# CHAPTER THREE

# REQUIREMENT AND ANALYSIS

## 3.1: Requirements Analysis in general

Requirements Analysis is the process of determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, analyzing, documenting, validating and managing software or system requirements. [1]

Requirements are description of how a system should behave or a description of system properties or attributes. It can alternatively be a statement of ‘what’ an application is expected to do. [1]

Requirements analysis includes three types of activities:

* Eliciting requirements: the task of identifying the various types of requirements from customers and users. This is sometimes also called requirements gathering.
* Analyzing requirements: determining whether the stated requirements are clear, complete, consistent and unambiguous, and resolving any apparent conflicts.
* Recording requirements: Requirements may be documented in various forms, usually including a summary list and may include natural-language documents, use cases, user stories, or process specifications. [1]

Requirements analysis can be a long and arduous process during which many delicate psychological skills are involved. New systems change the environment and relationships between people, so it is important to identify all the stakeholders, take into account all their needs and ensure they understand the implications of the new systems. Analysts can employ several techniques to elicit the requirements from the customer. These may include the development of scenarios (represented as user stories in agile methods), the identification of use cases, the use of workplace observation or ethnography, holding interviews, or focus groups (more aptly named in this context as requirements workshops, or requirements review sessions) and creating requirements lists. Prototyping may be used to develop an example system that can be demonstrated to stakeholders. Where necessary, the analyst will employ a combination of these methods to establish the exact requirements of the stakeholders, so that a system that meets the business needs is produced. [1]

In general, requirements are partitioned into functional requirements and non-functional requirements. Functional requirements are associated with specific functions, tasks or behaviors the system must support (e.g. the system must output the data in json format, or the system must provide the login function). While non-functional requirements are constraints on various attributes of these functions or tasks. The functional requirements address the quality characteristic of functionality while the other quality characteristics are concerned with various kinds of non-functional requirements (e.g. the system must be easily accessible, the quality must follow a particular standard). Because non-functional requirements tend to be stated in terms of constraints on the results of tasks which are given as functional requirements (e.g. constraints on the speed or efficiency of a given task)?

Software prototyping are used to gather the functional and non-functional requirements.

### 3.1.1: Software prototyping

Software prototyping refers to the activity of creating prototypes of software applications, i.e., incomplete versions of the software program being developed. It is an activity that can occur in software development and is comparable to prototyping as known from other fields. A prototype typically simulates only a few aspects of, and may be completely different from the final product. The main purpose of a prototype is to allow users of the software to evaluate developers' proposals for the design of the eventual product by actually trying them out, rather than having to interpret and evaluate the design based on descriptions. Prototyping can also be used by end users to describe and prove requirements that developers have not considered. [2]

### 3.1.2: Outline of the prototyping process

The process of prototyping involves the following steps

1. Identify basic requirements

Determine basic requirements including the input and output information desired. Details, such as security, can typically be ignored.

1. Develop Initial Prototype

The initial prototype is developed that includes only user interfaces.

1. Review

The customers, including end-users, examine the prototype and provide feedback on additions or changes.

1. Revise and Enhance the Prototype

Using the feedback both the specifications and the prototype can be improved. Negotiation about what is within the scope of the contract/product may be necessary. If changes are introduced then a repeat of steps #3 and #4 may be needed. [2]

## 3.2: Software Requirements Specification

A Software requirements specification (SRS), a requirements specification for a software system, is a complete description of the behavior of a system to be developed and may include a set of use cases that describe interactions the users will have with the software. In addition it also contains non-functional requirements. [3]

We will discuss the Use case diagram on the next page.

### 3.2.1: USE CASE Diagram



**Figure x: Use Case Diagram – Administrator/HR Administrator**



**Figure y: Use Case Diagram - Employee**

### Table x: Actors Grid

|  |  |  |
| --- | --- | --- |
| ID | Name | Related use cases |
| 1 | Administrator/HR Administrator | Manage User,  Manage Employee,  View Attendance,  Manage Pay Rate,  Manage Job Title,  Manage Employment Status,  Manage Job Category,  Manage Department,  Manage Overtime Pay Rate,  Manage Salary Adjustment,  View Overtime Chart,  View Total Work Hours Chart,  View Hourly Payroll Chart,  View Payslip |
| 2 | Employee | Update Personal Details,  View Job Details,  View Salary Details,  View Total Work Hours Chart,  View Hourly Payroll Chart,  View Overtime Chart,  View Payslip |

