2. Analysis

2.1 Introduction to Analysis

Analysis is the systematic and descriptive process of evaluating the complex domain and illustrating the complex domain as a simple module or sub-system by breaking them further apart. It is the first stage where all the primary requirements are identified and gathered for further analysis and problem solving. Without analysis, further improvement cannot be imagined.

Analysis phase helps in the identification and formulation of **Proof of concept (PoC)**. The identified requirements are then analyzed using a chosen analysis methodology.

2.2 Analysis Methodology

The preferred analysis methodology is **Object Oriented Analysis (OOA).** Object Oriented Analysis (OOA) is process of discovery where a development team understands and models the requirements of the system. In OOA requirements are organized as objects. It integrates all the process and data. But in others or traditional structural analysis both process and data are considered independently/separately. They use flow chart/structure charts for process and ER diagrams for data.

But in OOA some advance models are used. The common models used in OOA are: Use cases, Object models. Use cases describe pictures or overview for standard domain functions that the system must achieved. Object models describe the names, class relations, operations, and properties of the main objects. User-interface prototypes can also be created for better understanding (Cybarlab, 2017).

OOA Modellings

In OOA, there various modelling methodologies: Static Modelling, Dynamic Modelling and Functional Modeling.

* Static Modelling

It shows the relationship between the static constituents of the system. Static modelling are usually represented by Class diagram. Hence, the overall system’s structure is modelled in this modelling. In this analysis phase, initial class diagram is modelled with the identification of primary requirements.

* Dynamic / Functional Modelling

This modelling represents the behavioral aspects of the system. Dynamic and functional modelling can be generalized as a single modelling method where the control information are represented by series of events and operations that happen in the objects. Use case diagram, Sequence diagram, Activity diagram represents the dynamic model of the system.

2.3 Feasibility Study

In technical term, feasibility study is an assessment measure taken by the project owner in order to determine positive and negative outcomes, perform cost-benefit analysis and identify relevant factors that affect the project for its completion. There are various factors to consider in order to declare the project to be feasible. And they include social, operational, technical, legal, economical, etc.

|  |  |  |
| --- | --- | --- |
| **Factors** | **Summary** | **Feasibility Study** |
| Economic | Will it reduce organizational costs? | Yes |
| Will it provide economic benefits to the organization? | Yes |
| Can small organization afford the system? Is the product scalable as per the organization size? | May be |
| Social | Will there be a significant effect to the product by branding of similar other products? | May be |
| Will there be any need for further management of development time-box due to cultural and environmental factors? | May be |
| Is the proposed product sustainable in future market as per the organization trends? | Yes |
| Technological | Can rapid development of the product be achieved? | Yes |
| Can the product be compatible with emerging technologies? | Somewhat |
| Can we quickly mend the vulnerable dependency in the system? | Yes |
| Legal or Political | Can change in government regulation affect the product market value? | May be |
| Is there any chance of possible affect to the product due to change in trade regulations? | No |

2.4 Software Requirement Specification (SRS)

Software Requirement Specification is the document that describes the software system that is going to be developed with inclusion of functional and non-functional requirements along with relevant other use cases that finalizes the quality product. It is sometime called as Software or Requirement Engineering as well.

As the development methodology is preferred to be PXP (Personal Extreme Programming) which is endorsed by Agile methodology, the requirements are collected and recorded as **Product Backlog Items** where the user’s story are appropriately and concisely presented. In case of this project, the requirements are solely identified by me through the research of current academic organization situation.

Functional Requirements

These requirements are the ones that are very essential for the product to meet the behavioral requirement. It is the backbone of the proposed system. These requirements are focused with major priority.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependencies** |
| FR-01 | User Registration | Administrator should be able to add new user. | To entry new user so that different academic stakeholders can access their custom profiles. | N/A |
| FR-02 | User Login and Authentication | Registered users should be able to login after authentication is passed. | To validate and authorize the user login. | FR-01 |
| FR-03 | Organization’s details Entry | Users with granted permission should be able to add organization detail. | To add details in organization CMS | FR-02 |
| FR-04 | Organization’s detail Update | Permitted user should be able to edit existing details. | To edit existing details of organization. | FR-03 |
| FR-05 | View Organization’s details | All users should be able to view organization’s information | To display organization details. | FR-03 |
| FR-06 | Organization details Deletion | Allow granted user to delete existing organization information. | To remove obsolete organization details. | FR-03 |
| FR-07 | Student’s CMS | Granted user can add, edit, view and delete student records. | To create and manipulate student’s portfolio. | FR-02 |
| FR-08 | Academic Detail CMS | Permitted user (faculty staffs) should be able perform CRUD operation for academic details of the students. | To manage the academic information of the students. | FR-07 |
| FR-09 | Data Validation | Both client-side and server-side validation must be implemented. Form validation and entity (bean) validation must be integrated. | To maintain data integrity and correctness. | N/A |
| FR-10 | Centralized Student Information | Permitted user should be able to view every student details and transactions in the same web page. | To provide centralized summary details of every student. | FR-07, FR-08 |
| FR-11 | Access Control | Moderation of the system as per various authorization levels and permissions. | To filter the view page as per granted roles. | FR-02 |
| FR-12 | Tabular Representation | Representation of resource usage of the organization in tabular format. | To summarize the usage in tables. | FR-03 |
| FR-13 | Student’s analysis | Graphical representation of student’s analyzed information | To view the performance of the student. | FR-10 |
| FR-14 | Student Report Generation | Automated report generation of student details. | To trigger reports generation automatically. | FR-08 |
| FR-15 | Exam and Lecture notices | Automate notices in the organization. | To automate flow of notices within organization. | FR-03 |
| FR-16 | E-invoices | Automate creation and forward invoices through emails. | To automate invoice generation and delivery. | FR-07 |
| FR-17 | Multiple Sessions | Allow multiple sessions to run in runtime. | To allow multiple user use system concurrently. | FR-02 |
| FR-18 | Session Management | Invalidate stale sessions. | To nullify inactive sessions. | FR-02 |
| FR-19 | Intuitive UI | Functionality of the system should be easily achieved through convenient user interface. | To increase user’s productivity. | N/A |

Non-functional Requirements

These are the non-essential requirements which strengthen the developed system with additional features.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **Title** | **Description** | **Rational** | **Dependencies** |
| NFR-01 | Access Security | Deny deliberate and intrusive access faults | To implement access control as per authentication and authorization. | N/A |
| NFR-02 | Integrity | Assure data stored by the software is authentic, accurate and not corrupted. | To guarantee data authenticity. | N/A |
| NFR-03 | Flexibility | Easy software modification to adapt varying environments, and user needs. | To provide flexible configuration of the system. | N/A |
| NFR-04 | Portability | Easy transfer of software from current hardware or software environment to other one. | To ease software portability | N/A |
| NFR-05 | Reliability | Perform specified function consistently without failure. | To maintain consistency in the system. | N/A |
| NFR-06 | Maintainability | Easy detection and fix of a software system. | To easily maintain the software while in need. | N/A |
| NFR-07 | Scalability | Able to expand the processing and data handling capabilities. | To cope with business growth and needs. | N/A |
| NFR-08 | Availability | Active system in normal operating times. | To maintain system availability. | N/A |
| NFR-09 | Confidentiality | Protection of sensitive data from unauthorized users. | To safeguard confidential credentials. | N/A |
| NFR-10 | Usability | Convenient interaction of the user with the system. | To ease learning, operating process of the user. | N/A |
| NFR-11 | Efficiency | Effective handling of program load, throughput, response time. | To handle higher resource usage by the system. | N/A |
| NFR-12 | Interoperability | Able to facilitate and communicate the interface with other system. | To operate the system with other compatible system. | N/A |

2.4.3 MoSCoW Prioritization

|  |  |  |
| --- | --- | --- |
| **ID** | **Requirement** | **MoSCoW** |
| FR-01 | User Registration | M |
| FR-02 | User Login and Authentication | M |
| FR-03 | Organization’s detail Entry | M |
| FR-04 | Organization’s detail Update | M |
| FR-05 | View Organization’s detail | M |
| FR-06 | Organization details deletion | M |
| FR-07 | Student’s CMS | M |
| FR-08 | Academic Detail CMS | M |
| FR-09 | Data Validation | S |
| FR-10 | Centralized Student Information | M |
| FR-11 | Access Control | S |
| FR-12 | Tabular Representation | S |
| FR-13 | Student’s analysis | S |
| FR-14 | Student Report Generation | M |
| FR-15 | Exam and Lecture notices | W |
| FR-16 | E-invoices | M |
| FR-17 | Multiple Sessions | M |
| FR-18 | Session Management | M |
| FR-19 | Intuitive UI | W |
| NFR-01 | Access Security | M |
| NFR-02 | Integrity | S |
| NFR-03 | Flexibility | S |
| NFR-04 | Portability | S |
| NFR-05 | Reliability | M |
| NFR-06 | Maintainability | M |
| NFR-07 | Scalability | M |
| NFR-08 | Availability | M |
| NFR-09 | Confidentiality | M |
| NFR-10 | Usability | M |
| NFR-11 | Efficiency | M |
| NFR-12 | Interoperability | C |

2.4.4 Hardware Software Specification

Development Environment Specification

The specification of the system used while developing the product is mentioned below.

**Hardware Specification**

|  |  |
| --- | --- |
| **Item** | **Specification** |
| CPU | Intel® Core™ i5-4200U CPU @ 1.60GHz × 4 |
| RAM | 12 GB |
| Hard Disk | 500 GB |
| Screen Resolution | 1366 x 768 (16:9) |

**Software Specification**

|  |  |
| --- | --- |
| **Item** | **Specification** |
| Operating System (OS) | Ubuntu 18.04.2 LTS |
| Java | Oracle JDK 1.8.0\_201, JRE 1.8.0\_201 |
| Web browsers | Google Chrome v74.0.3729.131 (64-bit), Mozilla Quantum v66.0.3 |
| Integrated Development Environment | IntelliJ IDEA ULTIMATE 2019.1.1 |
| Database | MySQL Ver 14.14 Distrib 5.7.26, for Linux (x86\_64) |

Product Configuration/Deployment Specification

The hardware and software requirements for the product is specified below:

**Hardware Specification**

|  |  |
| --- | --- |
| **Item** | **Minimum Requirement** |
| CPU | Intel® Pentium™ 4 with a clock speed of 2 GHz or higher |
| RAM | 4 GB or higher |
| Hard Disk | 100 GB or higher |
| Screen Resolution | 1024 x 720 or higher |

