1.INTRODUCTION

1.1 Purpose

The purpose of this document is to capture, in natural language and at a functional level, the description and requirements of an effective project management for any academic institution. This is a functional description of those features required to address current requirements. A short discussion accompanies each requirement, to add the background and framework necessary to explain the functionality. It also describes nonfunctional requirements and other factors necessary to provide a complete and comprehensive description of the requirements for the software.

1.2 Scope

The pandemic has proved that the effective use of modern technology is a key to solve most of the problems effectively with minimal human interaction possible. In this regard “Converge” a collaboration tool which as a one point solution for academic project management needs, “Converge” a colab platform which connects students with their respective guides and provide one-to-one connection from there on and thereby making all the steps in a project development as facile as possible. There will be two stake holders in “Converge”, a student and a guide. Both stakeholders can register themselves and login to the system. holders. Documenting and processing has never been easier, with an intuitive UI, students can add the project details such as synopsis, documentation of each phase of development at ease. The status of the project will be constantly updated on completion of each phase so that any changes required can be made possible at earliest itself thereby reducing the operational overhead.

2.GENERAL DESCRIPTION

This section will give you an overview of the entire system. It will also describe what type of stakeholders that will use the system and what functionality is available for each type.(Converge) is meant to serve as a common tool where management of every academic projects can be carried out at ease. Our goal is to make a platform that can be used as a collaboration tool between students who are doing projects in their academics and the guides who guide them.

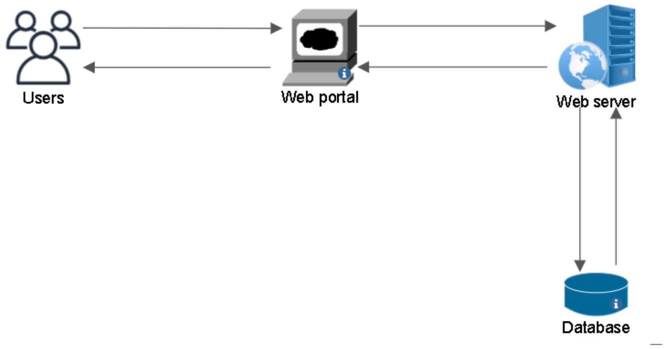


Figure 2.0 Block Diagram

Since this is a data-centric product it will need to store the gathered data somewhere. For that a database will be used, the web server will communicate with the database and process the data according to applied business logic and the user requests. As illustrated in the Figure 2.0 the users will request the request/update the data via web portal which will be processed by the web server and the results will be stored in the database as well as it will be displayed to user via Web portal.

One assumption about the project is that it will always be used on web browsers which have the latest rendering capabilities. In the case that project is delayed, there are some requirements that could be transferred to the next version of the application.

3.FUNCTIONAL REQUIREMENTS

This section includes the requirements that specify all the fundamental actions of the software system.

3.1 The Student class

ID:FR1

TITLE**:** Load the application the web browser

DESC**:** A user should be able to load the application in the web clients such chrome, edge, firefox etc.

ID:FR2

TITLE**:** User registration

DESC**:** Given that a user opened the application in the web browser, then the user should be able to register through the signup page in the portal. The user must provide the username, password, and email address.

ID:FR3

TITLE**:** User log-in

DESC**:** Given that a user has registered, then the user should be able to login to the application. The login session information will be stored on the client and the will be used as the authentication token for the future requests.

ID:FR4

TITLE**:** Retrieve password

DESC**:** Given that a user has registered, then the user should be able to retrive the his/her passwords by e-mail.

ID:FR5

TITLE**:** Update account details

DESC**:** Given that a user has registered, then the user should be able to update the non primary accont details.

ID:FR6

TITLE**:** User log-out

DESC**:** Given that a user has logged in, then the user should be able to login out of the application there by freeing any used resources.

ID:FR7

TITLE**:** Add projects

DESC**:** Given that a user has logged in, then the user should be able to add project along with all the required details.

ID:FR8

TITLE**:** Search projects

DESC**:** Given that a user has added the project, then the user should be able to search the projects that had been added.

ID:FR9

TITLE**:** Update the status

DESC**:** Given that a user has added the project, then the user should be able to change the status of each phase provided valid document is provided.