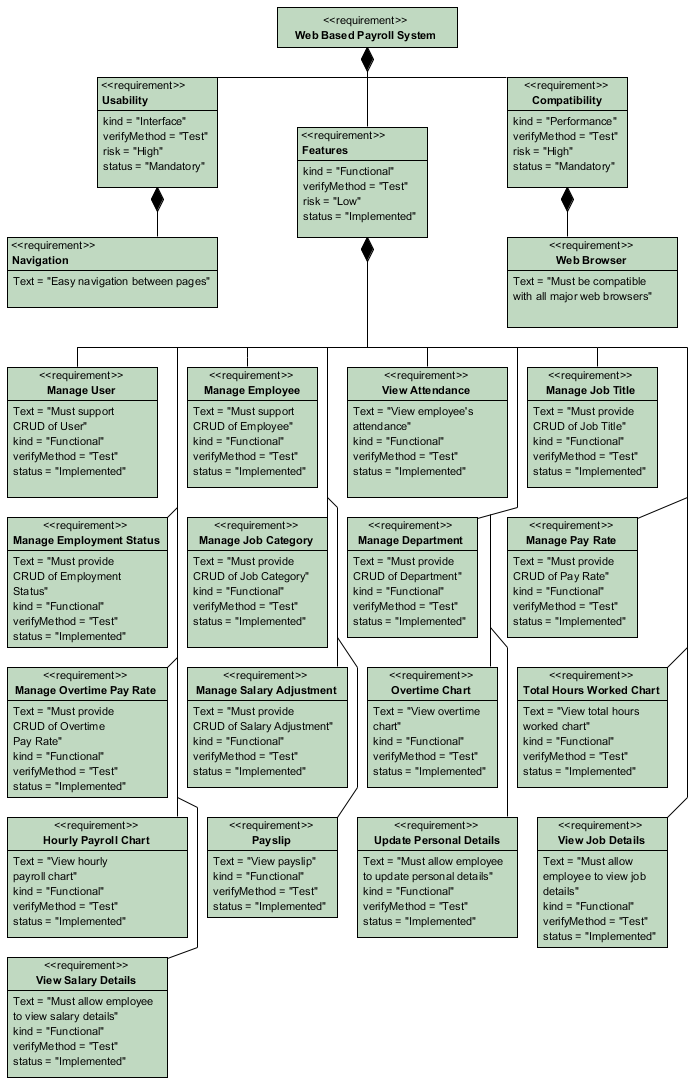
**Table y: Use Cases Grid**

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Name | Primary actors | Supporting actors |
| 1 | Manage User | Administrator/HR Administrator |  |
| 2 | Manage Employee | Administrator/HR Administrator |  |
| 3 | View Attendance | Administrator/HR Administrator |  |
| 4 | Manage Pay Rate | Administrator/HR Administrator |  |
| 5 | Manage Job Title | Administrator/HR Administrator |  |
| 6 | Manage Employment Status | Administrator/HR Administrator |  |
| 7 | Manage Job Category | Administrator/HR Administrator |  |
| 8 | Manage Department | Administrator/HR Administrator |  |
| 9 | Manage Overtime Pay Rate | Administrator/HR Administrator |  |
| 10 | View Overtime Chart | Administrator/HR Administrator,  Employee |  |
| 11 | View Total Work Hours Chart | Administrator/HR Administrator,  Employee |  |
| 12 | Manage Salary Adjustment | Administrator/HR Administrator |  |
| 13 | View Hourly Payroll Chart | Administrator/HR Administrator,  Employee |  |
| 14 | View Payslip | Administrator/HR Administrator,  Employee |  |
| 15 | Update Personal Details | Employee |  |
| 16 | View Job Details | Employee |  |
| 17 | View Salary Details | Employee |  |

