

Requirements document

codedCoffee



ITRW 213-REPORT 2

Marco Taljaard 26694379, Clarise bouwer 26152673, dylan jansen van vuuren 26009528, quintin jacobs 26147629, zander jagals 25855883, braam van der berg 25849182, stefan malan 26128225

Table of Contents

[1. Introduction 1](#_Toc448385632)

[2. Functional Requirements 1](#_Toc448385633)

[2.1 Login interface 1](#_Toc448385634)

[2.2 Main Program interface 2](#_Toc448385635)

[2.3 Stock interface 2](#_Toc448385636)

[2.4 Orders interface 3](#_Toc448385637)

[2.5 Sales interface 3](#_Toc448385638)

[3. Non-functional Requirements 4](#_Toc448385639)

[4. Candidate Systems Matrix 6](#_Toc448385640)

[5. Feasibility Analysis Matrix 8](#_Toc448385641)

[6. Use-Case 11](#_Toc448385642)

[6.1 Use-Case Glossary 12](#_Toc448385643)

[6.1.1 Actor List 12](#_Toc448385644)

[6.1.2 Use-Case List 12](#_Toc448385645)

[6.2 Use-Case Model Diagram 14](#_Toc448385646)

[7. Examples 16](#_Toc448385647)

[7.1 Fact-Finding Techniques 16](#_Toc448385648)

[7.1.1Observations 16](#_Toc448385649)

[7.1.2 Interview 16](#_Toc448385650)

[7.1.3 Research 17](#_Toc448385651)

[7.1.4 Questions 17](#_Toc448385652)

[7.1.5 Joint Requirements Planning 18](#_Toc448385653)

[7.1.6 Prototyping 18](#_Toc448385654)

[8. Closing 19](#_Toc448385655)

[8.1 Future & Further Planning 19](#_Toc448385656)

[8.1.1 PERT Chart 19](#_Toc448385657)

[8.1.2 Gantt Chart 21](#_Toc448385658)

[9. Summary 26](#_Toc448385659)

[Bibliography 27](#_Toc448385660)

List of Figures

[Figure 1: Use-Case Model Diagram 15](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385811)

[Figure 2: Database Prototype- Branches 19](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385812)

[Figure 3: Database Prototype- Employee 19](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385813)

[Figure 4: Database Prototype- Orders 19](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385814)

[Figure 5: Database Prototype- Stock 19](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385815)

[Figure 6: PERT Chart 20](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385816)

[Figure 7.1: Gantt Chart- 7 Feb 2016 - 20 March 2016 22](file:///C:\Users\Marco\Desktop\Requirements%20document.docx#_Toc448385817)

Figure 7.2: Gantt Chart- 20 March 2016 - 5 June 2016……………………………………………..23

Figure 7.3: Gantt Chart- 5 June 2016 - 21 Aug 2016………………………………………………..24

Figure 7.4: Gantt Chart- 21 Aug 2016 - 9 Oct 2016……………………………………………….…25

# 1. Introduction

Requirements analysis is defined as “the process of determining user expectations for a new or modified product” (Rouse 2007:1). In this report requirements analysis will be conducted on stockI.T, a system being developed by codedCoffee. The various steps of requirements analysis, including some examples and the fact-finding techniques used, will be discussed.

# 2. Functional Requirements

Functional requirements are descriptions of the activities and services that the system must perform or provide. The following are the functional requirements for stockI.T:

## 2.1 Login interface

|  |  |
| --- | --- |
| Inputs | * Gives the user the option of creating a new user and logging in with an existing account. * User enters his/her password and username. |
| Processing | * Cross-references data in a login database to verify the validness of the entered username and password. * Sends an email to the user when they forgot their password. * If a new user is being created, the manager’s authorization is required to enable the user. * Saves the username, date and time of the user logging in. |
| Outputs | * Notifies the user via a pop-up message when, depending on the validness of the entered username and password. |

## 2.2 Main Program interface

|  |  |
| --- | --- |
| Input | * User selects what he/she wants to do such as place an order, add stock, make a sale, etc. * The user has the option to log out. |
| Processing | * The system enables to the form that is requested by the user. * The system hides some components and shows others to change the layout of the interface, according to the request. * The program logs the user out of the system and returns to the login form. * Saves the username, date and time of when the user logs out. |
| Output | * The requested form is displayed. * Notifies the user that the system will log out. |

