Software Requirements Specification

for

Course Organizer System (COS)

Version 1.0 approved

Prepared by Aayush Prasad
18CS30002
aayuprasad@gmail.com

CSE, IIT Kharagpur

12th February , 2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version

1. Introduction

1.1 Purpose

The purpose of this document is to show the detailed explanation of the objectives, features, user interface and application of Course Organizer 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. The need for a Course Organizer System arises due to:

- **Time Management**: Nowadays, there is less time available and many tasks to accomplish. Hence, students cannot afford to waste too much time on looking through lists of courses to find the course which interests them. Thus, it is essential to have a Course Recommender which displays the appropriate non-matching slot courses.
- **Inadequate knowledge of available courses**: Due to inadequate knowledge of the available courses, students become disinterested to take up a course which interests them. This software will help fill that void and thus will ensure students are able to pursue the courses which will help them in future.
- **Selecting projects**: Projects helps students achieve their career goals/research goals. This software will help professors upload the projects they supervise and will let students develop their research interests. This will tackle the intimidation involved when meeting a Professor. The students will gain confidence and will become actively involved in the betterment of the society.

1.2 Addressing the need

A few ways in which the needs can be addressed are:

- **Time Management**: The Course Organizer System will use a web-based interface where one must login using a username and password and then perform various actions including searching, getting the breadth preference list, etc. This is a very fast and smooth process.
- **Inadequate knowledge of available course**: The course webpage will contain all the information relevant to the course. It will also include the course reviews by the students. In this way, students will gain the appropriate knowledge of a course before applying for it.
- **Selecting projects**: The projects will be present in the Professor webpage as well as the Department web page. The web-based user interface will ensure a smooth and comfortable experience.

1.3 Prospective Users

The prospective users of the Course Organizer System are:

- Students
  * UG
  * DD
  * Research Scholar
- TAs
- Professors

1.4 Document Convention

The standard font used throughout the document is Arial, with font size 11. The titles of the various sections of this SRS document have been represented in bold, with font size 14 and font Times. Important parts of the document have been indicated in bold.

1.5 Intended Audience and Reading Suggestions

This document is intended to assist the users when they use the software and for developers and the project managers to plan their project and implement the software required. This Software Requirement Specification document is divided into five subsections.
Section 1: Introduction
Section 2: Overall Description of the Software giving information about functions, user classes, operating environment, constraints 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

1.6 Product Scope

This software will help students select permissible subjects which they prefer. It will analyze the basic timetable (consisting of Depth subjects) of the students and would suggest which subjects they should take (Breadth, Additional) according to user's choice. The order in which certain subjects need to be taken up will be provided considering prerequisite subjects. The software will also contain the database of subjects (segregated by the years they were taught in), departments, projects and the members of the institute (students, Professors). Users will be needed to log in to the software to use it. The goal of this software is to save students' precious time. Instead of wasting time searching through numerous subjects offered, the student can easily obtain relevant information. It also facilitates important 3-way communication between student, teacher and TA. There is scope for usage by various educational institutions.

1.7 References

IEEE.IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specification. IEEE Computer Society, 1998.
https://goo.gl/nsUFwy

2. Overall Description

2.1 Product Perspective

The Course Organizer System will help the students stay informed about the courses they're interested in. It will also help Professors and TAs upload relevant information on the software for the students to see. The system is designed to provide as a one-stop solution for all the Software ambiguities related to course information. Before the start of a semester, a student would like to

look through lists of courses and decide which course he/she can take up. While studying for a test, a student requires the notes and the syllabus for the test. After the course, a student might feel like helping other students who are planning to take that course by providing review of the course describing how to study for that course. All the aforementioned three needs can be met by the Course Organizer System. It will also let users search for the various departments, courses, professors in the Institute to facilitate time-efficient communication.

2.2 Product Functions

The following functions are provided by the software:

- Create Account
- Account Login
- Search Course
- Search Department
- Search Professor
- Search Project
- Display Additional Subjects
- Display Order of Completion of Courses (For Minor)
- Add Student Review for Course
- Show Course Information
- Show Course Statistics
- Display Breadth Preference List
- Add Notes

2.3 User Classes and Characteristics

The System has the following user classes:

- B. Tech Student
- B. Tech + M. Tech (Dual Degree) Student
- Teaching Assistant
- Professor
- Research Scholar
- Administrator

The student can create account and login to do various actions i.e. display breadth preference list, display additional courses, upload notes, write course review, etc. The TA can create account and in addition to performing students' actions, they can upload course information. The Professor can create account and login to perform various actions like upload course information, notes, statistics, add project, etc. The administrator will ensure that the accounts created are genuine. The administrator can be the Head of the Department.

2.4 Operating Environment

Operating System: Windows 7,8, 8.1, 10
Language: Java
Whole system should be covered by internet connection.

2.5 Issue, Challenges and Design and Implementation Constraints

The issues/challenges to overcome while developing the system are as follows:

- The security issue: This software is meant for students and the Professors of an institute. An invalid user might create his/her account which might be difficult to trace. One primitive

solution to this problem would be to provide a password to the required valid users which would be required at the time of sign up.
- Making the user interface simple and easy to use.
- Designing the web page to ensure easy, fast and yet efficient way of browsing through software. Thus, the page should be lightweight and the Data Base Management, sorting and other processes should be carried out using efficient algorithms.
- Data Scraping Problem: The software will be required to provide additional subjects to the user based on the non-matching slots. To solve this problem, data scraping concept would be required and appropriate websites need to be searched which contain the required data.

The constraints are:
- Cannot be implemented as an Android/IOS App
- Doesn't work on outdated/slow servers
- No other languages except English is used in this implementation
- Advanced security features haven't been introduced.

2.6 User Documentation

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

2.7 Assumptions and Dependencies

It is assumed that the accounts created are genuine. Proper internet connection is assumed.

2.8 Project Planning

Week 1 & 2: The concepts required for the project will be studied. The project requires a good knowledge of Front-End Development tools like HTML, CSS, JavaScript, ReactJS, etc and also a good working knowledge of Java and a Database Management.

Week 3: The back-end for the project will be designed using Java. Many actions like Log in, sign-up will require codes at the back-end to process correctly.

Week 4: The designing of the front-end will start. This step will determine how our app will look on visiting the site. Various user interaction buttons and links will have to be integrated in the Web App. The concept of GUI-based programming will be required.

Week 5: The very basic 1st version with minimal features will be completed so that testing and debugging can begin. The most essential functions will be tested before proceeding further.

Week 6 & 7: The app will be updated and more features will be added. Detailed debugging will be done and the app will be improved as much as it can be. Finally, when the app will have all the required features and will work satisfactorily, the app will be deployed either on a Cloud based platform or a Local Server.

3. External Interface Requirements

3.1 User Interfaces

Works with Windows User Interface and Java is used to create the GUI. The interface will be user friendly. Users can use the functions without any technical difficulties.
1) GUI along with meaningful Frames and buttons.
2) Reports are generated as per requirements.
3) User Registration
4) Adding new contacts/users.
5) Signing in by user.

3.2 Hardware Interfaces

1) Hardware environment - Dual Core 2nd generation
2) System Configuration - RAM - 512 MB HDD 80 GB
3) Operating System -Windows XP/Vista/7/8/10/Linux
4) Student/Employee Database

3.3 Software Interfaces

The Course Organizer System will interface with a Database Management System (DBMS) that stores the information necessary for the System to operate. The DBMS must be able to provide, on request data concerning the course list, users (and their passwords) and available requirements. Additionally, it should take and archive data provided to it by the system. This data will include records of all courses, projects, reviews, user data executed by the System. The DBMS must store all data such that it can be used for accounting, as well as to show the statistics.
1) Front End - JavaScript, HTML and CSS
2) Back End - Java
3) Random code generating software as a unique code is generated every time attendance is taken.
4) Time Tracking (Example - Time Trex) as code is available for a short amount of time.
5) Captcha for preventing machine/robot usage
6) Integrity Checker

3.4 Communications Interfaces

There are no special communication interfaces requirements.

4. System Features

This subsection presents the identified functional requirements for the Course Organizer System. Where possible, the requirements have been demarcated based on their relevance to the users of the system, that is, students, TAs, Research Scholars, Professors, Administrator.

4.1 SignUp and Login

4.1.1 Description

Takes the necessary input from the users and creates an account and enables login

4.1.2 Stimulus/Response Sequences

The user will be able to create account and then login into the system. While logging in, if the user enters an incorrect password/userID, he/she is allowed to try logging in for another 4 times after which the system will be locked.

4.1.3 Functional Requirements

Sign-Up:

- Takes as input name, user type, roll number (in case of Student user), email address and password and registers account.

Login:

- Asks information of roll number, user type (UG Student, DD Student, Research Scholar, Teaching Assistant, Professor) and password and logs in.

4.2 Search Course/Department/Professor/Project

4.1.1 Description

Takes the necessary input of the department from a drop-down list and another input for Project/Professor/Subject

4.1.2 Stimulus/Response Sequences

If an invalid Subject/Professor/Project name is entered no result will be shown

4.1.3 Functional Requirements

Search Course: Takes first alphabet input to get to the relevant section in the drop down list.

Search Department: Takes first alphabet input to get to the relevant section in the drop-down list.

Search Professor: Takes first alphabet input to get to the relevant section in the drop-down list.

Search Project: In the department section, projects can be searched by scrolling through the drop-down list or providing the first alphabet to get to the relevant section of the drop-down list.

4.3 Display Additional Subjects

4.1.1 Description

Takes the necessary input from the users and displays the permissible additional subjects on screen based on slot.

4.1.2 Stimulus/Response Sequences

The additional subjects will be shown in alphabetical order

4.1.3 Functional Requirements

Enter timetable slots: The user enters the course and the semester number as the input for the system to obtain the timetable of that semester.
Display Additional Subjects: Based on the inputted timetable, the additional subjects which are in the available slots are displayed.

4.4 Display order of completion of Minor subjects

4.1.1 Description

Decides the order of completion of subjects in a Minor curriculum so that minor is obtained using Topological Sort of the graph (edges denoting prerequisites and nodes denoting courses).

4.1.2 Stimulus/Response Sequences

If completion of minor is not possible, error message to change the minor curriculum will be displayed

4.1.3 Functional Requirements

Enter department: Asks for department as input from the user.
Display order: Displays all the possible topologically sorted lists of subjects for the user to see.

4.5 Add Student Review

4.1.1 Description

Adds Student's review of a course

4.1.2 Stimulus/Response Sequences

If input is more than 200 words or maximum 20000 characters error message will be displayed.

4.1.3 Functional Requirements

Add review: Takes input as review (max 200 words) and adds it to the review section of the subject.
Show review: Shows all the reviews of the selected course.

4.6 Show subject information/statistics

4.1.1 Description

Takes subject name as input from the user and displays the subject description and previous year grade statistics

4.1.2 Stimulus/Response Sequences

The subject will be selected from a drop-down menu

4.1.3 Functional Requirements

Enter subject: User enter the name of the subject
Show info: The relevant information to the subject is shown.

4.7 Display Breadth Preference List

4.1.1 Description

Displays the different breadth preference orders based on percentage of students' scoring EX grade, scoring EX/A grade and scoring EX/A/B grade.

4.1.2 Stimulus/Response Sequences

The list may contain slots which may clash with the slots of additional courses.

4.1.3 Functional Requirements

Enter answers: Take as input the semester number, department, and the course
Show Statistics: Shows the preference lists based on the percentage of students' scoring EX grade, scoring EX/A grade and scoring EX/A/B grade.

5. Tentative Software/Hardware Environment

5.1 Front-End Development

Mostly Front-End Development includes using HTML, CSS, JavaScript, J query, AngularJS (Not everything necessarily). Of all the software tools, the three most used together or separately are HTML, CSS and JavaScript which also will be used.

HTML - Hypertext Mark-up Language (HTML) is the standard mark-up language for creating web pages and web applications. With Cascading Style Sheets (CSS) and software JavaScript – it forms a triad of cornerstone technologies for the World Wide Web. HTML describes the structure of a web page semantically and originally included cues for the appearance of the document. HTML elements are the building blocks of HTML pages. With HTML constructs, images and other objects such as interactive forms may be embedded into the rendered page. HTML provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items. HTML elements are delineated by tags, written using angle brackets. HTML can embed programs written in a scripting language such as JavaScript, which affects the behaviour and content of web pages. Inclusion of CSS defines the look and layout of content. Thus, we will use HTML for the purpose of creating, structuring and appearance of web pages.

CSS - Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a mark-up language like HTML. CSS is designed to enable the separation of presentation and content, including layout, colours, and fonts. This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple web pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content. Separation of formatting and content also makes it feasible to present the same mark-up page in different styles for different rendering methods, such as on-screen, in print, by voice (via speech-based browser or screen reader), and on Braille-based tactile devices. CSS also has rules for alternate formatting if the content is accessed on a mobile device.

JavaScript - JavaScript is a high-level, interpreted programming language that conforms to the ECMAScript specification. It is a language that is also characterized as dynamic, weakly typed, prototype-based and multi-paradigm. JavaScript enables interactive web pages and thus is an

essential part of web applications. The vast majority of websites use it. As a multi-paradigm language, JavaScript supports event-driven, functional, and imperative (including object-oriented and prototype-based) programming styles. It has APIs for working with text, arrays, dates, regular expressions, and the DOM, but the language itself does not include any I/O, such as networking, storage, or graphics facilities, relying for these upon the host environment in which it is embedded. Initially only implemented client-side in web browsers, JavaScript engines are now embedded in many other types of host software, including server-side in web servers and databases, and in non-web programs such as word processors and PDF software, and in runtime environments that make JavaScript available for writing mobile and desktop applications, including desktop widgets. Although there are similarities between JavaScript and Java, including language name, syntax, and respective standard libraries, the two languages are distinct and differ greatly in design; JavaScript was influenced by programming languages such as Self and Scheme.

5.2 Back-End Development

Back End Development mostly includes PHP, Java or any other server side scripting languages. For this project, Java is the language which will be used.
Java - Java is a general-purpose computer-programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that compiled Java code can run on all platforms that support Java without the need for recompilation. Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. The language derives much of its original features from SmallTalk, with a syntax similar to C and C++, but it has fewer low-level facilities than either of them.

5.3 Database Management System

MySQL - MySQL is an open source relational database management system (RDBMS). MySQL is a component of the LAMP web application software stack (and others), which is an acronym for Linux, Apache, MySQL, Perl/PHP/Python. MySQL is used by many database driven web applications, including Drupal, Joomla, phpBB, and WordPress. MySQL is also used by many popular websites, including Google (though not for searches), Facebook, Twitter, Flickr, and YouTube.

6. Non-functional Requirements

6.1 Performance Requirements

- The product will be based on local server.
- The product will take initial load time.
- The performance will depend upon hardware components
- Different databases are required

6.2 Safety and Security Requirements

- The whole system is secured. Account Creation is verified by the Admin.

- The system will use suitable transfer protocols.

6.3 Software Quality Attributes

- **Maintainability :** Different versions of the product should be easy to maintain. For development it should be easy to add code to existing system, should be easy to upgrade for new features and new technologies time to time. Maintenance should be cost effective and easy.
- **Usability :** This can be measured in terms of ease of use. Application should be user friendly. Should be easy to learn. Navigation should be simple.
- **Flexibility :** Should be flexible enough to modify. Adaptable to other products with which it needs interaction. Should be easy to interface with other standard 3rd party components.

6.4 Business Rules

This software should be use only after proper contract agreement with the company developing it. In case of any problems the user should immediately contact the project manager. The software should not be outsourced to any third party without prior permission.

6.5 Publicizing the system

- Reach out to influences through YouTube, Instagram, and other social media platforms and let their audience know about our project.
- Sharing on social media
  - Invest in FB Ads
  - Create a Facebook Group and add relevant people to it.
  - Join Relevant LinkedIn group
- Email Marketing
  - Facebook, Reddit, Whatsapp, Instagram, LinkedIn, etc
  - Send personal Email to an individual that introduce your idea and delivers your Call To Action.
  - We can use softwares like MailChimp for free email listing, creating a newsletterand then sending it out to people.

6.6 Cost of the product

The product will be freely available initially to gain popularity. Then we can earn through ads or make it available to established institutions or individuals at a reasonable price.

6.7 Other Requirements

- Licensing requirements: Applicable
- Legal, Copyright and Other notices: All rights reserved by our organization.
- Applicable Standards: It should be as per the industry standards

Appendix A: Glossary

1. Non-matching slots: Slots except the slots of the depth courses in a timetable.

Software Design Document

for

Course Organization System

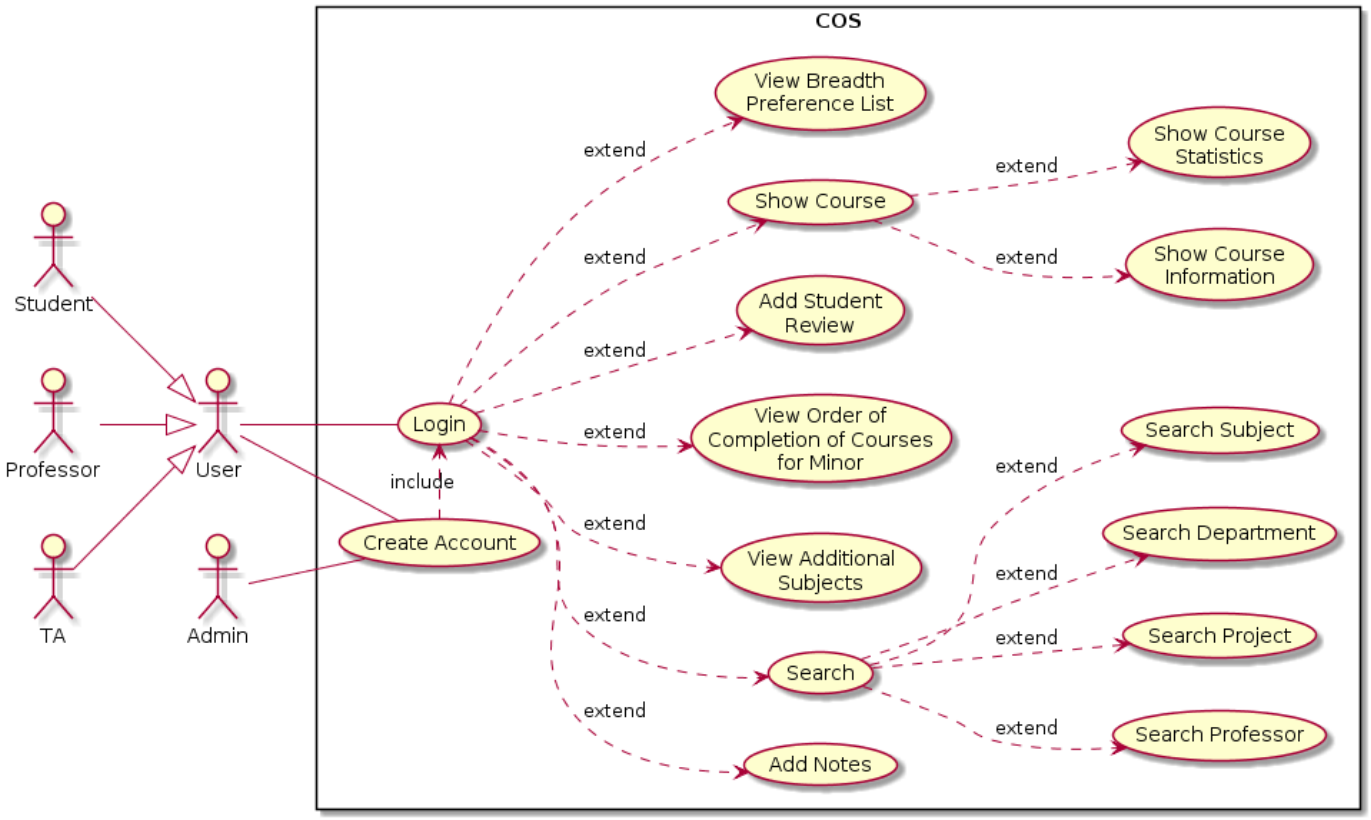
Version 1.0 approved

Prepared by Aayush Prasad
18CS30002
aayuprasad@gmail.com

CSE, IIT Kharagpur

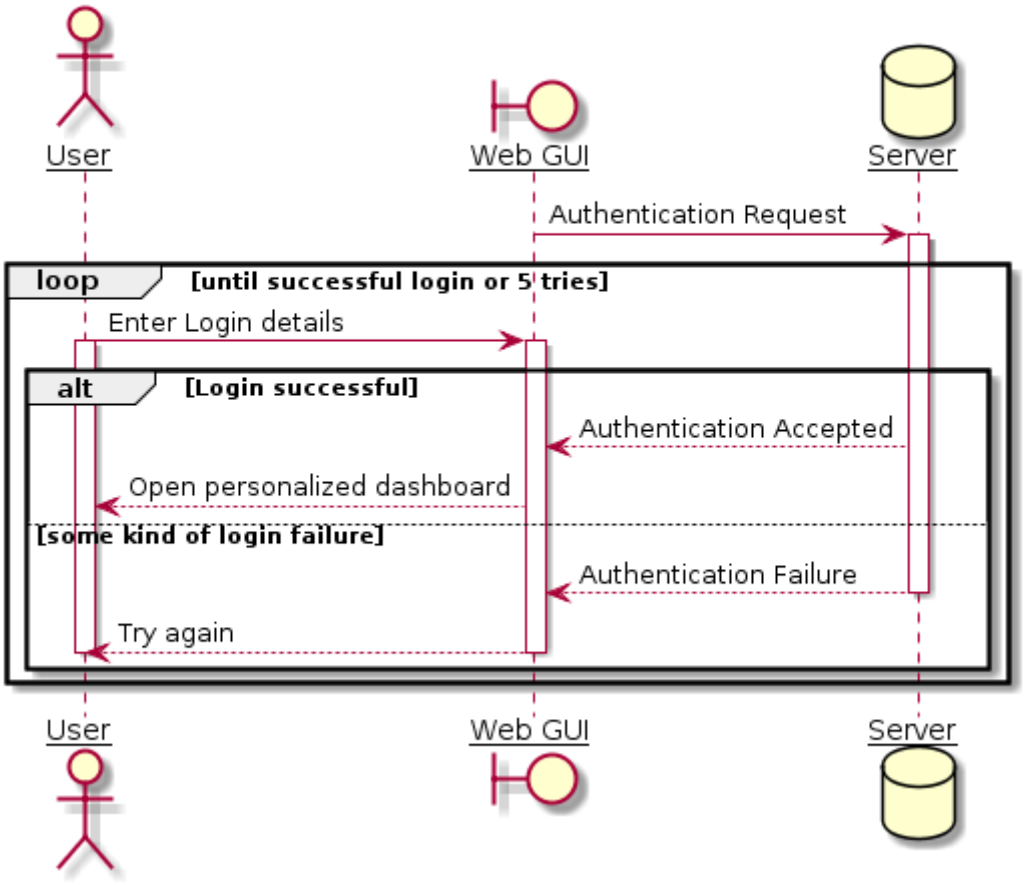
26th February, 2020