**Table z: Use Case Diagram Summary**

|  |  |
| --- | --- |
| Name | Documentation |
| Image1.png [Administrator/ HR](#MA0pDpSGAqACA1AL) Administrator | Manage employee details, manage salary structure for each employee, and generate employee payslip at end of month. |
| Image2.png [Manage](#Ji0JDpSGAqACAzOd) User | Create, Update, and Delete user. A user is required in order to access the system. |
| Image2.png [Manage](#V3TJDpSGAqACA0Gc) Employee | Create, Update, and Delete employee. Employee details include personal details, contact details, job details, salary details, and qualification details. |
| Image2.png [View](#nd3JDpSGAqACA0Yy) Attendance | View employee’s attendance record. |
| Image2.png Manage Pay Rate | Create, Update, and Delete pay rate. |
| Image2.png [Manage](#z6rZDpSGAqACA371) Job Title | Create, Update, and Delete job title. |
| Image2.png [Manage](#iGWZDpSGAqACA3eV) Employment Status | Create, Update, and Delete employment status. |
| Image2.png [Manage](#j5nZDpSGAqACA4Cr) Job Category | Create, Update, and Delete job category. |
| Image2.png [Manage](#_GvZDpSGAqACA4Jd) Department | Create, Update, and Delete department. |
| Image2.png [Manage](#0qu5DpSGAqACA5TJ) Overtime Pay Rate | Create, Update, and Delete overtime pay rate. |
| Image2.png [Manage](#eH05DpSGAqACA429) Salary Adjustment | Create, Update, and Delete salary adjustment. |
| Image2.png [View](#E615DpSGAqACA5r1) Overtime Chart | View monthly employee’s overtime chart. |
| Image2.png [View](#NeT5DpSGAqACA5zR) Total Hours Worked Chart | View hourly paid employee’s total hours worked chart. |
| Image2.png [View](#5zCFDpSGAqACA6p9) Hourly Payroll Chart | View employee’s payroll chart. |
| Image2.png [View](#AKA5DpSGAqACA4QH) Payslip | The home page of the user where all the friend's, group's, page's, activities appears. |
| Image1.png [Employee](#rQWpDpSGAqACA1CD) | The employee who can access system to update personal details, view salary details, view payroll charts, and generate payslip. |
| Image2.png Update Personal Details | Allows authenticated employee to update his/her personal details. |
| Image2.png View Job Details | Allows authenticated employee to view his/her job details. |
| Image2.png View Salary Details | Allows authenticated employee to view his/her salary details. |
| Image2.png View Total Hours Worked Chart | Allows authenticated hourly paid employee to view his/her total number of hours worked chart. |
| Image2.png View Hourly Payroll Chart | Allows authenticated hourly paid employee to view his/her hourly payroll chart. |
| Image2.png View Overtime Chart | Allows authenticated monthly paid employee to view his/her overtime chart. |
| Image2.png View Payslip | Allows authenticated employee to view/print his/her payslip. |
| Image3.png [Payroll](#ovOhDpSGAqACAyo2) System | A system which allows Administrator to manage employee details, manage salary structure for each employee, and generate employee payslip at end of month, and allows employee to update personal details, view salary details, view payroll charts, and generate payslip. |

### 3.2.2: Requirement Diagram



### 3.2.3: Operating Environment

In computing, an operating environment is the environment in which users run application software, whether by a command-line interface (such as in MS-DOS or the UNIX shell) or a graphical user interface (such as in the Macintosh operating system or a web browser). [4] Table X has briefly shows the operating environment of the Web based Payroll System.

**Table X: Operating Environment**

|  |  |  |
| --- | --- | --- |
| **No** | **Requirement** | **Reason** |
| 1 | Google Chrome | Google Chrome web browser best work with the application |
| 2 | Any Operating System with decent web browser | To access the system via local intranet or internet |
| 3 | Ruby on Rails with jQuery UI as the client side user interface, jQuery as the javascript library, and Highcharts to display charts | To develop the system |

### 3.2.4: Constraints and Dependencies

The Payroll System is dependent on jQuery [5], JQuery UI [6], and Highcharts [7] JavaScript library. jQuery is used to send Ajax request to Ruby code on the server side, resulting in a fast and asynchronous data transmission in a web browser. Meanwhile jQuery UI is used to create highly interactive web interface.

Highcharts is used to display intuitive and interactive charts in the payroll web application.

**3.2.4.1: Dependency on jQuery**

The system is completely dependent on the jQuery as the system is a fully ajax enabled web application. The goal of using jQuery is because it is fast, small, and feature-rich JavaScript library. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

**3.2.4.2: Dependency on jQuery UI**

The system uses this library to create the front end web interface. jQuery UI is built on top of jQuery JavaScript library that provides abstractions for low-level interaction and animation, advanced effects and high-level, themeable widgets.

This library is completely dependent on the jQuery library.

**3.2.4.3: Dependency on Highcharts**

Highcharts is a charting library written in pure JavaScript, which offers intuitive, interactive charts to a web application. The system uses this library to show various chart types, such as line, bar, and pie.

