

PROJECT PHASE 1

CONCEPTUAL DATABASE DESIGN



PROJECT TITLE

DORMITORY ALLOCATION SYSTEM

GROUP MEMBERS

1. Ababu Alemu	ATR/6421/10
2. Abel Alem	ATR/9260/10
3. Betemariam Assaminew	ATR/8821/10
4. Maedin Seid	ATR/1692/10
5. Natnael Fitsum	ATR/1627/10

SUBMITTED TO_TESFAMICHAEL G.
DEADLINE _April 17, 2021

Table of Contents

1. Introduction

- 1.1 Purpose
- 1.2 Objective
- 1.3 Scope of the Project
- 1.4 Overview of Project
- 1.5 About Our System

2. Requirements

- 2.1 Requirements Analysis
- 2.2 Hardware Requirement
- 2.3 Software Requirement

3. Data Design

- 3.1 Entity Relationship Diagram
- 3.2 Conceptual Schema
- 3.3 Conceptual Schema in UML Diagram

4. Entities and Attributes

- 4.1 Dorm Building
- 4.2 Proctor Staff
- 4.3 Student
- 4.4 Dorm
- 4.5 Locker
- 4.6 Bed

1. Introduction

This documentation, namely the SDD doc, provides instructions/aid on how to build the software and on how implement the database of the. It contains ER diagrams, conceptual schema, and other necessary requirements.

1.1 Purpose

The Software Design Document's goal is to provide a concise description of a system's design so that software development can start with a clear understanding of what needs to be installed and how it should be designed. The Software Design Document includes all of the details needed to provide a comprehensive description of the software and system that will be created.

1.2 Objective

- To deal with Dormitory Management System in an easy and an efficient manner.
- Create a database that is secure, fast and stable.

1.3 Scope of the Project

- Dormitory Management System is mainly designed for Universities.
- There will be certain criteria for the assignment of dorm to a student.
- The proctor checks the attested application forms of the students obtained from the internet and verify it with the student database.
- If the students are found eligible then they are allotted to the a dorm Room.

1.4 Overview of Project

The Dormitory allocation system is a web based system that aims to solve the current inefficient way of allocating dorms to students in universities. The current way of doing things involves filling out forms and submitting those forms which is very backwards and time consuming.

1.5 About OUR System

Although our university is one of the best in the country, the dorm management system is still a manual system. This system is an inefficient way of allocating dorms to students in universities. That's why computerized system is the best option. The computerized system would:

1. Be friendlier to students and the staff (proctor).
2. Improved service satisfaction.
3. Increases the dorm management efficiency.
4. Reduce the operational costs of the management.

1.5.1 THE ADVANTAGES OF THE SYSTYEM.

- The system enables easy and fast access to the student files.
- The system provides better data management facilities.
- The system provides performance evaluation of the employees/proctors to ensure maximum output from the employees.
- The system provides security measures to access to student's information lowering data security threats.
- Easy update of the student's records.
- Reduction of data entry and processing errors. Greatly reduce paper use at the dorm.

2. Requirement

2.1 Requirement Analysis

Requirements Analysis is the process of determining what the database is to be used for. It involves asking questions and having interview with students, university management staff and the proctors to identify what functionality they require from the database, what kinds of data they wish to process and the most frequently performed operations. This discussion is at a non-technical level and enables the database designers to understand the business logic behind the desired database. This way the database designers can understand what they are working with and work there way into solving there problem.

We can also analysis by doing some research online and finding the basic problems in this kind of university and finding a way in designing a better database sketch for solving this problems.

In working with requirement analysis we need to answer the following question those are?

- 1.What user want from the database?
- 2.What is going to be stored in the database?
- 3.What application are going to be built on top the database?
- 4.What are the most frequently asked queries?
5. Evaluate existing systems?
6. Consider security ?
7. Determine needed new forms /reports /queries?

1. What user want from the database?

Our user wants the data of the students which are to be allocated to the dorm there information will be stored in the database and our first user which is the proctor will use this specific data to allocate them to there dorm and also there will also be the data of the proctors to be controlled by the higher management staffs so they will also use this database to fetch the data off the proctors.

As a student one might require to know his/her building block placement together with the dorm number. And also may require to know his/her dorm mates. And the proctor may store the number of students in the building block under his/her management, know where a specific student is placed and also keep track of all the data under his/her authority. All these data input and output will be handled by the database efficiently.

