COMP39/9900 Computer Science/IT Capstone Project
School of Computer Science and Engineering, UNSW

Project Number: P20

Project Title: CSE space allocation management system

Project Clients: Arjun Radhakrishnan

Project Specializations: Software development; Computer Science and Algorithms.

Number of groups: 2

Background:

1. About the project

This proposal outlines the development of a Space Allocation Software System designed to streamline and optimize the allocation of physical spaces within K17, CSE. The primary objective is to create an intuitive, efficient, and scalable solution that ensures optimal utilization and future changes to the available spaces while meeting user needs and preferences. There are three aspects that CSE wishes the project would focus on:

i. CSE Meeting and Board Rooms space booking system (an upgrade from the current system that uses MS Outlook)

ii. Hot desk allocation

iii. New staff space allocation

2. Project Objectives/ Goals

- Efficient Space Utilization: Develop a software to capture the available spaces at K17 building including Meeting Rooms, based on specific criteria such as size, usage, and availability.

- Space allocation: Individual space allocation for staff who are using hot desks in level 3 area. Provide support to CSE admin team to allocate space for new staff joining.

- User-Friendly Interface: Create an intuitive interface that allows users to easily manage and request space allocations.

- Integration Capabilities: Ensure the software can integrate with existing systems like MS Teams, MS Outlook and the UNSW website.

- Data Analytics: Provide reporting and usage data that aids in continuous improvement..

Requirements and Scope:

Provide a space allocation management system that CSE staff can use to
i. Book meeting rooms and board Rooms (an upgrade from the current system that uses MS Outlook)
ii. Hot desk allocation for staff
iii. New staff space allocation which will be carried out by admin staff.

3. Scope of Work and deliverables

3.1. Requirements Analysis

- Stakeholder Interviews: Conduct interviews with potential users to understand their needs and preferences.
- Use Case Development: Develop detailed use cases to capture functional and non-functional requirements.
- Technical Feasibility Study: Assess the technical requirements and feasibility of integrating with existing systems.

3.2. System Design

- Define the system architecture, including data flow diagrams, plans to integrate database with existing database if needed, develop prototype focusing on ease of use and accessibility for users with a range of technical capabilities.
- Consider constraints like occupancy, scheduling conflicts, and user preferences.

3.3. Development

- Backend Development: Implement the server-side logic, database management, and API integrations.
- Frontend Development: Develop the user interface that can be easily integrated to existing university systems and that requires minimal upgrades.
- Integration: Integrate the system with existing tools such as MS Outlook, MS Teams and facility management systems (further information provided once the quote for the new AV system is approved).

3.4. Testing

- Unit Testing: Test individual components for functionality and reliability.
- System Testing: Conduct end-to-end testing to ensure all components work seamlessly together.
- User Acceptance Testing (UAT): Engage with a sample group of end-users to validate the system's functionality and usability.

3.5. Deployment

- Stage 1: Deploy the system final testing and validation by users (CSE Admin).
- Stage 2: Launch the system for all users to use once testing is final.
- Training & Documentation: Provide comprehensive training sessions and documentation for end-users (CSE admin team).

3.6. Maintenance and Support

- Training: Provide access and training for select staff to manage the system.
- Updates and Enhancements: Provide a detailed document on possible updates the system may need to undergo for the next five years. Include CSE staff to supervise the documentation.

Required Knowledge and skills:

Technical Skills

Software Development

Programming Languages

- Proficiency in languages such as Python, Java, JavaScript, or similar.
- Web Development: Knowledge of HTML, CSS, and JavaScript frameworks (e.g., React, Angular, Vue.js).

Backend Development

- Server-Side Languages: Proficiency in languages such as Node.js, Django (Python), Ruby on Rails, or similar.
- Database Management: Skills in SQL and experience with databases like MySQL, PostgreSQL, MongoDB, or similar.
- API Development: Experience in developing RESTful and/or GraphQL APIs.

Frontend Development

- UI/UX Design: Skills in creating intuitive and user-friendly interfaces.
- Frontend Frameworks: Experience with frameworks like React, Angular, or Vue.js.
- Integration
- API Integration: Experience with integrating third-party services and APIs (e.g., Google Calendar, Microsoft Outlook).
- Middleware: Knowledge of middleware technologies for smooth integration.

Expected outcomes/deliverables:

- The finished product is expected to be utilised for the long-term hence the knowledge of the source code shall be shared with the UNSW central IT team or CSE team.

- Training resources for staff/ training to select staff who could train the rest of the team

Supervision:

Arjun Radhakrishnan

Additional resources: