Software Requirements Specification

for

OpenSchool : on-demand voluntary mentoring platform

Version 1.0 approved

Prepared by

Urvi Ashturkar(111803037)

Mugdha Bapat(111803041)

Chetas Borse(111803045)

College of Engineering Pune - TY Comp Div. 1 Batch T1

21 February 2021

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Version one	20-1-2021	Getting started with the srs	1
Version two	24-1-2021	Added block diagram	2

1. Introduction

1.1 Purpose

OpenSchool is an online platform that will connect Indian school students (grades 1 to 12) to qualified volunteer mentors for one-time sessions either for help with school work or for more advanced discussion. It will help students overcome difficulties in school, pursue budding interests and understand different subjects in a broader, more practical light.

1.2 Document Conventions

This document uses Arial font throughout the entire document. Major headerlines use size 15 font. Subsections are denoted in italics. All other portions of this document use size 12 font. All sections and subsections are numbered accordingly. Functional and non-functional requirements follow a 1 to 10 level priority scale, where priority level 1 is of highest importance, and level 10 is of lowest importance.

1.3 Intended Audience and Reading Suggestions

This project is mainly useful for school students and is based on voluntary service by mentors. This document may be of interest to educational institutes and organizations.

1.4 Product Scope

We aim to initiate open discussion of Indian school students (grades 1 to 12) with experienced mentors, based on matching the needs of students to the expertise of the mentors. Students will be able to ask questions and discuss ideas beyond the scope of their curriculum. Students will gain exposure to mentors with diverse experiences, teaching styles and language backgrounds.
The mentors will be qualified individuals with a passion for sharing their knowledge, who will register on a voluntary basis. This service is not paid.

1.5 References

This document is built upon the following references:

- IEEE Software Specification Requirement template.

2. Overall Description

2.1 Product Perspective

This is a self-contained product. The learning environment for most school students is restricted by a standardized curriculum that reflects very little of what is going on in the real world.
There is a lack of quality teachers or even students' discomfort with the medium of instruction that makes students lose interest in some subjects. Different students are suited to different teaching styles and it is not possible for a single subject teacher at school to cater to the individual needs of each child.
Many experienced people in academia have a passion for teaching but are unable to make commitments because of other demands of senior positions. This product can connect them to students as volunteer mentors for one-time sessions either for help with school work or for more advanced discussion.
The product is a web application which can be accessed from anywhere. Hence, it can leverage the entire community of passionate teachers from all different backgrounds and locations. The product enables us to initiate open discussions with reliable mentors to facilitate the education of school students.

2.2 Product Functions

1. The application will offer an easy to use UI.
2. Students and mentors will be able to create their profiles.
3. Students will post a request for a mentoring session as per their requirements- subject, grade, preferred dates, preferred time slots, preferred language, type of session (help with schoolwork / advanced discussion).
4. Mentors who have registered as voluntary mentors for that subject will be notified of the request through email from within the app and an in-app notification.
5. Mentors will be able to confirm the slot with the student through email from within the app and an in-app notification.
6. The student will be able to see a list of all available mentors who have responded, along with their profiles and past reviews/credential badges.
7. The meeting link for the platform of choice will have to be shared. (No personal contact details exchanged.)
8. The student will be able to post a review of the mentor after the session.

2.3 User Classes and Characteristics

As the main aim of the application is to help the school student community, the application will be used by students and mentors. An administrator may check the qualifications of mentors. This user class is mainly moderation and checking the qualification of interested mentors. Hence, there will be three user roles:

1) Students: Students will have their own profile page. They can request to schedule a meeting on any of the subjects with any available mentor. After the meeting ends, students can submit ratings and reviews on the mentor. This class of users will be limited to school students - 1st to 12th.

2) Mentor: Mentors are the users who voluntarily want to teach and contribute to the student community. These users can confirm their availability when they are notified of a meeting request. Mentors have to submit their qualifications at the time of profile creation. If approved by the administrator, their profile gets created.

3) Administrator: The owners of the product appoint the set of administrators. Administrators have the privilege to check and verify mentors' qualifications. After the mentors upload their documents related to qualifications, they are verified by the administrator. If not approved, then the mentors get a message stating the reason. If approved, their account is created. Administrators also have the privilege to deactivate certain mentor or student accounts if serious reports are raised against them.

