1. Introduction
   1. Project Introduction

**Automated Academic Organization System (AAOS)** is the automated information system which will be specifically featured with the collective package of content management system, automated data-driven systems and data-analysis system. This information system will be useful for every academic institutions as it provides centralized information of all the stakeholders and helps in decision making from appropriate analysis of institutions’ data.

* 1. Justification of the project

In the world of technology, every sectors including business, education, management are powered by digital technologies. As per the research, the educational sectors are still using traditional way of storing the information despite emergence of new technologies. So, to address this problem, I’ve proposed an information system which will help in centralized storing of voluminous data appropriately and analyze it to generate helpful information.

* + 1. Background of the project

The proposed information system is supposed to work for both business and educational sectors. This software is focused on providing digital features for all the stakeholders including the business stakeholders, academic staffs, pupils in the educational institutions.

* + 1. Problem Statement

The major problem in most of the educational institutions is that the data stored are not digitalized or even if they are digitalized, the ultimate storage mechanism is isolated or they are not centralized. For an example, if a student who wants to enroll in a new degree, his/her fee clearance, academic performance should be checked separately from different departments. Moreover, most of the tasks such as record generation, document publishing are repetitive tasks which can be automated.

* 1. Description of the project

Automated Academic Organization System is proposed to be an enterprise application which will provide with the generic features of the educational institution. General features can be automated and data analysis can be undertaken effectively. In addition, additional features can be added as the system will be designed in modular way.

* + 1. Features of the project
* Content Management System for organizations, students, academic details.
* Tabular representation of resource usage in the organization. Example: Room Usage.
* Centralized Student Information.
* Access control and moderation of the system as per various authorization levels.
* Graphical representation of student’s analysis, academics achievements.
* Automation of tasks such as student report generation, exam notices, lecture notices, billing invoices through emails, etc.
  1. Overview of the project

Overall system will be designed to address the traditional problems in an educational institution. The three major parts of the system are content management system, automated information system and data-analysis system.

1. Scope
   1. Scope of the project

This project has wider scope in the market as the number of educational institutions are increasing exponentially every year. The proposed system is willing to effectively manage enormous number of information system in schools, colleges, and other educational institutions.

* 1. Limitations of the project
* The template cannot be customized dynamically as per the specific organization need.
* Limit in control of raw data, as the system do not support export features of data stored in the database.
* Use of core Java web components such as JSP regardless of emergence of new front-end tech stack such as Angular frameworks for effective client-side implementation.
  1. Aims of the project
* Build a centralized and robust system that can easily aid in management of the educational sectors with appropriate data analysis.
* Assist educational associations with strong information system.
  1. Objectives of the project
* To provide user-friendly and versatile interface for data entry.
* To provide analyzed and filtered document to various departments.
* To computerize practically all the transaction in the organization.
* To provide analyzed information through graphical representation.
  1. Overview of the scope

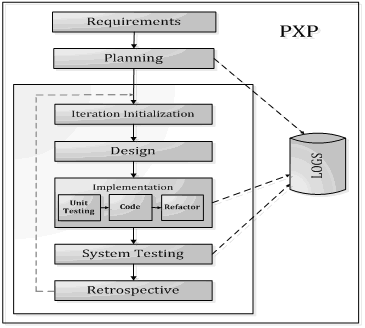
In spite of broad domain, due to major limitations the proposed product will flourish in the market slowly. However, it has huge fields for its application.

1. Development Methodology
   1. Description of the methodology

For the development of the proposed system, the preferred methodology is **PXP (Personal Extreme Programming).** The Waterfall methodology is not preferred because the application is scaled to be an enterprise level application where the software development steps should be iterative to cope with the changing nature of development methodology. However, general agile methodologies such as DSDM, Scrum are also not suitable as they relatively focus more on the team. As this project is an individual project, **PXP** is suitable methodology as it provides an agile way to design the system.