ID:FR10

TITLE**:** File upload

DESC**:** Given that a user has added the project, then the user should be able to relevant document after completion of each phase.

ID:FR11

TITLE**:** View the remarks

DESC**:** Given that a guide has added the remarks, then the user should be able to view the remarks.

3.2 The Guide class

ID:FR1

TITLE**:** Load the application the web browser

DESC**:** A user should be able to load the application in the web clients such chrome, edge, firefox etc.

ID:FR2

TITLE**:** User registration

DESC**:** Given that a user opened the application in the web browser, then the user should be able to register through the signup page in the portal. The user must provide the username, password, and email address.

ID:FR3

TITLE**:** User log-in

DESC**:** Given that a user has registered, then the user should be able to login to the application. The login session information will be stored on the client and the will be used as the authentication token for the future requests.

ID:FR4

TITLE**:** Retrieve password

DESC**:** Given that a user has registered, then the user should be able to retrive the his/her passwords by e-mail.

ID:FR5

TITLE**:** Update account details

DESC**:** Given that a user has registered, then the user should be able to update the non-primary account details.

ID:FR6

TITLE**:** User log-out

DESC**:** Given that a user has logged in, then the user should be able to login out of the application there by freeing any used resources.

ID:FR7

TITLE**:** Search projects

DESC**:** Given that a student has added the project, then the guide should be able to search the projects that had been added.

ID:FR8

TITLE**:** View the projects

DESC**:** Given that a student has added the project, then the guide should be able to view the details that are given by the student.

ID:FR9

TITLE**:** Download the files

DESC**:** Given that a student has added the project files, then the guide should be able to download and view them.

ID:FR10

TITLE**:** Accept the final report

DESC**:** Given that a student has completed all the phases of the project, then the guide should be able to accept the final report and mark is as complete.

ID:FR11

TITLE**:** Give remarks

DESC**:** Given that a student has added the project, then the guide should be able to give remarks to it.

ID:FR12

TITLE**:** Project status

DESC**:** Given that a student has added the project, then the guide should be able to view the status of it at any given point of time.

ID:FR13

TITLE**:** Download status report

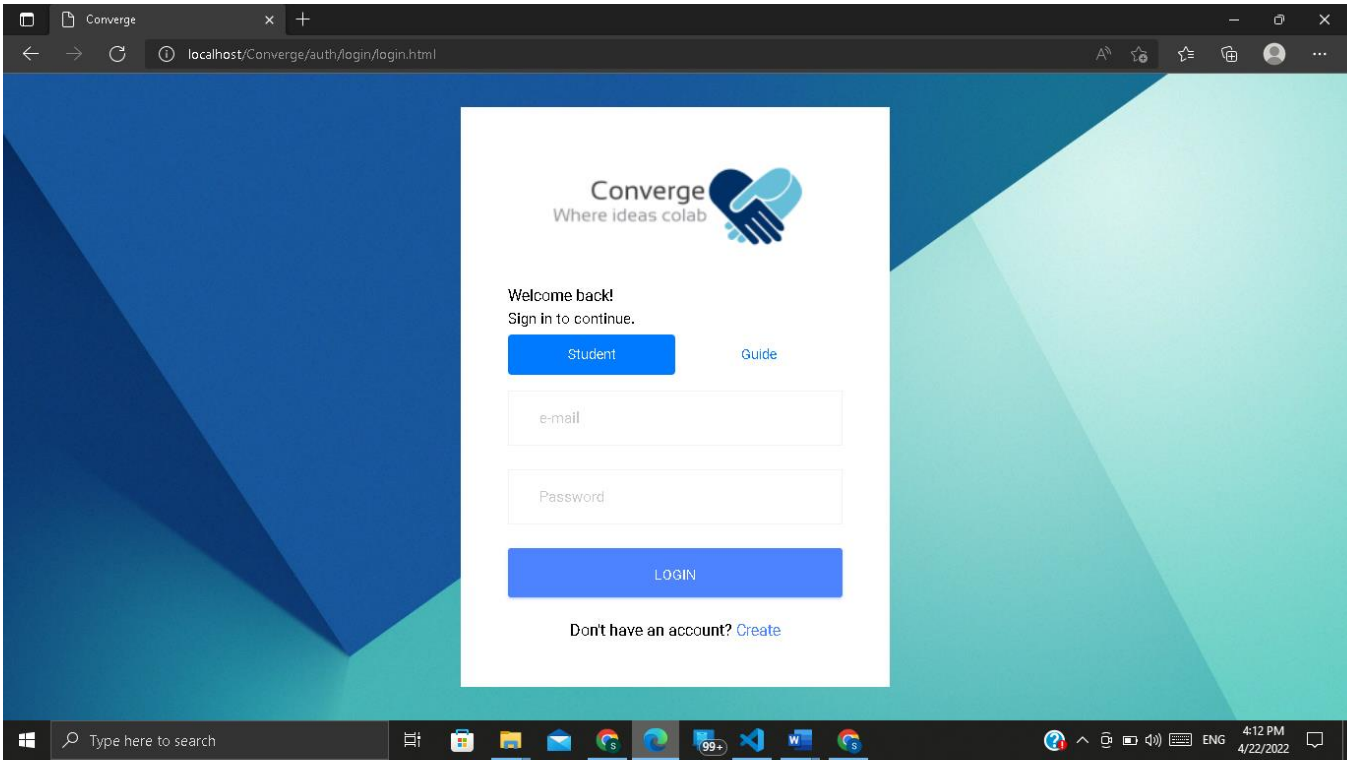
DESC**:** Given that a student has added the project, then the guide should be able to generate the csv report of the completed and pending projects.

