

Real-Time Office Seat Allocation System using Spring MVC framework

Bhargav N¹, Dr. Jayanthi P N²

¹Student, Dept. of Electronics and Communication Engineering, R.V. College of Engineering, Karnataka, India

²Assistant Professor, Dept. of Electronics and Communication Engineering, R.V. College of Engineering, Karnataka, India

bhargavn.ec17@rvce.edu.in, jayanthipn@rvce.edu.in

Abstract: Real-time Seat Allocation System is a web application that assists employees and employers in handling modern-day seating issues at work. Currently, seats are assigned to employees manually, which is a time-consuming process. This might cause issues if seats were not assigned or if none were available for some employees. Furthermore, during pandemics, social distance rules must be followed, making it difficult to manage the workforce. The proportion of individuals allowed in the workplace must adhere properly. As the firm grows, so does the number of employees, which necessitates the expansion of infrastructure, which costs money and time. These issues can be addressed to some extent by a shared office concept, in which employees have access to a desk and can work as needed. This article presents a responsive web application through which company employees can register using their company ID and reserve a seat if one is available on a specified day and time. Admins can add or remove offices, floors, and seats with certain permissions. To implement the business logic, the application uses Java, Spring MVC web framework with Tomcat server, renders the views using Java Server Pages, and uses front end technologies like HTML, CSS, and JavaScript for the front-end design of the User Interface. The web application built promotes the concept of using a shared workspace to maximize resource use. It also aids in increasing employee productivity by providing them with workplace seating and schedules flexibility.

Keywords: Spring MVC framework, Responsive web app, HTML5, CSS3, Bootstrap.

1. INTRODUCTION

The number of people working in the organization and companies is increasing every year. The organizations will have to accommodate all their employees at their workplace. Adding additional offices or floors to existing offices is an option but it would take time and cost to the company. Allocating seats to each employee, checking out the vacant seat, and reallocating is a tedious task when it is done manually. This can be resolved by using a seat allocation system. There have been other seat management systems like a seat management system for the library [1] which makes use of switches and exam hall seat arrangements system [2] for colleges.

When an employee comes to the office and he does not have a workspace or a seat

then the employee will have to sit and work from other places like the conference hall or cafeteria. The conference hall would be underutilized if only a few people are seated in it for not having a regular workspace. The web application would avoid this problem by letting the employee know whether there is a seat available for the time he would be going to an office. The employee can make an informed choice and work from home when no seat is available for his timeslot. The employees can book a seat of their choice and the time duration they would be working. Admins can update the application by adding new floors or seats or even offices when there is a change in infrastructure.

2. Proposed Solution

The implementation of the solution proposed by us makes use of front-end technologies and back-end technologies. It involves various phases starting with the requirements of the application, the planning phase followed by design and development phase. The application is tested to check whether it meets the requirements and performs the desired functionality. If any changes have to be made, they are done in the design and development and again checked. It follows scrum methodology for software development [3].

2.1 Responsive web design

If an application can adapt the appearance of a website based on screen resolutions, device orientation, and the device used to access the website, it is considered to be responsive [4]. Because the number of smartphones has grown over time, anything supplied via the internet should be shown effectively. It should be optimized so that the same information is delivered correctly on various devices by adjusting the layout in accordance with resolution [5]. The most prevalent technologies used for web application design are HTML, CSS, and JavaScript. With the help of Bootstrap [6], the problem with responsive site design can be solved. For layouts, buttons, forms, navbars, and other UI components, Bootstrap provides HTML5 and CSS3-based design templates. Bootstrap layouts are built on a grid framework that defines 12 columns in a single row by default. The Bootstrap utilities are used to create responsive layouts. CSS and JavaScript (JS) [7] can be used to further customize the components. Responsive web pages are designed with a mobile-first philosophy in mind. Because the functionality can be easily extended to larger devices with minimal adjustments.

We use HTML5 elements in our web application to create basic web pages like forms. CSS3 is used to improve the presentation of HTML elements by styling them. The jQuery library in JavaScript [8] is used in our application to conduct actions on events that occur on the webpage, such as a button click event or a focus event. The Bootstrap framework is used to create responsive web design and layout. It aids with the rapid creation of web pages. Additional styling is provided via Bootstrap themes.

