

IT314
Software Engineering

Name - Kanishk Kant

ID - 202001047

Functional requirements:

- **Feature for members to borrow and return books**

This requirement states that the LIS should have a system in place that allows members to borrow books from the library and return them when they are done. This could include creating an account, searching for books, reserving books, and checking out and returning books.

- **Notable feature which can be added**

The web-app notes down IDs of each student and saves their record in a database where it checks for all details saved and then displays it on a website. If no account has been created then the web-app must offer a feature to create a new account for the student.

- **Feature for members to extend borrowing dates**

This requirement states that the LIS should have a system that allows members to extend the due date of a borrowed book, as long as no other member has requested the book. This could include a feature to check the availability of a book and a button to request an extension.

- **Notable feature which can be added**

The web-app must have a feature to extend the return date for a student's book and update the same in the backend database.

- **Feature for members to fine borrowers for no returns or any damages**

This feature would enable the software to calculate the fine for any defaulter after the return date of the book has passed (after the warning duration has passed) or in any case the book has been damaged. The web app should also allow the librarian as an administrator to impose a fine if the book has been damaged.

- **Notable feature which can be added**
The web app must have a feature which gives two late returns warning and then starts calculating the fine in the backend for each day the borrower delays the returning date of the book.
- **Feature for library staff to handle book transactions**
This requirement states that the LIS should have a system in place that allows library staff to manage the day-to-day transactions of the library, such as checking books in and out and keeping track of overdue books.
- **Feature for the librarian to add and remove book records**
This requirement states that the LIS should have a system in place that allows the librarian to add new book records to the system when new books are purchased and to remove book records from the system when books are taken off the shelf.
- **Feature for non-members to browse and search books online**
This requirement states that the LIS should have a system in place that allows non-members to browse and search for books online, however only members can borrow or return books.
- **Restriction of issuing and returning books to valid members only**
This requirement states that only valid members of the LIS should be able to borrow or return books. This could include a login system and a verification process to ensure that only authorized members have access to these features.

Non-functional requirements

- **Web application using recent HTML 5**
This requirement states that the final deliverable of the LIS should be a web application that is built using the most recent version of HTML, which is HTML5.
- **Notable feature which can be added**
The website can use technologies such as CSS or some designing framework to improve the look and feel of the website to enhance and the UI must be easier to understand so as to enable every user to access it without any hassle.

- **Limited to running within the institute LAN**
This requirement states that the LIS should only be accessible within the institute's local area network (LAN) and not over the internet. This is to reduce the risk of unauthorized access to the system.
- **Notable feature which can be added**
The web-app must be shown as inaccessible if anyone outside the network tries to gain access to it.
- **Confidential information (such as passwords) should not be stored in plain text**
This requirement states that the LIS should not store any confidential information, such as passwords, in plain text. This could include using encryption or hashing to protect sensitive data. More simply stated, the encryption used to store the password must be one-way encrypted.
- **Notable feature which can be added**
The web app can have a function to store the passwords in a database which uses advanced encryption techniques to store the passwords in the form of hashes.
- **Secure handling of user's sensitive information**
This requirement states that the LIS should handle user's sensitive information in a secure manner and that the system should be designed to protect user's data from unauthorized access.
- **Notable feature which can be added**
A firewall should solve this requirement.

Q.2. Identify scope, features and non-functional aspects of the following problem.

Scope

- The scope of the problem is the overall goal or objective of the project. In this case, the goal is to create a mobile application that addresses the everyday needs of the hearing impaired community. This includes providing a solution that can recognize key sound events of interest to this community, such as car horns and babies, and providing immediate alerts and continual logging for these events.

Key features of the application

- **Recognition of warning signs**
The mobile application uses artificial intelligence to recognize key sound events of interest to the hearing impaired community. This includes sounds such as car horns and babies, which are critical for the user's safety and well-being.
- **Immediate alerts**
The mobile application provides immediate alerts for recognized sound events, so that the user can take immediate action.
- **Continual updates**
The mobile application continually logs sound events for the user, providing a record of the sounds they have encountered over time.
- **Optimized for Android**
The mobile application is optimized for Android devices, ensuring compatibility and optimal performance on these devices.
- **Low-latency**
The mobile application has low-latency, meaning that it works in real-time, providing the user with immediate feedback.
- **Integration with other devices**
The mobile application could potentially be integrated with other devices, such as hearing aids, to provide an enhanced experience for the user.

- **Machine Learning**

The mobile application could use Machine Learning to improve its performance over time, for example by learning the user's sound preferences and adapting the alerts accordingly.

- **Offline mode**

The mobile application should be able to work offline as well, so that the user can still receive sound alerts even if they are not connected to the internet.

- **Multi-language support**

The mobile application could support multiple languages, making it accessible to a wider range of users.

- **Cross-platform compatibility**

The mobile application could be developed to be compatible with different platforms such as iOS, to enable users with different types of devices to use it.

- **Voice recognition**

The mobile application could use voice recognition to enable users to interact with it using voice commands, making it more accessible for users with hearing loss.

- **Accessibility features**

The mobile application could include accessibility features, such as visual alerts, to make it more accessible to users with hearing loss.

Non-functional aspects

- **User experience**

The mobile application should be designed with the user experience in mind, making it accessible and easy to use for the hearing impaired community.

- **Performance**

The mobile application should have good performance, including low latency and real-time functionality, so that it can provide immediate feedback to the user.

- **Security**

The mobile application should protect user data and maintain the user's privacy.

- **Scalability**
The mobile application should be able to handle a large number of users without any issues.
- **Reliability**
The mobile application should function consistently and without errors, ensuring that it can be relied on by the user.
- **Compatibility**
The mobile application should work with Android devices, ensuring that it can be used by a wide range of users.