Software Testing Document
(CS360)

Stutor

Group Number: 8

Hassaan Hasan
Muhammad Zain Qasmi
Nawal Khurram
Abreeza Saleem
Ali Ahsan

Course: Software Engineering CS360

Instructor: Suleman Shahid

University: Lahore University of Management Sciences (LUMS

Version: 1.0
Date: (10/05/2017)
Number of hours spent on this document: 30

1 Change Log

1.1 Project Scope

Our project scope is same as that proposed in the SRS document. There have been no major changes or deviations from the initial objectives.

1.2 Change log

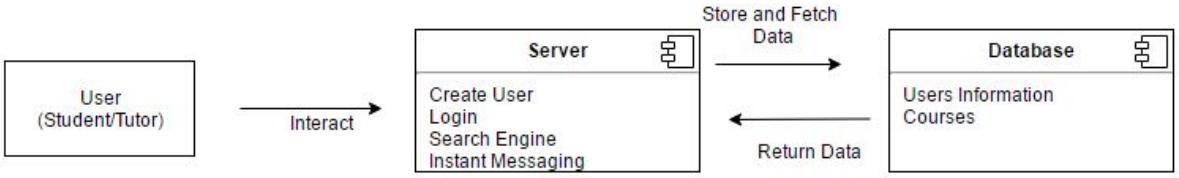
- Removed the option to login with Facebook.
- Able to extract user's location but not able to show nearby tutors.

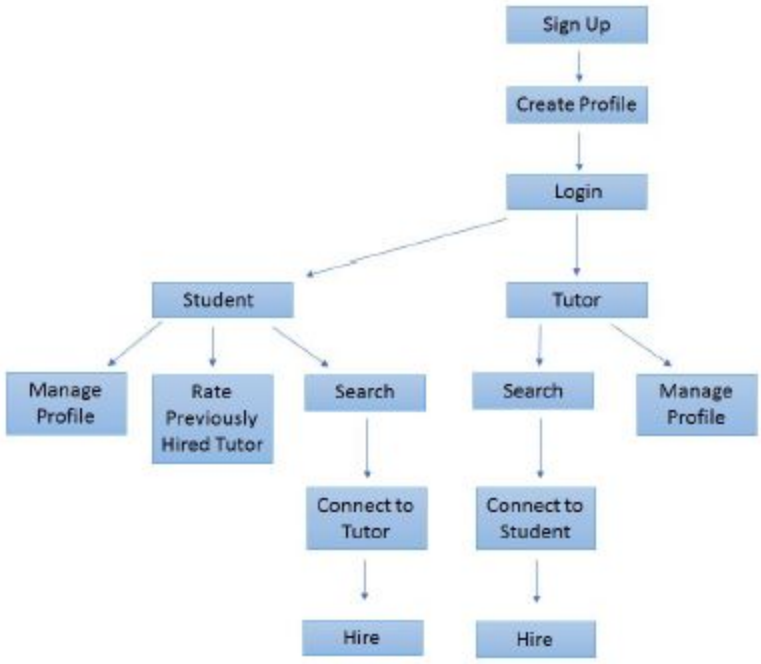
2 System Architecture

2.1 System overview

The user will connect with our server using a browser via HTTP connection. The system involves user belonging to two different categories : Students and Tutors. They will either register an account on Stutor or sign in using an email address. A signed in student will find nearby teachers using the Google Geolocation API for the course of their interest. They can then message the tutor and the rest is up to them to set up teaching. For teachers similarly, students interested in the teacher's course will be displayed. The follow up is the same as the students.

2.2 System Architecture





Server:
All the components are linked to this. We will be use NodeJS for building our application. The server is used to retrieve and update data in the database. The user interaction with the website sends requests to the server which handles them. It responds by sending the appropriate data back. It also handles the search queries.

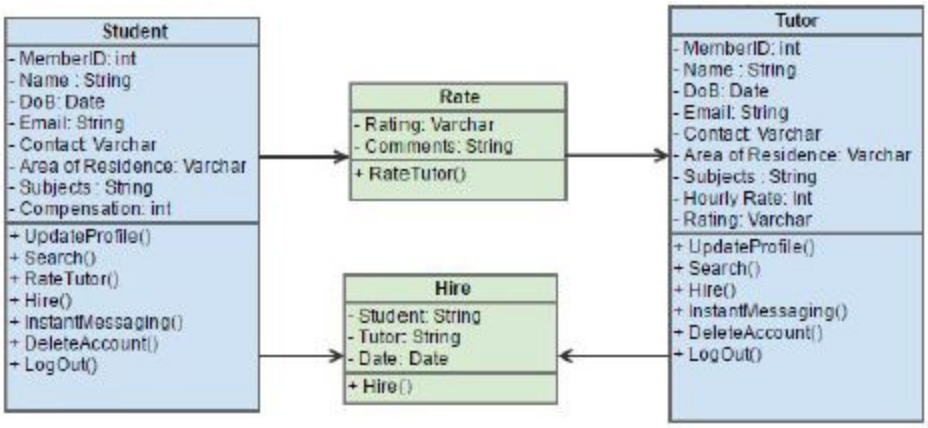
Database:
This will store all the student and tutor records. It will also be used to retrieve data based on the search queries. MongoDB will be used for building the database.

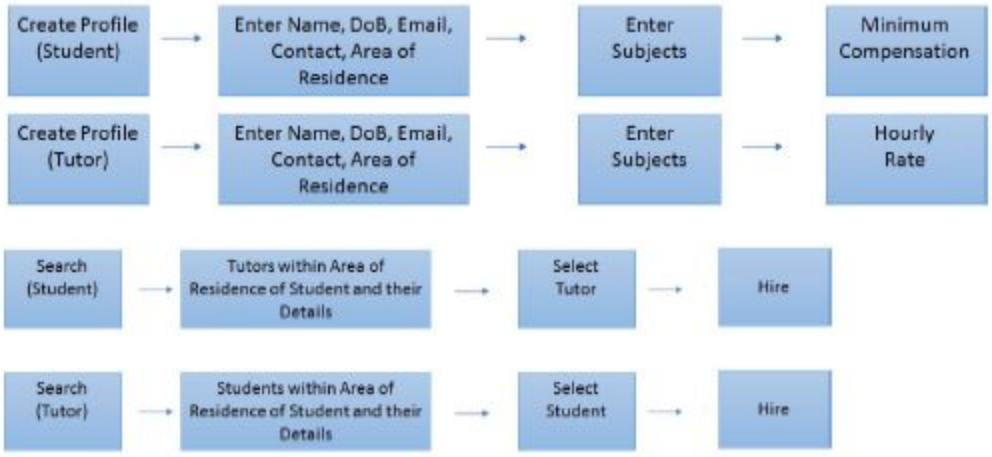
Search Engine:
This will be built using the search algorithm we have developed. It will be used for finding matches for students and tutors based on the various parameters and will output the result. This is the main functionality of our project.

User Interface:
The UI provides interaction between the user and the server. It is built using HTML and CSS.

Login and Sign Up:
We will design our own sign up system to implement this. This will be used to begin access to the application. It is a subcomponent of the server.

2.3 Subsystem Architecture





3. Final User Interface Design

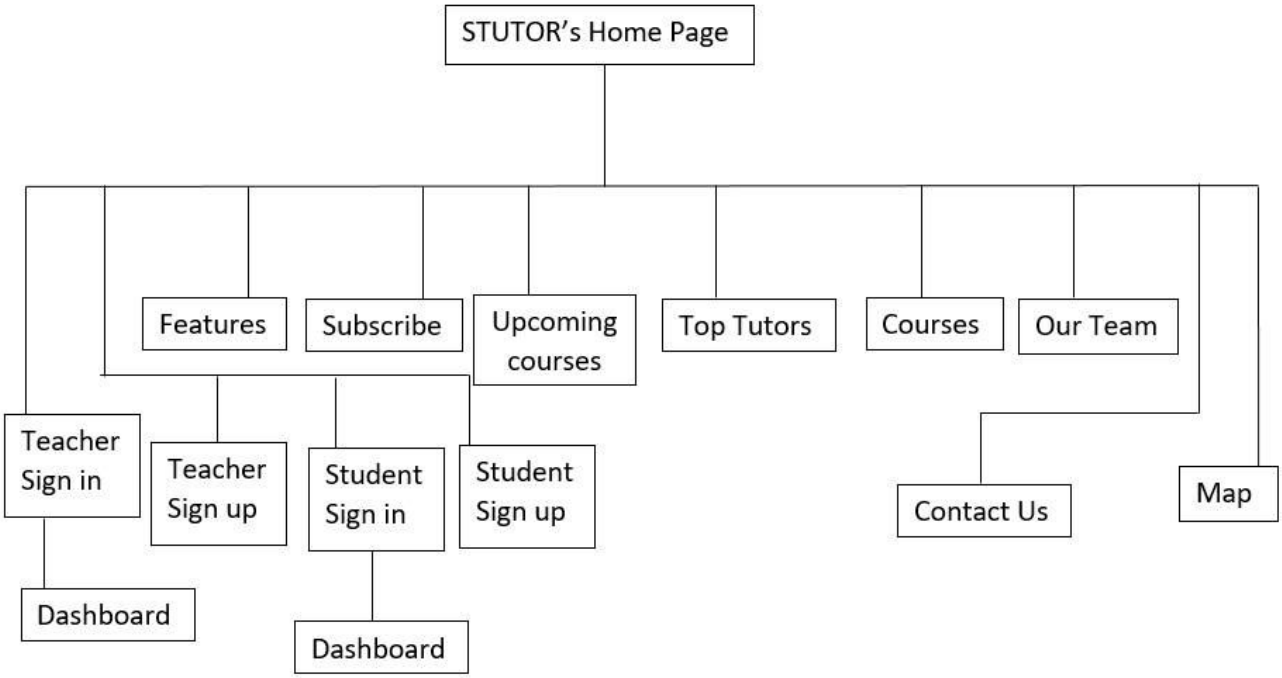
3.1 Final System Architecture

We used MVC architecture to implement our application, models providing the data base connectivity, Views provide the user interface and controller connects the screens(front end) with the database(back end). This provided our team a lot of flexibility, two of us (Zain and Hassaan) worked on the user screens without really worrying too much about the implications it could have on the backend. The backend team(Abreeza, Nawal and Ali) worked seamlessly on the server, followed the relationship schema and the proposed screens and so were not constrained by the progress at front end. This is the advantage MVC provided.

A Bootstrap template is used to provide a theme to our web app, it also provides better mobile responsiveness. HTML provides the basic static pages at the front end while CSS was used for user friendly presentation.

The MEAN Stack was used to implement the whole system. MongoDB to store and work with user data, NoSQL database such as MongoDB provides ease to the developers since the data revolves around JSON like objects (BSON) which are easy to interpret and work around with. AngularJS was selected due to its flexibility. Working at the front end i.e HTML pages went smoother. The server was programmed in JavaScript using the Node.js framework. Node is tailor made for event driven programming which suited our application needs. Express library from the npm manager was used for routing mechanisms inside Nodejs. It was used to handle get and post requests from the client.