4.INTERFACE REQUIREMENTS

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

4.1 User Interfaces

A first time user should see the login page when he opens the URL(Figure 4.1.1). If he is an existing user he can enter the credentials and move forward to dashboard else he need to click on create which will redirect to Signup page.

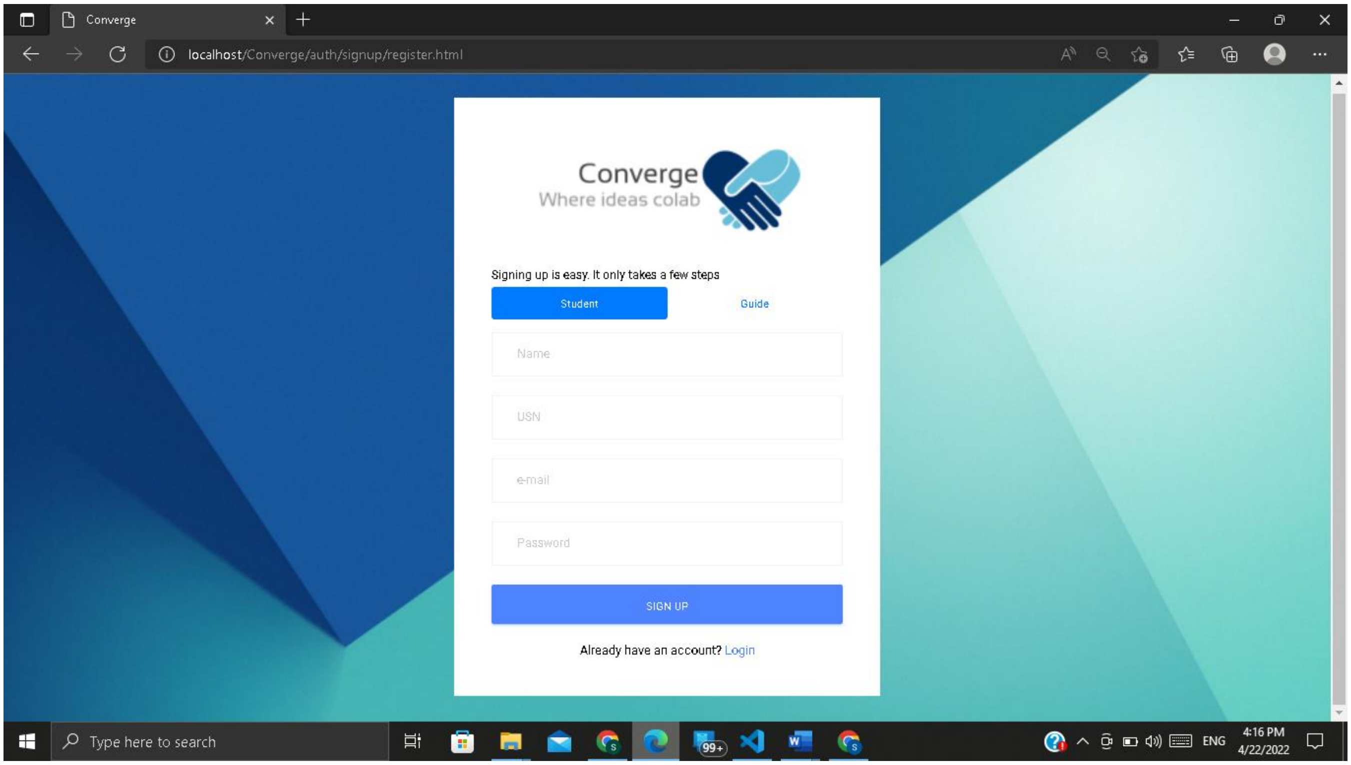