2. What is going to be stored in the database?

- Our database will store the data of the student which are name, age, student Id, mobile number, department, year of study this will be included in the database
- Our database will store the data of the proctor which are name ,Id, mobile number, dorm building this will be included in the database
- Our database will store the data of the dorm which are dorm numbers and building number this will be included in the database
- Our database will store the data of the locker in every room its attribute are locker number , dorm numbers and building number this will be included in the database
- Our database will store the data of the bed in every room its attribute are bed number , dorm numbers and building number this will be included in the database

3.What application are going to be built on top the database?

-We will built a website which will facilitate the allocation of dormitory and the management of students and proctors. So the application to be build in this database is a website build in APN.NET with languages of HTML5 and CSS3

4. What are the most frequently asked queries?

-We will be using different queries some of them are Create, Alter, Drop, Truncate in this specific

Create - queries are used to create a new database or table.

Alter - queries are used to modify the structure of a database or a table such as adding a new column, change the data type, drop, or rename an existing column, etc.

Drop - queries are used to delete a database or table. You should also be careful when using this type of query because it will remove everything, including table definition along with all the data, indexes, triggers, constraints and permission specifications for that table.

Truncate - queries are used to clean the table, remove all the existing records, but not the table itself.

The most used queries in our database is Create we will be accepting new students they will need to create a new table.

5. Evaluate existing systems?

-We tried to search if there is any existing system for dormitory allocation but we couldn't find any system this was our motivation to do this system.

6. Considering security issue?

-Database security refers to the range of tools, control, and measures designed to establish and preserve database confidentiality, integrity and availability.

-Database security measures include authentication the process of verifying if a users credentials match those stored in your database and permitting only authenticated users access to your data, networks and database platform in our project we will be using this authentication process in our project to make sure our system is secure and we have to deal with every request making sure they are a legit students and we will also make sure the data of student the system is encrypted.

7. Determine needed new forms /reports /queries?

-In our project we will be using different forms and reports to add tables to our queries some of them are snapshot query and continuous query.

-Every piece of data a **query**, **form**, or **report** uses is stored in one of your database tables. **Forms** allow you to both add data to tables and view data that already exists. **Reports** present data from tables and also from **queries**, which then search for and analyze data within these same tables. A search for a book

2.2 HARDWARE REQUIREMENTS

This section states the hardware requirement of the project in this section we have listed the minimum requirement needed to complete this project.

Processor	:	Pentium III or higher
RAM	:	128 MB or More
Hard Disk	:	20 GB or More

This are the basic minimum requirement needed for this project the hardware can be concluded this way the next is software requirement.

2.3 SOFTWARE REQUIREMENTS

This section also states the software requirement of the project in this section we have listed the minimum requirement needed to complete this project.