## 2.3 Stock interface

|  |  |
| --- | --- |
| Input | * User can add new stock into the system. * Manager can remove stock when necessary. |
| Processing | * The stock must be added into the database. * The stock must be removed from the database when the item is sold or removed by the manager. |
| Output | * Notifies the user that stock of a certain item is low. * Prompts the user for verification when an item is being removed or added. |

## 2.4 Orders interface

|  |  |
| --- | --- |
| Input | * Place an order into the system. * Ask for a report of the orders. * Employee can place a special order. |
| Processing | * Notifies the customer when the repair of his/her computer is complete. * Verifies that the special order’s stock is unavailable at the specific branch, in order to finalize the special order. |
| Output | * Notifies the customer via E-mail when his/her special order has arrived or has been delayed. * Notifies the customer via E-mail when his/her repair has been completed. * Displays the order report requested by the employee. |

## 2.5 Sales interface

|  |  |
| --- | --- |
| Input | * Manager can request a report of sales. * Employee inputs sale when a repair has been completed. * Employee enters the item being sold. |
| Processing | * Gets the data for the sales report. * Enters the sale’s detail into the sales database. |
| Output | * Print receipt for sale. * Display the sales report. * Prints the repair document for when the customer comes to retrieve his/her repaired item. |

# 3. Non-functional Requirements

|  |  |
| --- | --- |
| Non-Functional requirement | Description |
| *Performance* | * Speed   + Different processes :     - Inventory Control: Inventory on hand must be entered into the system. Once new stock arrives, it should be entered into the system. Stock that is sold should be removed from the system.     - Note: When an item’s stock falls below a certain limit, the system must notify the user that new stock should be ordered. Furthermore, customers should automatically be notified when their item(s) restoration is completed. The system user should also receive a notification when a “special order" has been placed.     - Speed ​​and quality: It is aimed to make the user interface as user-friendly as possible and thereby improving the speed, efficiency and quality of the system.     - Workers Identity Number: Every employee of Matrix Warehouse needs to use a worker identity number to sign into the system to use the system.   + Response Time     - The size of the inquiries, the internet speed, hardware of the computer and the quality of the system affects the response time of the system. In other words there may not currently be a fixed response time set for the system. The expected response time of the system is relatively high. |
| *Information* | * Output   + Lack of required information :     - Database requests: Employees who fail to enter all the necessary information, compromise the accuracy and performance of the system (Garbage in Garbage Out). * Input   + Data is entered inaccurately or incorrectly.   + Human error: The system requires data to be entered correctly, which can lead to important data being lost or mixed in with other data. |
| *Economics* | * Prices   + Prices: The current hardware of the computers will be upgraded, to use the system optimally. Some of the computers will need hardware and software upgrades. If the customer "version” of the system will be developed, there may a need for additional computers to be purchased. * Profit   + Although the purpose of the system is not to directly increase the company's profit, but rather to increase the efficiency and speed, will the company’s profit primarily stay unchanged. |
| *Security* | * Security Control   + Although the current system security is adequate, there is still an opportunity for improvement :     - The workers identity number could be improved upon by making the identity number more complicated and by verifying the identity number. |
| *Efficiency* | * To make the system more efficient, access to data should be simpler and easier. The database should also be designed in such a way that it is more simplistic and detailed oriented. To furthermore increase upon efficiency, the system’s user interface should be more user-friendly. The system will also be more effective if the accuracy of the data is improved. |
| *Service* | * Higher efficiency of the system will lead to faster and better customer service. The customer service will improve if the people who use the system, gained faster access to the data. |