Welcome back!
Sign in to continue.

Don't have an account? Create

Figure 4.1.1 Login Screen

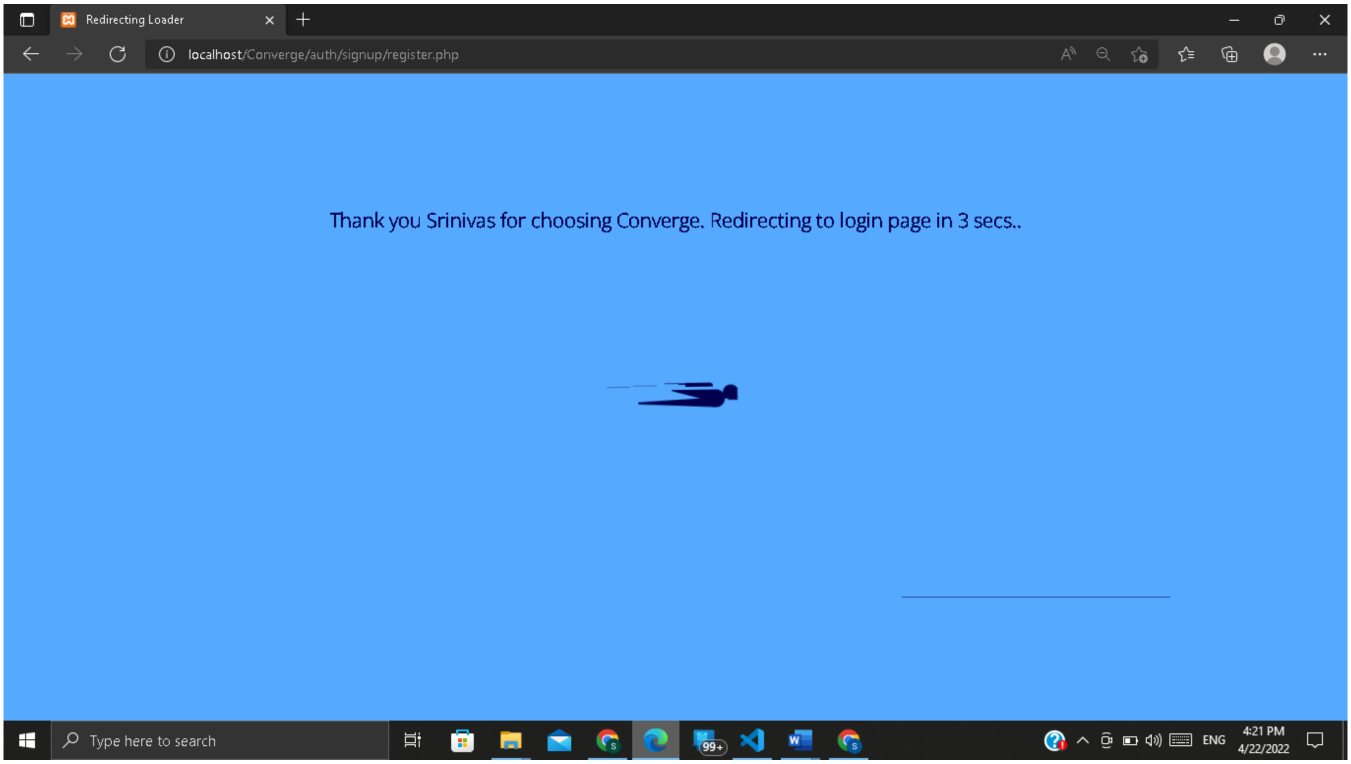
The signup page have several required fields (Figure4.1.2), user can sign up to the application based on their roles(Student/Guide),once the user provides all the required details and signup the user will be redirected to login screen (Figure 4.1.3).



