SOFTWARE REQUIREMENTS SPECIFICATION

for

Online Class Management System :: KGP Learnings

Version 1.0 approved

Prepared by :
1. Amit Kumar (20CS30003)
2. Mir Mohammad Wasif (20CS10035)
3. Nikhil Saraswat (20CS10039)

IIT Kharagpur

Prof. Sourangshu Bhattacharya Prof. Abir Das

16th march , 2022

Contents

Revision History

Version	Date	Author(s)	Reason For Changes
1.0.0	10.03.2022	Nikhil Saraswat, Amit Kumar, Mir Mohammad Wasif	Started the SRS documentation : Introduction, Overall Description, External Interface Requirements
1.1.0	15.03.2022	Nikhil Saraswat, Amit Kumar, Mir Mohammad Wasif	Modified/Added following points : System Features, Other Nonfunctional Requirements, Other Requirements, Appendix

1 Introduction

1.1 Purpose

The virtual classroom enables instructors to give web-based training to students and employees who are geographically dispersed. Live classroom lessons can be held in a virtual classroom. Aside from being user-friendly, a good virtual classroom has collaboration capabilities built in that engage students in active learning. For example, the virtual classroom allows for live sessions, and students who are uncomfortable expressing questions in front of professors may easily do so. In a virtual classroom, teachers may quickly post video lectures and give study material to all students, as well as issue assignments and class assessments and establish deadlines for the same and he/she can evaluate the assignments and tests and give marks for each individual student.
Students can attend online classes, they can get a message before start of the class and see the notices as the instructor puts them on the website and can ask doubts personally without any hesitation.
In this Online or virtual learning makes the education convenient and comfortable for both teacher and students. It saves time and money both. It will usher in the immense flexibility and sophistication in the existing learning platform structures, with the perfect blend of synchronous and asynchronous interaction. It provides a means of collaborative learning for the students.
The purpose of this document is to show the detailed explanation of the objectives, features, user interface and application of **KGP Learnings** System. Document provides the detailed profile of the external interfaces, performance considerations and design constraints imposed on the implementation. This document will further assist the various stakeholders by serving as a reference manual.

1.2 Document Conventions

The standard font used throughout the document is Lucida, with font size 12.
Then Titles of the various sections of this SRS document have been represented in bold, with font size 14 and font Lucida. Important parts of the document have been indicated in bold.

1.3 Intended Audience and Reading Suggestions

This SRS is for developers, project managers, documentation writers, testers and users. Further the discussion will provide all the interior , external, functional and also non-

functional informations about "Online Class Management System (OCMS)".
As mentioned above these people among audience are as following :-

- **Project managers** - Project managers are those who supervise the entire project.
- **Developers** - This group of specialists uses programming languages to put the developer's concept into action. They are in charge of all application components as well as graphical user interfaces.
- **Tester** - This group of people tests the produced system using specific test cases to assess its efficiency and estimate its performance.
- **Documentation writers** - Documentation writers create user manuals and other papers required for the correct configuration of a system in a certain operating environment.
- **OCMS Users** - People who want the system to be implemented in their educational institutions. They are accountable for the quality of the software requirement specification document by providing useful feedback on the original requirement papers.

This Software Requirement Specification document is divided into five subsections.
Section 1: Introduction
Section 2: A Overall description of the web-application, including information about features, user-defined classes, operating environments, limitations, and documentation..
Section 3: External Interface Requirements giving a brief introduction to user, hardware, software and communications interfaces.
Section 4: Gives functional requirements of different features.
Section 5: Provides a list of non-functional requirements
Section 6: Other requirements

1.4 Product Scope

KGP Learnings is an online learning environment that enables for real-time contact between the tutor and the students as they are engaged in learning activities.
In other terms, this kind of virtual classroom is a shared online place where learners and tutors collaborate at the same time. These conversations are often carried out via video-conferencing. Participants are given tools to deliver learning information in a variety of forms, as well as to carry out collaborative and individual activities. In this sort of engagement, the instructor plays the crucial function of moderator, guiding the learning process and supporting group activities and debates.
The advantages of having **KGP Learnings** are that students may watch courses at their leisure. They can take tests online, obtain progress reports, and participate in extracurricular activities. Online, faculty can grade test sheets, arrange tests, and execute

associated activities. The system will provide an easy-to-use web-based service that will allow management to effectively manage all resources.

