**Abstract**

**University Admission System** is a system which is used for applying admission to a University for a particular course. This system is a crucial and useful system for the students who are to apply for admission for certain university. This system helps students to apply for multiple courses, pay the admission fees, view the admission history, appear for the admission test, and print the receipt of fees.

This system is helpful for the Admission department as well, as it helps them to insert **Available** Admission courses, remove courses, print report for the admission, conduct test, and get the result of the test. This system

In this report we will discuss about the project and how it was developed throughout the given time period.

**Table of Content**

1. Introduction
   1. Existing System
   2. Problem Definition
   3. Proposed System
2. Software Development Methodology
3. Software Project Management
   1. Objective
   2. Scope
   3. Cost Estimation
4. Software Risk Management
   1. Strategy of Risk
   2. Risk Identification
5. Software Quality Management
   1. Concept
   2. Quality
   3. Quality Control
   4. Quality Checking Activity
   5. Cost of Quality
6. Software Configuration Management
7. Conclusion
8. References

**Introduction**

**Existing System:** The existing system allows you to apply for admission only once without login, which makes the database bulky. The existing system does not have a user profile, thus user cannot login to see the details of admission. The existing system is not “mobile-friendly”, meaning, it is not responsive. The existing system allows administrator to print reports. The flow of the current system is that, the user visits the website, apply for the course and pay the appropriate fees to that.

**Problem Statement:** The current system has many problems such as; it does not have an integrated quiz for admission. It is not mobile friendly. It does not have admin panel, authorities have to ask system administrator to make changes. It does not have a good user interface.

As per our preliminary research, the current system for the admission process for the university is open for all, due to which the University cannot keep a track of students who have visited the admission page and did not applied for admission. The new system targets user in such a manner that, the student (user) will have to first register an account and based on which the student (user) will be able to apply for admission in different courses.

The second problem in the system is that, there is lack of authentication for each module; anyone can directly access any module, which risky as per the University point of view. So in order to avoid this problem, the secondary target will be to add authentication as security.

Another problem in the current system is that it is not ‘mobile-friendly’ aka ‘responsive’, so the aim will be to re-design the current system such that it can be accessible from any device.

**Proposed System:** We as a team have proposed a system which overcomes most of the problems in the existing system. By definition, University Admission system will provide facility to various kinds of users of this system respectively. For students, this system will help in such a manner that, a student will be able to view course details of the respective course, which will include, course duration, course syllabus, university details, along with these facilities, a student will be able to register for certain courses.

After registration, a student will be able to apply for his/her choice of course. A student then also can have the option to pay the one-time registration fees online.

For, Administrative department, this system is helpful for tracking admission for their respective department and according to that they will be able to generate reports.

For, Accounts department, this system is helpful for tracking the payment status of the students who have registered for the courses.

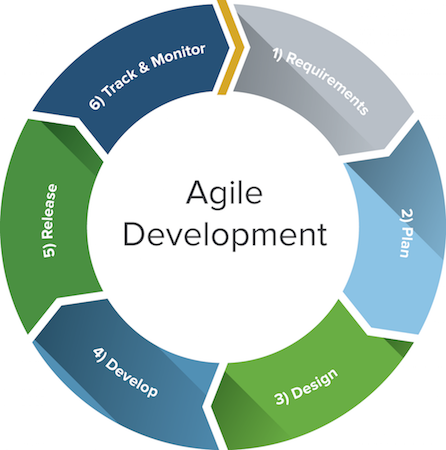
Moreover, this system will have a Super Admin who will be able to manage all the respective admin, meaning Super Admin will be able manipulate the roles of admins such as Add Admin and Delete Admin

**Software Development Methodology**

We have developed this project by following the **AGILE** methodology.

As software development is a **continuous** process, AGILE methodology is the best option to select among all Software Development Technology.

Below is how AGILE methodology works: -



**Software Project Management**

**Objective:**

The Objective of this project was to set goals and achieve those goals in the given time period.

We were in continuous communication with the client and were getting constant constructive feedback.

As a part of continuous development we used to Develop - > Test -> Debug -> Deploy and repeat the same process.

We did proper planning and followed that plan throughout the project.

**Scope**

The scope of project was to **develop** and **deploy** an Admission system for the University, about which the details are mentioned in **Introduction** part.

The purpose of the registration process is to determine which students will be taking courses within the University, and for the administration to keep its records up-to-date. From the student’s point of view, registration enables them to acquire the necessary authorized membership of the University, It is usually the case that students will register for particular courses, or modules, at the same time, and the information collected is used by members of the teaching staff to construct class lists.

For the ease we divided the project in smaller parts i.e. **Design – Develop (includes Debug) – Test.**

**Cost Estimation**

Cost Estimation is considered as the most crucial of all, as it depends more than one element.

We ensure our end product which is deliverable is in reasonable size, so that it does not faces performance issues.

The said software has passed all possible manual testing, it is hack proof as it is using Object Oriented methodology and is using latest hack proof methods such as PDO and MySQLi.

This software is fully open-source, i.e. anyone can use it/ distribute/ and contribute.

The said software has MIT License.

Any additional software used to develop this project had not expense as we have used all open source libraries and software.

Miscellaneous expenses such as Travel, Skill, and Training and support etc. are not part of this project.

We are using **CoCoMo** (Constructive Cost Model) as a Cost Estimation model.

**Software Risk Management**

In this section we will discuss possible risks which occur or may occur or have occurred during the development of this software. By definition, software risk means, a possibility of suffering from loss in software development process is called a software risk.

Generally risk is handled by the Project Manager and the Team, client or end-user has nothing to do with Risk which occurs during development. The known risks which have occurred are as bellows

**Compatibility Issues:** Software which is developed in certain said environment i.e. Windows, Linux or Mac OS is limited to that environment only. We cannot run that software on other platforms. This risk can be managed by making that as platform independent.

**Time Constraint:** Time is a critical factor in a project, which is limited during the project. We need to deliver the product in the given time. To solve this constraint we divided the work, so that each and every one of us gets equal work and no one gets burdened with work.

**Budget:** Budget gets decided during the planning phase, and cannot get altered or changed during the project. We cannot get over-budget in the middle of the project. To avoid this issue we did project planning properly and made a project charter.

Above are the “**known**” issues which we faced during the project and resolved for the same.

**Strategy of Risk:** In a small project like this, the risks are low and limited but risks still occur. As discussed above the risks which occurred, are of “**dual nature**”, meaning it affects both the developer (team) and the client.