EPOKA UNIVERSITY

Software Engineering 2023
Lecture Prof. Aida Bitri / Lecture Prof. Edison Rrashketa
Spring Semester 2023

Software Engineering Project
Katia Haveri
Library Management System for Universities/Schools

1. Development Model

To develop my software, I will use the agile model as I believe that it is the best option in my case. The agile model is suitable to design a library management system, because it is made to handle complicated projects which requirements are always changing, making it the best fit for the dynamic and developing demands of a university and school library. It is also very helpful, because developing a Library management system requires extensive collaboration and feedback from stakeholders such as librarians, students, and teachers, which can be easily facilitated by this model as stakeholders can be involved in every step of the development process. The agile approach leads also to a more user-centric system, as it can prioritize features and functionalities based on the needs of the users. This model places a strong emphasis on iterative development and short, frequent releases that can be swiftly tested and improved, making it possible to respond to feedback and make changes in real-time, which results in a better final product.

2. Requirements

[1] User Requirements

Functional requirements

- Create user account: The users (students, faculty, and staff) should be able to create an account with unique usernames and passwords.
- Login into the user account: The users should be able to login into their account.
- Edit/Delete account: The user should be able to make changes into their account or delete it.
- Search for materials: The users should be able to search in the library catalog by title, author, subject, or keyword.
- Books availability: The user should be able to see which books are available to borrow.
- Borrowed/ Returned books: The users should see their borrowing list and their due dates. Librarians can view which books are borrowed and which have been returned.
- Cataloging books: The users should be able to view the materials in different categories, while the librarian users should be able to add and edit the materials.
- Renew the book due date: The user with the permission of the librarian can extend the due date.
- Reports: The user (librarian) should be able to create reports for different reasons
- Suggestion/Reviews: The users should be able to leave reviews or suggest books.

- Mobile usage: The user should be able to use the system from smartphones and tablets

Non-functional requirements

- Security: The user's personal information should be secure and protected from unauthorized access
- Availability: The user should be able to use the system 24/7, with minimal downtime and maintenance windows.
- Fast response and performance: The user should be able to quickly access the materials they need
- Support and training: The users should have a guide to use the system and train them if needed.

[2] System requirements

Functional requirements

- User-friendly interface: The system should be as simple and understandable as possible in order to be easy to use by all.
- Register/Login: The system should provide the option to register and login only using the schools/universities email.
- Categorization: The system should provide specified categorization of materials so that it can be simple to find what user is looking for.
- Borrowing system: The system should allow users to borrow books that are available.
- Search function: The system should be able to offer the search functionality so that user can find their desired materials by title, author, subject, or keyword, but also librarian should have the search function in some other features (ex: searching for a specific user).
- User management: The system should be able to give the librarian the right to create and edit user profiles and control borrowing rights as part of the system's user management capabilities.
- Material circulation management: The system should give permission to the librarian to control the circulation of materials, including lending and returning them, extending dues, and handling material overdue fees.
- Reports: The system should be able to create reports and analytics to assist librarians in more efficiently managing the library's holdings.
- Database: The system should be able to access the created database to get the user and book information.
- Review/Suggestions control: The system should give the librarian the right to control reviews and suggestions.

Non-functional requirements

- Security: The system should be secure and protect the user's personal data by using user authentication. It can be done by sending an confirmation email for many function that may be involved in the system. To confirm that the user is a member of the school, the photo of the student id may be also required.
- System usage: Both technical and non-technical users should be able to easily use the system thanks to its user-friendly interface and intuitive design.
- Performance: The system should be able to manage several requests from users at once without experiencing any major delays or outages.
- Scalability: Over time, as library users and materials increase, the system should be scalable and able to handle the growth.
- Maintainability: The system should be created and constructed in a way that makes it simple to keep up with changes over time, such as frequent software upgrades.
- Performance during high usage periods: The system should be able to maintain a constant level of performance even under severe loads, such as: when many users enter the system at the same time or when materials are loading.
- Portability: The system should be developed to work on a number of platforms and operating systems in order to enable access from a range of devices.

[3] Software Requirements

- Operating system: MacOS operating system is chosen to create the software, but it can be used by all other operating systems
- Database: MYSQ is employed as a database because it is simple to manage and retrieve records using straightforward English-language queries that are simple to read and write.
- Development tools and programming language: The whole website's code will be written in HTML, which is then developed with CSS, java script for style, and PHP for server-side scripting.