# 4. Candidate Systems Matrix

The candidate systems matrix is used to “summarize and compare the attributes of alternative solutions generated” (Hoffer *et al.*, 2005:247). The following is the candidate systems for the system being developed by codedCoffee:

|  |  |  |  |
| --- | --- | --- | --- |
| Characteristics | Candidate 1 | Candidate 2 | Candidate 3 |
| Brief Description | Do nothing: The software and business processes continue as it has been done till this point. | A system is developed that can be used to maintain and manage an inventory and sales database. | The same as candidate 2 plus an additional part is added which customers can use as a “self-help station” to make “special orders” and view stock. |
| Portion of System Computerized | N/A | Fully supports the current system’s business processes. Plus more efficient interaction with the system and the automatic notification of customers to pick up repairs. | The same as candidate 2, plus the customers would be able to check stock themselves and if stock is not available, make a “special order”. |
| Benefits | The business processes won’t be disturbed and employees wouldn’t have to be trained to use the new system. | The system would be much easier to use. The employees’ work would be a lot less time consuming and easier to perform. | The same as candidate 2, plus client service would be greatly improved upon. |
| Technology | N/A | PC with modem. | The same as candidate 2. |
| - Servers and Workstations | N/A | Technically architecture dictates third generation i3 processors and MS Windows 7 class servers and workstations (clients). | The same as candidate 2. |
| -Output Devices and Implications | N/A | (1) Standard laser printer (depending on branch size).  (1) PRINTRONIX bar-code printer (includes software & drivers) | The same candidate 2, plus network for verification of client’s special order. |
| -Input Devices and Implications | N/A | Keyboard and mouse.  Considerations:  (2) PSC Quickscan laser barcode scanners (depending on number of cashiers). | The same as candidate 2. |
| -Storage Devices and Implications | N/A | MS SQL Server DBMS with PC hard disk (at least 500GB). | The same as candidate 2. |
| -Software Tools Needed | N/A | MS Visual Studio  MS Access  MySQL  C#  Candidates:  Google Chrome  Notepad  HTML5  CSS (Cascading Style Sheets) | The same as candidate 2. |
| -Application Software | N/A | Custom Solution | The same as candidate 2. |
| Interfaces | N/A | PC Monitor (Access Interfaces)  Candidates:  Web Pages | The same as candidate 2. |
| Processes | Same as current processes. | Processes for repairs will be changed.  Processes for communication with headquarters. | The same as candidate 2, plus  customer order processing will be changed. |
| -Method of Data Processing | N/A | Batch processing in combination with remote bath and real-time. | The same as candidate 2. |

# 5. Feasibility Analysis Matrix

Feasibility is a measurement of how practical and beneficial an information system will be to an organization. During system analysis there are three feasibility checkpoints. These checkpoints include the scope definition checkpoint, problem analysis checkpoint and decision analysis checkpoint. After these checkpoints there will be a decision made if the system development works shall continue or be cancelled.

At the scope definition checkpoint, the accuracy of the problems and opportunities are measured. The problem analysis checkpoint focus more on the detailed measurement of the current system’s problem analysis. The last feasibility checkpoint is the decision analysis checkpoint. At this checkpoint, the software, hardware, input methods and output methods requirements and the detailed estimation of the development cost and is available.

There are six tests for feasibility analysis; legal feasibility, operational feasibility, cultural feasibility, economic feasibility, technical feasibility, and schedule feasibility.

Legal feasibility is a measurement, regarding the existing legal and organization’s policy, of how well a solution can be implemented.

Operational feasibility is a measurement of how well the requirements of the system meets the solution. It is also a measurement of how well the system requirements was identified in the requirements analysis phase.

Cultural feasibility, also called political feasibility, is related to operational feasibility. The cultural feasibility is a measurement of how the users feel about a proposed system. The concern with this feasibility is whether the system is adaptable to the organizational environment.

Economic feasibility is a measurement of to identify the costs related to the development project and to identify the financial benefits.

Technical feasibility is a measurement of the availability of technical expertise as well as the practically of the technical solution.

Schedule feasibility is a measurement to find out how reasonable a project’s timetable is. The main concern is if the tasks will be completed within the deadlines. It is very important to define headlines for every phase of a project.