1.5 References

IEEE. IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

2 Overall Description

2.1 Product Perspective

This is a replacement for existing system(Like Microsoft Teams). This platform will help professors and students to interact with each other for any academic need/activity. Additionally, the frontend will be used by the users to access/submit/upload assignments , join classes ,add video lectures , providing grades or any other purpose. In the backend, our software engine will allow us to aggregate the data, store the data, and produce results which will be of value to the users.
KGP_learnings , web application will be built on top of Django + Python and interface with the SQLite3 server to store and pull relevant data.
The backend will be the SQLite3 server that will store and push the data that is being used by the application. The backend will be primarily used by professors.

STUDENTS	TEACHERS
can see class notice	write notice for all students
submit assignment	upload assignment
can access teachers list	see student list
attend live class	add/remove student
add reminders	organise classes
can personally message others(teachers+students)	can personally message others(teachers+students)
change personal details	change personal details

2.2 Product Functions

- Maintaining Database for
    1. Details of students and professors
    2. semester lectures, study materials and assignments
    3. marks distribution enumerate
- Professor's End
    1. upload assignments/study material/recorded lectures
    2. access student list

3. evaluate assignments and assignment grades
4. write a class notice
5. can personally contact any student
6. schedule live classes
7. can organise tests

- Student's list
  1. submit assignments
  2. access study material/recorded lectures
  3. access teacher list
  4. see assignment's grades
  5. access class notice
  6. can personally contact any professor and ask doubts
  7. attend live classes
  8. can give tests

- Common Interface/Website
  1. can create account
  2. change personal details
  3. change passwords
  4. Contact us
  5. Feedback form

2.3 User Classes and Characteristics

The System has the following user classes:

- Professor
- Student

The Student may create an account and then attend classes, submit assignment, access recorded lectures and much more from the desktop after logging in. The Professor sees the assignments and evaluates it. The student has a certain deadline to give his or her tests and assignments. The professor notifies the students regarding live classes. The professor may change the deadline, monitor list of submission, and ensure that each student has submitted or not.

2.4 Operating Environment

- Windows 7
- Windows 8
- Windows 10
- Linux
- Mac OS X

This will be a web based system.We will actually have a server to run. Stores all functions and custom user data. Custom ends will just be a GUI.
All the modules mentioned in requirements.txt are required to be installed for application to work.

2.5 Design and Implementation Constraints

- The system has dependency on the database and the hotel's existing traditional system.
- Internet connection is a constraint for this system because system is available from cloud therefore customer needs to have good network connection to connect to our web interface
- Advanced security features haven't been introduced.
- This software was developed in Django(python framework)+ HTML+CSS+JavaScript.
- This software requires specific modules to be installed before hand.
- Django version $\geq$ 2.2 is required.
- For windows operating system , Microsoft visual c++ build tools $\geq$ 14 is required.

2.6 User Documentation

License and User Manual will be provided along with the software.

2.7 Assumption and Dependencies

Assumption: The user will have the basic resources required for this web application :-

- Active Internet Connection
- A browser to access internet

- A desktop system or a cell phone to work on

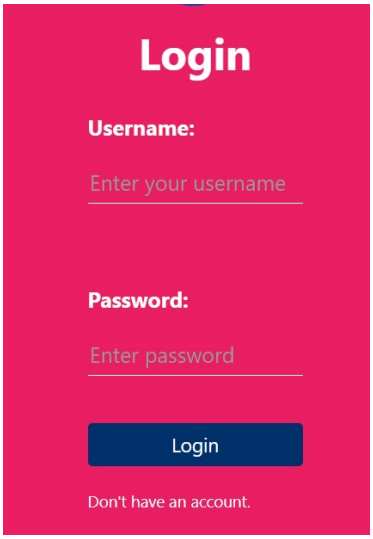
Other than these, the web application software depends on various APIs, which is required.

3 External Interface Requirements

External interface requirements are types of functional requirements. They're important for embedded systems. And they outline how our product will interface with other components.

