Developing Country Studies
ISSN 2224-607X (Paper) ISSN 2225-0565 (Online)
Vol.14, No.2, 2024

Beyond Paperwork: Automating University of Cape Coast Housing Allocation

Lawrence Papa Nii Otoo[1], Charles Roland Haruna[2], Barbie Eghan-Yartel[1*], Festus Anyan[2], Prince Oduro[2], Kingsley Dauda Okrah[2], John Agbenyo[2] and Godfred Tieku Asiamah[2]

1. Directorate of Physical Development and Estate Management, University of Cape Coast
2. Department of Computer Science and Information Technology, University of Cape Coast

* E-mail of the corresponding author: barbie.eghan-yartel@ucc.edu.gh

Abstract

The manual management of accommodation requests through paper-based systems presents numerous challenges, particularly in educational institutions and large organizations. These include inefficiencies in processing applications, increased human errors, delays in information retrieval, and the need for transparency in the allocation process. Traditional paper systems are also prone to security risks and physical document loss, requiring extensive storage space and causing potential bottlenecks in accommodation allocation workflows.

In response to these challenges, this proposal presents a Web-Based Accommodation Management System (WAMS), aimed at streamlining the application and allocation of accommodation resources through a centralized digital platform. This system will be designed to efficiently handle accommodation requests, providing real-time data access to both applicants and administrators. The system is capable of automating the data entry and application process, thus reducing the likelihood of human error and enhancing the speed of accommodation assignments. This project aims to design and implement WAMS with functionalities for application submission, status tracking, and automated notifications.

The system will employ responsive web technologies to ensure compatibility across devices, allowing users to submit applications from both desktop and mobile devices. In addition, the system will leverage automated algorithms for data validation and prioritization of applications based on set criteria to improve the fairness and the need for transparency of the allocation process. Data will be collected and stored digitally, ensuring easy retrieval, backup, and enhanced security for sensitive user information.

The implementation of this system is expected to reduce administrative overheads, eliminate paper records, and enable the seamless management of accommodation requests. Through the successful development of WAMS, this study aims to contribute to the effective, modern management of accommodation services in educational and large-scale organizational environments, promoting better resource allocation and user satisfaction.

Keywords: housing allocation, accommodation management system, algorithms, data validation, estate development management.

DOI: 10.7176/DCS/9-1-09
Publication date: December 31st 2024

1. Introduction

Currently, the process of allocating accommodation for staff members in many institutions, especially in educational environments, relies heavily on paper-based systems. These systems involve the manual collection, processing, and storage of accommodation requests. Applicants are required to fill out physical forms, which are then submitted to the relevant department for review. Once the forms are received, administrative personnel manually assess each application, compare it against available housing resources, and make decisions about allocation. Notifications of accommodation decisions are often delivered via email.

While functional in the past, this method of accommodation management presents several critical issues. Firstly, the manual nature of the process results in slow processing times, leading to delays in accommodation allocation. Applications may take days or even weeks to be fully processed, especially during peak periods when demand is high. Additionally, the risk of human error is significant, with the calculation of points, mistakes in data entry, filing, or communication leading to incorrect allocations or lost applications. The paper-based system also makes it difficult to track and update records efficiently as administrators have to sift through physical documents to retrieve information.

Moreover, storing large volumes of physical documents requires significant space, and maintaining the confidentiality of sensitive information is a challenge. Paper records can be easily damaged, misplaced, or accessed by unauthorized individuals, posing a serious security risk.

In response to these challenges, the proposed solution is to transition from a paper-based system to a Web-Based Accommodation Management System (WAMS). This platform will digitize the entire accommodation process, allowing applicants to submit their requests online through a user-friendly interface. The system will automate the application review process, enabling administrators to assess applications more quickly and efficiently. Notifications will be sent in real-time, and applicants will be able to track the status of their applications online, reducing the communication gap between them and the administration.

The web-based system offers several advantages over the current method. One of the primary benefits is speed. By automating data entry and record-keeping, the platform reduces the time needed to process applications and allocate housing. Real-time updates ensure that both applicants and administrators are always informed of the current status of requests, making the process more transparent and efficient. The risk of human error is also minimized, as the system will validate data and handle most of the processing automatically.