2.2 Shared Workspace model

The optimization of the physical environment in the office is what workplace management [9] entails. To work efficiently, every work environment requires administration in some fashion. Employee dissatisfaction and unnecessary financial costs might emerge from inadequate space usage due to a lack of proper management. Seat allocations, or where each employee must sit to complete his or her work, are an important part of workplace

management. Workspaces come in many different forms. Collaborative workspace [10] makes use of existing office space in a way that best matches the company's needs. Employees without a fixed workstation in a shared workspace have access to a private place where they can work on a long-term or as-needed basis. Employees at a shared workspace share the same area but at separate times. It saves costs on additional infrastructure. Our online application is based on the concept of a shared workspace where employees can reserve a seat for a specific time slot. Another employee can book the same seat at a different time without overlapping.

2.3 Logical Architecture

A system's logical architecture, as depicted in Figure 1, is made up of a set of connected technical concepts and principles that support the system's logical operation. The logical architecture is divided into three tiers: Web tier, Service tier, and Repository tier. The Web tier represents the Presentation layer. It takes care of incoming client requests and responds appropriately. A front-end web application is one in which the user interacts with the presentation layer. Any key user action, such as submitting a form, causes the data to be transmitted to the business layer.

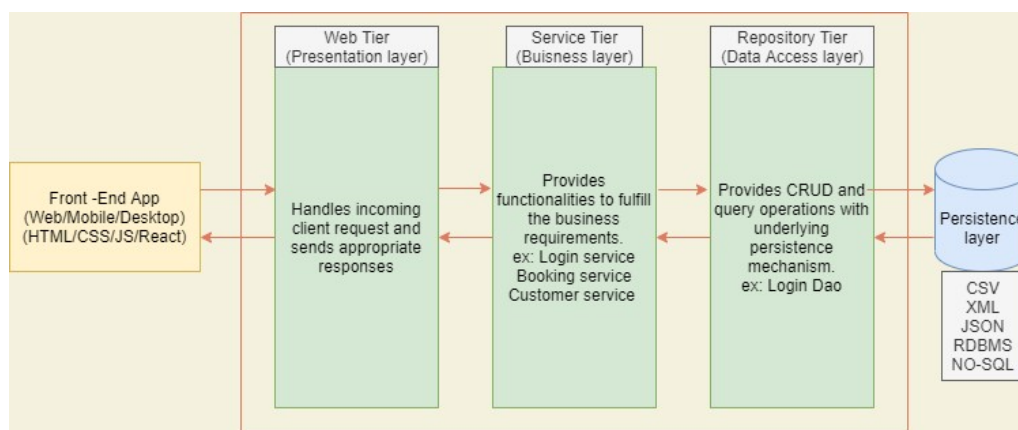


Figure 1. Logical Architecture

The Business layer is represented by the Service tier. It has the functionality to meet the needs of the business. Login service, booking service, and Customer service are just a few examples. Here, all operations are carried out. For each operation, a separate service has been defined. Rather than having a generalized function, they have a specific one. It's in charge of connecting to the database and carrying out all database-related tasks. The data access layer interacts with the business logic. The Data Access layer is made up of the Repository tier. With the support of the underlying persistence system, it performs CRUD activities. A database or an Extensible Markup Language (XML) or JavaScript Object Notation is used as the persistent layer (JSON). It is in charge of storing data and assisting the Repository Tier.

2.4 Data Definition

To manage the data, Java entities were constructed. The tables in the databases are linked to these entities. Employee entity in Table 1 stores all employee information such as employee id, name, department, and email. When a new employee record is created, new entries are created.

Table 1. Employee Entity

Column Name	Data Type	Description
Emp id	Integer	Primary Key
Employee name	Character	Employee name
Department	Character	Department name
Email	Varchar	Employee email
Phone	Integer	Contact number
Password	Varchar	Password for the profile

The number of offices available for booking, the number of floors in an office, and the number of seats on each floor can all be updated by the administrator. The office entity as seen in Table 2 contains the office id, which is a primary key that ensures the entity is unique, as well as the total number of floors in the office.

Table 2. Office Entity

Column Name	Data Type	Description
Office id	Integer	Primary Key
Total floors	Integer	Number of floors in the office

There are a limited number of floors in each office. Through one-to-many mapping, the office object is linked to the floor entity. Because there are multiple floors associated with a single office, the floor object is mapped to the office using many-to-one mapping. As illustrated in Table 3 the floor entity has a florid, floor Color, and total seats on the floor. A foreign key connects the office table and the floor table. Each floor has a large number of seats. Through one-to-many mapping, the floor entity is linked to the seat entity.

Table 3. Floor Entity

Column Name	Data Type	Description
Floor id	Integer	Primary Key
Floor Color	Varchar	Color of the Floor
Total Seats	Integer	Number of seats in the floor
Office OfficeId	Integer	Foreign Key

The seat object has two fields: seat Id and seat Status as shown in Table 4. Seat Status is used to show whether a seat is free or reserved. A particular floor has a large number of seats. Through many-to-one mapping, the seat entity is linked to the floor entity. The entities of the office, floor, and seat are all related. Depending on the status of a seat we can make informed decision. If all seats are occupied then the employee cannot go to the office for that particular slot during which all seats are occupied.

Table 4. Seat Entity

Column Name	Data Type	Description
Seat id	Integer	Primary Key
Seat Status	Integer	Status of the seat
Floor FloorId	Integer	Foreign Key

2.5 SpringMVC

The Spring WebMVC framework [11] is designed around a Dispatcher Servlet that handles all the HTTP requests and responses as shown in Figure 2.

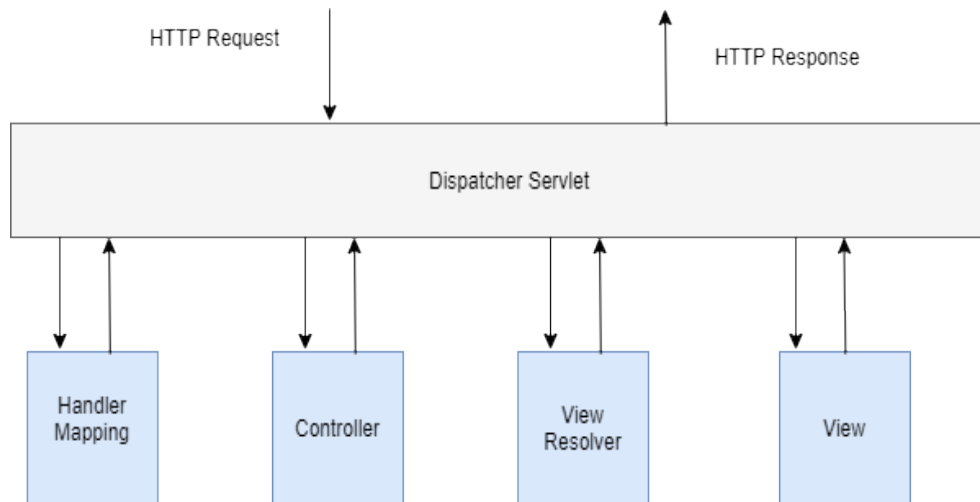


Figure 2. Processing Workflow of Spring MVC

When the Dispatcher Servlet receives an HTTP request, it examines the Handler Mapping to determine which Controller to contact. The Controller receives the request and, depending on the GET or POST method used, calls the corresponding service methods. The Dispatcher Servlet receives the view name from the service function, which sets model data based on given business logic. The user sees the view, which is a Java Server Page (JSP). The View Resolver takes the request's configured view and passes it to the dispatcher servlet. When a view is complete, the Dispatcher Servlet sends the model data to the view, which is then rendered in the browser. The data is stored and retrieved using the H2 database. While keeping essential functionality, Spring JPA minimizes code and streamlines the data access layer. The Query Domain Specific Language (DSL) allows us to design Java interface methods that use specific keywords and JPA entity properties to conduct the work required to appropriately implement queries without having to supply much real coding. It assists us in easily writing custom queries.

3. Results

A landing page, login page, new employee registration page, User dashboard, booking form, Admin dashboard, adding new office form, editing office form, user information page, and office details page are all included in the created web application. Figure 3 depicts the landing page. It includes a brief description of the product as well as a sign-up or login option.