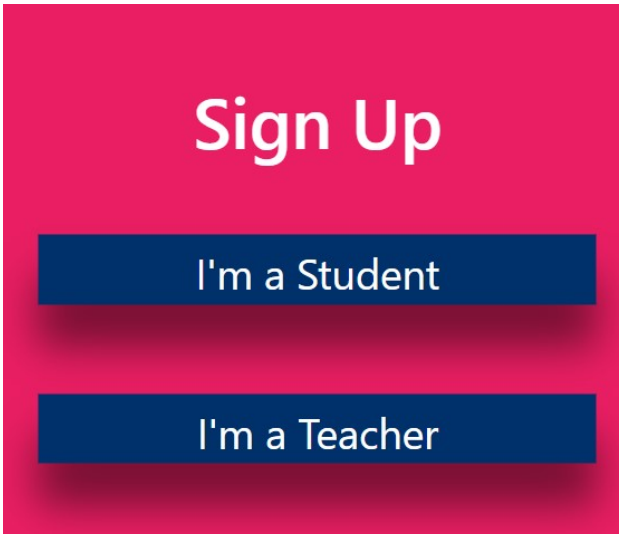
3.1 User Interfaces

1. Login : The login portal will allow users to log in to their respective account using username and password. After successful login user will be redirected to the homepage.

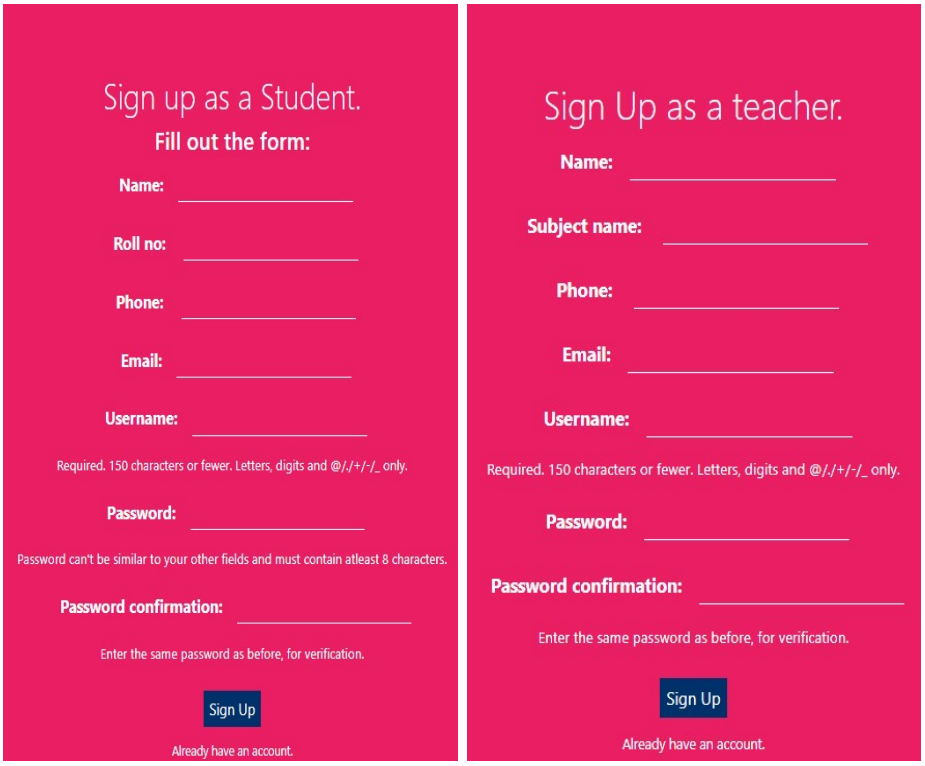


login page

2. Sign Up : The Sign Up portal will allow students/Professors to sign up for an account with name, roll no., phone no., email, username and password. It will allow teachers to sign up for an account with name, subject, phone no., email, username and password.

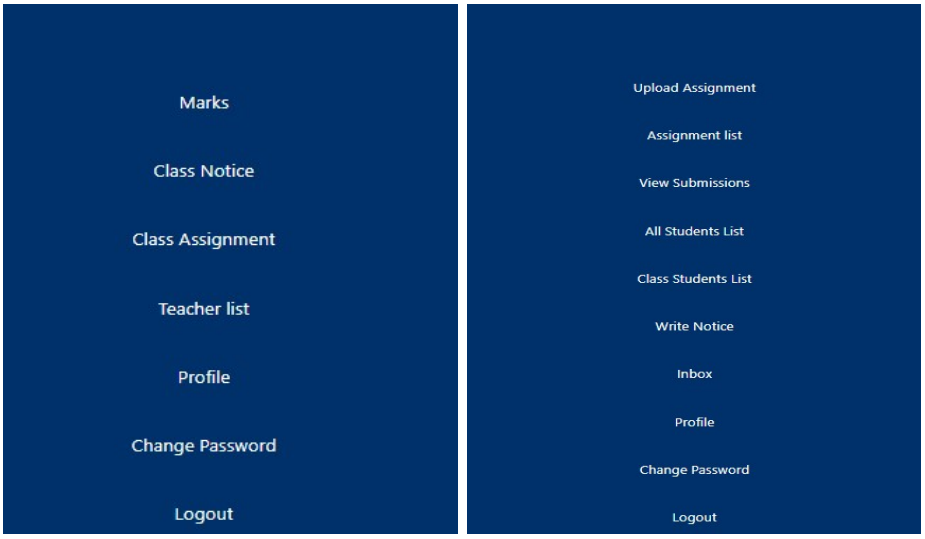


sign up option



sign up interface

3. Available Features : There are different different features which will appear for student and professors.



Features for Student and Professor respectively

Note: We can add more features in future.