The following are each of these six tests, performed on stockI.T:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Feasibility Criteria:** | **Weight:** | **Candidate 1** | **Candidate 2** | **Candidate 3** |
| Description: |  | Do nothing: continue current business process. | Design a new stock managing application using Visual Studio, MySQL, Notepad and Chrome. | The same as candidate 2,plus  design a customer self-help application using Visual Studio, MySQL, Notepad and Chrome. |
| Operational Feasibility: | 15% | N/A | Supports user-required functionality fully.  Score: 100 | Supports user-required functionality fully.  Score: 100 |
| Cultural Feasibility: | 10% | N/A | No foreseeable problems.  Score: 100 | No foreseeable problems.  Score: 100 |
| Technical Feasibility: | 20% | N/A | Solution requires writing application using Visual Studio, MySQL, Notepad and Chrome.  Technical staff has training in Visual Studio, and will easily be able to learn MySQL and Chrome.  Score: 85 | Although the current technical staff is used to working with Visual Studio, it will be easy to learn the use of MySQL, Notepad and Chrome.  Score: 75 |
| Economic Feasibility: | 30% | N/A | All the software being used is free.  The cost to develop will include the salary of 7 system analysts. The salary of each system analyst will be approximately R30 000. The cost over nine months will then be R 1 890 000 for labor.  Score: 60 | All the software being used is free.  The salary of each system analyst will be approximately R30 000. The cost to develop will include the salary of 7 system analysts. The cost over eight months will then be R 1 680 000 for labor.  Score: 70 |
| Schedule Feasibility: | 15% | N/A | 8-9 months  Score: 70 | 8 months  Score: 75 |
| Legal Feasibility: | 10% | N/A | No foreseeable problems.  Score: 100 | No foreseeable problems.  Score: 100 |
| **Weighted Score:** | 100% | N/A | 80.5 | 82.25 |

# 

# 6. Use-Case

A use-case is defined as a “behaviorally related sequence of steps (a scenario), both automated and manual, for the purpose of completing a single business task” (Bentley, 2007:246). The following is related to the use-cases in relation to stockI.T:

## 6.1 Use-Case Glossary

### 6.1.1 Actor List

A user-case actor is anything or anyone that has to interact with the system in order to exchange information. The following are the actors involved in the stockI.T system:

|  |  |
| --- | --- |
| Term | Description |
| Customer | An individual that makes purchases or uses the services provided by the company. |
| Employee | An individual that is employed by the company to fulfill the activities required by the company. |
| Manager | An individual that participates in managerial activities for the company. |
| Supplier | A company that provides stock to the company in exchange for money. |

### 6.1.2 Use-Case List

|  |  |  |
| --- | --- | --- |
| Use-Case Name | Use-Case Description | Participating Actors and Roles |
| Add Stock | This use-case describes the event where the user adds stock. Extends and initiates the verify stock use-case. | * Employee (Primary business) * Manager (Primary business) |
| Cancel Order | This use-case describes the event where the user cancels an order. Depends on the revise order use-case. | * Manager (Primary business) * Supplier (External server) |
| Login | This use-case describes the event where a user logs into the system. Initiates the verify-user use-case. | * Employee (Primary business) * Manager (Primary business) |
| Make Product Inquiry | This use-case describes the event where either the manager or customer makes an inquiry about a specific product. | * Customer (External receiver) * Manager (Primary business) |
| Make Purchases | This use-case describes the event where the customer makes a purchase. Extends the remove stock use-case. | * Customer (External receiver) * Employee (Primary business) |
| Make Purchase History Inquiry | This use-case describes the event where the manager makes an inquiry about the purchase history to the supplier. | * Manager (Primary business) * Supplier (External server) |
| Place Order | This use-case describes the event where either the manager or employee places a new order. Depends on the revise order use-case. | * Manager (Primary business) * Employee (Primary business) |
| Place Special Order | This use-case describes the event where the customer initiates a special order. Extends the place order use-case. | * Customer (External receiver) * Employee (Primary business) |
| Remove Stock | This use-case describes the event where the manager initiates this use-case via the verify stock use-case to remove stock. Depends on the verify stock use-case. | * Manager (Primary business) |
| Revise Order | This use-case describes the event where the manager or supplier revise the details and legitimacy of an order. Depends on the place order use-case. | * Manager (Primary business) * Supplier (External server) |
| Update Stock | This use-case describes the event where the manager initiates this use-case via the verify stock use-case to update the details of the stock. Depends on the verify stock use-case. | * Manager (Primary business) |
| Update User | This use-case describes the event where the manager adds or removes a user or changes a current user’s details. | * Manager (Primary business) |
| Verify Promotion | This use-case describes the event where the customer wants to make a purchase based on a promotion. Extends the make purchases use-case. | * Employee (Primary business) |
| Verify Stock | This use-case describes the event where the manager wants to verify numbers and legitimacy of stock. | * Manager (Primary business) |
| Verify User | This use-case describes the event where a user that is attempting to log in, is validated for existence. | * None |