3.2 Final Information architecture

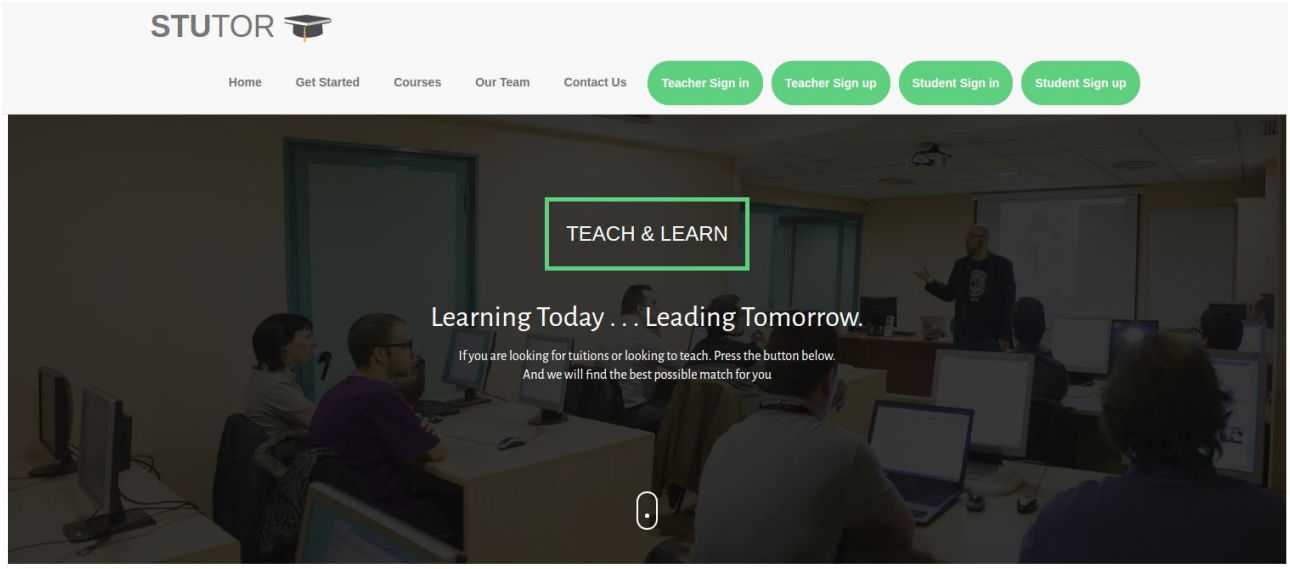


The user can directly access the following from the homepage:

- Features
- Subscribe
- Upcoming courses
- Top tutors
- Courses
- Our team
- Contact Us
- Map

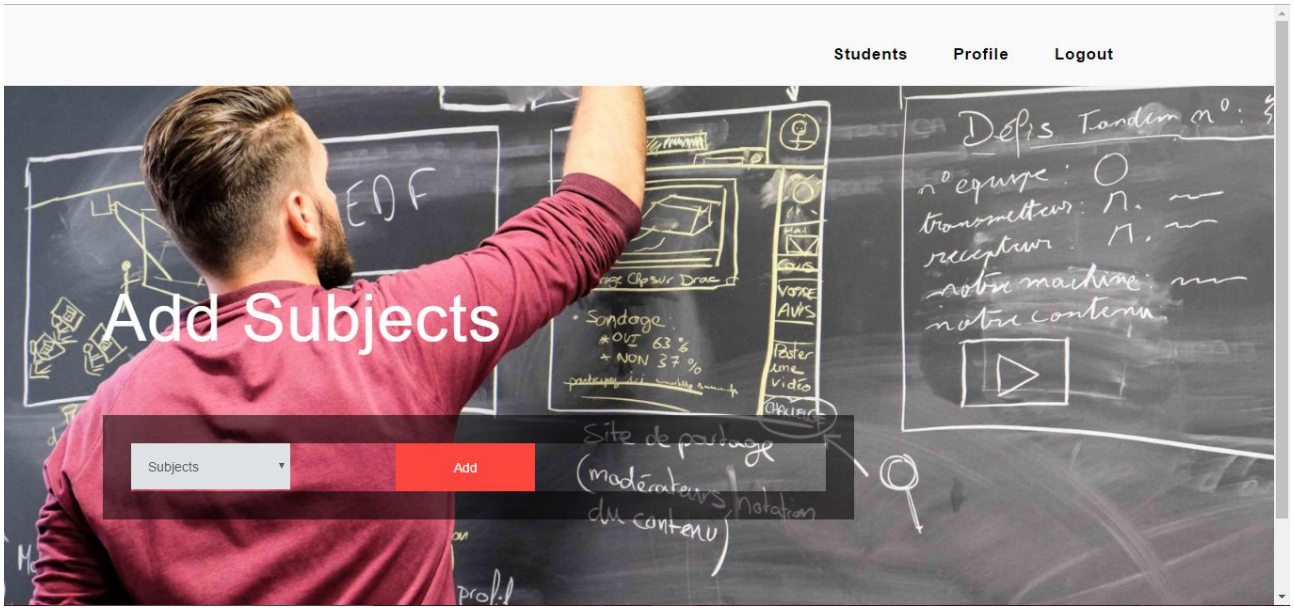
The user can also sign up and sign in from the homepage. He/she will be asked for the necessary credentials before proceeding. Then, if the user is a tutor, he/she will be directed to a page asking him/her to add the courses he/she wants to teach. If the user is a student, he/she will be directed to a search page for courses. Relevant tutors will be listed.