3.2 Hardware Interfaces

Our web application will be accessed via a web browser on a desktop or laptop computer. Assuming the user is working in a standard computer environment, the user can interact with the system via mouse clicks and keyboard inputs.
There are no other special hardware interface requirements.

3.3 Software Interfaces

The GUI will be made by HTML, CSS,Javascript. Python would be used for the processing on the server. Database will be made using SQLite3.
Class Manager uses the multi-user concept of Django where student and professor are different types of user and have different functionalities.

3.4 Communication Interfaces

The communication will take place across a network in an encrypted format. When accessing our web application via a web browser on a laptop/desktop, the user will use HTTPS to access the web interface. Our system will be accessed by the user via our website.

4 System Features

This subsection presents the identified functional requirements for the online class management system. Where possible, the requirements have been demarcated based on their relevance to the users of the system, that is, professor and students.

4.1 SignUp & Login

4.1.1 Description and Priority

This is the initial step for each user to do in order to begin using these services. By joining up for these services, the user must create an account.

4.1.2 Stimulus/Response Sequences

The user must provide his personal, educational, and professional information. The system will establish an area for the user and his data.

4.1.3 Functional Requirements

- Requirement 1: Request user information.
- Requirement 2: Allocate memory space in the server.
- Requirement 3: A function that provides the user with a secure username (unique) and password.
- Requirement 4: A function that acts as an extension to keep the user logged in.
- Requirement 5: A function for creating a user's resume and personalising his dashboard.

4.2 Uploading Assignments/Video Lectures/class notes

4.2.1 Description and Priority

This will be the most used features of our web application. Because after every class lecture,notes and probably assignments needs to be uploaded.This is most important feature after login and sign up.

4.2.2 Stimulus/Response Sequences

Required files needs to be submitted by professor.The files will be retrieved by the students.

4.2.3 Functional Requirements

- Requirement 1: Request server information.
- Requirement 2: Allocate memory space in the server.
- Requirement 3: A function that provides the user with a option to upload any file.
- Requirement 5: A function for creating a user's resume and personalising his dashboard.

4.3 Live classes and calendar

4.3.1 Description and Priority

Live online classes have gained a huge popularity in this era, so it is one of the most important features of our web application. According to scheduled time professors will set the timings of the respective classes and also update it in the calendar. The calendar will be accessible to the students. It will work as a reminder for them.

4.3.2 Stimulus/Response Sequences

The professor should schedule the live classes and upload that in the calendar.

4.3.3 Functional Requirements

- Requirement 1: Needed a virtual classroom API.
- Requirement 2: Function that schedule live classes.

4.4 Maintainance

4.4.1 Description and Priority

Web application maintenance is just as vital as web application creation. That's because if online apps are infected with malware, they may not work as expected, rendering the entire development process pointless. Delaying this crucial action will only prove to be too costly and inconvenient for any firm. In terms of web traffic and earnings, obsolete technology and incompatible data can cause a significant setback.

4.4.2 Stimulus/Response Sequences

As per contemporary web trends, corrective efforts such as technological upgrades, debugging, patch updates, and functional optimization are required.

4.4.3 Functional Requirements

TBD

4.5 Feedback

4.5.1 Description and Priority

This is the last step for the user before exiting from the online portal.The visitor would be asked to fill up a form with experience details, which would be helpful to upgrade the service quality. Priority: High

4.5.2 Stimulus/Response Sequences

Professors and Students will have a feedback form in the home page which can be use to report the bugs and suggestions to improve the product.

4.5.3 Functional Requirements

Requirement 1: Memory to store the feedback details in the server

Requirement 2: Feedback form to collect database

5 Other Nonfunctional Requirements

5.1 Performance Requirements

1. The software will be built on a local server.
2. The product will be very fast depending on processor.
3. Hardware components will influence performance.
4. The response of server should be fast (within 2 seconds) when coupled with high speed internet.

5.2 Safety Requirements

1. There is a Feedback form available where users can report any bugs they have encountered so that the developers can fix it in the next release.
2. There is no danger from any threat that the system may issue. Because it is a web-based application, there will undoubtedly be internet-based risks. These should be kept to a minimum as much as feasible. Before a user may upload documents or photographs, a virus scanner will scan them.

5.3 Security Requirements

1. The system is designed with login capabilities so that only approved Institute personnel may access it. Permissions to read, write, and change are provided based on who the knowledge is intended for.
2. After authenticating the user's credentials, some of the data can be viewed.

5.4 Software Quality Attributes

1. **Maintainability**
   The product's many versions should be simple to maintain. It should be simple to add code to an existing system and to upgrade for new features and technologies as they become available. Maintenance should be inexpensive and simple.

2. **Flexibility**
   It should be adaptable to other products with which it must engage. It should be simple to integrate with other standard third-party components(APIs).
   Popular browsers such as Chrome and Firefox should execute the web application without a hitch.This web application offers both basic and advanced functionality to its customers.

3. **Usability**
   This may be judged in terms of usability. The application should be simple to use. It should be simple to learn. The navigation should be straightforward.
   It may be utilised by both specialists and average people because to its well-designed and user-friendly interface.

5.5 Business Rules

This web application should only be used if a suitable contract agreement has been reached with the firm that created it. In the event of an issue, the user should notify the project manager right away. Without prior consent, the web application should not be outsourced to a third party.

6 Other Requirements

1. Copyright and licensing are necessary.
2. An internet platform will be required to host the database.

7 Appendix A:Glossary

KGP Learnings - Name of the web application

HTML - HyperText Markup Language

CSS - Cascading Style Sheets

SQLite3 - SQLite is atomicity, consistency, isolation, durability (ACID) compliant. This embedded relational database management system is contained in a small C programming library and is an integral part of client-based applications.