2.4 Operating Environment

The platform will function as a web-based application and will not use the user's local storage. User data will be stored on the server's database safely. The front-end will be implemented using a suitable framework/library like React and the backend will consist of a backend framework like Express. The runtime used will be Node.js.

2.5 Design and Implementation Constraints

The verification of the authenticity of the profile information given by new registering mentors is left to the discretion of the admin. Hence along with the mentor-student design, an additional admin user needs to be set up who can verify the details submitted by the mentors.

2.6 User Documentation

A walk-through video will be provided on the user's dashboard. The video displayed on the dashboard will vary depending on the class of the user, that is, different videos will be available for mentors and students.

2.7 Assumptions and Dependencies

Audio and video-communication required for the meeting is not a feature of the software. The meeting links shared by the mentor with the student are external to the software. These meeting links can be of any video-communication platform, for example, Google Meets, Zoom, Discord, etc. Hence, the smooth and efficient working of the meeting is dependent on the assumption that these external video-communication services work correctly.

3. External Interface Requirements

3.1 User Interfaces

The user interface will be a web application created using React.js, connected to the backend and database using an API. The application will have an easy to use, functional and modern UI which will be developed using the Bootstrap frontend framework.
There will be a generic landing page and separate dashboards for students, mentors and administrators.

3.2 Hardware Interfaces

The web application will offer cross-platform support on mobile phones, laptops and desktop devices. As the application will not involve any installation or downloads, it will work on all compatible modern browsers. The backend will consist of a server built in Express.js with a Node environment, and a MySQL database.

Minimum operating system requirements -

- Windows7 and above
- Windows Server 2008 and above
- SmartOS
- Ubuntu 14.04 LTS and above
- MacOS 10.7 and above
- Contemporary Unixes

The above requirements are not exclusive.

3.3 Software Interfaces

The main web application will be connected to a MySQL database on the server. All profile information of the registered users will be stored and retrieved from the database. It will also hold information about the requests made by students, mentor responses and the meetings scheduled.

Requirements to run the server -

- Express v4.17.1
- MySQL v2.18.1
- React v17.0.1
- Bootstrap4
- Python3

Minimum web browser requirements for users -

- Google Chrome 49 and above
- Mozilla Firefox ESR
- Internet Explorer 11 and above
- Safari 10 and above
- Microsoft Edge 14 and above

The above requirements are not exclusive.

3.4 Communications Interfaces

The web application, running on any of the compatible modern web browsers, will communicate with the server via http in the development phase, and via https on deployment. Requests will be made using the Axios library.
E-mails are to be sent from within the application in the following scenarios -

1. To admin for verification on registration by a mentor.
2. To mentor on successful creation or rejection of the new account.
3. To mentor on being selected by a student for a session.
4. To students when the mentor shares the meeting link.

4. System Features

4.1 Interests

4.1.1 Description and Priority

Interests include preferred languages and subjects which can be included in a user's profile. Interests give an idea of the topics and preferred teaching medium for a particular user. This helps in scheduling sessions accordingly.

4.1.2 Stimulus/Response Sequences

In the profile menu, there is an option named interests where a user can add his/her preferred languages and subjects.

4.1.3 Functional Requirements

LOGGED IN : To access this feature the user needs to be logged into their account.

STUDENT : Students can add only subjects as the language of instruction can be given in the request.

MENTOR : Mentors can add subjects as well as languages as both are needed in broadcasting of requests to specific mentors.

4.2 Requests

4.2.1 Description and Priority

The student users will be able to request a session in any topic of their interest. This feature is of high priority as the whole system revolves around students being able to request on demand tutoring.

4.2.2 Stimulus/Response Sequences

Once a student logs in the web app, he/she will be provided with an option to request a session from the home page. The student can either request a session from a specific teacher based on their ratings or broadcast the request to all the mentors tutoring the subject. The requests get broadcast to only those mentors with interests matching the language_of_instruction and subject of the request.

4.2.3 Functional Requirements

LOGGED IN : To access this feature the user needs to be logged into their account.

STUDENT : To access this feature, the user should be a student.

FIELDS : Students need to fill all the fields in the form so that mentors can get a clear idea of the request.

INTERESTS : At least one mentor in whole database needs to have the subject and language of this requested session as an interest in their profile.

4.3 Scheduling

4.3.1 Description and Priority

Once a request has been sent/broadcast, the mentors can respond to the request following which the meeting is scheduled.

4.3.2 Stimulus/Response Sequences

Once a mentor logs in to his/her account, he/she can see a tab with requests. In this tab, he can choose the requests he wants to accept and can send a confirmation for a meeting with the respective student on a click of a button.

4.3.3 Functional Requirements

If a student broadcasts his request, there is a possibility that 2 mentors approve the requests at consecutive times. In this case the request of the first mentor is processed and the second mentor is displayed with a message “Request has already been approved”,(Note that this case arises only when 2 mentors fetch the requests page simultaneously. If a mentor fetches a page from the server after the request has been approved by some other mentor, he will not see the request).

LOGGED IN : To access this feature the user needs to be logged into their account.

MENTOR : To access this feature, the user should be a mentor.

4.4 Sessions

4.4.1 Description and Priority

Users will be able to see the upcoming sessions, past sessions and the pending requests for a session.

4.4.2 Stimulus/Response Sequences

Once the users log in the web app, they see a dashboard with upcoming sessions showing the sessions scheduled by a mentor(This step happens after scheduling), sessions taken by the user in the past and also the pending request tab(only for students) which includes the requests still not approved by any of the mentors.

4.4.3 Functional Requirements

LOGGED IN : To access this feature the user needs to be logged into their account.

PENDING REQUESTS : This option is available only for the students.

4.5 Recommendations

4.5.1 Description and Priority

The student users will get mentor recommendations based on the subjects they like. This list includes top 10 mentors based on their ratings. A student can directly request a session from the recommended mentor and in this case, the request won't be broadcast..

4.5.2 Stimulus/Response Sequences

The recommendation list will be a carousel in the homepage showing the profile of the mentors along with a schedule meeting option.

4.5.3 Functional Requirements

LOGGED IN : To access this feature the user needs to be logged into their account.

STUDENT : To access this feature, the user should be a student.

4.6 Ratings

4.6.1 Description and Priority

Each mentor in the system will have rating points associated with them. The rating points system is like an achievement for a mentor. These points are used to filter the recommendation list, help judge a mentor, etc.

4.6.2 Stimulus/Response Sequences

After the session ends, students can go to the sessions tab and rate the mentor from there. After rating the mentor, the concerned session automatically moves from upcoming sessions to past session.

4.6.3 Functional Requirements

LOGGED IN : To access this feature the user needs to be logged into their account.

STUDENT : To rate a mentor, you should be a student.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

1. User interface needs to be intuitive for the users.
2. Attractive user interface is preferable especially for school students.
3. The software is specific to school students, however it should be scalable to students and mentors across the country.
4. The server should be maintained regularly for a smooth user experience.
5. Reliability of mentors during the meeting (last minute cancellation, poor quality teaching, reviews) should be taken into consideration.

5.2 Safety Requirements

1. Mentors are expected to understand the needs of the school students and should teach and behave accordingly in the meeting. Any slang or poor choice of words must be avoided in the meeting.
2. Students need to be aware of the review feature where they may raise objections against certain mentors.
3. Users need to agree to the terms and conditions that may be required by the external companies providing audio-video communication services for the meeting.

5.3 Security Requirements

1. No contact details of the users will be exchanged at any point.
2. Personal information shared at the time of profile creation is solely for students or mentors to view before scheduling meetings. This information will not be shared with any third-party.
3. Qualification documents uploaded by the mentor will be only used for maintaining the quality of mentors teaching the students. The documents can be only accessed by the administrator.
4. Audio or video permissions needed during the meeting are not a part of the software. These permissions are completely a responsibility of the external companies providing audio-video communication services for the meeting.

5.4 Software Quality Attributes

- Notifications: Notifications should be immediately sent to mentors after a session is requested. Similarly, students should receive notifications after their session is confirmed.
- Correctness: The users should feed the data correctly on the portal. The data fed by the mentors will be verified by the administrator.
- Maintainability: The administrator should be able to maintain the records and data regularly, for optimum performance.
- Usability: The portal should be able to satisfy the maximum number of students (users) using the services. Basic knowledge about the portal should be sufficient for the users to interact and use it to the fullest.

5.5 Business Rules

In the following 2 cases -

1. A student raises serious objections in the reviews given to the mentor.
2. A mentor sends false qualification information which is revealed later in some circumstances.

The administrator can deactivate the mentor's account.

Any conversation during the course of the meeting is not a responsibility of the product owners and the administrator. However, such objections can be raised by the students in the review section or by the mentors using the contact provided to them.

Usage of this software is completely free - profile creation, scheduling meeting, submitting reviews, getting notifications is completely free of cost. If charges are incurred for the meeting conducted by the external service provider, that is not a concern of this product.

6. Other Requirements

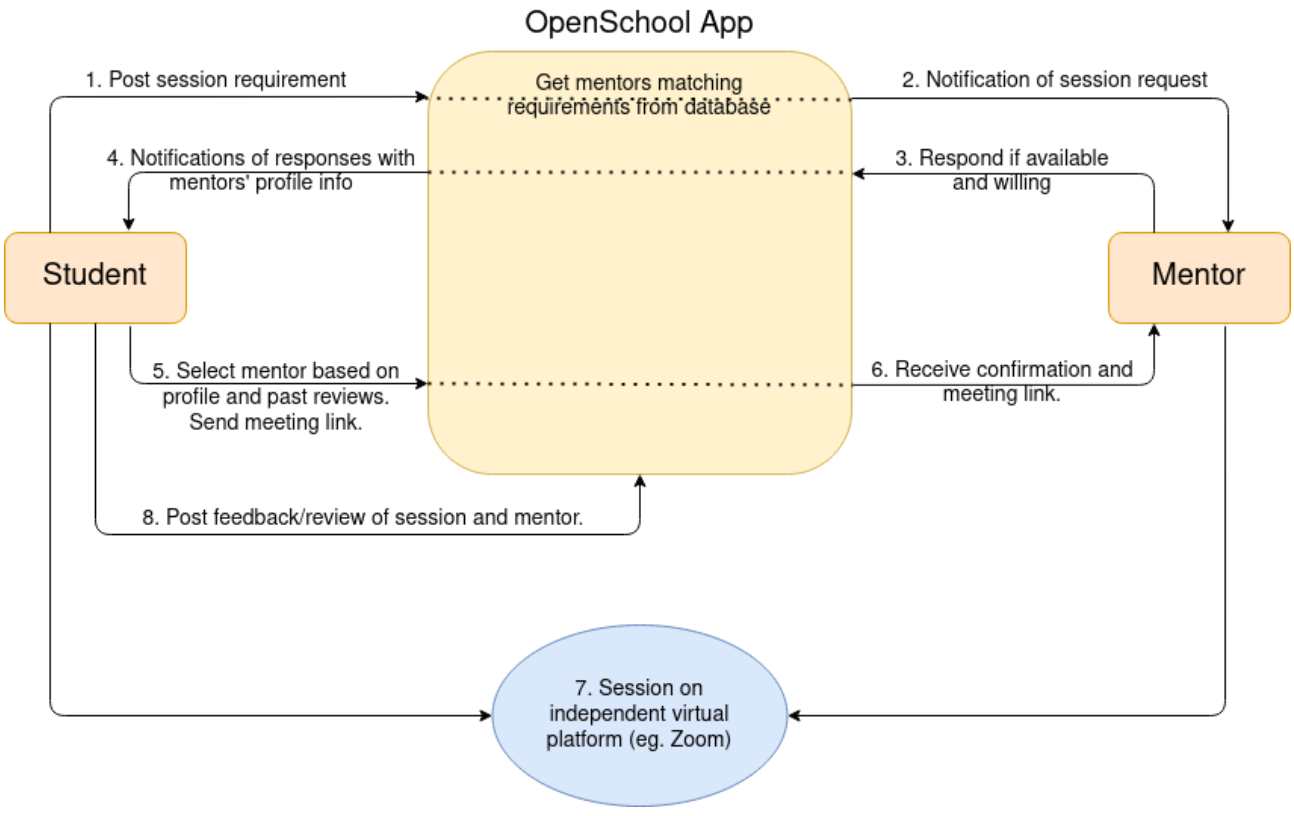
None identified.

Appendix A: Glossary

- Mentors - People who voluntarily sign up for conducting sessions on request and teaching the student as per their requirements.
- HTML- HyperText Markup Language
- CSS- Cascading Style Sheets
- Bootstrap- Frontend toolkit
- Express- JavaScript Web Framework Express.js
- Node- Node.js is a JavaScript run-time built on Chrome's V8 JS Engine
- MongoDB:-No SQL database

Appendix B: Analysis Models

Block Diagram of the application is as shown below -



Appendix C: To Be Determined List

<Collect a numbered list of the TBD (to be determined) references that remain in the SRS so they can be tracked to closure.>