Personal Extreme Programming (PXP) is a software development process designed to be applied by software engineers individually. (ResearchGate, 2019)

PXP is based on PSP (Personal Software Process) where it keeps the basic principles of PSP and also introduces subset of XP (Extreme Programming) practices. XP is an agile software development framework. The PXP methodology is an iterative process that provides more flexible development methods. The phases of PXP is as shown in the figure below:



Img 1 - Personal Extreme Programming

The PXP practices and their integration is discussed below as: (Alpha Epsilon, 2017)

|  |  |  |
| --- | --- | --- |
| Practices or Principles | Integration in the project | Explanation |
| Decide which features will be of maximum value to the business | Yes | This practice will be effectively taken in analysis phase. |
| Aim for small and often releases | No | With time constraints, currently there is no different modules for small releases. |
| Strive for single and consistent system of names | Yes | Appropriate consistent names for required packages and other system modules. |
| Simple Design | Yes | Simplest possible design will be undertaken as it is an individual project. |
| Testing Driven Development | Yes | Testing module will be implemented. |
| Refactor in refactor branch and put into production branch after completion | No | Refactoring and release-based production is impossible for single module and due to time constraints. |
| Continuous Integration | No | Will not be implemented as the project is planned to be delivered collectively at last. |
| Pair Programming | No | It is an individual project. |
| Collective Code Ownership | Yes | Code base will be appropriately maintained as it is a solo development. |
| Stop working when you’re tired or no longer productive | Yes | Implements 40-Hour-Week rule to balance the productivity |
| On-site Customer | No | The project is just an academic project with no ties with real customers. |
| Coding Standards | Yes | Adherence to a self-imposed coding standard to have a consistent code base |

* 1. Design pattern

Singleton

This design pattern is more likely to be used in user authentication as a session can have only one instance of the user.

Factory Method

Likely to be used in the session management of the JDBC connection.

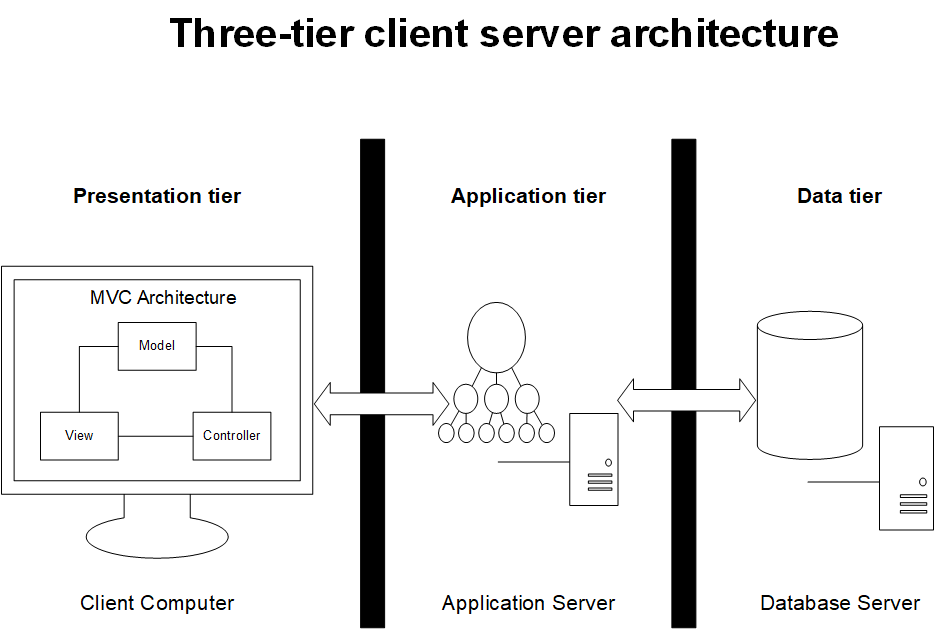
MVC

Model View Controller (MVC) design pattern provides specific software development method and also solve architectural complexities by dividing the complete system into three major divisions: Model consists of the entities that are directly connected to the database, View consists of graphical components such as web forms and Controller is where program logics are implemented.

* 1. System Architecture

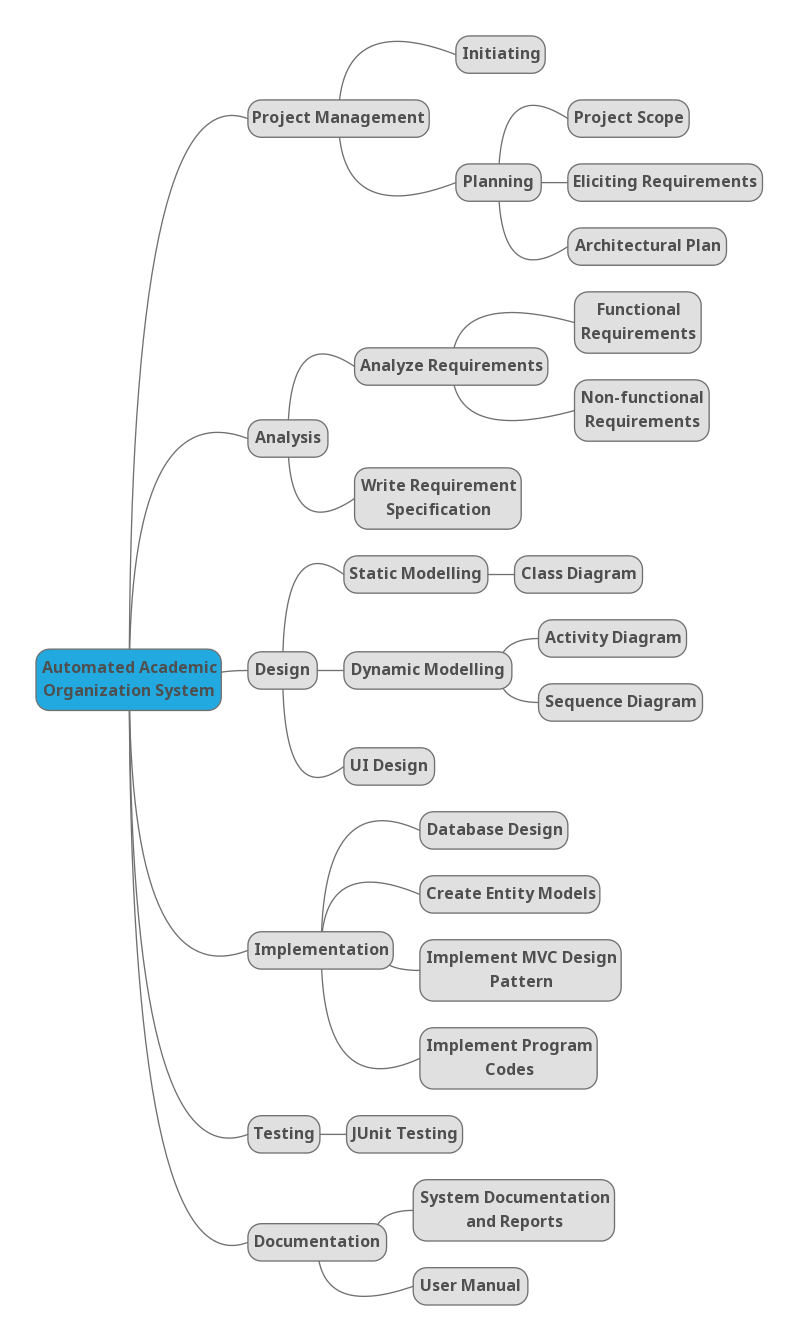
Three-tier architecture

It is a client-server software architecture pattern in which there are three major modules: Presentation tier, Logic or Application tier and Data tier. In client-server architecture, numerous clients request and receive service from the server. In three-tier architecture, the client side is provided with the user interface to allow the user to request services of the server. This layer is normally called as the **Presentation layer or tier.** The **Application tier** and the **Data tier** are generally in the server-side as the request are processed in the server with appropriate business logic and data.