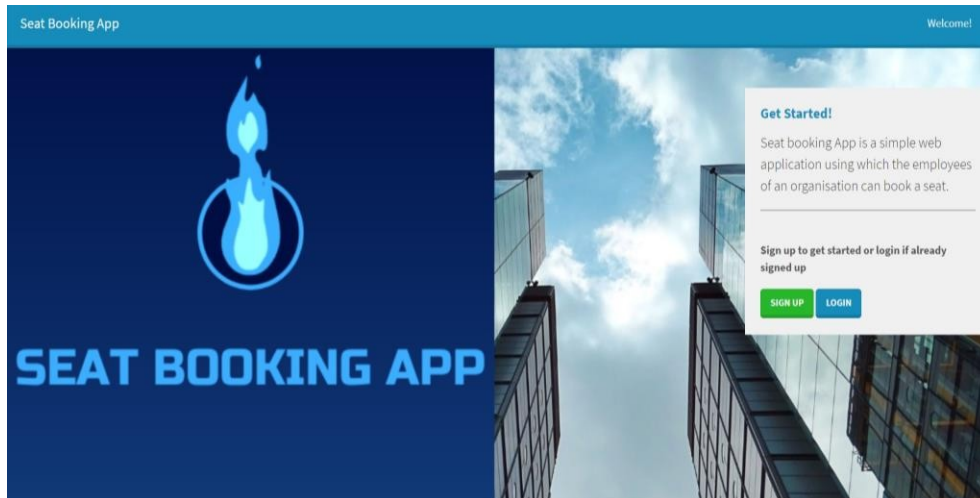


Figure 3. Landing page

If the employee is a first-time user, he must first register. Name, employee ID, department, email ID, phone number, and password are all required fields on the registration screen displayed in Figure 4. The password must include an upper-case letter, a lower-case letter, digits, and a special character, and it must be at least six characters long. The majority of the fields in the form are verified and must be filled out in the correct way. A ten-digit phone number is required. After completing the sign-up process, the employee is directed to a landing page where he can log into the app.

Figure 4. Registration page for new Employee

As seen in Figure 5 the login page has two fields: username and password. The username is the email address. Both fields are required; if one is left blank, an error notice will be displayed. When incorrect credentials are entered, the login screen displays the error message "Invalid User".

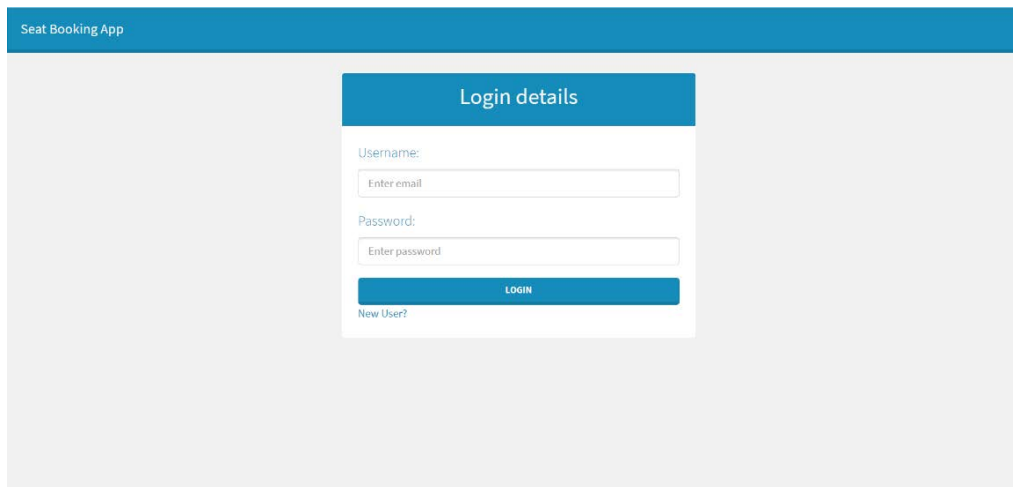


Figure 5. Login page

3.1 Employee functionalities

After logging into the application, the employee is directed to the employee dashboard. The employee's name appears in the top right corner of the page. The employee has the option to log out of the application at any time. The employee's information is displayed in the profile tab. Figure 6 depicts the employee dashboard. By selecting "New Booking," the employee can reserve a seat. A future date, start time, end time, office, floor, and seat must all be chosen by the employee. The booking information is shown to the employee. The employee can confirm or make modifications.