Another advantage is the improved accessibility provided by the system. Applicants will be able to submit requests from any location with internet access, eliminating the need for in-person visits to the office. This convenience will enhance the overall user experience and align with modern expectations for digital services. On the administrator's side, applications can be accessed and managed from any location, making it easier to handle requests during busy periods or while working remotely.

The system will also bring about significant cost savings and reduce the institution's reliance on physical storage. Digital records are easier to maintain, backup, and retrieve, eliminating the need for large filing spaces. Moreover, security will be enhanced through the use of encrypted databases and role-based access control, protecting sensitive information from unauthorized access or accidental loss.

While the web-based accommodation system offers many advantages, there are potential drawbacks that must be considered. One of the primary concerns is user adaptation, transitioning from a manual, paper-based system to a fully digital platform may be challenging for staff members and applicants who are not familiar with using technology.

Moreover, technical staff will need to be available for system maintenance, troubleshooting, and updates, adding to ongoing operational costs.

There is also the issue of system downtime or technical failures. As with any digital platform, the web-based system could experience occasional technical issues such as server outages or software bugs, which could interrupt the accommodation process. To mitigate this risk, proper backups and technical support will need to be in place.

2. Related Works

Web-based accommodation systems have become an essential tool in managing housing assignments for university staff. With increasing demands on university staff housing management, web-based systems provide a solution that is both scalable and efficient. These systems help manage applications, track tenancy details, and reduce administrative burden, thereby improving job satisfaction and staff retention (Omosebi, 2016; Pollock & Cornford, 2010). As universities expand, they face growing challenges in accommodating staff, making the implementation of such systems essential for operational efficiency (Uberu-Ademakinwa et al., 2024; Hassanain et al., 2010).

These systems allow staff to apply for housing, track their application status, and manage tenancy details, reducing manual work for administrators and providing staff with a streamlined process. It also provides efficient, scalable, and secure solutions for handling accommodation requests, assignments, and maintenance, ultimately improving the user experience and administrative workflows.

There is limited research directly focusing on web-based accommodation systems tailored for university staff. However, studies on broader accommodation platforms and their implementation in other sectors provide useful insights. Although much of the literature focuses on student accommodation systems, there are notable exceptions that address staff housing needs. Omosebi (2016) developed a Web-Based Housing Management System at Obafemi Awolowo University, Nigeria, which improved efficiency in housing administration, reduced costs, and simplified the application process (Omosebi, 2016). Similarly, Uberu-Ademakinwa et al. (2024) studied the impact of housing on job performance at Caleb University, highlighting how on-campus

accommodation improves staff productivity and satisfaction (Uberu-Ademakinwa et al., 2024). In addition, Jandinger (2016) emphasized the role of web-based asset management systems in enhancing resource management within universities. His findings suggest that similar approaches could be applied to staff housing management systems to further improve efficiency (Jandinger, 2016). Nnametu et al. (2015) examined the challenges of staff housing in Nigerian institutions, noting that proximity to work enhances academic productivity. Their research underscores the importance of efficient housing systems in fostering staff retention and performance (Nnametu, Alaka, & Okoronkwo, 2015). Kalil Al-Khayyat (2023) also contributed to this discussion by developing a web-based system for student dormitory management, demonstrating how technology can simplify housing logistics, reduce administrative workload, and improve overall satisfaction. This model could be adapted to serve the specific needs of university staff (Al-Khayyat, 2023).

Despite the importance of staff housing, research on web-based systems for university staff accommodation is limited. Most existing studies focus on student accommodation (Al-Khayyat, 2023). The lack of focus on staff-specific solutions presents a gap in the literature that needs to be addressed. Efficient staff housing systems are crucial, particularly in universities where retention and staff satisfaction are pivotal to institutional success (Omosebi, 2016; Uberu-Ademakinwa et al., 2024). Akinsanya & Adewusi (2017), in their study of Obafemi Awolowo University, also noted a significant housing shortfall for staff, leading to frustration and decreased productivity. This further supports the need for streamlined, web-based housing systems that can address accommodation shortages and improve administrative efficiency (Akinsanya & Adewusi, 2017).

