



**ÇANKAYA UNIVERSITY
FACULTY OF ENGINEERING
COMPUTER ENGINEERING DEPARTMENT**

Project Report

Version 1

CENG 407

Innovative System Design and Development I

Mobile School Management System

**Oğuz GÜL 201411025
Furkan TÜRKMEN 201511064
Burak Emrullah GÜNEŞ 201611026**

Advisor: *PHD Murat SARAN*

Table Of Contents

Abstract	2
1. Introduction	3
2. Literature Search	3
2.1 Introduction	3
2.2 Background	4
2.3 Similar Application	5
2.4 Conclusion	5
3. Software Requirements Specification	5
3.1 Introduction	6
3.2 Overall Description	7
3.3 Specific Requirements	14
4. Software Design Description	28
4.1 Introduction	28
4.2 Design Considerations	30
4.3 Architecture	31
4.4 Process Design	32
4.5 Database Design	40

Abstract

In today's world the usage rate of mobile devices are constantly increasing in all kinds of educational areas. This report expresses the need for a mobile application in such areas of education. This application will be named Mobile SChool Management System and its aim is to create a bridge between Educational Institutions and Student-Parent-Teacher-Administrative Employee. MSMS will be a bridge for students and parents to communicate more closely and easily with their schools. All kinds of users will be able to get the content depending on their user type on the system. For example announcements, news, grades, attendance, syllabus of students etc.

Keywords: Educational Institutions, Mobile, Student , Parent, Teacher, Educational Institution Manager, System Administrator, Sms, Email, Notification

Özet

Özellikle pandemi nedeniyle tüm dünyada eğitim ve öğretim faaliyetlerinde mobil cihazların (notebook, tablet, cep telefonları) kullanım oranları artmaktadır. Mobil Okul Yönetimi Projesi Eğitim kurumlarının veliler, öğrenciler, öğretmenler ve idari çalışanlar ile haberleşmesini kolaylaştırmak amacıyla tasarlanacaktır. Hali hazırda kullanılan sms ve/veya e posta türü iletişim servisleri güncelliklerini yitirmiş durumda ve birçok kullanım ve teknolojik sorunlar içermektedirler. Hazırlanacak “Mobil Okul Yönetimi” Projesi Okul-Veli-Öğrenci paydaşlarını içeren kapsamlı bir okul uygulamasıdır. Mobil Okul Yönetimi sayesinde okullar ile öğrenci ve veliler çok daha yakın bir iletişimde olabileceklerdir. Proje sayesinde Eğitim Kurumlarının veliler, öğrenciler, çalışanları ile mobil odaklı bir yazılım üzerinden basit ve hızlı bir şekilde iletişim kurmaları sağlanacaktır.

Anahtar Kelimeler: Eğitim Kurumları, Mobil, Öğrenci, Veli, Öğretmen, Eğitim Kurumu Müdürü, Sistem Yöneticisi, Sms, E-Posta, Bildirim

1.INTRODUCTION

This project has two different platforms on the Web and Mobile side. Website available for all users so all users can process on the application. Another side is mobile applications which all users will be able to activate so you will be able to call functions on the mobile side. This application thanks to synchronization at the same time of users and then users can communicate with each other at the same time. People in particular want to get information instantly, and they want to respond to that information at the same time. Therefore, this project will work in a synchronized way in terms of handling and reporting school information.

2.Literature Research

Abstract

Today, especially due to the pandemic, the usage rates of mobile devices are increasing in all kinds of education for example such as High School, University, Primary School and training activities all over the world. Mobile School Management System aims to create a bridge between Educational Institutions and Student-Parent-Teacher-Administrative Employee.

Currently used Sms and / or Email type communication services between Educational Institutions and End Users are outdated and typically have many technological problems. Through Mobile School Management Application students and parents will be able to communicate much more closely and easily and any time they want with their schools. The user will be able to get content depending on their user type. For example for the Parent user type there will be some content only available to that user type. Including announcements, news, check attendance of all students, syllabus of all students etc.

2.1)Introduction

Today, mobile apps and websites have been developed to meet the needs of many people. With the Interactive Mobile School Management system, our goal is to ensure that schools and students and parents communicate much more closely.

Through this application and system, we aim to enable educational institutions to communicate with parents, students and employees in a simple and fast way through a mobile-oriented software. In addition, the parent will be able to track the student's course attendance, activity attendance and grade follow-up, as well as accounting work. Activity and announcement tracking, grade information system, mutual surveys and intensive evaluation system will be provided by the student and teacher. In addition, a web system will be made where administrative staff and school administrators can manage the mobile application. In this way, instant communication will be provided for the school parent trio of students.

Our project's web system is C# and .We'll use the .NETCore native language. In the mobile part, we will use the Flutter language to be used on all platforms.

2.2)BACKGROUND

2.2.1) Mobile School Management System

When researching online school system applications, the EDUPAGE system appears as a similar one. Although the mobile application part of the system that we will develop will be similar to the web-based part of our system, we can observe that it covers a lot of background software work.

Thanks to this system, it communicates interactively with students, parents and teachers, in short with the school. Our system also provides both mobile and web parts with API service providers and other databases simultaneously, using more efficiently to communicate with the school and students, parents and teachers simultaneously.

2.2.2) Mobile Application Development

When developing Mobile applications, languages such as Flutter, Kotlin, Swift, and React Native are mainly used. We will use the Flutter language in the mobile application that we will develop. Because this language will be the most ideal language option for our project, both because it provides a workable environment in two platforms (IOS & Android) and because it contains framework tools on this platform, since there will be a web part in our project too.

2.2.3) Flutter

Flutter is a mobile application SDK that includes framework, widgets and other tools, providing developers with application enhancements for the Android and iOS platform. Developed by Google, Flutter is open source and available for free to all developers and designers.

Using Flutter, developers can save the cost and time of developing and delivering cross-platform mobile applications that target both platforms. Interface designers, on the other hand, can develop more successful application designs that adapt to the world of Android and IOS using Flutter. So Flutter is a native application structure that we just mentioned and is slightly newer than other languages supported by Google.[1]

2.2.4) Visual Code

Visual Studio Code is an IDE developed by Microsoft and is used on platforms such as Windows, Linux and macOS. Because we will use .NET in our project, using vs code when developing the project will be the IDE that we will most benefit from when developing the project.

2.3) Similar Applications

2.3.1) EduPage

EduPage is a school management system integrated with the world's most known planning software. EduPage offers many features such as entering curriculum, tracking attendance, room booking, assigning homework. Integrations with other school processes are very important as the program changes daily.

2.4) Conclusion

There are not many applications that handle school management like our project. In today's world the schools usually send information via SMS or E-mail services. These services are generally One-way. This means they are usually about an announcement and don't require any response from the receiver side. This makes the sender side wondering if the receiver even received that message or if they agree about what the message presents. Our goal is to create the bridge between the school's management and the end users just like teacher, student, parent. Users will be able to get every information they need from this mobile application. For example the parents will be able to check their child's attendance, exam notes, check news about the school or the class, get announcements, solve surveys about the school or the teacher, check their child's syllabus etc.

3. Software Requirements Specification

PREFACE

This document contains the Software Requirements Specification (SRS) of Mobile School Management System (MSMS), which aims at a project that will bring Parents, Students, Teachers and Institution, System Administrator users together in a common system. MSMS SRS is prepared according to IEEE STD 830-1998, IEEE Recommended Practice for Software Requirements Specifications. This document involves all tools, functions, user characteristics and system requirements of the product.