## 6.2 Use-Case Model Diagram

Use-case modelling is defined as “the process of modelling a system in terms of business events, who initiated those events and how the system responds to those events” (Bentley, 2007:245). A use-case model diagram is a graphical depiction of the interactions between external users and systems, and the system. The following diagram depicts the relationships and interactions between the actors and use-cases:

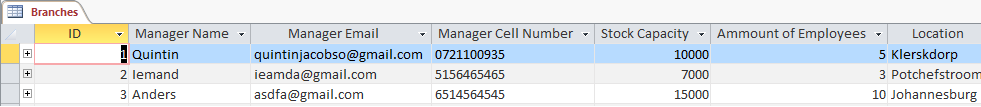


Figure : Use-Case Model Diagram

# 7. Examples

## 7.1 Fact-Finding Techniques

Fact-finding is defined as the “process of collecting information about system problems, opportunities, solution requirements, and priorities” (Bentley, 2007:165).

The Fact-finding techniques that were used for this project thus far are:

* Observations of the work environment
* Interviews
* Research and site visits
* Questions for the employee
* Joint Requirements Planning
* Prototyping

### 7.1.1Observations

Bentley (2007:218) defines observations as a “fact-finding technique wherein the systems analyst either participates in or watches a person perform activities to learn about the system”.

After observing Sean Smith using the system to make a sale, the following observations were made concerning the current system:

* After Sean showed a customer an item from the database, it was clear that the interface was “sluggish”.
* Although the current system did accomplished the main task, it is very outdated and not very user-friendly. It immediately became a concern that was in need of a rework.
* After observing one of the cashiers at Matrix Warehouse, it became obvious that the system lacked a “wow”-factor.
* One of the observations that was made, was that in order to accomplish certain tasks, an additional program had to be opened.

### 7.1.2 Interview

Defined as a “fact-finding technique whereby the systems analyst collects information from individuals through face-to-face interactions” by Bentley (2007:222).

Sean Smith was interviewed using an unstructured interview (interview conducted with few specific questions and with only a general goal in mind) and was generally asked about the system, mainly concerning the system’s effectiveness and user friendliness of the system. He replied saying that “the system works perfectly, it just lacks user friendliness and an improved graphical user interface”. Part of the main task immediately became clear.

### 7.1.3 Research

After the interview with Sean Smith, research was done about the improvement of graphical user interface (GUI) and user friendliness. It was found that a simple, to the point interface in combination with tooltips and help menus, was the best choice. This will allow system users to better understand the system. It will also reduce the amount of training required to “master” the system.

### 7.1.4 Questions

Sean Smith was asked a series of open ended questions and was willing to cooperate. Here follows a list of questions asked:

1. What is your opinion about Matrix Warehouse’s current information system?
2. What would you like to see changed about Matrix Warehouse’s current information system?
3. Are you happy with the current information system?
4. What are the other employees of this branch’s opinions about the current information system?
5. What do you like about the current information system?
6. What do you not like about the current information system?
7. If a new interface is developed, what would be the most comfortable way of operating the system?
8. If there is anything you want to see completely removed from the system, what would it be?

His answers were short and to the point. Here follow his answers:

1. “It is a well working system but feels ‘old’.”
2. “I would like to see a new layout and a better interface.”
3. “It gets the job done, and that’s what matters.”
4. “They agree with me that the system works but feels ‘ancient’.”
5. “I like the fact that it performs the task as well as any other.”
6. “I don’t like the user interface and the fact that there are no guide lines to help us when we struggle.”
7. “A system that minimizes the use of lengthy dropdown-menus, but instead uses shortcuts and better input methods, like buttons.”
8. “Definitely the use of additional programs. I would like the system to be able to complete all the tasks, without me needing to wait for another program to open.”

### 7.1.5 Joint Requirements Planning

Joint requirements planning (JRP) is defined as a “process whereby highly structured group meetings are conducted for the purpose of analyzing problems and defining requirements” (Bentley, 2007:229).