3.3 Final Screens



Main Page

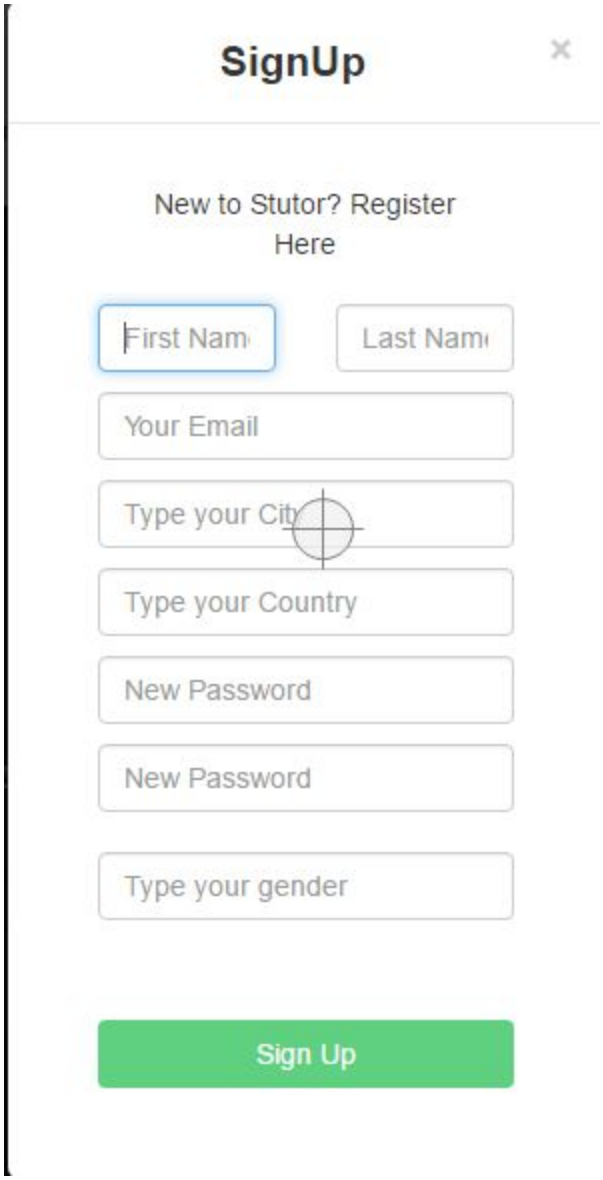
- The navigation bar at the top, route the user to several distinct functionalities. From the sign up pages for both teacher and student to sign in pages. Also the buttons for Get Started, Courses, Our Team and Contact Us are present.



Teacher's Page

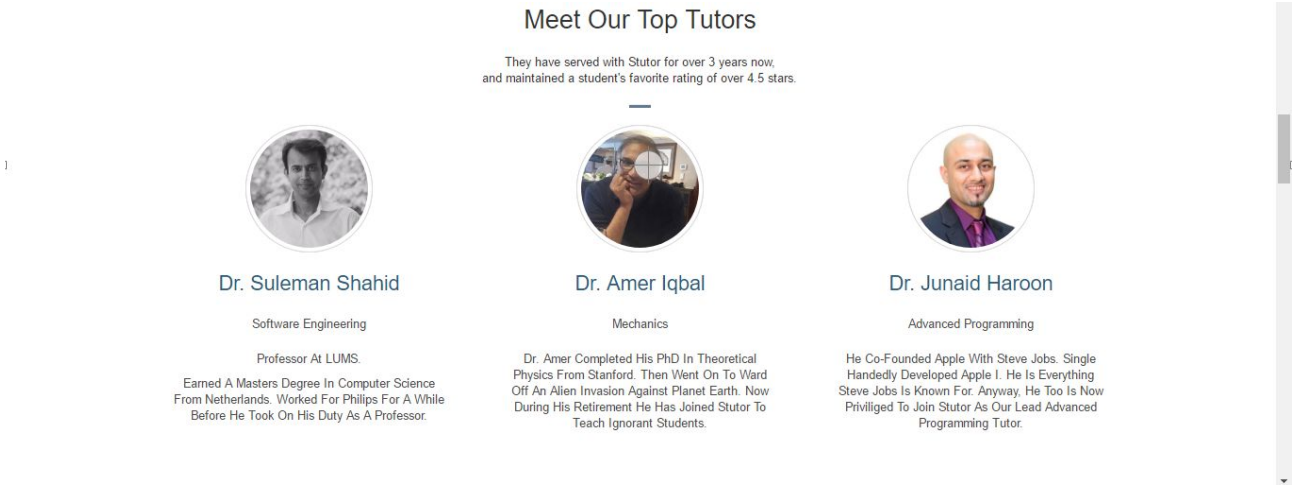
- The three buttons at the navigation bar provides routing to webpages of student records, teacher profile and log out from app respectively.