3.1 INTRODUCTION

The subsections of this section will provide an overview of the entire Software Requirements Specification (SRS) document. In this section, it contains a document documentation that includes the application stages with MSMS requirements.

3.1.1 PURPOSE

This document consists of technical explanations required for all software. The document content is prepared according to IEEE standards and includes descriptions of all functions about the Mobile School Management System(MSMS), about requirements, and about users.

3.1.2 SCOPE

Mobile School Management System (MSMS) The first is a Web - based system, and the second is a mobile application-based system. In the content of this system, there are three main user actors; the student, parent and teacher, as well as school management and school computing management are these users. In this way, it is to collect all school content under one roof and prepare interactive system content.

The aim of the project is;

- All kinds of communication , activities and homework , etc.in a virtual environment between the student, parent and teacher triangle, using situations effectively one-on-one.
- Publish all the content of the school administration and provide information about the school and its activities to other users.
- School computing management is to provide services to users about announcements,events, notifications and system requirements for the effective operation of this system.

3.1.3 GLOSSARY

Abbreviation	Meaning
MSMS	Mobile School Management System
IEEE	Institute of Electrical and Electronics Engineers
SRS	Software Requirements Specifications

3.2 OVERALL DESCRIPTION

In this section, the needs and basics of the system are explained in detail. In this section, it consists of three main sections, and each section contains subcategory sections within itself.

3.2.1 PRODUCT INTERFACES

MSMS is a totally self-contained system. Also it is not dependent on any hardware heavy system.

3.2.1.1 SYSTEM INTERFACES

MSMS is only connected to the E-School system which provides grades and information about the students that use MSMS.

3.2.1.2 USER INTERFACE

In the user interface, there are Web and Mobile-Based user types for these, users simply need to use devices that contain simple computer hardware and basic mobile hardware. Since there are three types of main users, they can see their own homework, announcements and notes when entering the system as a student. When logged into the system as a parent, the user can get information about the student, about the school and about the announcements and activities that the teacher has put in the system, etc., and also create an appointment system with the school administration and teacher through the application.

3.2.1.3 HARDWARE INTERFACES

The requirement on the Web side includes the server/client side requirement for hardware.

Server side

This system is expected to run on 64 bit based devices and on a 30-40 gb hard disk system.

Client side

This system is available to all personal computers, all operating systems such as linux, windows etc.

Mobile side

The requirement hardware for the mobile base is that minimum contains a requirement to run an application. For this purpose, it contains up-to-date software and contains at least 4 GB memory.

3.2.1.4 SOFTWARE INTERFACE

This system is on the web side ASP.NET because it will be developed with all https requests, it will be able to work over all computers through the TCP protocol, the system can return to the user over the web.

Since this system will also be mobile-based, MSMS is an application that can run on both platforms using flutter software.

3.2.2 PRODUCT FUNCTIONS

In this section, the contents of the functions that users can do in the system are explained in detail.

3.2.2.1 Student Functions

The user he/she can see exam results, see the curriculum, see announcements, see news, get notifications, reservations, see and answer surveys, and give feedback.

Student_Login Function

Student_ExamResult Function

Student_Curriculum Function

Student_Announcement Function

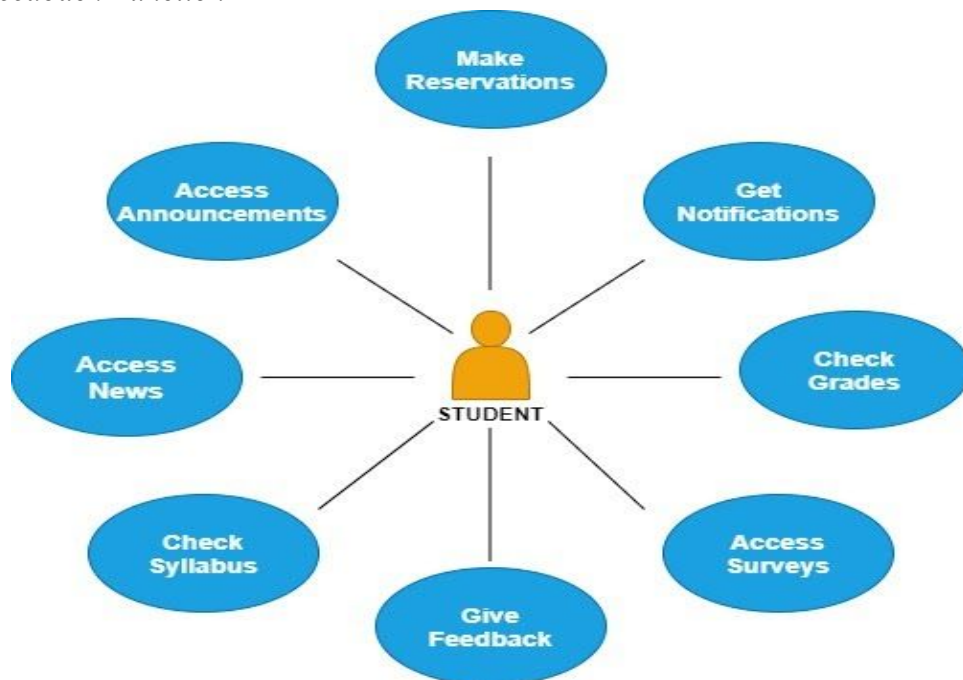
Student_News Function

Student_Notification Function

Student_Reservation Function

Student_Survey Function

Student_Feedback Function



3.2.2.2 Teacher Functions

After logging the system, the teacher performs certain functions of his / her own thanks to the system. These functions respectively;

Teacher_Login Function

Teacher_Reservation Function

Teacher_Settings Function

Teacher_Feedback Function

Teacher_Survey Function

Teacher_Syllabus Function

Teacher_Announcements Function

Teacher_News Function



3.2.2.3 Parent Functions

After the parent logging the system, it performs certain functions of its own thanks to the system. These functions respectively;

Parent_Login Function

Parent_StudentData Function

Parent_Feedback Function

Parent_Reservation Function

Parent_Survey Function

Parent_Setting Function

Parent_AccountingOperations Function

Parent_Announcements Function

Parent_News Function



3.2.2.4 School Management Functions

After logging into the system, the administrative employee performs its own specific functions thanks to the system. These functions respectively;

SchoolManager_Login Function

SchoolManager_Announcements Function

SchoolManager_Notification Function

SchoolManager_News Function

SchoolManager_ViewReservation Function

SchoolManager_ViewSurvey Function

SchoolManager_Settings Function

SchoolManager_Feedback Function



3.2.2.5 School IT Management Functions

After the School IT Management logging the system, it performs certain functions of its own thanks to the system. These functions respectively;

SchoolITMan_Login Function

SchoolITMan_DataTransfer Function

SchoolITMan_SchoolAdd Function

SchoolITMan_ClassAdd Function

SchoolITMan_SectionAdd Function

SchoolITMan_GrupAdd Function

SchoolITMan_UpdateEducationLogo Function

SchoolITMan_Notification Function

SchoolITMan_Reservation Function

SchoolITMan_AbsenceNotice Function

SchoolITMan_ExamResult Function

SchoolITMan_Syllabus Function

SchoolITMan_AccountingOperations Function

SchoolITMan_Announcements Function

SchoolITMan_Report Function

SchoolITMan_News Function

SchoolITMan_Management Function

SchoolITMan_Settings Function



3.2.3 USER CHARACTERISTIC

All users must have at least Internet browsing skills to use the system, as well as skills that can be used for smartphones and tablets.