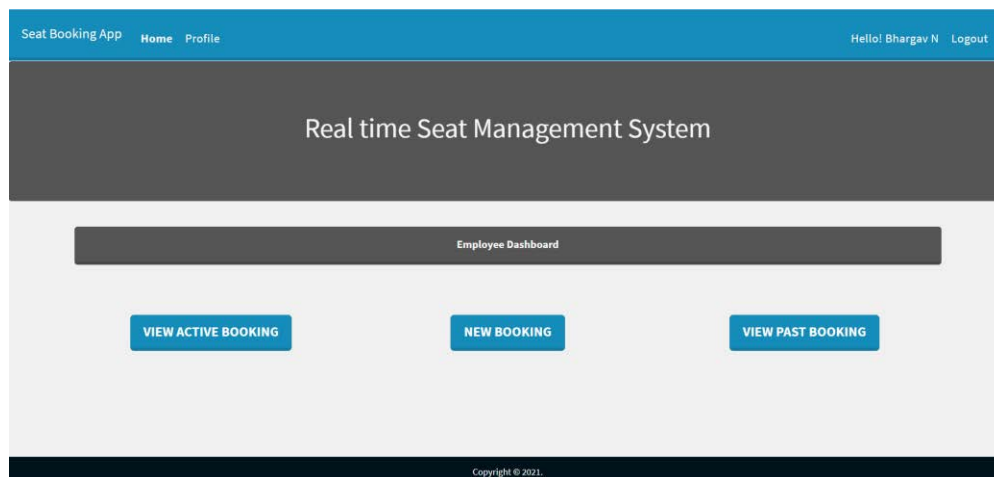


Figure 6. Employee Dashboard

3.2 Admin functionalities

As demonstrated in Figure 7, the admin has a separate dashboard than the employee. Admins can see a list of offices, create new ones, delete old ones, and update existing ones. To create a new office, the administrator must first choose the number of floors for the new office. The highest number of floors that an office can have is four, while the minimum number is one. A maximum of 100 seats and a minimum of 1 seat can be found on each floor. Each floor is designated a color solely for the purpose of identification. Admins

adding a new two-story office in Figure 8. The green color has been given to Floor 1, which has 5 seats. Floor 2 has 5 seats as well, all of which are red in color. When the admin clicks on new office, he or she is asked for confirmation before the office is added and shown in the list of offices. The new office can be modified by changing the floor color or raising or decreasing the number of floors or seats. The web application also allows you to view the details of each office. Any of the offices in the list can be changed or deleted. When an employee tries to make a booking, the admin's adjustments are mirrored for him. The employee can view the additional office or floor added to an office, as well as the change in the number of seats in a floor, in the booking area.

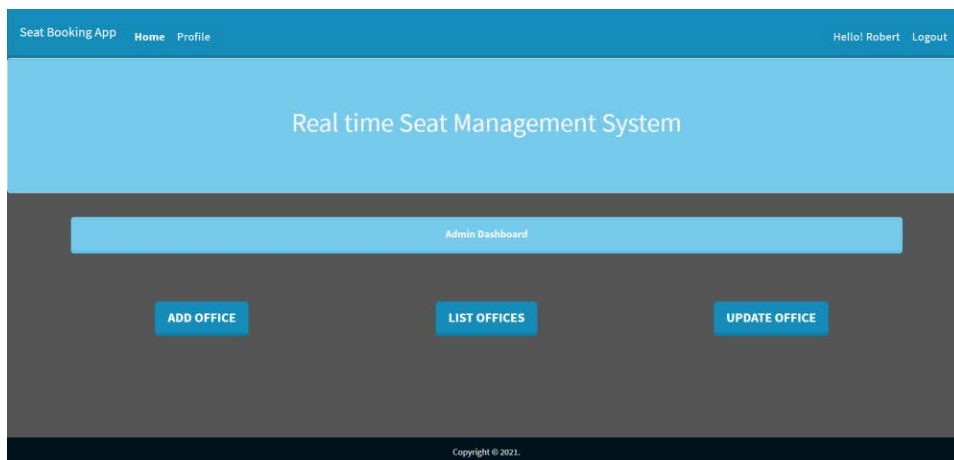


Figure 7. Admin Dashboard

Figure 8. Adding new Office

4. Conclusion

Every year, as the firm grows in size, the difficulty of finding a suitable workspace develops. The corporation cannot continue to add new buildings or floors because it will incur costs and require time. As the population grows, new buildings will have an impact on the environment. Instead, we can embrace a shared office paradigm. The developed application aids in the resolution of issues related to workspace management. It permits various employees to sit in the same seat at various times. Some employees can work from

home, but when they need to, they can schedule a seat and use the office's resources. Adopting scrum approach [12] also reduces complexity and speeds up development. It has also aided in good coordination when working in a group [13]. The tool is especially beneficial to bigger businesses that are dealing with these difficulties. The application simplifies employee seat assignment and lowers the company's commercial lease costs.

