# CHAPTER 3: REQUIREMENTS ANALYSIS

## Introduction

This chapter entails an analysis of the resources – software, hardware and others – that are needed for the proposed system to function or work properly. It also provides a brief summary of the existent system.

## Current System

Here you’re showing an understanding of the existing system before you start proposing your own or an update of what is there.

### Context Level Diagram

### Process Flow Diagram

### Use-case

## Feasibility Study

A feasibility study is an analysis of the practicality and viability of a proposed solution while emphasising on the potential hurdles that can be met. This section looks to critique the applicability of the proposed system with respect to various aspects such as profitability, technicality and whether it is economic.

### Technical Feasibility

This refers to an analysis of the technical functionality of the proposed system.

#### Hardware Requirements

The following are the hardware required for the project to run:

A development PC with these minimum requirements: 2GB installed RAM and a Core i3 Processor with processing speeds of 1.5GHz and 60GB of disk space. The table below illustrates a comparison of the minimum, recommended and available hardware requirements for the system to run.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Minimum** | **Recommended** | **Available** |
| **Installed RAM** | 2GB | 8GB | 4GB |
| **Processor** | Core i3 @ 1.5GHz | Core i7 @ 3.0GHz | Core i5 @ 2.5GHz |
| **Hard Disk** | 60 GB | 150GB | 1TB |

Technically speaking, in terms of hardware, the proposed project is feasible since the available hardware surpasses the minimum hardware required for it to run.

#### Software Requirements

The following software requirements are needed for the project to begin running:

Any Python 3 version, which is the programming language used to develop the system. Django, the Python Web Framework (and associated libraries to be installed using the pip command), Hyper-Text Mark-up Language (HTML), Cascading Style Sheets (CSS) and JavaScript (JS), the front-end technology for User Interface design, and Visual Studio Code, a text editor, for writing the code, as well as any modern browser for displaying the rendered results obtained or retrieved from the database.

|  |  |  |  |
| --- | --- | --- | --- |
| **Requirement** | **Minimum** | **Recommended** | **Available** |
| **Python** | 3.1.0 | 3.8.2 | 3.8.2 |
| **Text Editor** | Notepad++ | Visual Studio Code | Visual Studio Code |
| **Web Browser** | Microsoft Edge | Mozilla Firefox | Mozilla Firefox |

The recommendations above are given with particular concern on the RAM of the PC in use. They require less RAM to run and are therefore ideal for any PC with minimal hardware requirements.

### Economic Feasibility

This section analyses the development cost of the project with regards to the purchase of the tools that are required for the entire development process as well as other additional costs. It tries to assess the cost-benefit of the given project.

|  |  |  |  |
| --- | --- | --- | --- |
| **Item** | **Quantity** | **Cost** | **Capital Cost** |
| Samsung Laptop | 1 | Free | Researcher |
| Python Software | 1 | Free (Open Source) | Freeware(Open License) |
| Visual Studio Code | 1 | Free (Open Source) | Freeware(Open License) |

## Requirements Analysis

Requirements analysis entails the process of evaluating whether customer expectations for a proposed system are achievable. It elaborates on the services that the system has to provide while highlighting constrains that need to be accounted for during operation.

### Functional Requirements

Functional requirements describe the services that the proposed system is supposed to provide; how it should respond to user inputs as well as its expected behaviour in particular scenarios.

The following are the proposed system’s expected functional requirements:

#### Generation of hashes

* The system should be able to generate hash values or checksums using the MD5 algorithm, for every file in the specified directory.

#### Storage of hashes

* The system should be able to push the generated hash value, along the file’s associated properties, to the database.

#### Retrieval and display of hashes

* The system should be able to retrieve the stored information from the database and display it on the User Interface.

#### Comparison of hashes

* The system should perform a scan and compared the stored checksum of a file against the checksum of the file’s current state.

#### Alerting

* The system is supposed to send an alert to the user if a file’s checksum has changed.

#### User Registration and Login

* The system should allow for users to sign up for new accounts and login to already existing accounts.

### 3.3.1.1 Context Level DFD

### 3.2.1.2 DFD Level 1

### 3.2.1.3 DFD Level 2 (Optional)

### 3.2.1.4 Use-case

### 3.2.2 Non-functional Requirements

These are constraints on the services or functions offered by the system. They include timing constraints, constraints on the development process, and constraints imposed by standards. Non-functional requirements often apply to the system as a whole, rather than individual system features or services.

### 3.2.2.1 Performance

The code base of the system allows for robust performance by providing features such as minimal page loading time.

### 3.2.2.2 Usability

The system has a User-Friendly interface that is easy to use and navigate around.

### 3.2.2.3 Security

With regards to security, the system will be able to provide the following features:

Authentication

* The system will authenticate users and only authenticated users be granted access to particular features.

Encryption

It will store passwords securely using the SHA1 encryption algorithm.

### 3.3 Interface Requirements

### 3.4 Technical Requirements

### 3.5 Assumptions

### 3.6 Conclusion