**Table x: Chart used**

|  |  |  |
| --- | --- | --- |
| **Category** | **Chart Type** | **Description** |
| Overtime Chart | Bar Chart | Shows monthly paid employee’s overtime record for a particular year. |
| Total Work Hours Chart | Line Chart | Shows hourly paid employee’s total working hours worked record for a particular year. |
| Hourly Payroll Chart | Pie Chart & Bar Chart | Shows hourly paid employee’s payroll amount for a particular year. |

**3.2.5: System Features requirement**

Section below will illustrate the required features of the Payroll System. The feature requirements fall under the function requirements because they interact directly with the user. [8]

#### 3.2.5.1: Feature 1: User CRUD functions

Create, Read, Update, and Delete User. This feature allows administrator to create or maintain users which allow to access the system. Each user can be enabled or disabled. If a user is disabled, the user will not able to access the system. The administrator can search by username, employee name, user role, and status.

#### 3.2.5.2: Feature 2: Employee CRUD functions

Create, Read, Update, and Delete Employee. This feature allows administrator to create or maintain employees in the system. Each employee can be associated with a user to allow them to access the system. The administrator can search by employee, employment status, department, staff id, and job title.

**3.2.5.3: Feature 3: View Attendance**

This feature allows the administrator to view employee’s attendance record.

The administrator can search by employee and working date.

**3.2.5.4: Feature 4: Job Title CRUD**

Create, Read, Update, and Delete Job Title. This feature allows administrator to create or maintain job titles in the system. Each employee should have a job title. The administrator can search by keyword.

**3.2.5.5: Feature 5: Employment Status CRUD**

Create, Read, Update, and Delete Employment Status. This feature allows administrator to create or maintain employment statuses in the system. Each employee should have an employment status. Example of employment status would be Confirmed, Probation. The administrator can search by keyword.

**3.2.5.6: Feature 6: Job Category CRUD**

Create, Read, Update, and Delete Job Category. This feature allows administrator to create or maintain job categories in the system. Each employee should have a job category. The administrator can search by keyword.

**3.2.5.7: Feature 7: Department CRUD**

Create, Read, Update, and Delete Department. This feature allows administrator to create or maintain departments in the system. Each employee should belong to a department.

The administrator can search by keyword.

**3.2.5.8: Feature 8: Pay Rate CRUD**

Create, Read, Update, and Delete Pay Rate. This feature allows administrator to create or maintain pay rates in the system. The administrator can define pay rate for each hourly paid employee, for each year. The administrator can search by staff id, year, and month.

**3.2.5.9: Feature 9: Overtime Pay Rate CRUD**

Create, Read, Update, and Delete Overtime Pay Rate. This feature allows administrator to create or maintain overtime pay rates in the system. The administrator can define rate for each year. The administrator can search by year.

**3.2.5.10: Feature 10: Salary Adjustment CRUD**

Create, Read, Update, and Delete Salary Adjustment. This feature allows administrator to create or maintain salary adjustments in the system. The administrator can define adjustment for a particular employee, for a particular year. The administrator can search by staff id, year, and month.

**3.2.5.11: Feature 11: Overtime Chart**

Shows monthly paid employee’s overtime record for a particular year. The administrator can see the chart for a particular employee and for a particular year. It also has option to select months for the chart to be generated.

**3.2.5.12: Feature 12: Total Work Hours Chart**

Shows hourly paid employee’s total working hours worked record for a particular year.

The administrator can see the chart for a particular employee and for a particular year. It also has option to select months for the chart to be generated.

**3.2.5.13: Feature 13: Hourly Payroll Chart**

Shows hourly paid employee’s payroll amount for a particular year. The administrator can see the chart for a particular employee and for a particular year. It also has option to select months for the chart to be generated.

## **References**

|  |  |
| --- | --- |
| [1] | Wikipedia, "Requirements analysis - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Requirements\_analysis. [Accessed March 2013]. |
| [2] | Wikipedia, "Software prototyping - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Software\_prototyping. [Accessed March 2013]. |
| [3] | Wikipedia, "Software requirements specification - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Software\_requirements\_specification. [Accessed March 2013]. |
| [4] | Wikipedia, "Operating environment - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Operating\_environment. [Accessed March 2013]. |
| [5] | jQuery, "jQuery," [Online]. Available: http://jquery.com/. [Accessed March 2013]. |
| [6] | jQuery UI, "jQuery UI," [Online]. Available: http://jqueryui.com/. [Accessed March 2013]. |
| [7] | Highcharts JS, "Highcharts - Interactive JavaScript charts for your webpage," [Online]. Available: http://www.highcharts.com/. [Accessed March 2013]. |
| [8] | Wikipedia, "Functional requirement - Wikipedia, the free encyclopedia," [Online]. Available: http://en.wikipedia.org/wiki/Functional\_requirements. [Accessed March 2013]. |