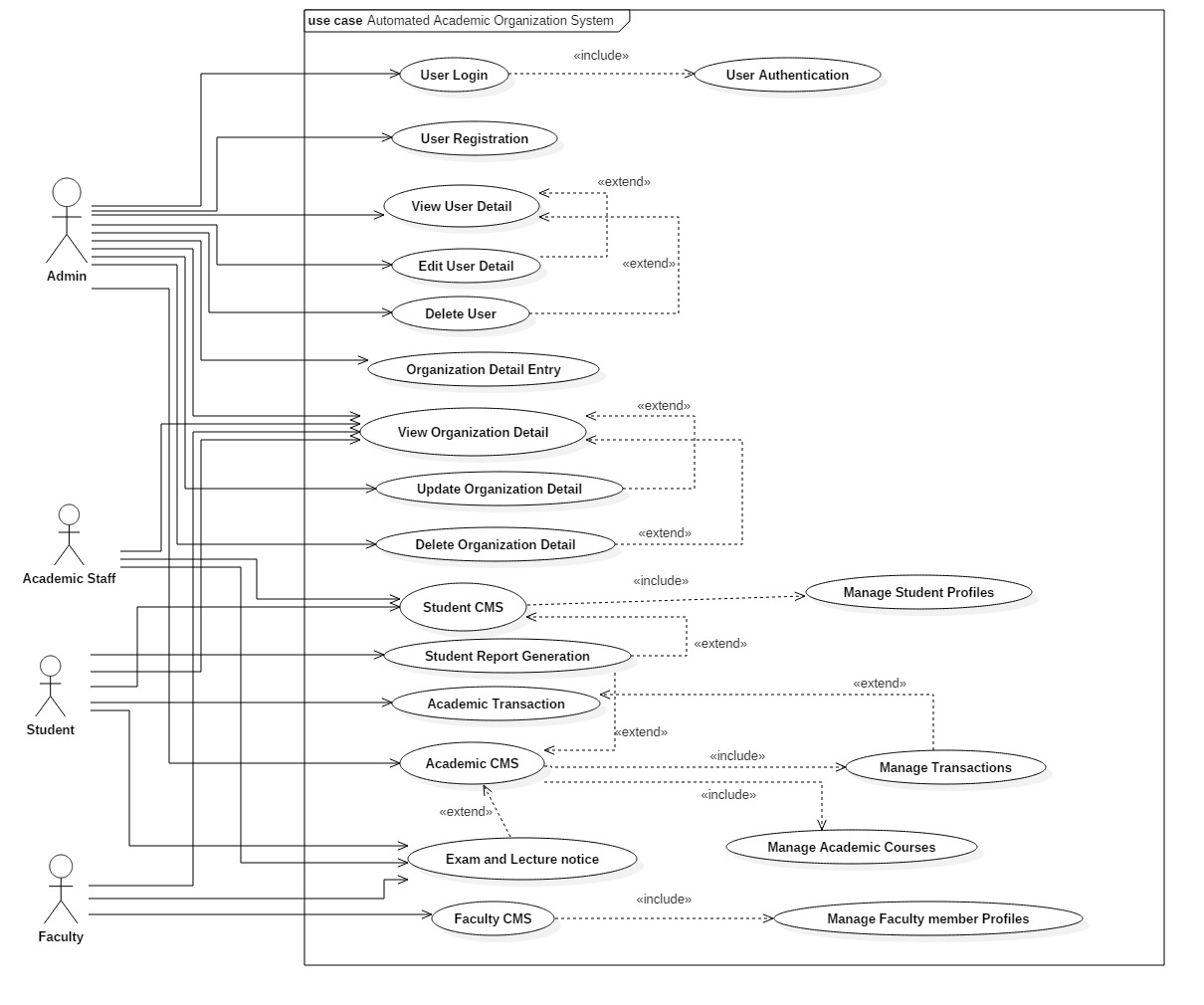
**Software Specification**

|  |  |
| --- | --- |
| **Item** | **Minimum Requirement** |
| Operating System (OS) | Microsoft® Windows® 7 or Windows® 10 or other Linux distros |
| Java | JAVA(TM) SE Runtime Environment (Oracle JRE 1.8.0\_201) |
| Web browsers | Microsoft® Internet Explorer® v11 or Google Chrome v74 (64-bit) or Mozilla Quantum v66 |
| Web Server | Apache Tomcat 9.0.2 |
| Database | mysql Ver 14.14 Distrib 5.7.26, for Linux (x86\_64) |

2.5 Use-case diagram

Use-case diagram also known as behavioral diagram presents the system’s high level requirement analysis by clearly displaying the interaction of the user (known as the actor) with different parts of the system.

The diagram below shows the use-case for Automated Academic Organization System where four different users are presented with different access in the system.



Img 1: Use-case diagram

Use-case Justification

2.6 Initial Class Diagram (NLA)