Use Case Diagram

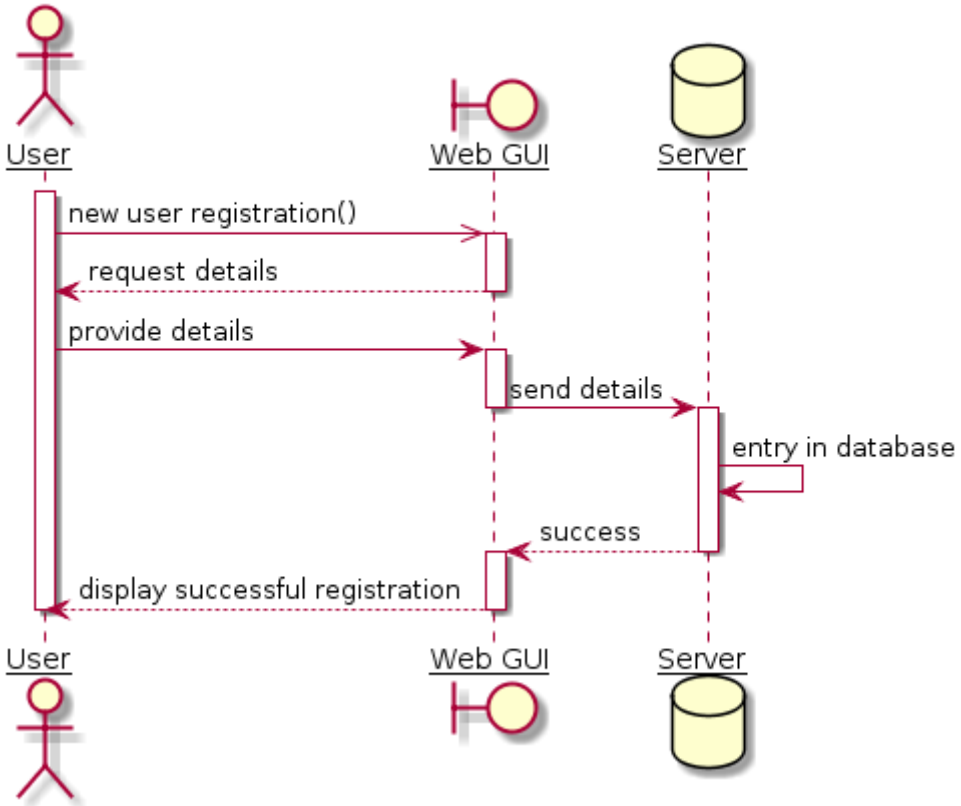


Sequence Diagrams

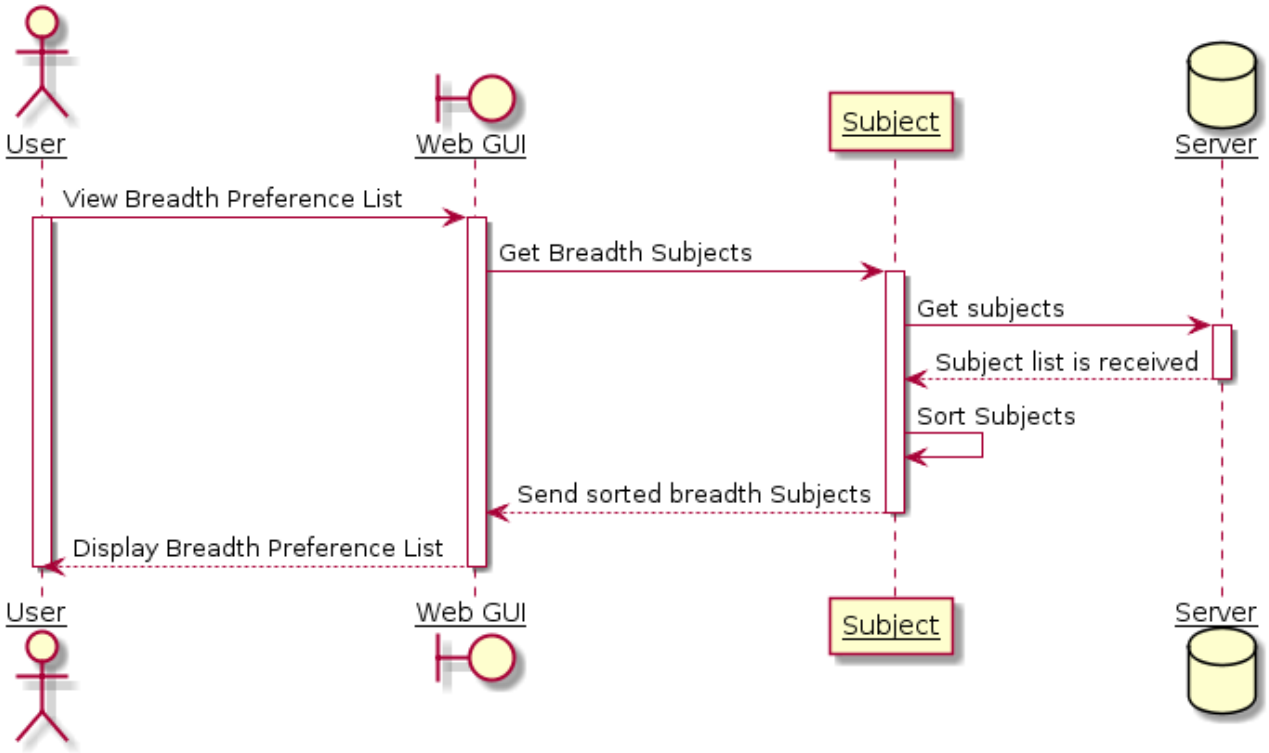
Login - Sequence Diagram



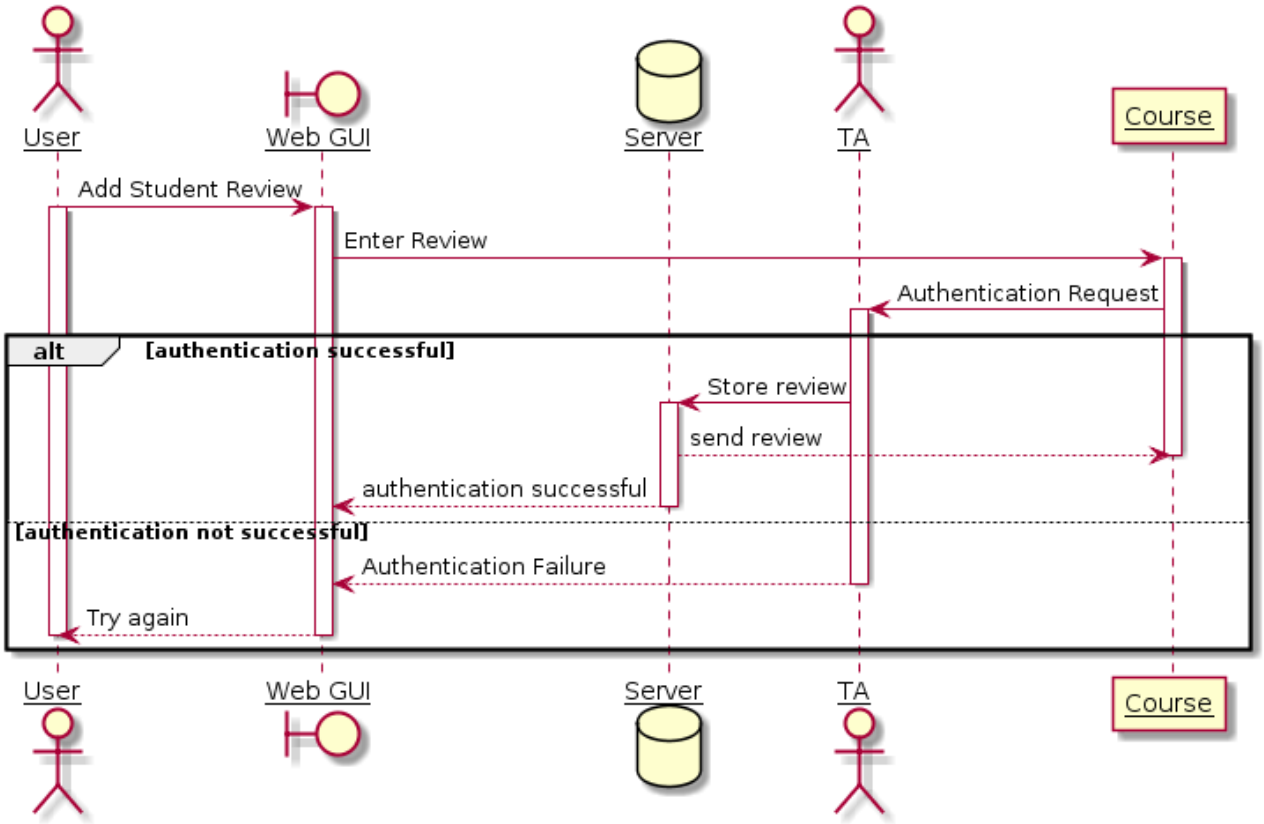
Create Account - Sequence Diagram

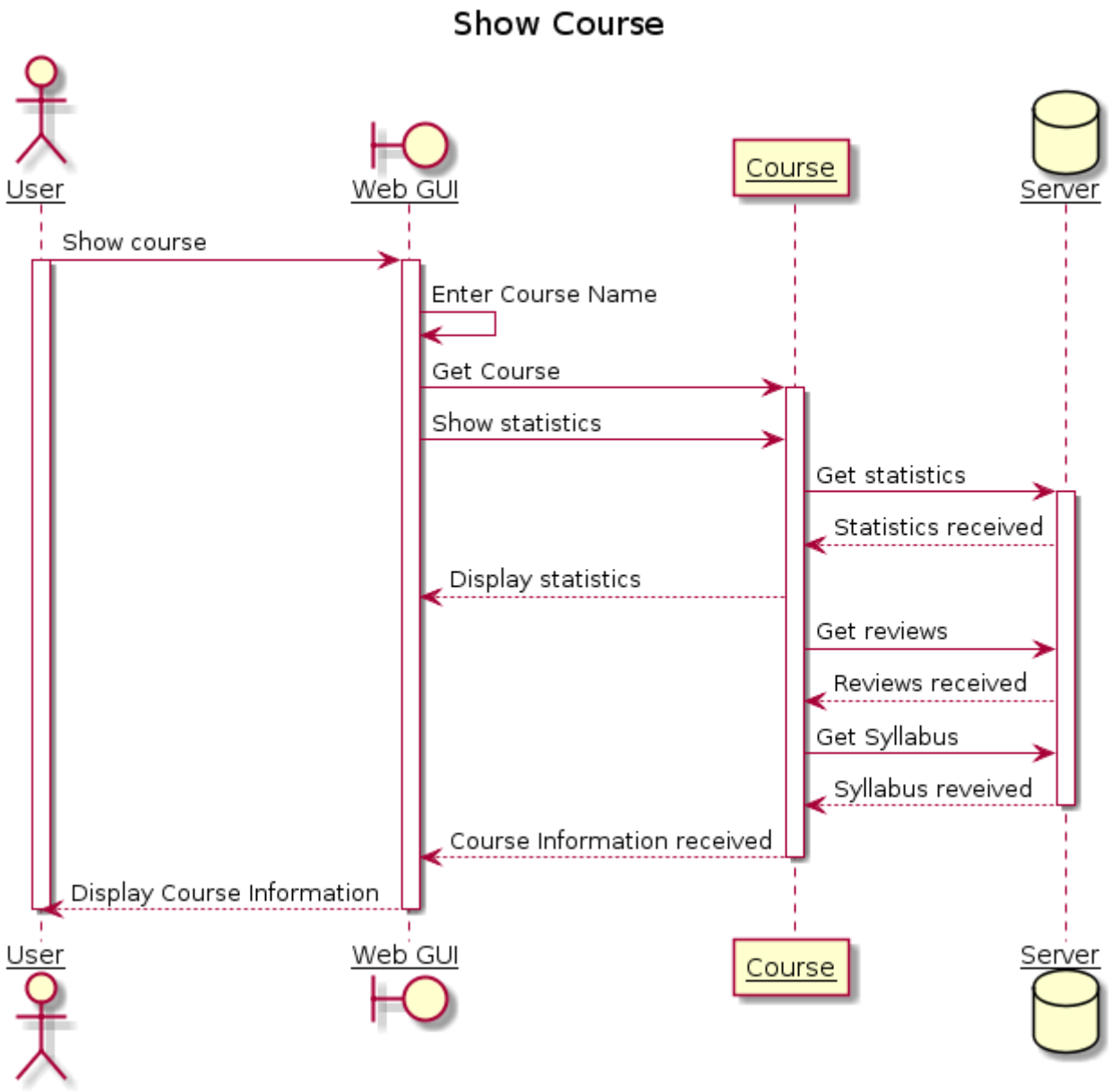


View Breadth Preference List - Sequence Diagram

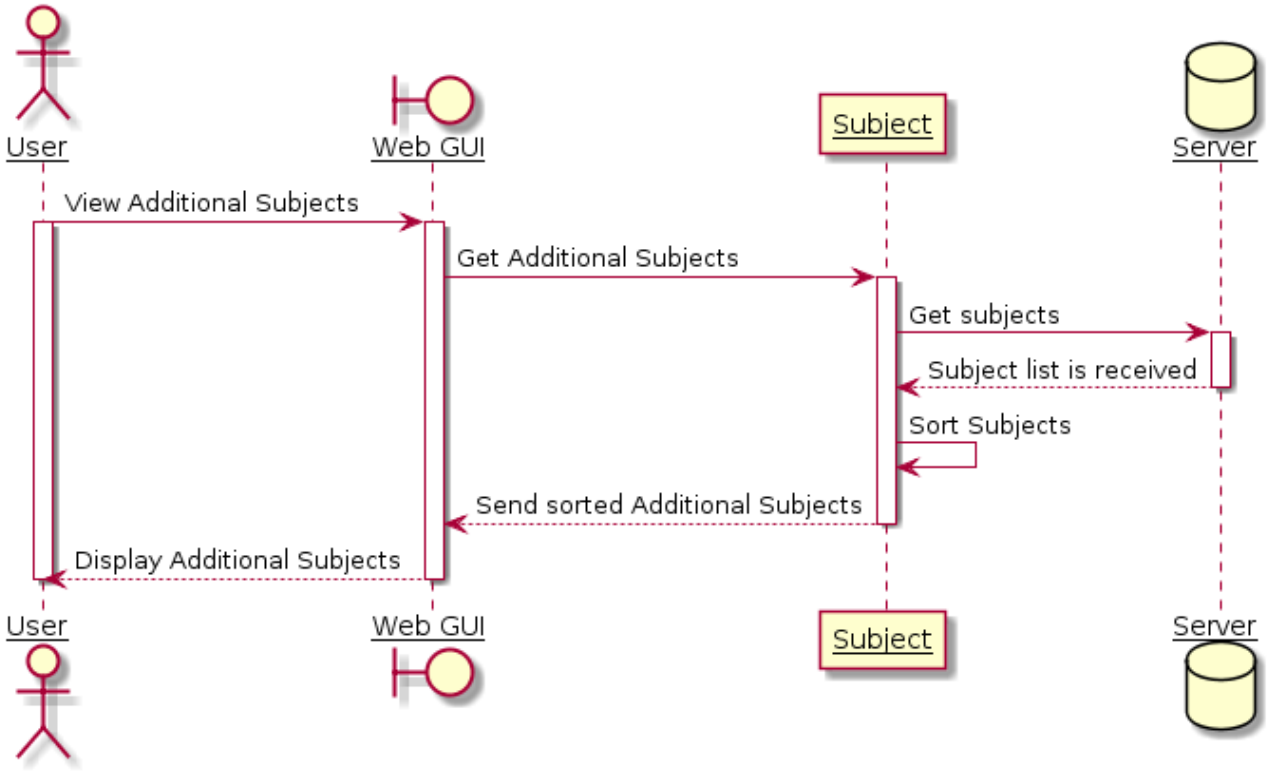


Add Review