- Tutor can add subjects he/she is interested in teaching using the drop down list shown over the image. Add button will add the course selected to the teacher records.



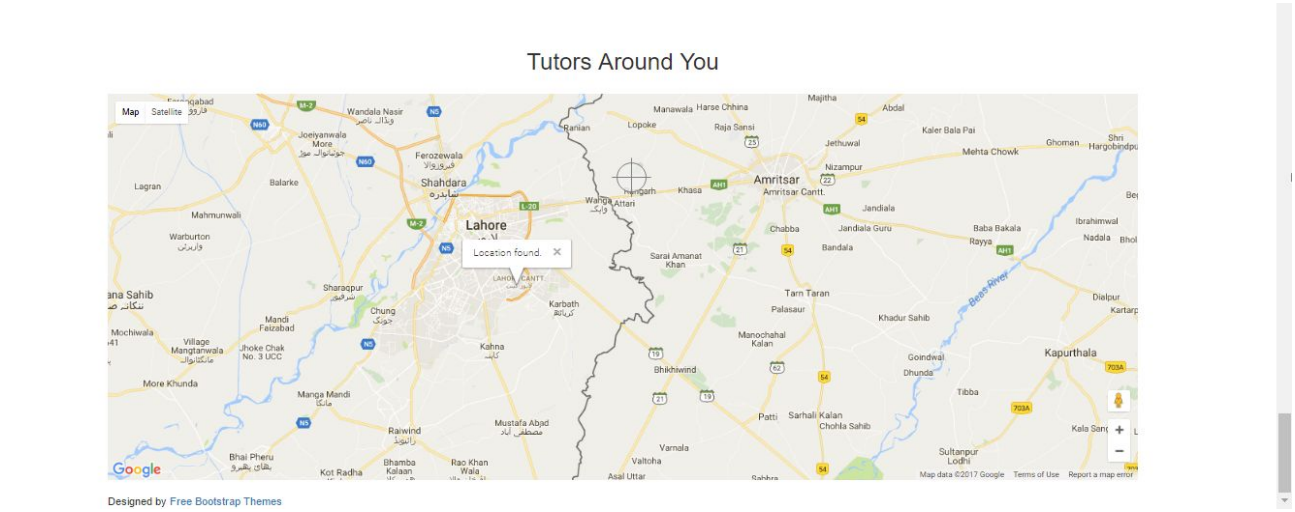
Signup Page:

- The sign up page is the same for tutors and students.
- 8 text boxes with labels are provided to aid user in entering data.
- Sign up button sends the data to the serve.



Trending Tutors

- Our main page also shows the top rated tutors. We wanted to incorporate the rating feature in our app but we failed to do so due to lack of time.



Map

- Map added to show the location of the user, given the user permits us to use their location.
- The idea was to show the nearby tutors to students or students to tutors, depending on who logged in. Couldn't implement it in the limited time we had.

4. Test Case Specifications

Test case ID	*Test Objective*	*Precondition*	*Steps:*	*Test data*	*Expected result*	*Post-condition*
1	*Successful Employee login to Stutor portal*	*1. A valid User account to login to be available* *2. Stutor site is launched on a compatible http2 browser.*	*1. In the login Panel, enter the username*	*"A valid username"*	*The user is logged in successfully.*	*User is redirected to his profile home page.*
2	*Error message on unsuccessful Employee login to Stutor Portal*	*1. A valid Stutor Username to login to be available* *2. Stutor site is launched on a compatible browser*	*1. In the login Panel, enter the username* *2. Enter the Password for the Stutor User account in the password field* *3. Click "Login" button*	*"A valid username"* *"A invalid Password"*	*An Error message is displayed and the user is not logged in to the Stutor portal. "<Login Failed>"*	*User redirected to a fresh login page*
3	*Find Tutors*	*1. Select Course*	*1. Goto search page by clicking on "Getting*	*"A valid course"*	*Query successfully executed.*	*User redirected to a search result page with all the Tutors teaching the specific*

			Started"			*course in given location.*
		2. Select Location	*2. In the search bar enter your course.* *3. In the second search bar enter your location*	*"A valid locatio n"*		
4	*Edit Student and Tutor profile*	*1. User should exist in the database*	*1. Goto your profile update page* *2. Select Edit Profile*	*"Valid attribu tes should be filled with valid data type"*	*Error in case of invalid data types such as using string in Date.*	*Profile updated successfully in the databases.*
5	*Successfull y logout user*	*1. User should be logged in*	*1. Click on Logout button*	-	*The user is logged in successfully.*	*User is redirected to the main page of the website.*

5. Results and discussion

5.1 Future work

We haven't tested how scalable our product is. Maximum no. of users tested after deployment was ten i.e. five tutors and five students. This limitation also arose from the fact we were using free version of Heroku.