3.3 SPECIFIC REQUIREMENTS

3.3.1 External Interfaces

Name of item: The SQL file, which includes user's information

Description of Purpose: This file's purpose is to contain information

Source of Input: Structured Query Language(SQL) File

Valid Range, Accuracy, and/or Tolerance: The accuracy and integrity of the file is provided by SQL. Any incorrect data is not tolerable. In such a case, the school IT management should be able to fix such errors.

Timing: As long as the user's keep using the system.

3.3.2 Functions

3.3.2.1 Students Functions

3.3.2.1.1 Student_LoginFunction

***Introduction:** Students should be able to login to the system.*

***Input:** student_id, student_pass.*

***Process:** The student logs in to the system. If the student enters wrong login credentials there will be an error message displaying.*

***Output:** error message to the user will be displayed.*

3.3.2.1.2 Student_ExamResult Function

***Introduction:** Students should be able to get an exam result .*

***Input:** None.*

***Process:** The student can show all exam results.*

***Output:** List of all exam results.*

3.3.2.1.3 Student_Curriculum Function

***Introduction:** Students should be able to get a curriculum .*

***Input:** None.*

***Process:** The student can show all the curriculum.*

***Output:** List of all curriculum.*

3.3.2.1.4 Student_Announcement Function

***Introduction:** Students should be able to get an announcement .*

***Input:** None.*

Process: *The student can show all the announcements.*

Output: *List of all announcements.*

3.3.2.1.4 Student_News Function

Introduction: *Students should be able to get news .*

Input: *None.*

Process: *The student can show all the news.*

Output: *List of all news.*

3.3.2.1.5 Student_Notification Function

Introduction: *Students should be able to get notification .*

Input: *None.*

Process: *The student can show all the notifications.*

Output: *List of all notifications.*

3.3.2.1.5 Student_Reservation Function

Introduction: *Students should be able to get reservations .*

Input: *None.*

Process: *The student can show all the reservations.*

Output: *List of all reservations.*

3.3.2.1.6 Student_Survey Function

Introduction: *Students can participate and do surveys within the system.*

Input: *None.*

Process: *The student can show all the surveys.*

Output: List of all surveys.

3.3.2.1.7 Student_Feedback Function

Introduction: Students can receive and give feedback within the system.

Input: None.

Process: The student can show all the feedback.

Output: List of all feedback.

3.3.2.2 Teacher Functions

3.3.2.2.1 Teacher_Login Function

Introduction: Teachers should be able to login to the system.

Input: teacher_id, teacher_pass.

Process: The teacher logs in to the system. If the teacher enters wrong login credentials there will be an error message displaying.

Output: error message to the user will be displayed.

3.3.2.2.2 Teacher_Reservation Function

Introduction: Teachers should be able to get reservations .

Input: None.

Process: The teacher can show all the reservations.

Output: List of all reservations.

3.3.2.2.3 Teacher_Settings Function

Introduction: Teachers should be able to get settings .

Input: None.

Process: The teacher can show all the setting systems.

Output: List of all setting systems.

3.3.2.2.4 Teacher_Feedback Function

Introduction: Teachers can receive and give feedback within the system.

Input: None.

Process: The teacher can show all the feedback.

Output: List of all feedback.

3.3.2.2.5 Teacher_Survey Function

Introduction: Teachers can participate and do surveys within the system.

Input: None.

Process: The teacher can show all the surveys.

Output: List of all surveys.

3.3.2.2.6 Teacher_Syllabus Function

Introduction: Teachers should be able to get a syllabus .

Input: None.

Process: The teacher can show all the syllabus.

Output: List of all syllabus.

3.3.2.2.7 Teacher_Announcements Function

Introduction: Teachers should be able to get an announcement .

Input: None.

Process: The teacher can show all the announcements.

Output: List of all announcements.

3.3.2.2.8 Teacher_News Function

Introduction: Teachers should be able to get news .

Input: None.

Process: The teacher can show all the news.

Output: List of all news.

3.3.2.3 Parent Functions

3.3.2.3.1 Parent_Login Function

Introduction: Parents should be able to login to the system.

Input: parent_id, parent_pass.

Process: The parent logs in to the system. If the parent enters wrong login credentials there will be an error message displaying.

Output: error message to the user will be displayed.

3.3.2.3.2 Parent_StudentData Function

Introduction: Parents can access and view all students' data.

Input: None.

Process: The parents can show all the students' data.

Output: List of all students' data.

3.3.2.3.3 Parent_Feedback Function

Introduction: Parents can receive and give feedback within the system.

Input: None.

Process: The parent can show all the feedback.

Output: List of all feedback.

3.3.2.3.4 Parent_Reservation Function

Introduction: Parents should be able to get reservations .

Input: None.

Process: The parent can show all the reservations.

Output: List of all reservations.

3.3.2.3.5 Parent_Survey Function

Introduction: Parents can participate and do surveys within the system.

Input: None.

Process: The parent can show all the surveys.

Output: List of all surveys.

3.3.2.3.6 Parent_Setting Function

Introduction: Parents should be able to get settings .

Input: None.

Process: The parents can show all the setting systems.

Output: List of all setting systems.

3.3.2.3.7 Parent_AccountingOperations Function

Introduction: Parents can realize accounting operations related to educational institutions.

Input: None.

Process: The parent can see all the accounting transactions.

Output: all accounting operations are listed in the system.

3.3.2.3.8 Parent_Announcements Function

Introduction: Parents should be able to get an announcement .

Input: None.

Process: The parent can show all the announcements.

Output: List of all announcements.

3.3.2.3.9 Parent_News Function

Introduction: Parents should be able to get news .

Input: None.

Process: The parents can show all the news.

Output: List of all news.

3.3.2.4 School Management Functions

3.3.2.4.1 SchoolManager_Login Function

Introduction: School managers should be able to login to the system.

Input: SchoolManager_id, SchoolManager_pass.

Process: The school manager logs in to the system. If the school manager enters wrong login credentials there will be an error message displaying.

Output: error message to the user will be displayed.

3.3.2.4.2 SchoolManager_Announcements Function

Introduction: School managers should be able to get an announcement .

Input: None.

Process: The school managers can show all the announcements.

Output: List of all announcements.

3.3.2.4.3 SchoolManager_Notification Function

Introduction: School managers should be able to get notification .

Input: None.

Process: The school manager can show all the notifications.

Output: List of all notifications.

3.3.2.4.3 SchoolManager_News Function

Introduction: School managers should be able to get news .

Input: None.

Process: The school manager can show all the news.

Output: List of all news.

3.3.2.4.4 SchoolManager_ViewReservation Function

Introduction: The school managers can see the reservations.

Input: None.

Process: The school manager can show all the reservations.

Output: List of all reservations.

3.3.2.4.5 SchoolManager_ViewSurvey Function

Introduction: The school managers can see the survey.

Input: None.

Process: The school manager can show all the surveys.

Output: List of all surveys.

3.3.2.4.6 SchoolManager_Settings Function

Introduction: School managers should be able to get settings .

Input: None.

Process: The school managers can show all the setting systems.

Output: List of all setting systems.

3.3.2.4.7 SchoolManager_Feedback Function

Introduction: School managers can receive and give feedback within the system.

Input: None.

Process: The school managers can show all the feedback.

Output: List of all feedback.

3.3.2.5 School IT Management Functions

3.3.2.5.1 SchoolITMan_Login Function

Introduction: School IT managers should be able to login to the system.

Input: SchoolITManager_id, SchoolITManager_pass.

Process: The school IT manager logs in to the system. If the school IT manager enters wrong login credentials there will be an error message displaying.

Output: error message to the user will be displayed.

3. 3.2.5.2 SchoolITMan_DataTransfer Function

Introduction: school IT manager performs data transfer between student, parent and teacher.

Input: None

Process: school IT manager can observe data transfers that it performs.

Output: school IT manager can list all data transfers it performs.

3. 3.2.5.3 SchoolITMan_SchoolAdd Function

Introduction: school IT manager can add school.

Input: None

Process: school IT manager can see the schools it adds.

Output: school IT manager can list all the schools it adds.

3.3.2.5.3 SchoolITMan_ClassAdd Function

Introduction: school IT manager can add class.

Input: None

Process: school IT manager can see the class it adds.

Output: school IT manager can list all the classes it adds.

3. 3.2.5.4 SchoolITMan_SectionAdd Function

Introduction: school IT managers can add sections.

Input: None

Process: school IT manager can see the section it adds.

Output: school IT manager can list all the sections it adds.

3.3.2.5.5 SchoolITMan_GroupAdd Function

Introduction: school IT managers can add groups.

Input: None

Process: school IT manager can see the section of the group.

Output: school IT manager can list all the group it adds.

3.3.2.5.6 SchoolITMan_UpdateEducationLogo Function

Introduction: school IT managers can update the logo and information of educational institutions.

Input: None

Process: school IT manager can see the information it updates.

Output: school IT manager can list all the information it has updated.

3.3.2.5.7 SchoolITMan_Notification Function

Introduction: School IT managers should be able to get notification .

Input: None.

Process: The school IT managers can show all the notifications.

Output: List of all notifications.

3.3.2.5.8 SchoolITMan_Reservation Function

Introduction: school IT managers manage reservations.

Input: None.

Process: The school IT managers can show all the reservations.

Output: List of all reservations.

3.3.2.5.9 SchoolITMan_AbsenceNotice Function

Introduction: school IT managers manage absence notices.

Input: None.

Process: The school IT managers can show all the absence notices.

Output: List of all absence notices.

3.3.2.5.10 SchoolITMan_ExamResult Function

Introduction: school IT managers manage exam results.

Input: None.

Process: The school IT managers can show all the exam results.

Output: List of all exam results.

3.3.2.5.11 SchoolITMan_Syllabus Function

Introduction: school IT manager can perform syllabus operations.

Input: None.

Process: The school IT managers can show all the syllabus.

Output: List of all exam syllabus.

3.3.2.5.12 SchoolITMan_AccountingOperations Function

Introduction: school IT manager can perform accounting management operations.

Input: None.

Process: The school IT manager can see all the accounting transactions.

Output: all accounting operations are listed in the system.

3.3.2.5.13 SchoolITMan_Announcements Function

Introduction: school IT manager can perform announcement management operations.

Input: None.

Process: The school IT manager can see all the announcements.

Output: all announcement operations are listed in the system.

3.3.2.5.14 SchoolITMan_Report Function

Introduction: The school IT manager can do reporting.

Input: None.

Process: The school IT manager can see all the reports.

Output: all reports operations are listed in the system.

3.3.2.5.15 SchoolITMan_News Function

Introduction: The school IT manager can perform news operations.

Input: None.

Process: The school IT manager can see all the news.

Output: all news operations are listed in the system.

3.3.2.5.16 SchoolITMan_Management Function

Introduction: The school IT manager can create other types of administrators.

Input: None.

Process: The school IT manager can see all the other types of administrators.

Output: all other types of administrators operations are listed in the system.

3.3.2.5.17 SchoolITMan_Settings Function

Introduction: School IT managers should be able to get settings .

Input: None.

Process: The school IT managers can show all the setting systems.

Output: List of all setting systems.

3.3.3 PERFORMANCE REQUIREMENTS

This system is created for unlimited school capacity. There is no cap in the amount of schools or users in this system, but in the first phase there will be 1 school to try out the system. Users should get results within 5-10 seconds in the final form of the system.

3.3.4 SECURITY REQUIREMENTS

This system is only available to some schools and their inside users. Users information can be viewed only by the users that have authority to do so. Each user has different privileges in the system for security measures.

4. Software Design Description

4.1 INTRODUCTION

In this documentation file, it is a documentation content that describes the parts of the project that cover the design and architectural goals of the project. Mobile School Management System's design and database includes operation diagrams.

4.1.1 Purpose

This document has been created to indicate details in the design part of Mobile School Management System Mobile School Management System (MSMS). At the same time, this document was created according to the IEEE 1016-2009. Those expected to use this system are the Mobile School Management System (MSMS)'s students, teachers, school management, school IT management and system management but school IT management and especially the system administrator will perform operations on the back, not the front of the application.

4.1.2 Scope

This SDD document contains general definitions and project features as content. It covers the architectural designs within the project and the content and design parts of these designs. These designs consist of process design, use case diagram, sequence diagram and database design. For those who study this content, project formation becomes more understandable and easy.

4.1.3 Glossary

TERM	DEFINITION
SDD	Software Design Description
MSMS	Mobile School Management System
IEEE	Institute of Electrical and Electronics Engineers
UML	Unified Modeling Language
MSSQL	Microsoft SQL Server
DB	Database

4.1.4 Overview

This SDD is divided into five sections to provide target readers with a complete and comprehensible perception of the system. The first part is mostly about the scope and purpose of the document. The second part specifies design topics and consists of two parts. In the first section, design assumptions, dependencies, and constraints of the systems are defined. The second section provides design goals and guidelines for the reliability, availability, portability and extensibility of the system. In addition, the tools used for project design are explained. In the third section, the architectural design of the application and the detailed description of the modules are detailed. In the fourth section, this section covers what users can do and which users have access to what. Interaction diagrams detailing how operations are performed for each user type are available here. The sequence of interaction is shown visually using the vertical axis in the sequence diagram. In the fifth section, there is detailed information about the database design of the project.

4.4.2 DESIGN CONSIDERATIONS

4.2.1 Approach

While developing the project, we have decided to use MSSQL for creating and maintaining the database in future operations. It provides the data to be classified to where it belongs. It has a structure that is flexible for maintaining and inserting or deleting new data to database structure.

4.2.2 Tools Used

Microsoft Visual Studio 2019	Programing,testing,developing website
Microsoft SQL Server	For Database Management
Microsoft Office Word	Documentations
https://app.diagrams.net/	Diagrams, UML diagrams, Sequence Diagram

4.2.3 Constraints

4.2.3.1 Time

Ceng 407 project we have exactly a period to finish the documentation. To advance in this project, we will adopt the waterfall model and advance the project. As this method progresses step by step, we will make every feedback we receive without going to the previous step, and thus we will complete the general design templates of our project.

4. 2.3.2 Performance

Performance is an important criterion for each software product. Since the MSMS project will be run by local customers in the web browser and is a multi-user process, users internet speeds and hardware-wise, RAM and CPU quality are factors that affect the performance.

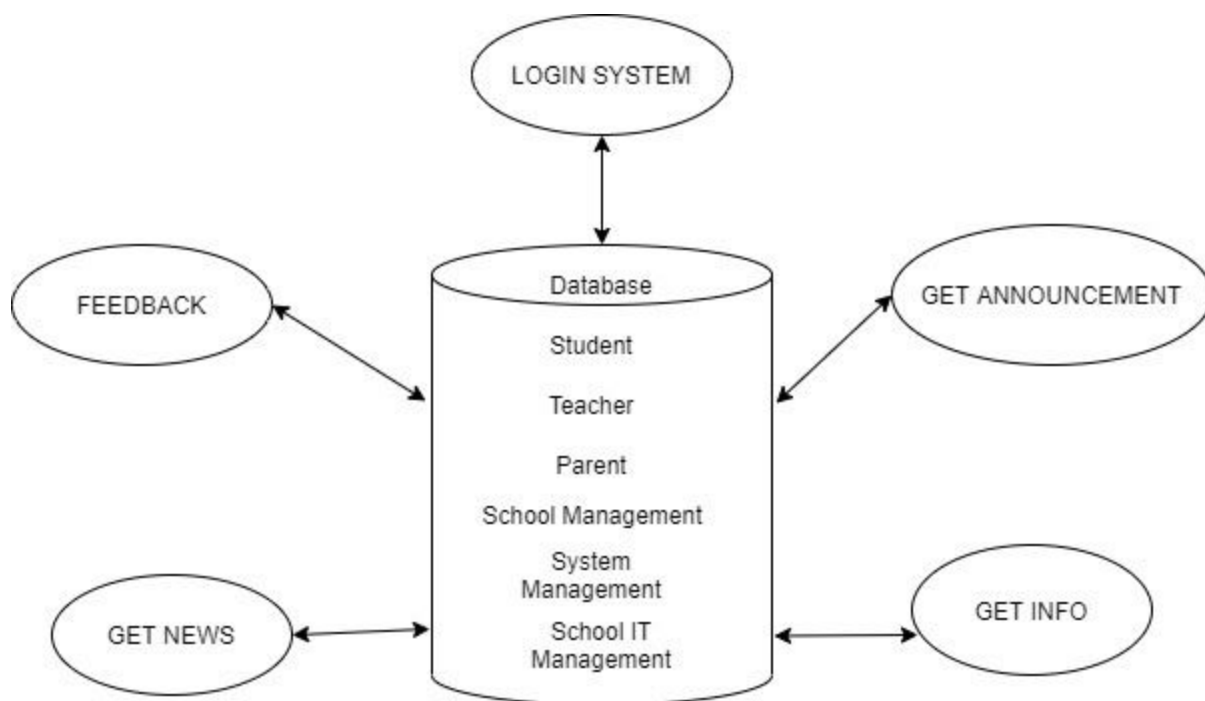
4.2.4 Assumptions and Dependencies

In designing this project, we had to make some basic assumptions about software and hardware. Our program is designed to run on all operating systems. So it is enough to have any computer and internet connection to use a web browser

4.4.3 ARCHITECTURE

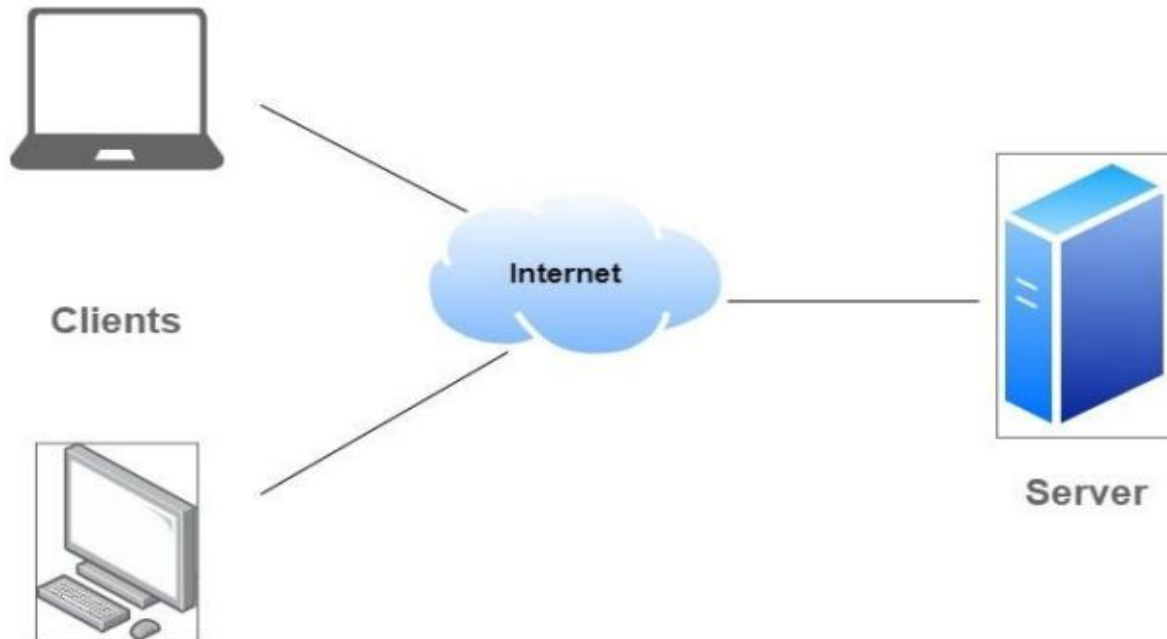
4.3.1 Software Architecture

The database will be used to contain the data of students, parents, teachers, school management, system management, school IT management, and will help to add new data or make changes to the data. More importantly our database will contribute to the functions that are defined for each user type to be used. Database is the cornerstone of our project that keeps all the necessary information in our system. We might call our architecture type the Repository Architecture Style.



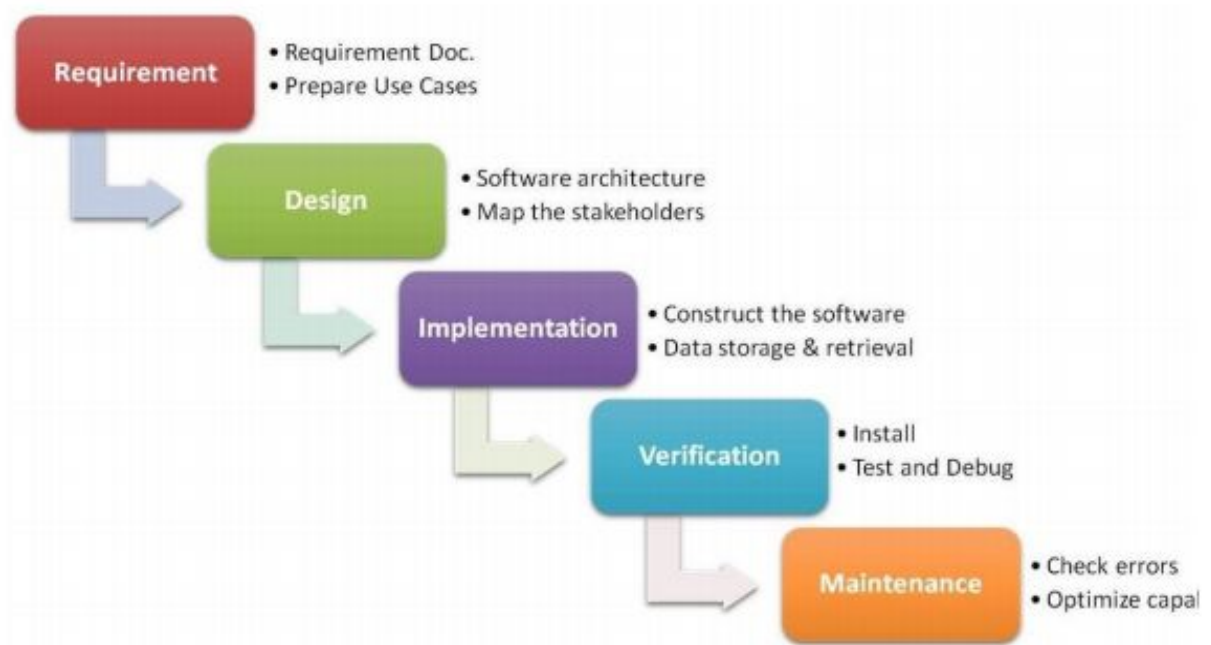
4.3.2 Hardware Architecture

We use the Client-Server model for our project. In the client-server architecture, when the client computer sends a request for data to the server through the internet, the server accepts the requested process and delivers the data packets requested back to the client.