Signing up is easy. It only takes a few steps

Already have an account? Login

Figure 4.1.2 Signup page



Thank you Srinivas for choosing Converge. Redirecting to login page in 3 secs..

Figure 4.1.3 Redirection page

4.2 Hardware interfaces

Since the web portal doesn’t have any designated hardware, it does not have any direct hardware interfaces. The hardware connection to the database server is managed by the underlying operating system on the web server.

4.3 Communication interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems for the web portal. The HTTPS protocol will be used to facilitate communication between client and server.

4.4 Software interfaces

The webapp gets the information from the users. The communication between the database and the web portal consists of operation concerning both reading and modifying the data, while the communication between the database and the webapp consists of only reading operation. An apache web server will accept all the requests from client and forward it accordingly.

5.PERFORMANCE REQUIREMENTS

The request should be processed within one second which is standard acceptable limit. The response time should be maintained while having at least 200 concurrent users. This statement provides a general sense of reliability when the system is under load. It is important that a substantial number of users be able to access the system at the same time, since an academic portal is important to the courses that employ it. The times when the system will be under the most stress are likely during assignment submissions. Therefore, it must be able to handle at least 200 concurrent users while maintaining one second response time.

6.OTHER NON-FUNCTIONAL ATTRIBUTES

6.1 Security

Passwords will be saved encrypted in the database in order to ensure the user's privacy and the user's IP will be logged everytime. The system will be protected against vulnerabilities such as SQL injection attacks.Proper measure will be taken to ensure the third party data storing entities will ensure the correct policies to secure user data.

6.2 Reliability

The reliability of the overall program depends on the reliability of the separate components.

6.3 Maintainability

MySQL is used for maintaining the database and the Apache server takes care of the site. In case of a failure, a re-initialization of the program is recommended.

6.4 Portability

The application is windows-based and should be compatible with other systems. Apache, PHP and MySQL programs are practically independent of the OS-system which they communicate with. The end-user part is fully portable and any system using any web browser should be able to use the features of the application.

6.5 Availability

The system should be available at all times, meaning the user can access it using a web browser, only restricted by the down time of the server on which the system runs. In case of a of a hardware failure or corruption, a replacement page will be shown. Also in case of a hardware failure or database corruption, backups of the database should be retrieved with the MySQL server and saved by the administrator.

7.OPERATIONAL SCENARIOS

This section describe a set of scenarios that illustrate, from the user's perspective, what will be experienced when utilizing the system under various situations.

7.1 Operational Scenario 1 – User Login

A registered user wants to log into system.

Purpose	User logs in to system using existing profile.
User	A user with an existing profile.
Input data	Profile username and password.
Output data	Corresponding page data.
Pre-condition	User is not logged in to a profile, input profile exists in data base, user password matches profile

Post-Condition	User's computer has been supplied with appropriate cookie, page data is appropriate for selected profile
Basic flow	Webpage looks up profile data and returns the matching cookie. Webpage is updated to match new user data.

7.2 Operational Scenario 2 – Content Sharing (Upload Files)

A user logs into the system and wants to share some content.

Purpose	A user wants to share some data (pdf, ppt etc).
User	A legitimate user logged into the system
Input data	The file to be shared.
Output data	File ready to download by other users.
Pre-condition	User is Logged in; file exists on user's Computer.
Post-Condition	Any other person to whom the content was made available is able to download it
Basic flow	The user uploads a file to be shared using the upload box.The file then gets uploaded to the server and desired users are able to download it after logging in.

7.3 Operational Scenario 3 – Search Result

A user wants to search for a project using a keyword.

Purpose	A user wants to search for a particular keyword.
User	Any user of the Converge.
Input data	The keyword to be searched.
Output data	Search results.
Pre-condition	User is Logged in.