OCMS - Online Class Management System

IEEE - Institute of Electrical and Electronics Engineers

HTTPS - Hypertext Transfer Protocol Secure

API - Application Programming Interface

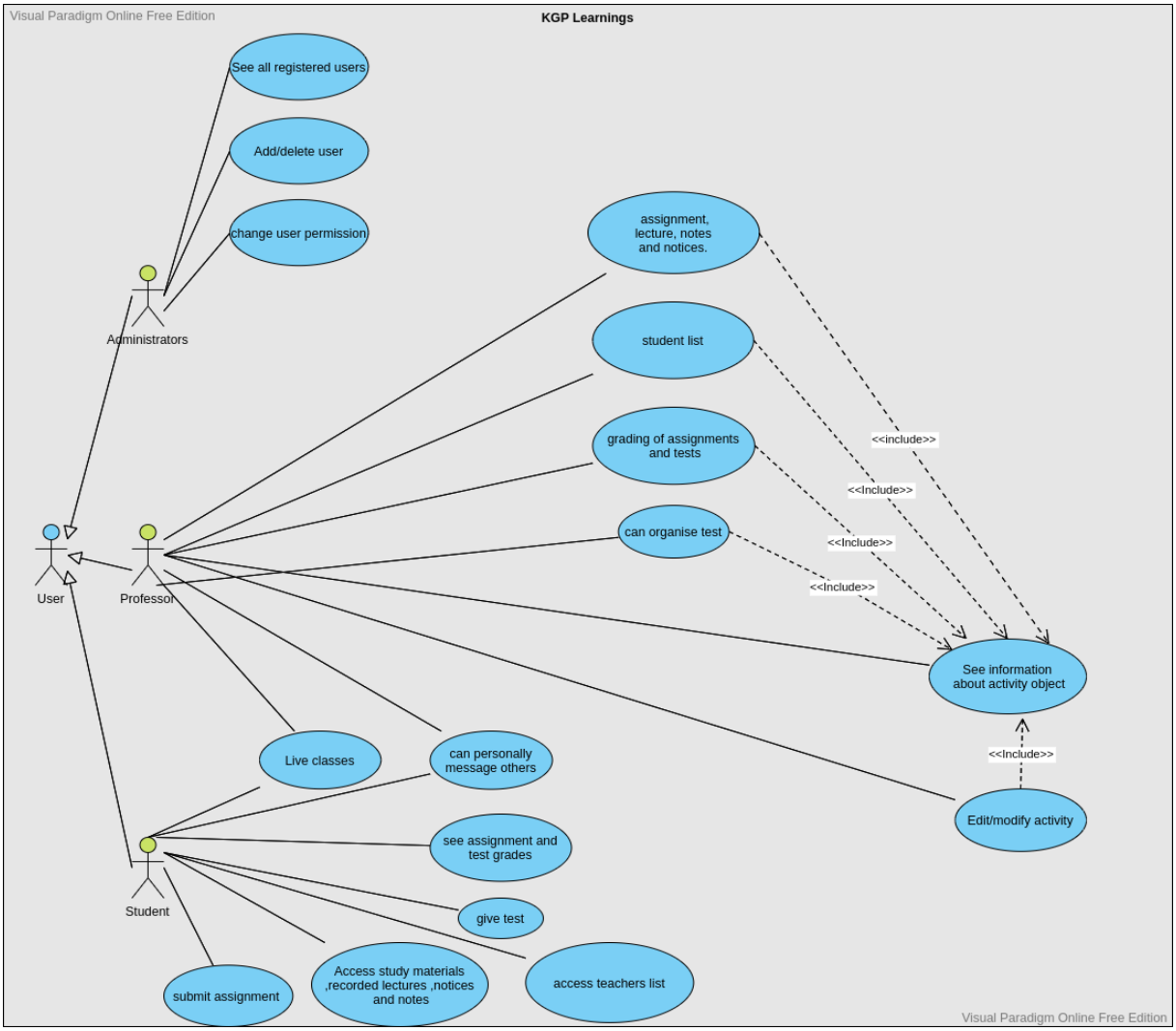
UML - Unified Modelling Language

TBD - To Be Determined

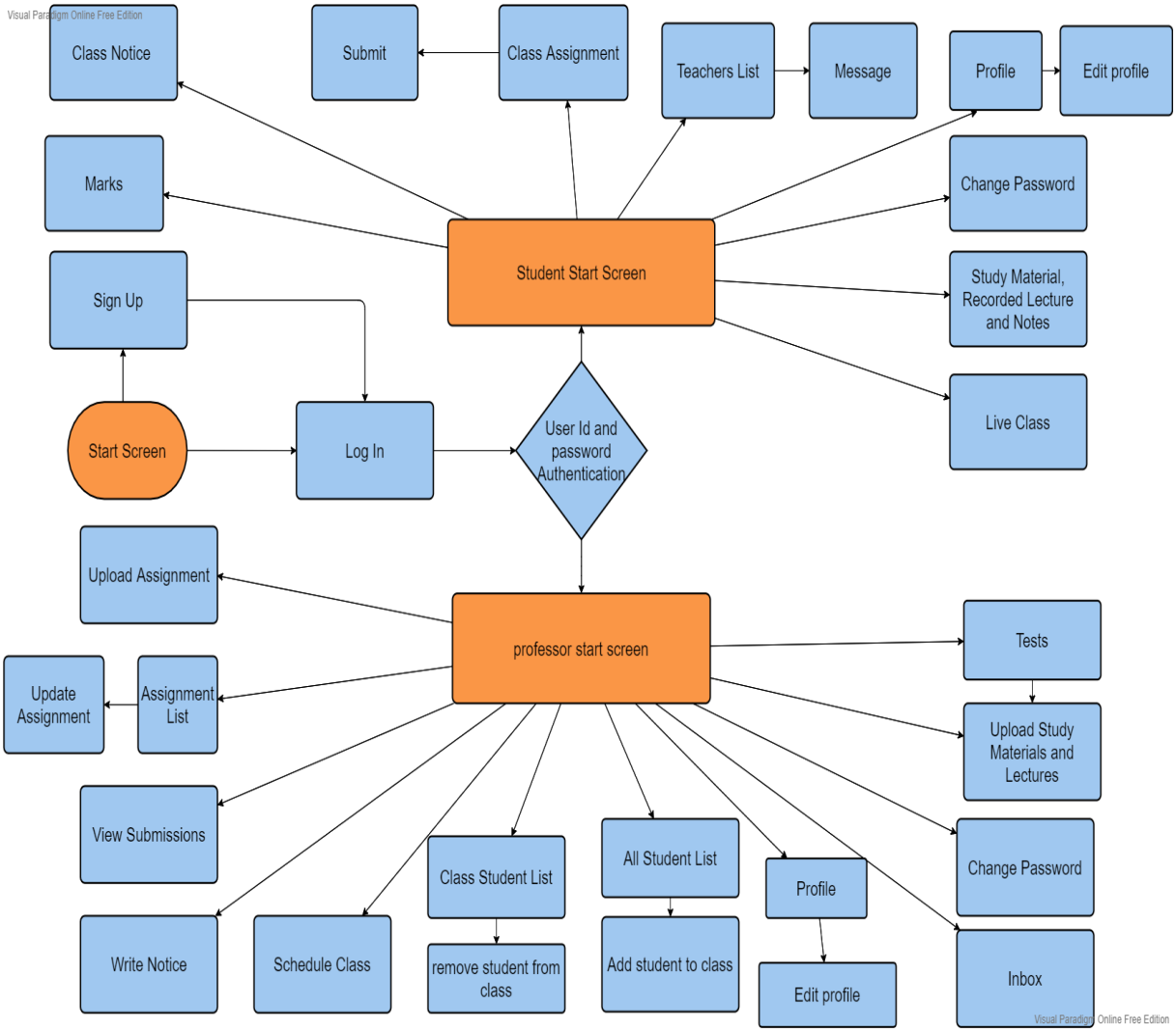
8 Appendix B:Analysis Models

8.1 UML Diagram

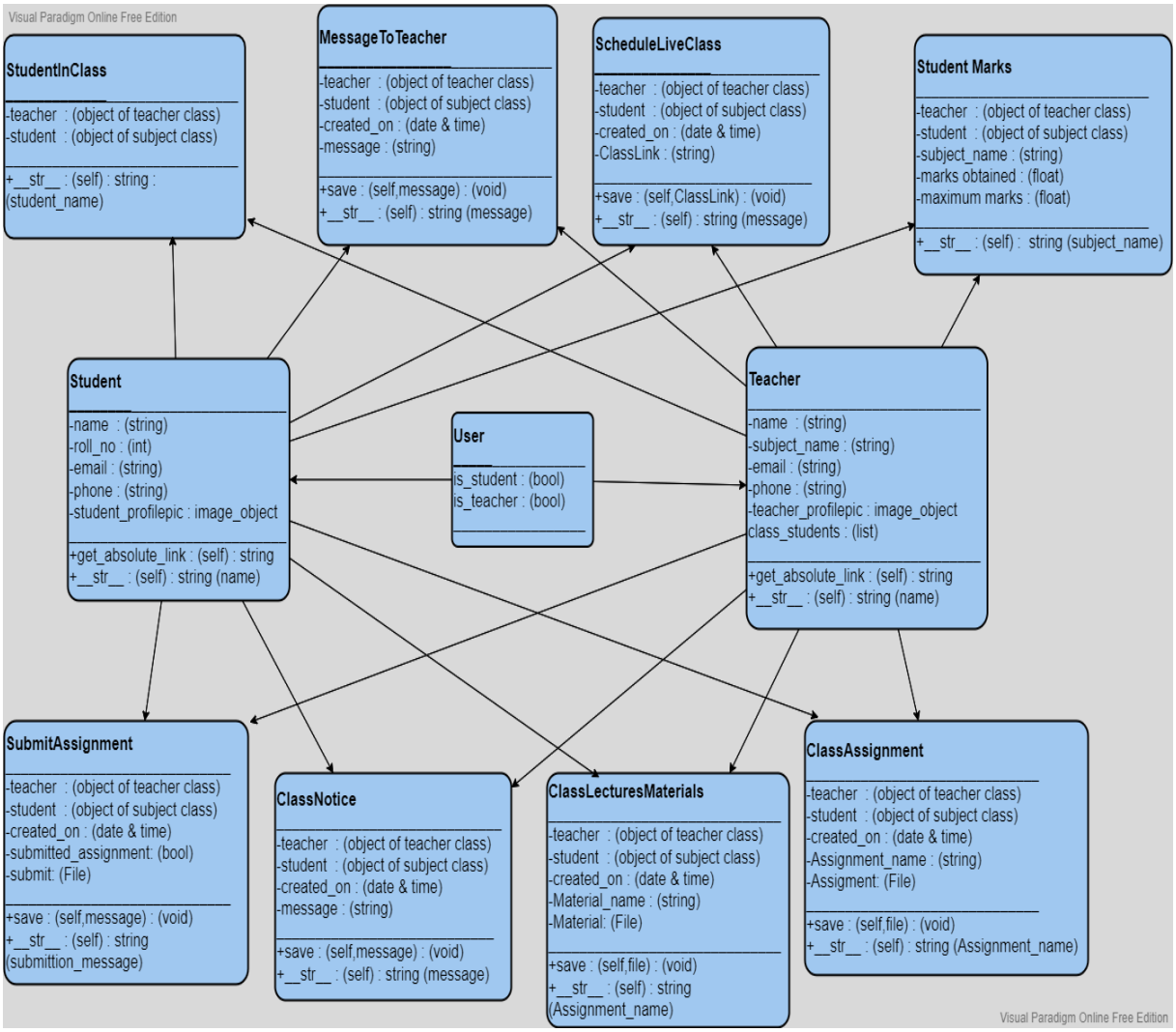