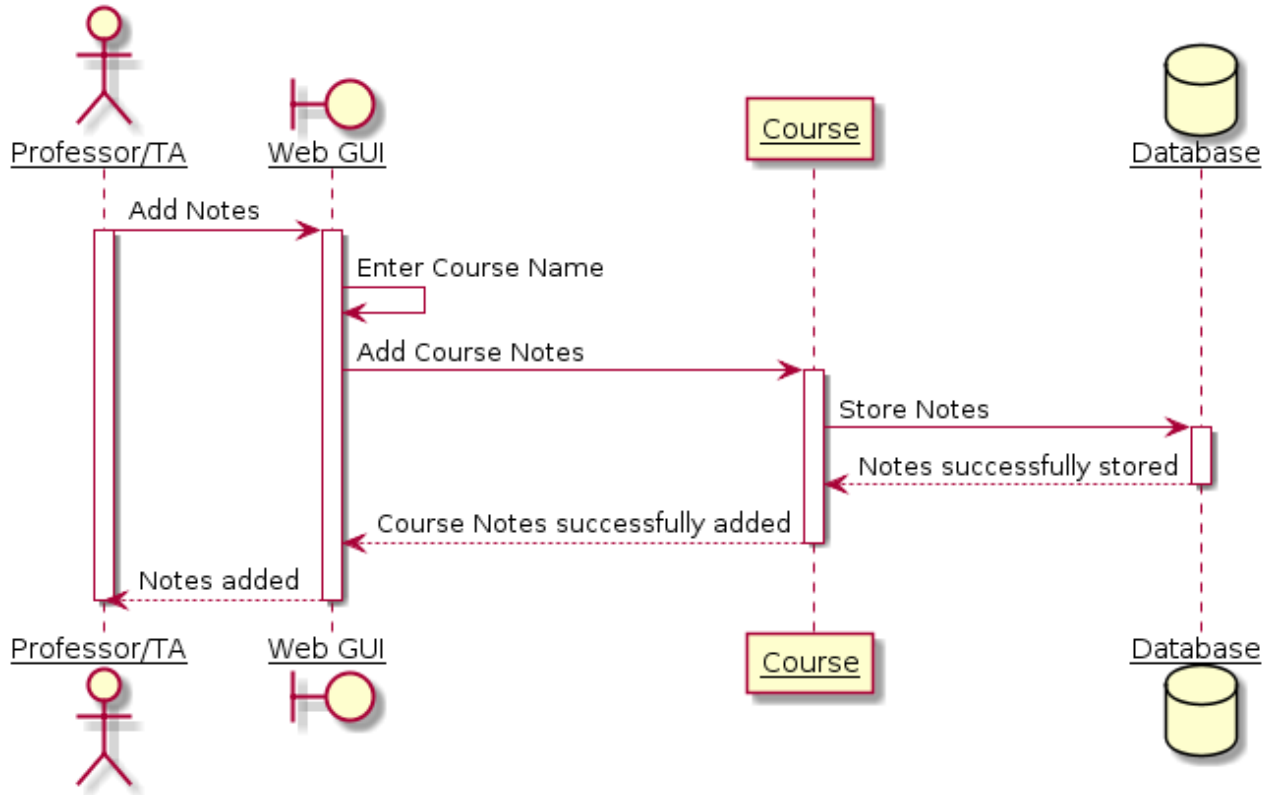


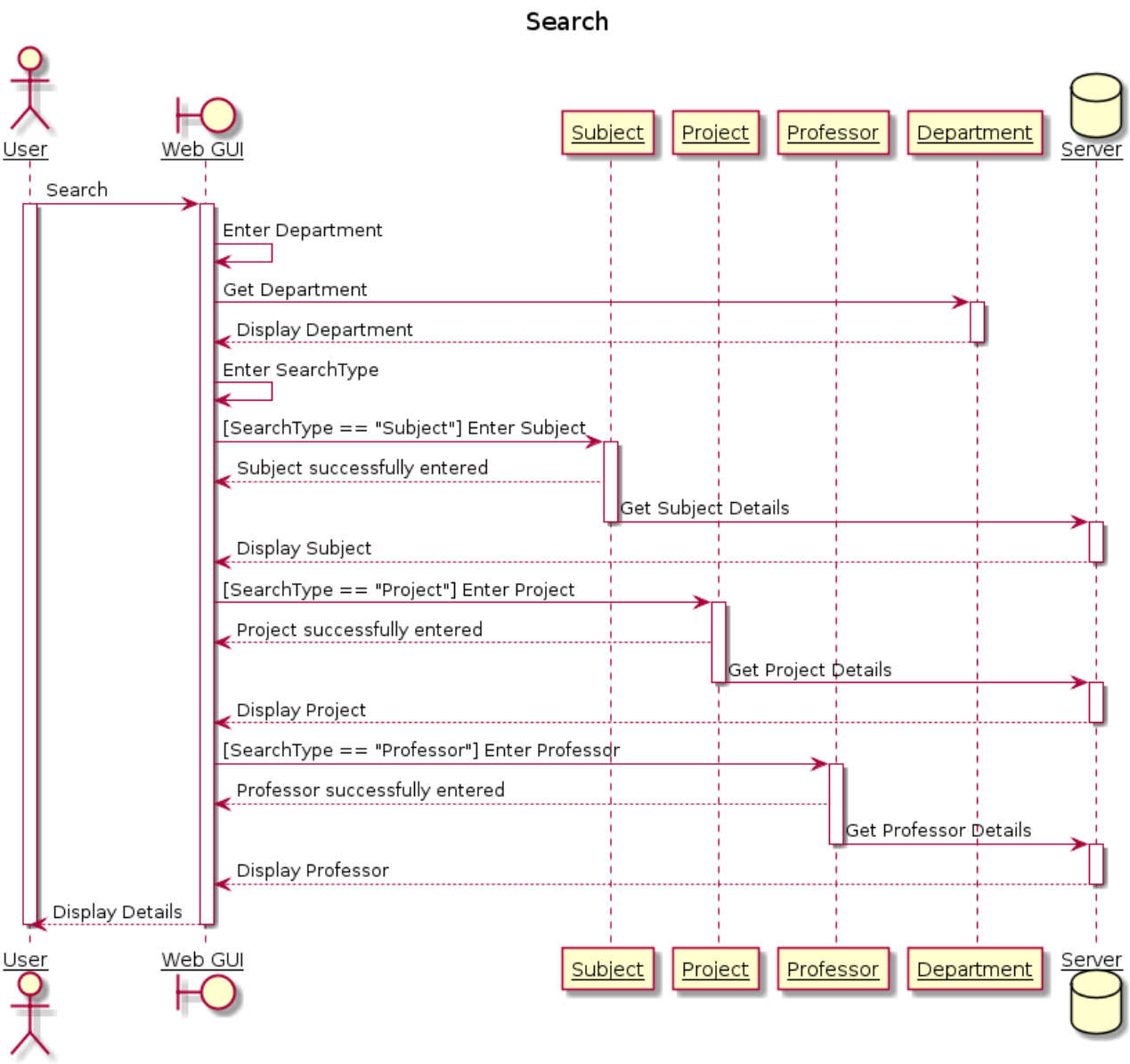

Show Course
User
Web GUI
Course
Server
Show course
Enter Course Name
Get Course
Show statistics
Get statistics
Statistics received
Display statistics
Get reviews
Reviews received
Get Syllabus
Syllabus reveived
Course Information received
Display Course Information
User
Web GUI
Course
Server

View Additional Subjects - Sequence Diagram



Add Notes




Search
User
Web GUI
Subject
Project
Professor
Department
Server
Search
Enter Department
Get Department
Display Department
Enter SearchType
[SearchType == "Subject"] Enter Subject
Subject successfully entered
Get Subject Details
Display Subject
[SearchType == "Project"] Enter Project
Project successfully entered
Get Project Details
Display Project
[SearchType == "Professor"] Enter Professor
Professor successfully entered
Get Professor Details
Display Professor
Display Details
User
Web GUI
Subject
Project
Professor
Department
Server

Class Diagram

Visibility Icons

Character	Icon for field	Icon for method	Visibility
-	□	■	private
#	◇	◆	protected
~	△	▲	package private
+	○	●	public