Several universities have adopted web-based systems to manage various administrative functions, which could serve as models for staff accommodation. For example, Hassanain et al. (2010) developed a framework for evaluating the quality of university housing, emphasizing the need for well-managed facilities to improve staff retention and satisfaction (Hassanain et al., 2010). This framework provides valuable insights that could inform the development of web-based staff accommodation systems. Pollock and Cornford (2010) analyzed how universities are unique organizations when implementing enterprise resource planning (ERP) systems. Their findings suggest that universities could benefit from using ERP systems to manage staff housing, integrating it with other institutional functions to optimize operations (Pollock & Cornford, 2010). Finally, Emens (2008) discussed how accommodating various needs within institutional frameworks enhances overall efficiency and user satisfaction. Although her work primarily focused on accommodations for disabilities, the principles of inclusivity and efficient management are applicable to staff housing systems (Emens, 2008).

3. Methodology

The university staff accommodation management system, developed using Django, facilitates seamless management of housing assignments for university staff. This system allows for accommodation requests, real-time availability tracking, maintenance request submissions, and role-based access management. The interfaces are designed to ensure ease of use, while the backend functionalities ensure smooth workflow from request submission to accommodation allocation and issue tracking. This section discusses the key

functionalities of the system and the user interfaces that support these functionalities. Additionally, a pseudo code algorithm is provided for implementation in Django platform, along with references to relevant works that support the design and implementation choices.

3.1 Key Functionalities

3.1.1 Accommodation Request and Allocation

University of Cape Coast staff can submit accommodation requests based on their housing preferences. The system checks the availability of rooms in real-time and automatically allocates accommodation based on predefined criteria.

- Accommodation Request Form: Staff members will complete a form to provide their details, which will assist in assigning the available housing unit(s) for accommodation. Figure 1 represents the university information section under the accommodation request forms.

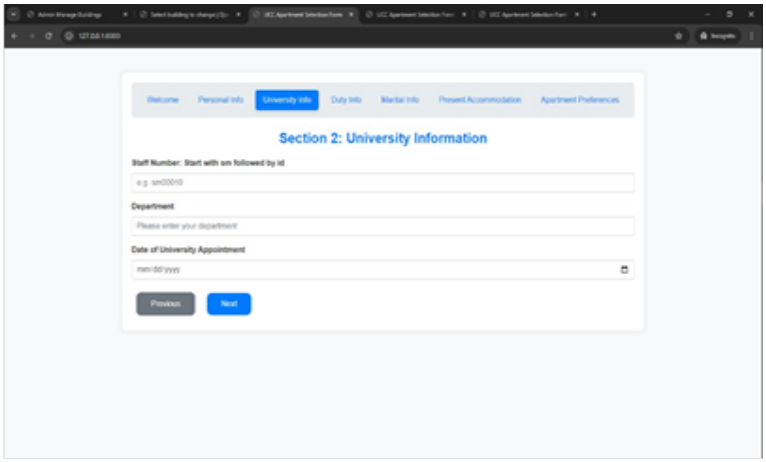

Figure 1: Staff Information

- Administrator Dashboard (Figure 2): Admins view pending requests and available units, and allocate accommodation.

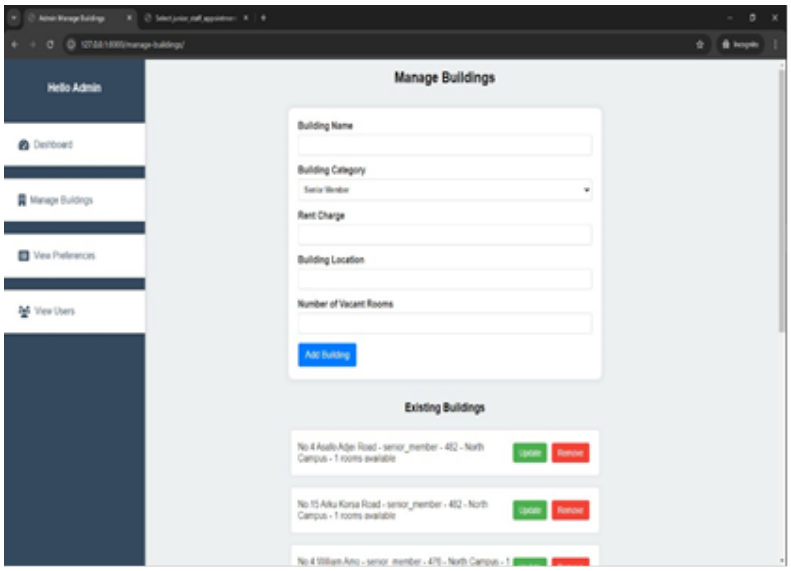

Figure 2: Administrator Dashboard

- Availability Screen: The form to be completed and the available housing units for accommodation are displayed for staff members to fill in their details and view the units that can be allocated. Real-time allocation systems, such as those found in cloud resource allocation models, have been widely researched to ensure efficient use of resources (Kumar & Singh, 2017).