8.1.1 Use Case Diagram



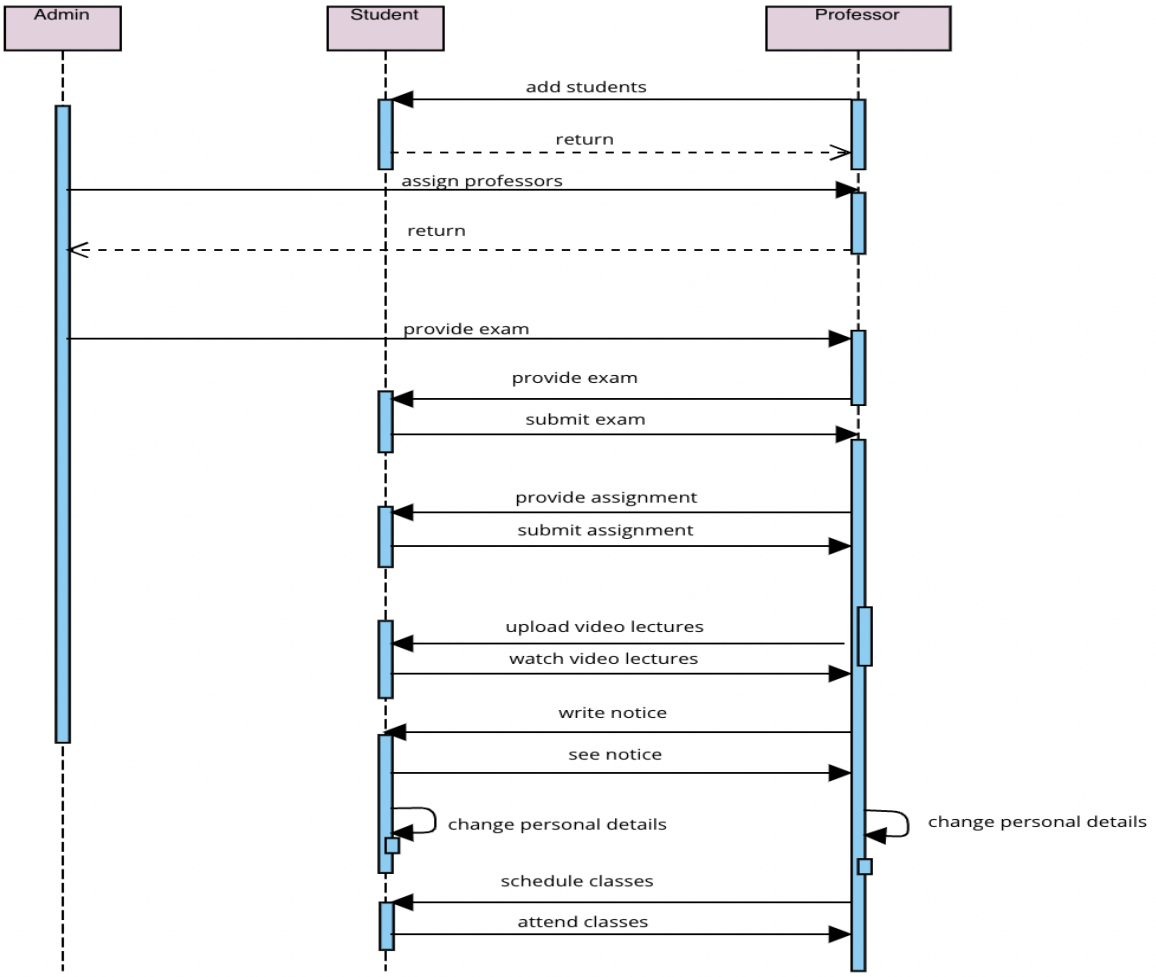
8.1.2 Interface Diagram



8.1.3 Class Diagram



8.1.4 Sequence Diagram



9 Appendix C:To Be Determined List

a) 4.4.3