**4. System Design**

**4.1. Introduction**

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This system design section of this CSE Department Academic Resource Management System (CRMS) contains a detailed description of the objects defined in the previous chapter; the software requirement section. It will attempt to define methods, properties, accessories and to a certain extent, provide algorithms or ways of approaching the coding or programming process.

**4.1.1. Purpose of the System Design**

This software design section describes the architecture and system design of CRMS showing what methods and directions to be followed in order to develop the system. It expresses the essential building blocks that are necessary to build the system. It also provides the basic implementation aspects of any software. As this system is primarily an academic resource management system; it should be equipped with some sort of structure and organization throughout the system, such that the CRUD (Create Read Update Delete) rule as well as MVC (Model View Controller) layout must be maintained. As this standardized methods are implemented, the system would be open to highly qualified functionality, refactoring, update and upgrades.

**4.1.2. Design Goals**

The design goals describe the major qualities that a system needs to achieve and address through the design process. Generally, there are a lot of design goals to be achieved, yet the following are some of the design goals that are attempted to be attained for this particular system; namely CSE department’s academic resource management system.

1. Reliability

As the system is a resource management system; those resources should be reliable as well as the system. The system should provide reliable service to the users meaning providing the proper and authentic material in a very reliable, convenient, effective and efficient manner. It should be a well trusted source and portal of resource in the department’s community.

1. Maintainability

The system should always be open to monitors and changes. If a certain feature in the system needs to be modified, that modification should address only that specific feature without affecting the overall functionality of the system which can be achieved by using the full stack development method as well as a framework approach.

The System should also be capable to add or enhance a certain functionality, it should be achieved by refactoring the existing system and bring about that change through minimal effort and inconvenience.

Adaptability is another aspect of maintainability which addresses issues like new releases through other platforms. The system should be simple enough to be adapted or outfitted into a new platform; this case is also being practiced as the CRMS web system is being adapted into an android mobile application in the near future, which also attains the portability goal as well since the app will be more reaching and easy for access.

1. Scalability

Scalability is also another major goal that needs to be attained through the design process, since the system can be scaled into more departments, schools and the entire university. So it should follow a certain set of parameters in order to assert this issue by using a more general, objective and broad classes and categories to maintain a scalable standard.

1. Robustness

As any given system, flaws and erroneous activities might occur and this must be addressed by redirecting to an error page, provide troubleshooting option and providing contact page, through all this the system should not crash or fail, it should handle such errors as it is supposed to.

1. Performance

The system should operate with in optimum response time and memory space. The system should respond to requests in a maximum duration of 10 to 20 seconds. It should also meet a certain memory and storage standard since it provides resources; that’s why as a primary storage a state of the art cloud storage (Google Drive / One Drive) is used since it operates on a higher performance rate. There is also a secondary storage site that is the Web server’s storage.

1. Low Cost

Since the system is developed by students with no proper budget in place, it should be developed and deployed within reasonable and low cost. This can be achieved by using free development methods like open source web & framework development methods and free domain name and also free web hosting services.

1. Understandability

The system should be well understood by the users and satisfy their need as much as possible. So in order to achieve that it must be presented with proper language, proper grammar, unambiguous format and user-friendly interface.

1. Rapid Development

As the system follows the rapid development model (RAD), it should be developed at an earliest convenience that is less than three months, so any tool that is deemed fit to achieve this would be utilized such as scaffolding tools, basic frameworks and simpler layouts.

1. Flexibility

The system should bring a well thought out plan to implement a certain task; for instance provide a search option, perform single and bulk operations, and provide a navigation and directive.

1. Well Documented

The system should be well documented especially through the design process since it provides the very fundamental information about the task commenced to develop the system. So it should have a proper documentation as well as a final product documentation that can serve as a user manual.

**Design Trade-offs**

Efficiency vs Portability: As the system tries a more efficient state, it would lose its portable features in the process. In order to provide an efficient experience, CRMS would lose its portable features, since it would consist a more constrained and predetermined path of operation.

Rapid Development vs Functionality: As the system is developed in a very short period of time, it would miss some functionalities and also discard some features on the way to meet the deadline.

**4.1.3 Definitions, Acronyms and Abbreviations**

Cloud Service Provider (CSP): A business entity that offers computing, storage, or software services powered by VMware cloud platforms to consumers through a private or public network.

Consumer (or customer or end user): Someone who consumes the services offered by the client or directly by the system.

Service: A means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks.

Cloud Management Platform (CMP): A suite of integrated products that provide management for public, hybrid, and private cloud environments (includes self-service interfaces, provisioning systems, metering and billing, and workload optimization).

ACS Auto-Configuration Server

SQL Structures Query Language

API Application Programmer Interface

ASF Apache Software Foundation

GUI Graphical User Interface

**4.1.4 References**

*At the end of the documentation.*

**4.1.5 Overview**

The following sections of this system design chapter would consist the software architecture- high level software architecture representation, the system design process, the subsystem decomposition, API layouts, hardware and software mapping, database setup, access control, boundary and constraint conditions as well as subsystem services.