Operating System: : Windows XP, VISTA, 7

Server : MySQL 2010

Front End : ASP.NET

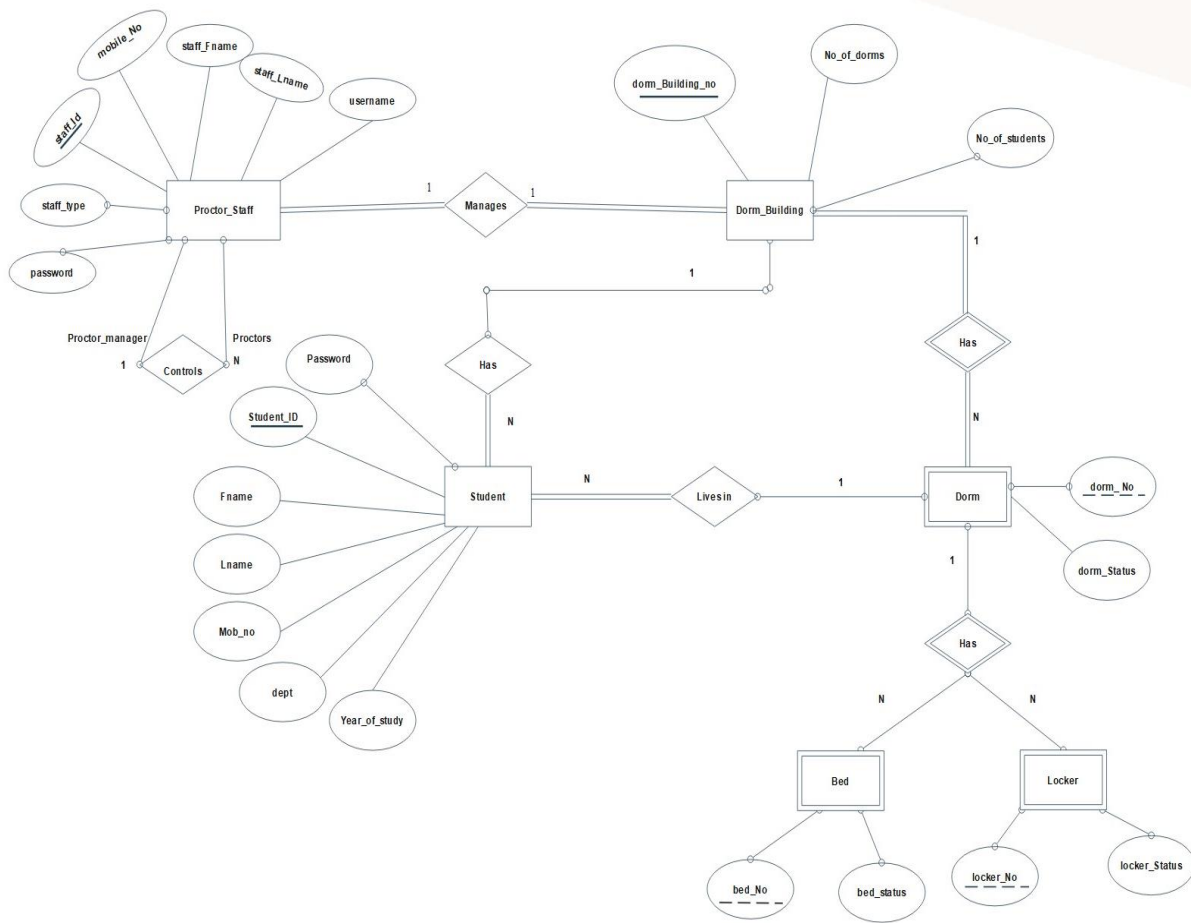
Scripts : JavaScript

Language : HTML5&CSS3

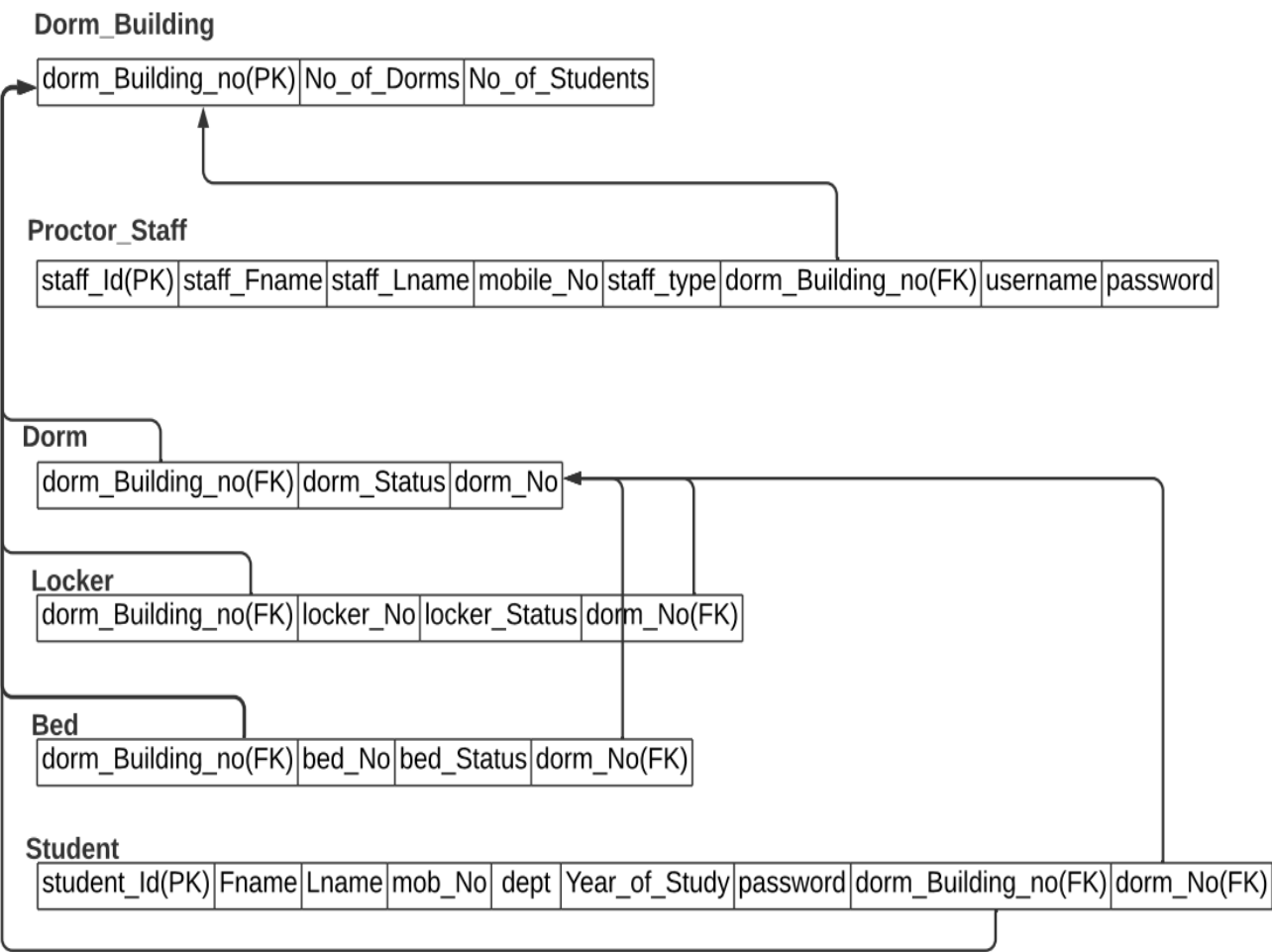
This are the basic minimum software requirement needed for this project to be concluded. We have listed what front end we will be using also the languages we will be using and we described we will be using MYSQL for the web server.