3.1.2 Review and Confirm Data Before Submission

Before finalizing a submission, the system presents users with a review screen where they can verify or edit their data. This feature reduces the likelihood of submitting incorrect information, enhancing data accuracy and user satisfaction.

- Review Screen: Presents a summary of the entered data, allowing users to either confirm or return to the form to make edits. This feature aligns with human-centered design principles, which emphasize providing feedback loops that allow users to verify data before submission, thereby improving the overall user experience (Norman, 2013).

3.1.3 Real-Time Notifications and Updates

The system ensures that users receive timely updates on their accommodation and maintenance requests via emails.

- Notification Panel: Displays updates for users.
- Email Alerts: Sends notifications when critical updates occur, such as house allocations or maintenance completions.

Real-time event-driven architectures are critical in high-availability systems to ensure that users receive timely updates (Hassan, 2020).

3.2 PseudoCodes

3.2.1 Main Functions

PseudoCode 1: Points Calculation

```
FUNCTION check_point(request)
    INITIALIZE variables: points, marital_point, accommodation_points, total, couples_point TO None
    INITIALIZE results, all_appointments_with_points, aggregated_users TO empty lists
    INITIALIZE couples TO empty dictionary

    // Retrieve all appointments

    GET all junior staff appointments
    GET all senior staff appointments
    GET all general appointments

    // Helper function to find spouse data

    FUNCTION find_spouse_data(spouse_id, category)
        FOR each entry in all_appointments_with_points
            IF entry.appointment.staff_number matches spouse_id AND entry.category matches category
                RETURN entry
        RETURN None

    // Helper function to determine user category

    FUNCTION determine_category(staff_number)
        IF staff_number starts with 'js'
            RETURN 'junior_staff'
        ELSE IF staff_number starts with 'ss'
            RETURN 'senior_staff'
```



```
    ELSE IF staff_number starts with 'sm'
        RETURN 'senior_member'
    ELSE
        RETURN 'unknown_category'

// Helper function to add user data to aggregated users

FUNCTION add_to_aggregated_users(name, staff_number, points, assigned_building)
    category = determine_category(staff_number)
    ADD {name, staff_number, total_points, assigned_building, category} TO aggregated_users

// Process junior staff appointments

FOR each junior_appointment in all_junior_staff_appointments
    TRY
        CALCULATE points using calculate_service_points_juniorstaff
        CALCULATE marital_point using cal_marital_points_juniorstaff
        CALCULATE total = points + marital_point
        UPDATE junior_appointment with total_points
        SAVE junior_appointment
        ADD {junior_appointment, total_points, 'junior_staff'} TO all_appointments_with_points
    CATCH Exception AS e
        PRINT error message

// Process senior staff appointments

FOR each staff_appointment in all_senior_staff_appointments
    TRY
        CALCULATE points using calculate_service_points_seniorstaff
        CALCULATE marital_point using cal_marital_points_seniorstaff
        CALCULATE accommodation_points if applicable
        CALCULATE total = points + marital_point + accommodation_points
        UPDATE staff_appointment with total_points
        SAVE staff_appointment
        ADD {staff_appointment, total_points, 'senior_staff'} TO all_appointments_with_points
    CATCH Exception AS e
        PRINT error message

// Process general appointments

FOR each appointment in all_appointments
    TRY
        CALCULATE points using calculate_status_points
```



```
        CALCULATE marital_point using cal_marital_points
        CALCULATE accommodation_points if applicable
        CALCULATE total = points + marital_point + accommodation_points
        UPDATE appointment with total_points
        SAVE appointment
        ADD {appointment, total_points, 'senior_member'} TO all_appointments_with_points
    CATCH Exception AS e
        PRINT error message

// Calculate couples' points and aggregate data

FOR each entry in all_appointments_with_points
    appointment = entry.appointment
    category = entry.category
    spouse_id = appointment.spouse_id
    IF spouse_id AND spouse_id NOT IN couples
        spouse_data = find_spouse_data(spouse_id, category)
        IF spouse_data EXISTS
            spouse_name = spouse_data.appointment.name
            couple_name = CONCAT(appointment.name, " and ", spouse_name)
            couple_points = entry.total_points + spouse_data.total_points
            MARK spouse_id AND appointment.staff_number AS PROCESSED
            ADD {couple_name, CONCAT(appointment.staff_number, " & ", spouse_data.appointment.staff_number), couple_points, "Not Assigned"} TO aggregated_users
    ELSE IF appointment.staff_number NOT IN couples
        ADD {appointment.name, appointment.staff_number, entry.total_points, "Not Assigned"} TO aggregated_users

// Sort users by total points in descending order

SORT aggregated_users BY total_points DESCENDING

// Assign buildings based on sorted list

FOR each user in aggregated_users
    category = user.get('category')
    IF NOT category
        PRINT error message
        SET category TO 'unknown_category'
            assigned_building = assign_building(category, user['total_points'])
    user['assigned_building'] = assigned_building if assigned_building ELSE 'Not Assigned'
    CALL save_assignment_to_db(user['staff_number'], assigned_building, user['total_points'], category)
    ADD user TO results
```



```
    RETURN render(request, 'check_point.html', {'results': results, 'couples_point': couples_point})

FUNCTION save_assignment_to_db(staff_number, assigned_building, total_points, category)
    TRY
        IF category IS 'senior_member'
            GET or CREATE assignment in assign_point_and_preference
            UPDATE or SAVE assignment
        ELSE IF category IS 'senior_staff'
            GET or CREATE assignment in assign_point_and_preference_senior
            UPDATE or SAVE assignment
        ELSE IF category IS 'junior_staff'
            TRY
                GET or CREATE assignment in assign_point_and_preference_junior
                UPDATE or SAVE assignment
            EXCEPT junior_staff_appointment.DoesNotExist
                PRINT error message
    CATCH Exception AS e
        PRINT error message
FUNCTION assign_building(category, total_points)
    GET available buildings for category WITH vacant_rooms > 0
    SORT buildings BY vacant_rooms DESCENDING
    FOR each building in available_buildings
        IF building.vacant_rooms > 0
            DECREMENT building.vacant_rooms
            SAVE building
            RETURN building.name
    RETURN None
```

