.

7. Evaluation

7.1 Evaluation of proposed information system

This document describes the design of Web based human resources management system used by organizations to manage their employees. Web based human resources management system has ten modules, separated from each other. In this document there are selected the main modules to analyse and design their main functionalities. Modules that are selected as a case study are Personnel management, Leave management, and Payroll management. Personnel management contains employee information which can be managed from HR employees, Leave management is a module which records employee leave request, and Payroll management is a module which is related with employee salary calculation. The main functionalities of these modules are depicted.

Web based HRMS is designed using UML-web based engineering methodology which uses UML notation and diagrams to represent different phases of system design. UWE has several phases which are followed while designing this proposed web based HRMS. The first phase is requirement analysis which uses UML use case and activity diagrams to depict behaviour of the system.

This document contains several use case diagrams for each module, which represent the main functionality of a particular module, also for better understanding the behaviour of users towards web based HRMS. After requirement of specification is finished it continues with conceptual model of the web based HRMS which is second phase of the UWE. Conceptual model provides a visual specification of the domain relevant information for the web based HRMS that mainly comprises the content of each module which is presented by UML class diagrams. There are used these two phases to create define a navigation structure of the web based HRMS. Related to the requirement analysis and the conceptual model, the navigation structure of web based HRMS is modelled.

Furthermore, following UWE methodology, the presentation structure is modelled, the central element of the presentation in web based HRMS is a page as a visualization unit.

Each page contains different nodes and links. The presentation model is created based on navigation structure which describes the behaviour-oriented aspects of the user interface of our Web based HRMS. With this model are included all aspects of web information design which are part of UWE methodology. While constructing the system using UWE methodology, it was needed to have a better presentation of the interface of the system. There is presented implementation model, which is a new step that contains conceptual design of web user interface, from which there are created different views which describe the behaviour of the administrator when he interacts with the system.

Finally, there has been an evaluation while comparing the proposed system with the existing systems with the same functionality, where there is found that the web based HRMS is more productive and meets more requirements for use by small and medium size organizations.

8. Conclusion

In this thesis the use of UML notation and diagrams and UWE methodology to design a Web based HRMS has been elaborated. UML is a standardized software modelling language but does not cover all aspects of web information design. For this reason it was needed to use an extension of UML such as UML-web based engineering which is entirely focused in design of web information system aspects. In the case study elaborated in this thesis a design of main modules of Web based HRMS has been created, where there are present all phases of web information system from a design view. The system is dedicated to small and medium organizations and it is designed to be developed as Software as a Service. This technology allows organizations to use a single information system, while they have branches separated in different places. Users can connect through internet to the system. This document also covers the research and analysis of books and papers related to UML and design method for web information system UML - web based engineering, furthermore, a detailed description of human resources management system and design of Web based HRMS is included. The work has been organized as follows.

The research and analysis of books and papers of the study are presented in Section 3. The UML notation and diagrams has been described, followed by an explanation of UML-web based engineering extension for design of web systems which has been used to design proposed web based system. In this section the description of the technology Software as a Service (SaaS) is also included, which suggests the future development and implementation of designed system. While in Section 4 a description of human resources management system including all modules related to it has been presented. This section describes the importance of human resources management system to be implemented in the organizations and the relation between modules of the system, and other systems which are not part of the HRMS. Furthermore, both sections explained above are included in section 5 which includes a practical work. This section covers the overall design of the web based human resources management system. Starting with the requirement analysis of the system explained by UML use case and activity diagrams. And then continuing with the development of conceptual model of each module described by class diagrams, followed by navigation model, presentation model and implementation design explained by the proposed web design.

9. Bibliography

Anerousis, N. & Mohindra, A., 2006. The Software-as-a-Service Model for Mobile and Ubiquitous Computing Environments. *Mobile and Ubiquitous Systems: Networking & Services, 2006 Third Annual International Conference on,* Volume 2, pp. 1-6.

Baumeister, H., Koch, N. & Mandel, L., 1999. Towards a UML extension for hypermedia. *UML 99' The Unified Modeling Language,* p. 614–629.

Bezivin, J. & Muller, P.-A., 1998. *The Unified Modeling Language.* Mulhouse(France): Springer-Verlag Berlin Heidelberg.

Booch, G. et al., 2007. *Object-Oriented Analysis and Design with Applications.* 3rd ed. s.l.:Addison-Wesley.

Booch, G., Rumbaugh, J. & Jacobson, I., 1999. *The Unified Modeling Language User Guide.* 1st ed. s.l.:Adison-Wesley.

Borges, R. M. & Mota, A. C., June 2007. Integrating UML and Formal Methods. *Electronic Notes in Theoretical Computer Science,* Volume 187, pp. 98-112.

Bray, T. et al., 2006. *Extensible Markup Language (XML) 1.1,* s.l.: W3C.

Chou, C. D., 2007. Analysis of a new information systems outsourcing practice:software-as-a-service business model. *International Journal of Information Systems and Change Management,* pp. 392-405.

Conallen, J., 2002. *Building Web Applications with UML.* 2nd ed. s.l.:Addison Wesley.

Eriksson, H.-E. & Penker, M., 2000. *Business Modeling with UML: Business Patterns at Work.* s.l.:John Wiley & Sons.

Fowler, M., 2003. *UML Distilled: A Brief Guide to the Standard Object Modeling Language.* 3rd ed. s.l.:Addison-Wesley.