Post-Condition	Search results
Basic flow	User logs in, Enters the keyword in the search box, clicks the search button and gets the search results.

7.4 Operational Scenario 4 – Content Sharing (Download Files)

A user wants to download a particular file.

Purpose	A user wants download a particular file.
User	Any user of the Converge.
Input data	Request to download a particular file.
Output data	File is downloaded on user's device.
Pre-condition	User is Logged in and file exists.
Post-Condition	The user has downloaded the file successfully.
Basic flow	User logs in, selects the file which he wants to download. The file is then transferred from the server to the user's device.

8.PRELIMINARY USE CASE MODELS AND SEQUENCE DIAGRAMS

8.1 Use Case diagram

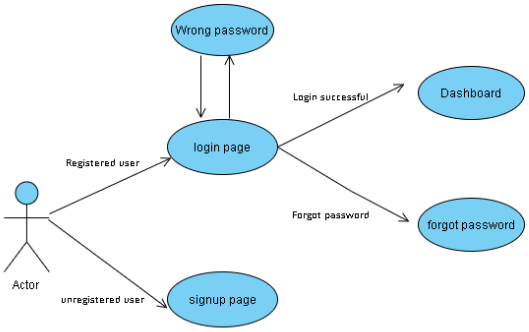


Figure 8.1.1 Use Case for User Login