Img 2 Three tier Client Server Architecture (with MVC implementation)

1. Project Planning
   1. Work Breakdown Structure



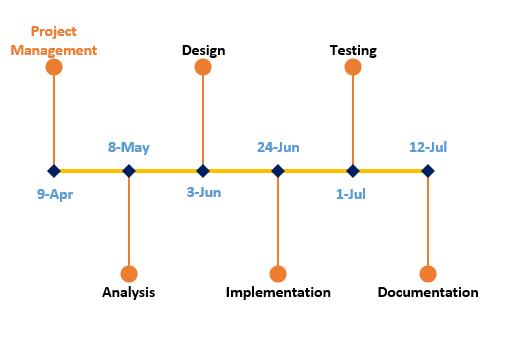
Img 3 WBS

WBS Time Estimation

|  |  |
| --- | --- |
| **Activity** | **Time Estimation (days)** |
| **Project Management** | **15** |
| Initiating | 1 |
| Planning | 14 |
| **Analysis** | **29** |
| Analyze Requirements | 18 |
| Write Requirement Specification | 11 |
| **Design** | **26** |
| Static Modelling | 5 |
| Dynamic Modelling | 8 |
| UI Design | 13 |
| **Implementation** | **21** |
| Database Design | 2 |
| Create Entity Models | 5 |
| Implement MVC Design Pattern | 3 |
| Implement Program Codes | 11 |
| **Testing** | **7** |
| JUnit Testing | 7 |
| **Documentation** | **11** |
| System Documentation and Reports | 8 |
| User Manual | 3 |
| **Total** | **109** |

Work Breakdown Structure (WBS) represents the project’s activity at different structure. It generally denotes the detail information of the works performed in various level. Generic tasks are shown in higher level whereas specific tasks are shown in the lower level of particular generic task. The time estimation of each task is also presented and are subject to change.

* 1. Milestone



Img 4 Project Milestones

There are six major milestones in this project. And they are discussed briefly below:

* Project Management

The first milestone of this project is the finalization of the planning processes, which includes **project scope, eliciting requirements, and architectural plan**.

* Analysis

The next major milestone comprises of detail analysis which includes **functional** and **non-functional requirements analysis**, and then produce **requirement specification**.

* Design

This milestone includes various activities related to system modelling. It comprises of various success criteria such as **static modelling, dynamic modelling,** and **UI design**.

* Implementation

This milestone ensures that the implementation of the proposed system is carried out properly. It includes **database design, creation of entity models, implementation of design patterns,** and **program codes**.

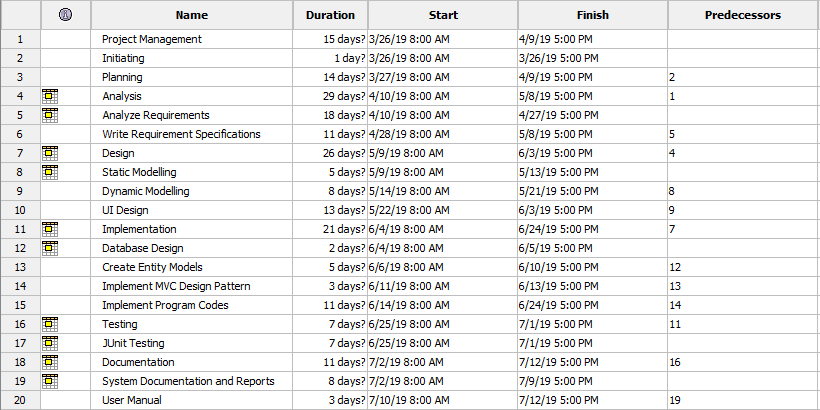
* Testing

This is a vital milestone which ensures that the developed system works and integrates properly. This milestone specifically implements **Junit Testing**.