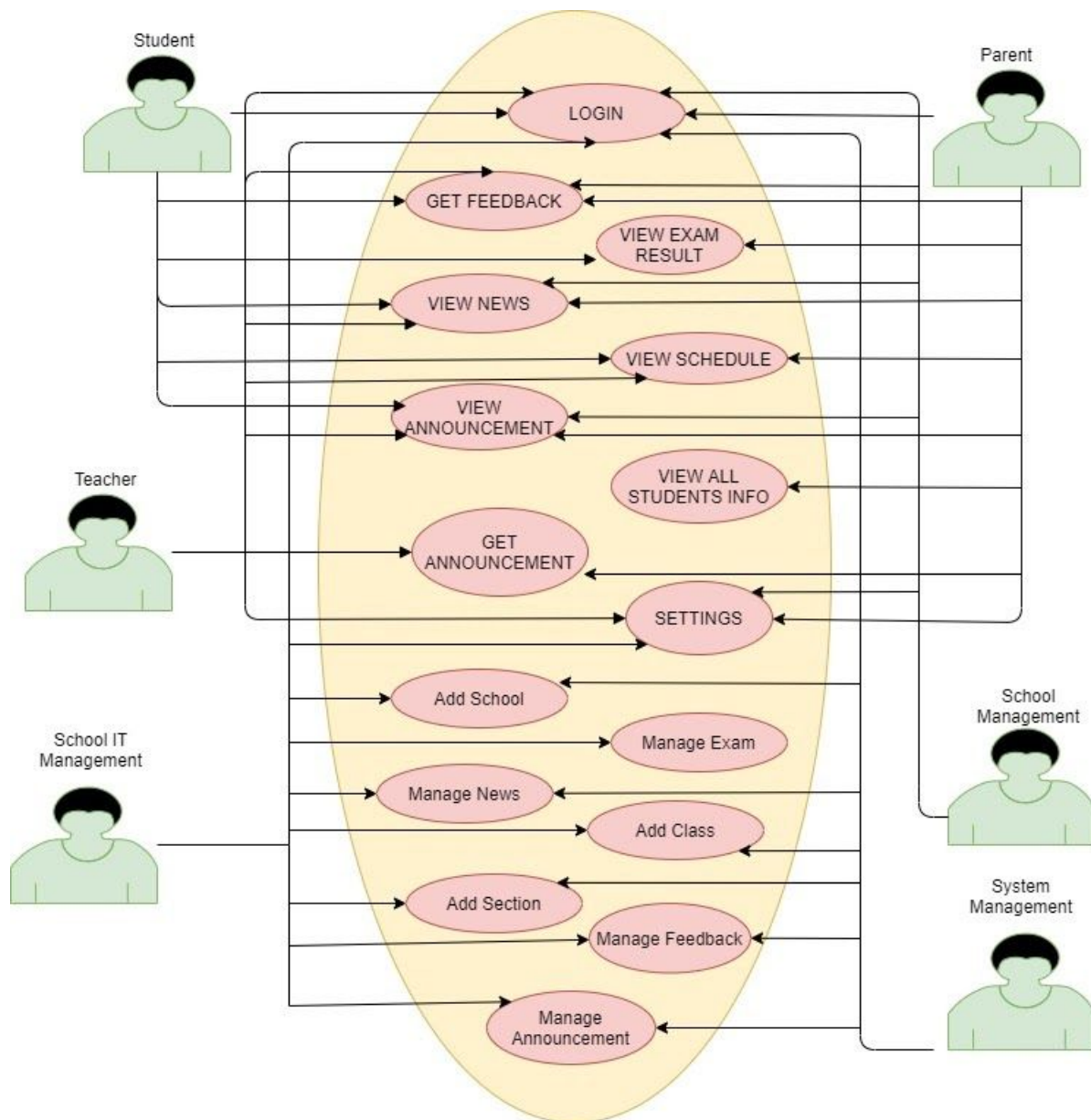


4.4 PROCESS DESIGN

In a project, process management is the cornerstone of the project. We used the waterfall model as it is always more effective to progress step by step in our project and facilitates the process management.

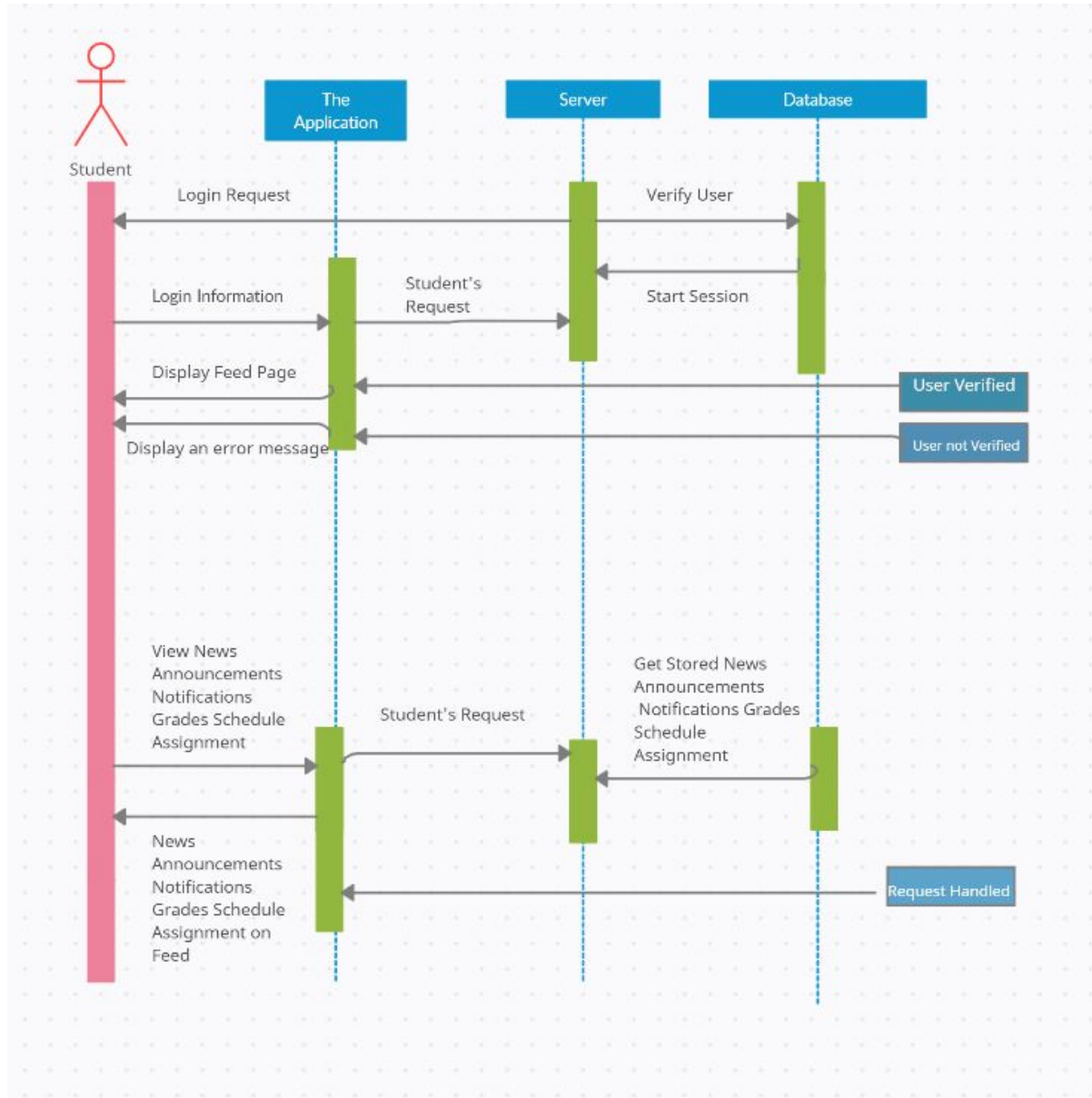


4.4.1 Use Cases

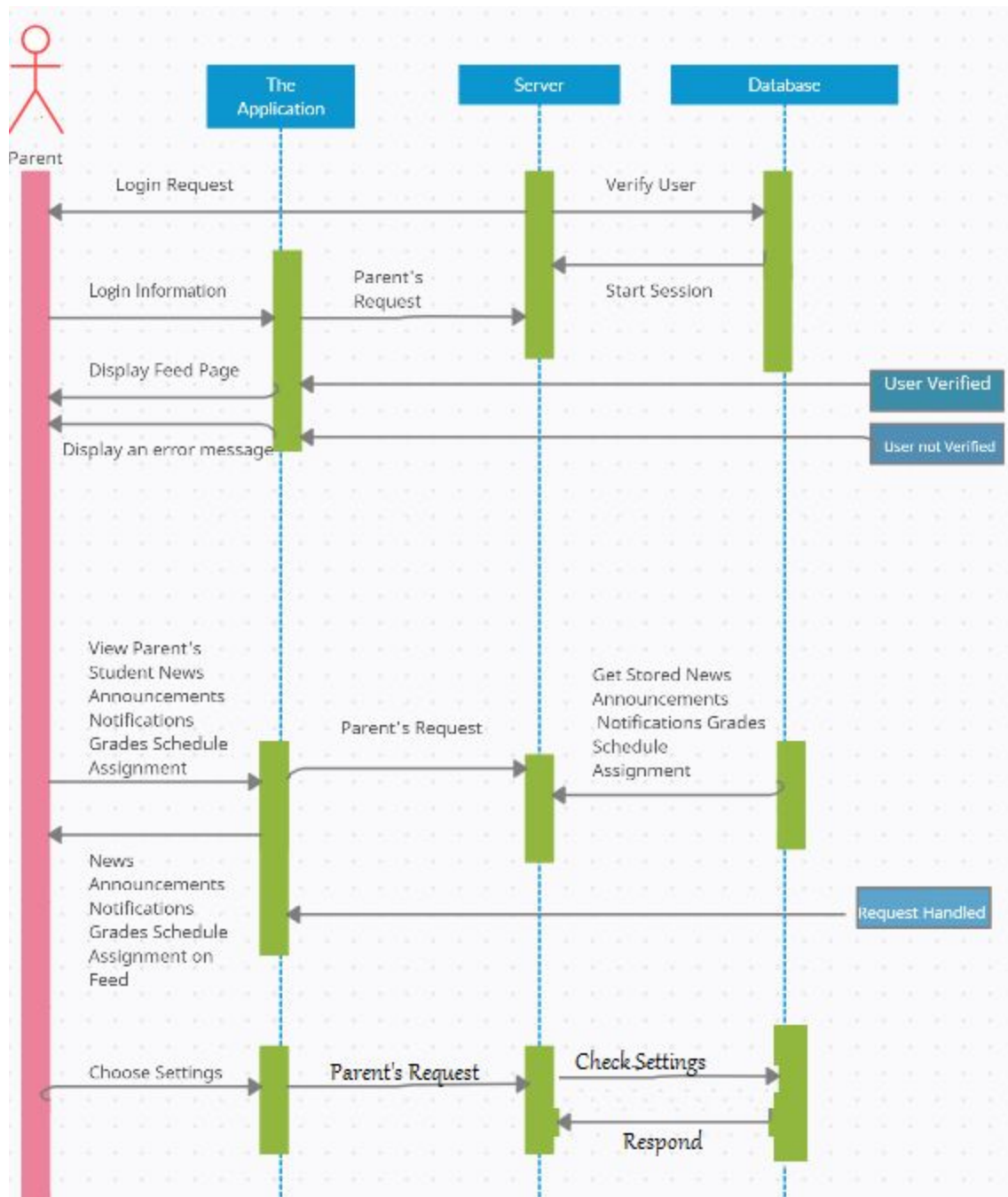


4.4.2 Sequence Diagrams

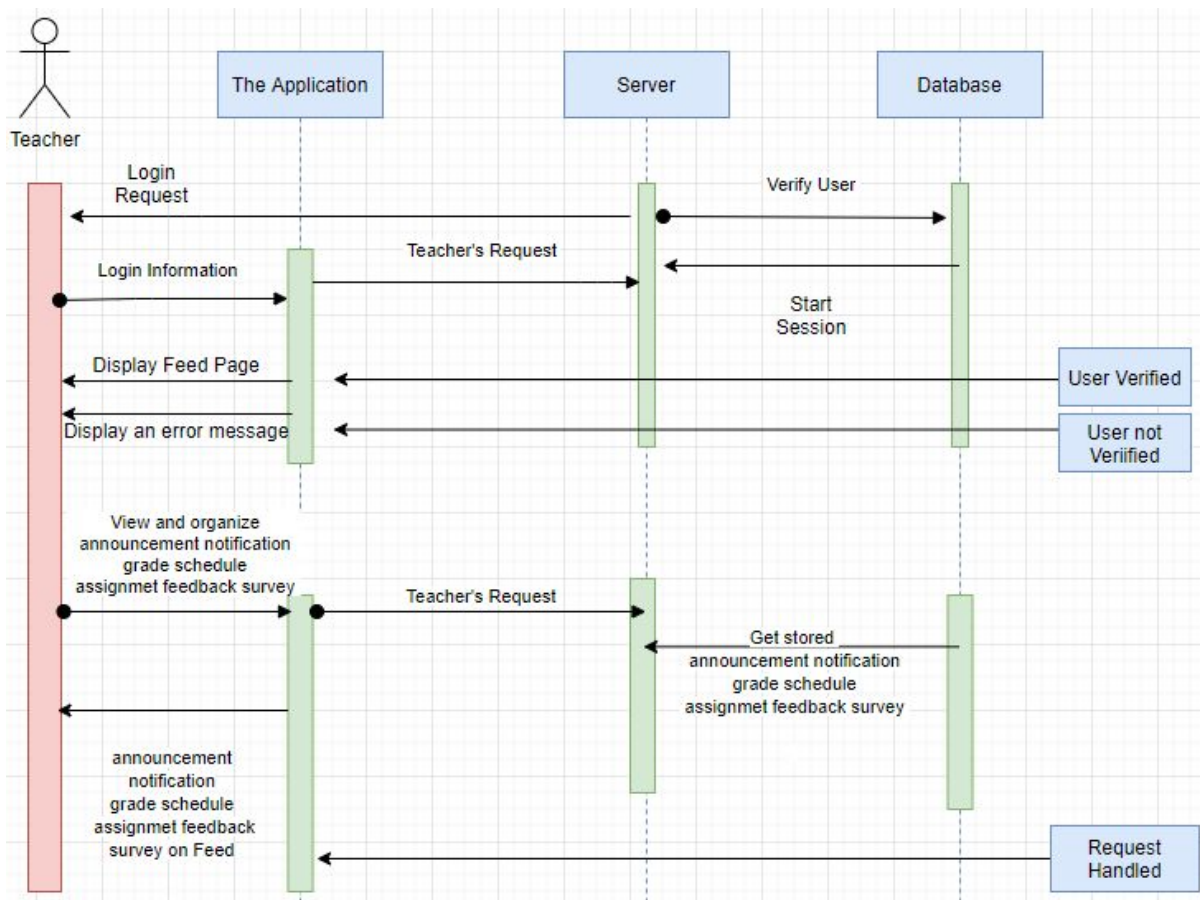
4.4.2.1 Student Diagram



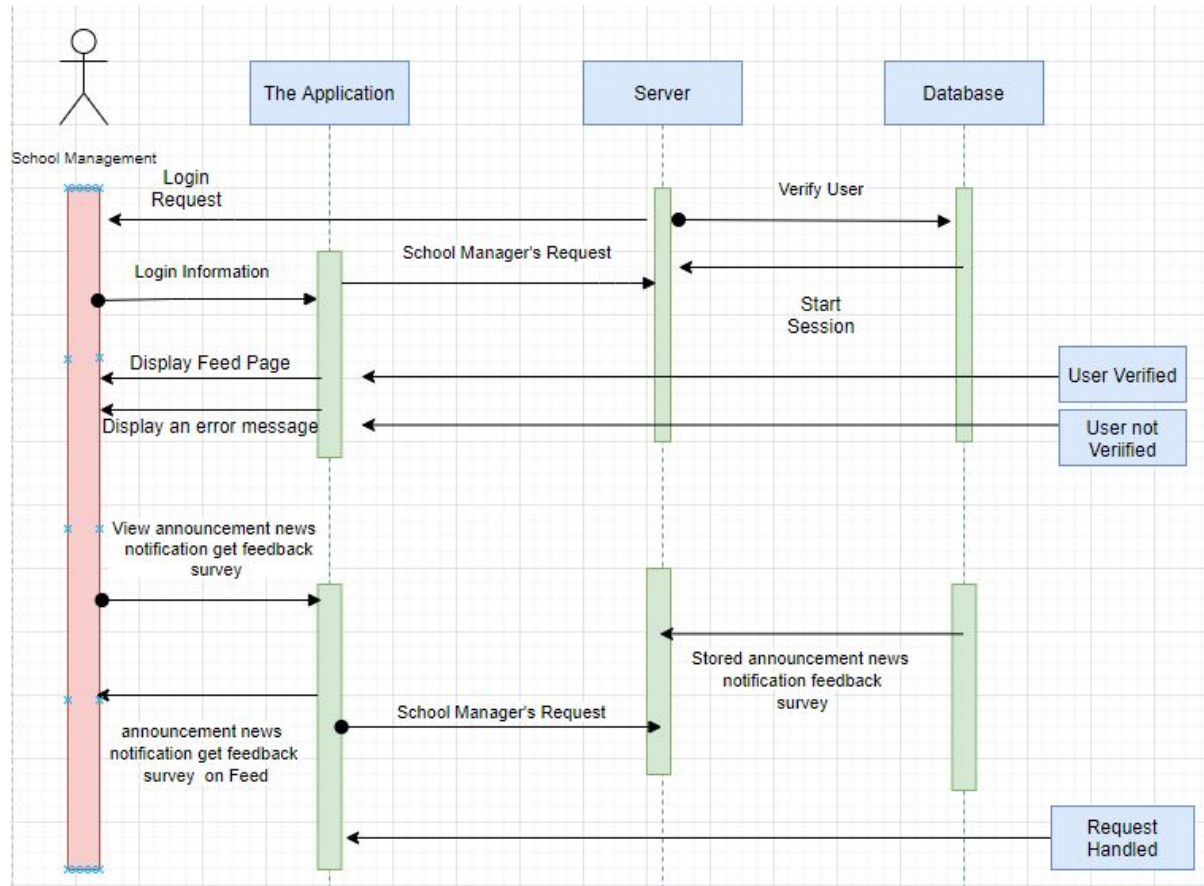