5. Future Work

The application can be improved more in the future. When employees book a seat and utilize it without squandering their booking, a reward system can be implemented to the application to award them with useful points. To strengthen security, additional security measures such as two-factor authentication or three-factor authentication can be implemented. Personalized content can be sent to each app user based on how frequently they book. Additional features, such as booking statistics and specific staff information, can be added to admin. We have only examined scheduling for a single day; however, an employee may schedule for many days. The created web application includes basic booking features. Additional functionality or features can be added to improve the application's usability.

6. REFERENCES

- [1] Y. Jing, L. Mao, and J. Xu, "Research on a library seat management system," in 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), 2019, pp. 297–301. doi:10.1109/ICISCAE48440.2019.221639.
- [2] S. Aravinth, G. Pavithra, M. Myvizhimalar, D. Divya, and M. Rathinakrithika, "exam hall seating arrangement system using php," International Journal of Innovative Research and Technology, vol. 1, no. 11, 2014.
- [3] V. T. Faniran, A. Badru, and N. Ajayi, "Adopting scrum as an agile approach in distributed software development: A review of literature," in 2017 1st International Conference on Next Generation Computing Applications (NextComp), 2017, pp. 36–40. doi: 10.1109/NEXTCOMP.2017.8016173.
- [4] N. Li and B. Zhang, "The design and implementation of responsive web page based on html5 and css3," in 2019 International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2019, pp. 373–376. doi:10.1109/MLBDBI48998.2019.00084.
- [5] J. Lee, I. Lee, I. Kwon, H. Yun, J. Lee, M. Jung, and H. Kim, "Responsive web design according to the resolution," in 2015 8th International Conference on u- and e-Service, Science and Technology (UNESST), 2015, pp. 1–5. doi: 10.1109/UNESST.2015.11.
- [6] W. Jiang, M. Zhang, B. Zhou, Y. Jiang, and Y. Zhang, "Responsive web design mode and application," in 2014 IEEE Workshop on Advanced Research and Technology in Industry Applications (WARTIA), 2014, pp. 1303–1306. doi: 10.1109/WARTIA.2014.6976522.
- [7] E. Perakakis, A. Levenko, and G. Ghinea, "Html5 and companion web technologies as a universal platform for interactive internet tv advertising," in 2012 International Conference on Telecommunications and Multimedia (TEMU), 2012, pp. 214–219. doi: 10.1109/TEMU.2012.6294721.
- [8] M. Vidakovic, S. Cosic, O. Cosic, I. Kastelan, and G. Velekic, "One solution for execution of JavaScript in java ee application servers," in 2018 Zooming Innovation in Consumer Technologies Conference (ZINC), 2018, pp. 177–180. doi:10.1109/ZINC.2018.8448646.
- [9] M. Nithin and V. Suma, "Workspace management and hot-seating," in 2017 International Conference on Intelligent Computing and Control Systems (ICICCS), 2017, pp. 900–903. doi: 10.1109/ICCONS.2017.8250595.

- [10] X. Xu, G.M. Luis, A. Lobov, and L. Martinez Lastra, "Multiple ontology workspace management and performance assessment," in 2015 IEEE 13th International Conference on Industrial Informatics (INDIN), 2015, pp. 1063–1068. doi:10.1109/INDIN.2015.7281882.
- [11] Y. Hou, "Design and implementation of the framework for spring+springmvc+mybatis in the development of web application," in 2017 International Conference on Computer Technology, Electronics and Communication (ICCTEC), 2017, pp. 368–371. doi:10.1109/ICCTEC.2017.00085.
- [12] A. Srivastava, S. Bhardwaj, and S. Saraswat, "Scrum model for agile methodology," in 2017 International Conference on Computing, Communication and Automation (ICCCA), 2017, pp. 864–869. doi:10.1109/CCAA.2017.8229928.
- [13] W. Sisomboon, N. Phakdee, and N. Denwattana, "Engaging and motivating developers by adopting scrum utilizing gamification," in 2019 4th International Conference on Information Technology (InCIT), 2019, pp. 223–227. doi:10.1109/INCIT.2019.8911976.