In the first stage of JRP, the roles of each person in the group and what work would be assigned to each person, was decided. Afterwards the current system’s problems and the main requirements of the system were discussed in detail. It was decided that improving the user friendliness, interface and the speed of the system was essential. These were the main problems and needed to be taken care of first. The group leader immediately started developing a GUI prototype and asked others for opinions and reviews.

### 7.1.6 Prototyping

The process of building a system’s model, is called prototyping.

After discussing the main problems at hand, work was started on prototyping a GUI so that problems faced in the future could be easily overcome during the final development phase. A database prototype was also created, with the same intentions as the GUI prototype. The following figures are of the basic database prototype:

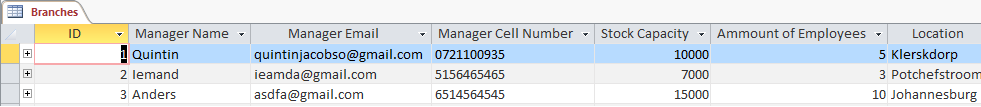
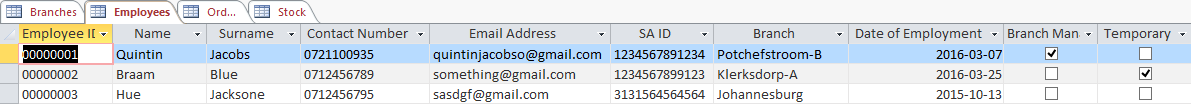


Figure : Database Prototype- Branches

Figure : Database Prototype- Employee



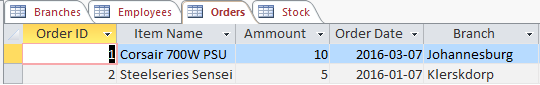
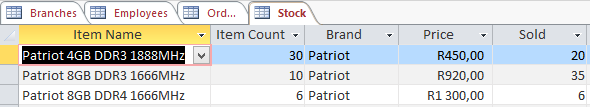


Figure : Database Prototype- Orders

Figure : Database Prototype- Stock



# 8. Closing

## 8.1 Future & Further Planning

In this section the future planning of the entire development of the system will be discussed and graphically depicted

### PERT Chart

A PERT chart is a graphical network model that depicts the project’s tasks’ interdependencies. A PERT chart’s succession of reliant tasks, which determines a project’s earliest date of completion, is known as the critical path. The following figure represents stockI.T’s PERT chart:

Figure : PERT Chart

### 8.1.2 Gantt Chart

When a bar chart is depicts project tasks against a calendar, it is known as a Gantt chart. The following sequence of figures represents the Gantt chart for the development of stockI.T:

Figure .1: Gantt Chart- 7 Feb 2016 - 20 March 2016

Figure 9: Gantt Chart- 20 March 2016 - 5 June 2016

Figure 7.2: Gantt Chart- 20 March 2016 - 5 June 2016

Figure 7.3: Gantt Chart- 5 June 2016 - 21 Aug 2016

Figure 7.4: Gantt Chart- 21 Aug 2016 - 9 Oct 2016

# 9. Summary

During the requirements analysis done for this report, has all the functional and non-functional requirements become clear. Several possible system candidates were identified and analyzed, to the conclusion that candidate 2 is the most feasible of the three, only due to the fact that candidate 3 would require more time. If time is well managed and it is determined that there is enough time, candidate 3 will become the lead candidate.

Through the feasibility analysis conducted it has been determined that the system is feasible in all aspects. As a result of the use-case model diagram, have all the relationships and interactions between the various actors and use-cases been determined and thoroughly made clear. This will allow codedCoffee to better apply this knowledge to the development of the system.

Several examples of how information has been gathered till this point in time and the related data, have been included to give a better understanding of how the problem statement and solutions have been reached. As depicted in the Gantt chart, will the solution be reachable within the giver time-frame. Thus has codedCoffee deemed the completion of stockI.T as possible.

# Bibliography

Bentley, L & Whitten, J. 2007. System Analysis and Design Methods. 7th ed. McGraw Hill.

Hoffer J.A., George J.F., Valacich J.S. 2005. Modern System Analysis and Design. 4th ed. Prentice-Hall Inc.

Rouse, M. 2007. Requirements analysis (requirements engineering). <http://searchsoftwarequality.techtarget.com/definition/requirements-analysis> Date of access: 13 April 2016.