G. Kappel, B. P. S. R. & Retschitz, W., 2004. *Web Engineering: Systematische Entwicklung von Web-Anwendungen.* s.l.:dpunkt-Verlag.

Gogolla, M. & Kobryn, C., 2001. *UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools: 4th International Conference.* Toronto: Springer.

Gomaa, H., 2011. *Software and modeling and design.* Cambridge: Cambridge University Press.

Grässle, P., Baumann, P. & Baumann, H., 2005. *UML 2.0 in Action.* s.l.:Packt Publishing.

Hendrickson, A. R., 2003. Human Resources Information Systems: Backbone Technology of Contemporary Human Resources. *journal of Labour Research,* 24(4), p. 381–394.

Hogg, K., Chilcott, P., Nolan, M. & Srinivasan, B., 2004. An evaluation of Web services in the design of a B2B application. *Proceedings of the 27th Australasian conference on Computer science,* pp. 331-340.

Isakowitz, T., Stohr, E. A. & Balasubramanian, P., 1995. RMM: a methodology for structured hypermedia design. *Communications of the ACM,* Volume 8, pp. 34-44.

Ivanka Menken, G. B., 2009. *Software as a Service and Web applications.* s.l.:Emereo Publishing.

Kaplan, J., 2007. SaaS: Friend Or Foe?. *Business Communication Review,* pp. 48-53.

Kappel, G., Proll, B., Reich, S. & Retschitzegger, W., 2006. *Web Engineering: The Discipline of Systematic Development of Web Applications.* s.l.:WILEY.

Kavanagh, M. J., Thite, M. & Johnson, R. D., 2011. *Human Resource Information Systems: Basics, Applications, and Future Directions.* 2nd ed. s.l.:SAGE Publications, Inc..

Koch, N., 2006. *Transformation techniques in the model-driven development process of UWE.* s.l., Workshop proceedings of the sixth international conference on Web engineering, ACM.

Koch, N., A.Knapp, Zhang, G. & Baumeister, H., 2008. UML-based web engineering. An Approach Based on Standards. *Web Engineering: Modelling and Implementing Web Applications. Springer Londo,* pp. 162-191.

Koch, N., Kraus, A. & Hennicker, R., 2001. *The authoring process of the uml-based web engineering approach.* s.l., First International Workshop on Web-Oriented Software Technology.

Lano, K., 2005. *Advanced Systems Design with Java, UML and MDA.* s.l.:Butterworth Heinemann.

Liao, H. & Tao, C., 2008. An Anatomy to SaaS Business Mode Based on Internet. *International Conference on Management of eCommerce and eGovernment,* pp. 215-220.

Liu, G., Jiang, H. & Geng, R., 2010. Software Design on a SaaS Platform. *Computer Engineering and Technology (ICCET), 2010 2nd International Conference on,* Volume 4, pp. 355-358.

Muller, P.-A., Bresch, D. & Studer, P., 2004. Model-Driven Architecture for Automatic-Control: An Experience Report. *The Unified Modeling Language,* pp. 260-274.

Noor, M. & Razali, R., 2011. Human Resources Information Systems (HRIS) for military domain - a conceptual framework. *Iernational Conference on Electrical,* p. 17–19.

OMG, 2010. *Unified Modeling Language: Superstructure specification, Version 2.4, ptc/2010-11-14,* s.l.: Object Management Group.

OMG, 2011. *OMG Unified Modeling Language(OMG UML),: Infrastructure specification, Version 2.4.1, formal/2011-08-05,* s.l.: Object Management Group.

Pilone, D. & Pitman, N., 2005. *UML 2.0 in a Nutshell.* s.l.:O'Reilly.

Ruël, H., Bondarouk, T. & Looise, J. K., 2004. E-HRM: Innovation or Irritation. An Explorative Empirical Study in Five Large Companies on Web-based HRM. *Management Revue,* 15(3), p. 364–381.

Rumbaugh, J., Jacobson, I. & Booch, G., 2004. *The Unified Modeling Language Reference Manual.* Boston: Addison-Wesley.

Schwabe, D. & Rossi, G., 1995. The object-oriented hypermedia design model. *Communications of the ACM,* Volume 38, pp. 45-46.

Schwinger, W., Koch, N. & I, 2006. Modeling Web Applications. In: *Web Engineering.* s.l.:WILEY, pp. 39-64.

Spence, C., Devoys, J. & Chahal, S., 2009. *Architecting Software as a Service for the Enterprise,* USA: Intel Corporation.

Stevens, P., Whitlle, j. & Booch, G., 2003. *UML 2003: The Unified Modeling Language.* San Francisco: Springer.

Vrana, I., 2009. *Projecting of Information Systems with UML.* Prague: CULS.

Waters, B., 2005. Software as a service: A look at the customer benefits. *Journal of digital asset management,* Volume 1, pp. 32-39.

Wazlawick, R. S., 2014. *Object-Oriented Analysis and Design for Information Systems.* 1st ed. s.l.:Elsevier Inc..

Weilkiens, T., 2008. *Systems Engineering with SysML/UML: Modeling, Analysis, Design.* s.l.:Morgan Kaufmann.

Werner, C., Kraatz, S. & Hogrefe, D., 2006. UML Profile for. *System Analysis and Modeling: Language Profiles,* Volume 4320, pp. 1-18.