4.4.2.2 Parent Diagram



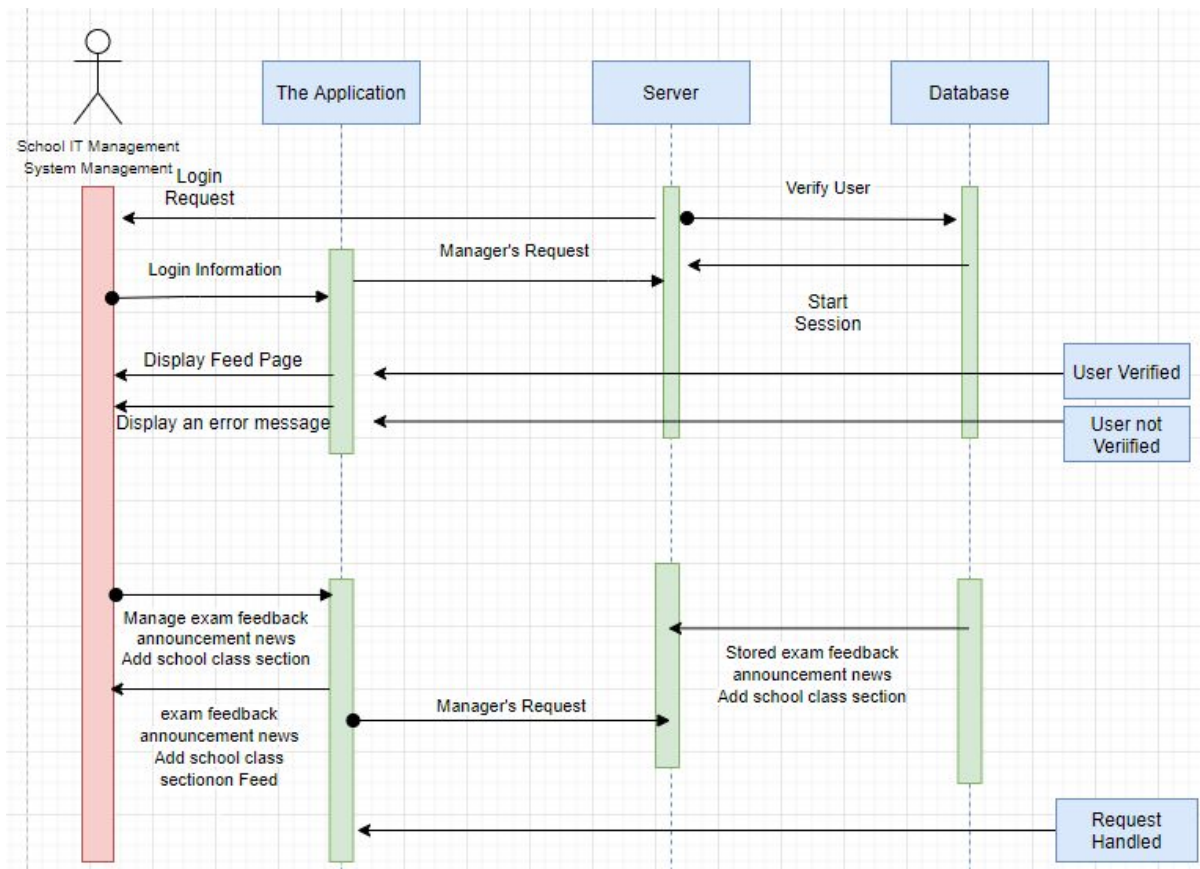
4.4.2.3 Teacher Diagram



4.4.2.4 School Management Diagram



4.4.2.5 School IT Manager / System Management (Admin)



4.5. DATABASE DESIGN