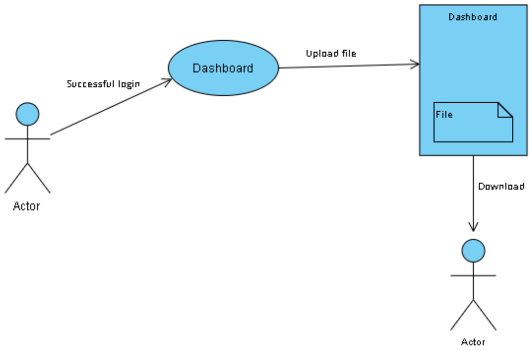


Figure 8.1.2 Use Case for file upload and download

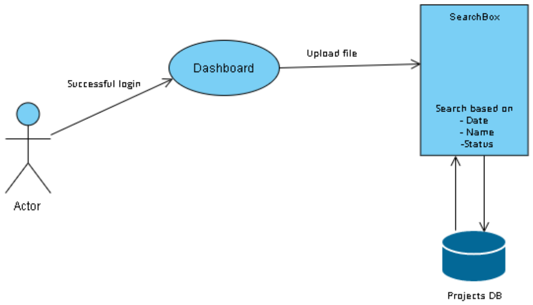


Figure 8.1.3 Use Case for searching

8.2 Sequence diagram

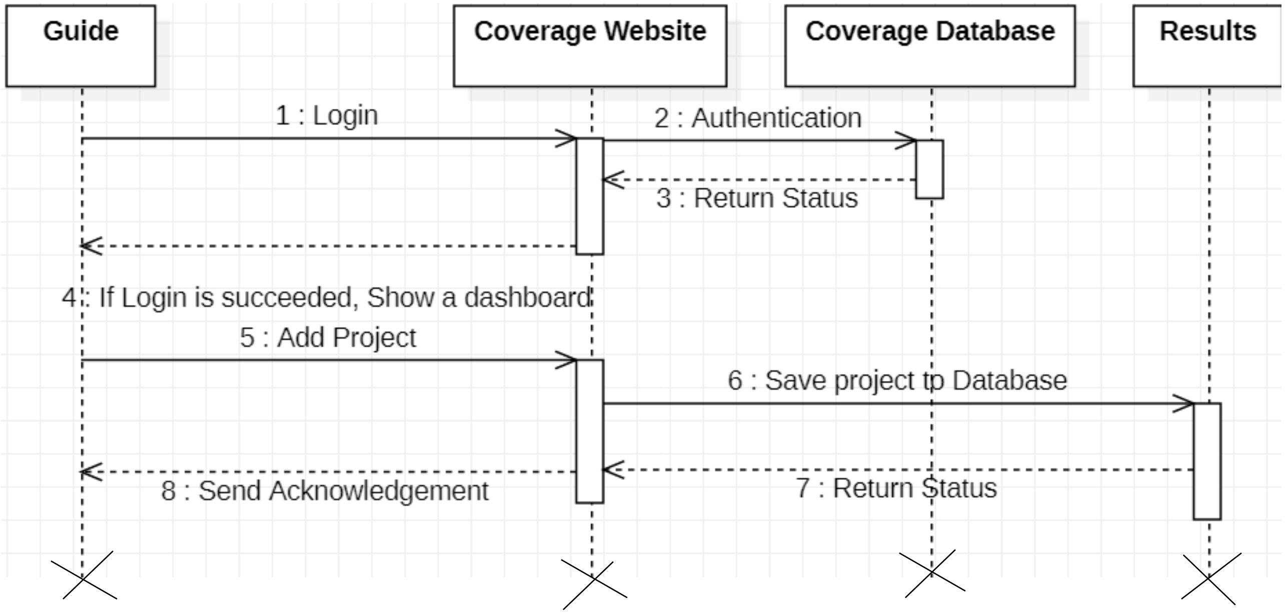


Figure 8.2.1 Sequence diagram for Guide

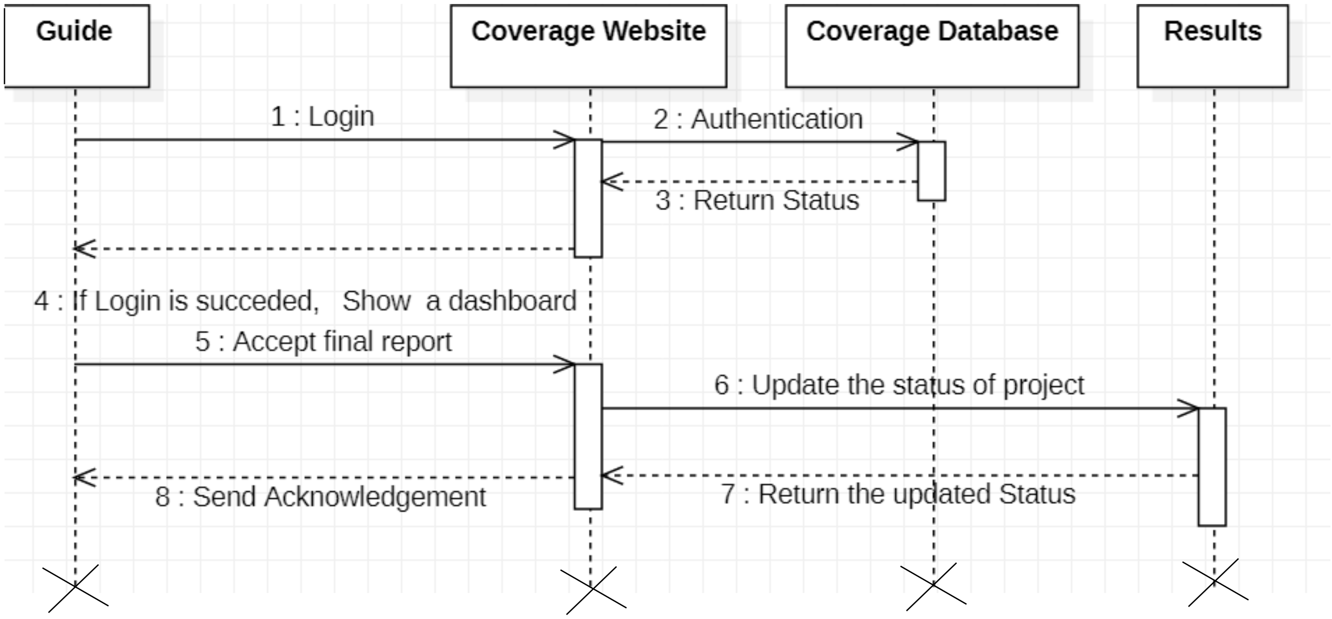


Figure 8.2.2 Sequence diagram for Student

9.DEFINATIONS, ACRONYMS AND ABBREVIATIONS

9.1 Converge

Converge is software meant to help facilitate the collaboration between students with their respective guides there by ensuring efficient utilization of the resources available.

9.2 Guide

Guide is a user who can be a professor who has the ability to manage the projects under his/her supervision.

9.3 Student

Student is a user who can create the projects under the supervision of a particular guide.