3. Data Design

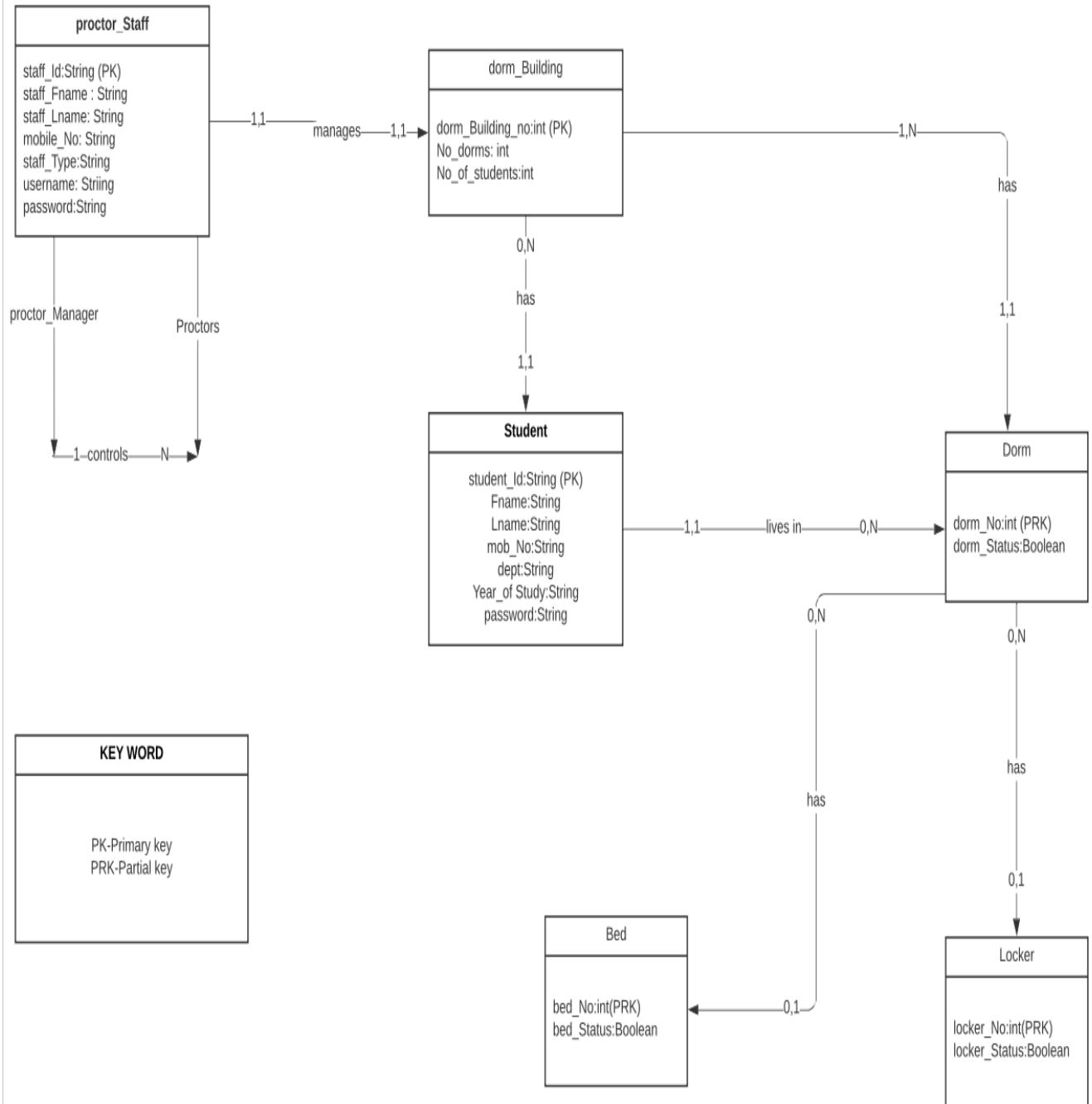
3.1 Entity Relationship Diagram



3.2 Conceptual Schema



3.3 Conceptual Schema in UML Diagram



4. Entities, Attributes and Relation

This segment of the document is intended to make the design more easy and understandable for everyone. It explains document explains the entities used in the project, their attributes and how they will work together.

Entities

1. Dorm Building
2. Proctor Staff
3. Student
4. Dorm
5. Locker
6. Bed

4.1 Dorm Building (Block)

A University may have many dorm buildings and each building is represented using this 'Dorm Building' entity. It takes part in the following relationships.

1. Proctor Staff manages dorm building.
2. Dorm building has Students.
3. Dorm building has Dorms.

Attributes

Name	Data Type	Type
Dorm_Building_no	Integer	Primary Key attribute
No_of_Dorms	Integer	Non Prime attribute
No_of_Students	Integer	Non Prime attribute

4.2 Proctor Staff

Every dorm building has an administrator and is represented using the 'Proctor' entity. Proctor entity takes part in following relationships.

1. **Proctor Staff** manages dorm building .

Attributes

Name	Data Type	Type
Staff_Id	String	Primary Key attribute
Staff_Fname	String	Non prime attribute
Staff_Lname	String	Non prime attribute
Mobile_No	String	Prime attribute
Staff_Type	String	Non prime attribute
username	String	Prime attribute
password	String	Prime attribute
dorm_Building_no	String	Foreign key attribute

4.3 Student

Every dorm building has students and which are represented by the 'student' entity. Student entity takes part in the following relationships.

1. Dorm building has **Students**.
2. **Students** lives in Dorm.

Attributes

Name	Data Type	Type
student_Id	String	Primary Key attribute
Fname	String	Non Prime attribute
Lname	String	Non Prime attribute
Mob_No	String	Prime attribute
dept	String	Non Prime attribute
Year_of_Study	String	Non Prime attribute
Password	String	Non prime attribute
dorm_Building_no	Integer	Foreign Key attribute
dorm_No	Integer	Foreign Key attribute

4.4 Dorm

Every dorm building has dorms and they are represented using 'dorm' entity. Dorm entity participates in the following relationships.

1. Dorm building has **Dorms**.
2. Student stays at **Dorms**.
3. **Dorms** has bed.
4. **Dorms** has locker.

Attributes

Name	Data Type	Type
dorm_Building_no	Integer	Foreign Key attribute
dorm_Status	Boolean	Non Prime attribute
dorm_No	Integer	Prime attribute

4.5 Locker

Every room has locker and they are represented using 'locker' entity. Locker participated in following relationships.

1. Dorm has **Locker**.

Attributes

Name	Data Type	Type
dorm_building	Integer	Foreign Key attribute
locker_No	Integer	Prime attribute
locker_Status	Boolean	Non Prime attribute
dorm_No	Integer	Foreign key attribute

4.6 Bed

Every dorm has bed and they are represented using 'bed' entity. Bed participated in following relationships.

1. Dorm has **Bed**.

Attributes

Name	Data Type	Type
dorm_Building_no	Integer	Foreign Key attribute
bed_No	Integer	Prime Attribute
bed_Status	Boolean	Non Prime attributes
dorm_No	Integer	Foreign Key attribute