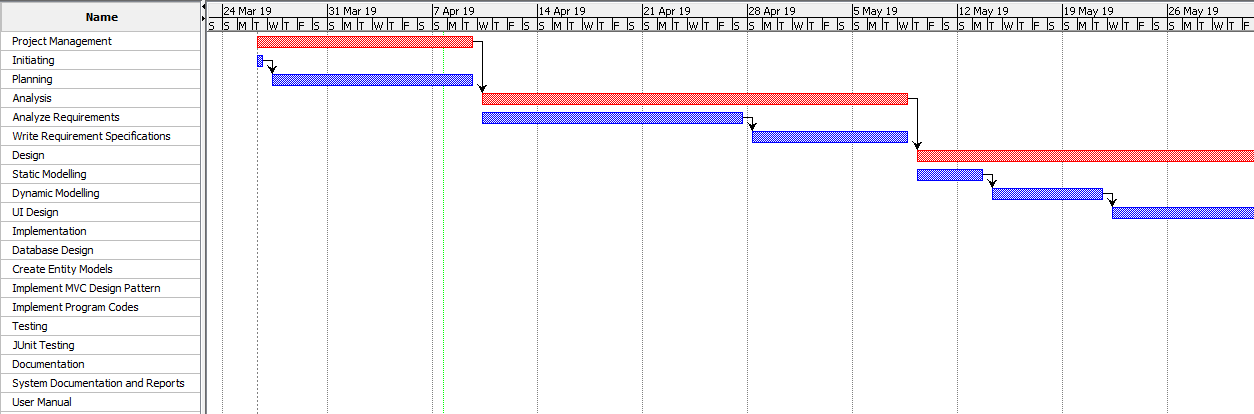
* Documentation

This is the final milestone and it certifies the completion of the project with the appropriate **documentation** and **user manuals.**

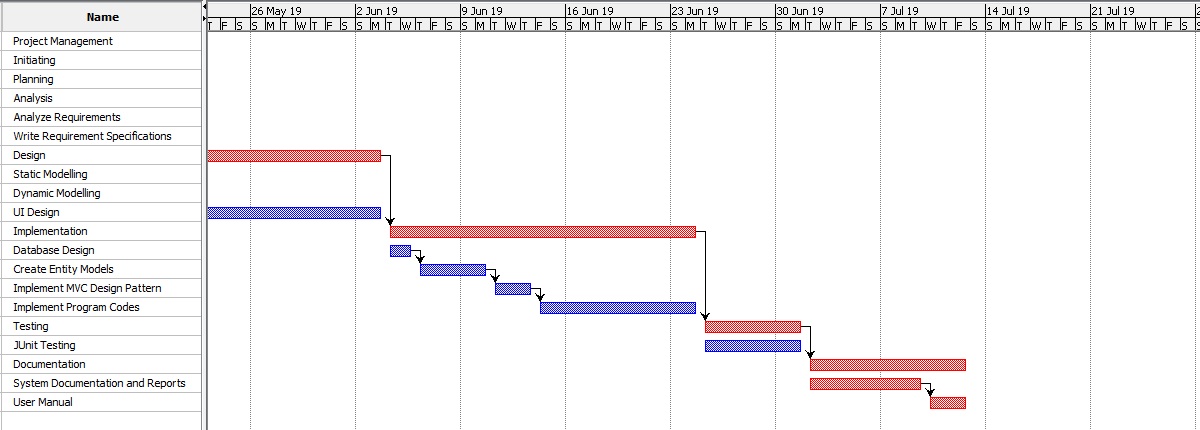
* 1. Gantt Chart



Img 5 Gantt Chart 1



Img 6 Gantt Chart 2



Img 7 Gantt Chart 3

1. Risk Management

Project risk management is the process of identifying, analyzing and then responding to any risk that arises over the life cycle of a project to help the project remain on track and meet its goal. Risk management isn’t reactive only; it should be part of the planning process to figure out risk that might happen in the project and how to control that risk if it in fact occurs. (PROJECT MANAGER, 2017)

The lifecycle of the risk management process is as shown in the figure below: (INTLAND SOFTWARE, 2014)