Geo-location system works on the website where the user is shown his/her location. Future work includes implement the location based recommendations where the student will be shown potential tutors close to him/her on the map.

5.2 Discussion

Our project allows students and tutors to sign up and create their profiles. It also allows students to search or courses in the city they want to find a tutor in. We were not able to implement the geo-location feature where a student is shown tutors in close proximity to him/her on the map. Chat box system for students and tutors once they choose each other is not implemented. HO

Appendix A – Phase Contribution Statement

Name	*Contributions in this phase*	*Approx. Number of hours*	*Remarks*
Abreeza Saleem	*Change Log and Test Case Specifications*	*18*	
Ali Ahsan	*Final UI Design and System Architecture*	*19*	
Zain Qasmi	*Test Case Specifications*	*20*	
Nawal Khurram	*Final UI Design and Results and discussion*	*20*	
Hassaan Hasan	*Test Case Specifications*	*18*	

Appendix B – Project Contribution Statement

Name	***Project phase***	***Contributed to***	***Approx. Number of hours***	***Remarks***
Abreeza Saleem	*Proposal*	*Project Organization*	*5*	
	SRS Document	*Introduction and Definitions*	*7*	
	SDS Document	*Worked on Section 4.4*	*10*	
	Actual Software Development	*Back End (Creation of Database and Linking)*	*49*	
	Testing document	*Change Log and Test Case Specifications*	*18*	
	Tool presentation	*Introduction and Features in Manual*	*8*	
	Final presentation	*Introduction and Problem*	*4*	

Name	***Project phase***	***Contributed to***	***Approx. Number of hours***	***Remarks***
Ali Ahsan	*Proposal*	*Schedule and Budget Summary*	*5*	
	SRS Document	*Non-functional requirements*	*7*	
	SDS Document	*Worked on Section 3*	*10*	
	Actual Software Development	*Back End (Database and Routing)*	*50*	
	Testing document	*Final UI Design and System Architecture*	*19*	
	Tool presentation	*Conclusion and How to get in Manual*	*9*	
	Final presentation	*Proposed Solution*	*3*	

Name	***Project phase***	***Contributed to***	***Approx. Number of hours***	***Remarks***
Nawal Khurram	*Proposal*	*Project description and Purpose, Scope and Objectives*	*7*	
	SRS Document	*Functional Requirements and External Interface*	*7*	
	SDS Document	*Worked on Sections 2, 4.1,4.2,4.5*	*10*	
	Actual Software Development	*Back End (Algorithm and Linking) and Front End*	*51*	
	Testing document	*Final UI Design and Results and discussion*	*20*	
	Tool presentation	*Usage and Demo, Use Cases in Manual*	*10*	
	Final presentation	*System Architecture and Tools and Technologies*	*4*	

Name	***Project phase***	***Contributed to***	***Approx. Number of hours***	***Remarks***
Zain Qasmi	*Proposal*	*Project Deliverables*	*6*	
	SRS Document	*Document designing and Use Case Design, Table, Diagram*	*7*	
	SDS Document	*Worked on Section 5*	*10*	
	Actual Software Development	*Front End (Pages and UI Design)*	*49*	
	Testing document	*Test Case Specifications*	*20*	
	Tool presentation	*Main Features*	*9*	

	Final presentation	*Demo*	*3*	

Name	***Project phase***	***Contributed to***	***Approx. Number of hours***	***Remarks***
Hassaan Hasan	*Proposal*	*Assumptions and Constraints and Expertise in a specific tool*	*7*	
	SRS Document	*Overall description*	*7*	
	SDS Document	*Worked on Section 5*	*10*	
	Actual Software Development	*Front End (Pages and UI Design)*	*48*	
	Testing document	*Test Case Specifications*	*18*	
	Tool presentation	*Pros and Cons*	*8*	
	Final presentation	*Testing Results and Discussion*	*4*	