Img 8 Risk Management Lifecycle

For the risk management of this project, the steps are generic compared to the lifecycle mentioned above. The major steps include:

1. Identification of risks
2. Risk and Vulnerability assessment
3. Identification of risk reduction measures
4. Prioritization of risk reduction measures

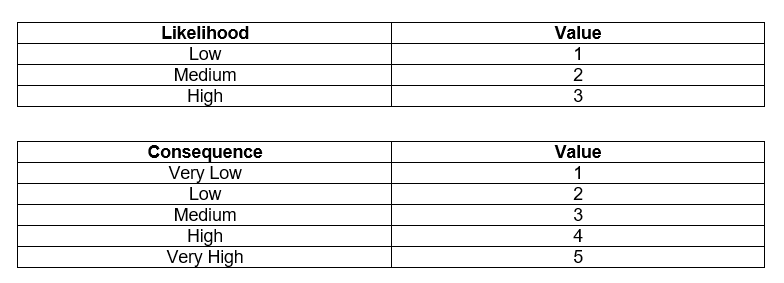
The risk assessment is performed on the basis of following factors:

* Consequence
* Likelihood
* Risk

The relation between these factors is given by,

Risk (Impact) = Consequence x Likelihood

Hence, with the reference of following table, the risk analysis is carried out as:

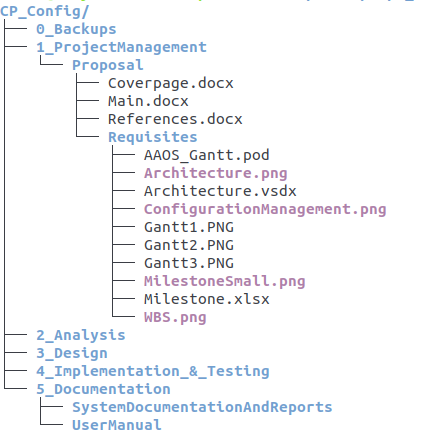


Img 9 Risk Analysis Lookup Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Risk** | **Likelihood** | **Consequence** | **Impact** | **Remarks** |
| Addition of extra features that were not included in project scope | 1 | 5 | 5 | Strictly adopt requirement specifications. |
| Deadline Overruns | 2 | 4 | 8 | Strictly follow development methodology to ensure feature delivery with respect to time constraint. |
| Hardware failure | 2 | 4 | 8 | Apply continuous integration methodology to repetitively push new commits to remote repository |
| Natural calamities | 1 | 5 | 5 | Cloud Backup |
| Vulnerabilities in project dependencies | 2 | 4 | 8 | Should use scalable dependencies with no major milestone. |
| Difficulty in implementing complex functionalities | 3 | 4 | 12 | Get advice and help from web community and experts. |
| Growth of Project Scope | 2 | 5 | 10 | Strictly implement development methodology. |
| Inflexibility in design making change request impossible | 2 | 4 | 8 | Possibly generate dynamic templates. |

1. Configuration Management

The complete project directory structure is configured as per various stages of Software Development Lifecycle as shown in the figure below:



Img 10 Project Directory Structure

The complete project resides in the given directory structure and also pushed in remote repository hosted in GitHub which can be accessed from <https://github.com/elwyncrestha/AAOS>. The root folder of remote repository provides the implementation codes along with the directory named **CP\_Config** where given project is stored as per the directory structure mentioned above.

1. Conclusion

An information system has been proposed in this document which will help academic organization to manage their activities effectively with the proper storage and analysis of their data. Moreover, this system also provides paper-less academic system as it is featured with automated task executer. The proposal is initiated with the problem domain, enlisting all the scopes and coverage of the proposed information system. From that the problem statement was identified. With a standard problem statement, scopes were listed along with the limitations. In addition, the development methodology was chosen unlike the traditional approach to cope with the complex requirements. With the help of WBS, time estimation and milestones were determined. Moreover, the possible risks were identified and their mitigation measures were also identified through risk management. Ultimately, the project directory structure was discussed briefly in configuration management.