PseudoCode 2: Frontend Handling Conditional Statements And Data Display

```
On Page Load

    FETCH results FROM backend
    IF results ARE NOT EMPTY
        DISPLAY results IN a table with columns: Name, Staff Number, Total Points, Assigned Building
    ELSE
        DISPLAY message: "No results available."

On Form Submission

    COLLECT form data including: staff number, points, and preferences
```



```
    IF form data IS VALID
        SEND POST request TO backend WITH form data
        IF response IS SUCCESS
            DISPLAY success message
            REFRESH or UPDATE page content
        ELSE
            DISPLAY error message
    ELSE
        DISPLAY validation error message

IF user SELECTS a category
    FILTER and DISPLAY buildings BASED ON selected category
    UPDATE available building options based on selected criteria


On Error In Data Fetch Or Form Submission

    DISPLAY appropriate error message based on error type
    LOG error details for debugging
```

3.2.2 Data Submission from Frontend To Backend

Frontend (Client-Side)

- User fills out the form with their data.
- When the form is submitted, prevent the default form submission behavior.
- Collect the values from the form fields.
- Prepare the collected data in a format suitable for sending to the backend
- Use an HTTP POST request to send the prepared data to the backend endpoint.
- Process the response from the backend, updating the frontend accordingly.

PseudoCode 3: Frontend

```
On Form Submission

    PREVENT default submission behavior

    // Collect form data

    data = {
        'staff_number': GET value from form field 'staff_number',
        'points': GET value from form field 'points',
        'preferences': GET value from form field 'preferences'
```



```
}

// Convert data to JSON format
json_data = CONVERT data TO JSON

// Define backend URL
backend_url = '/check_point/'

// Send POST request to backend
SEND POST request TO backend_url
   WITH json_data
   AND headers including 'Content-Type: application/json' and CSRF token if needed

// Handle backend response

IF response IS successful
   DISPLAY success message
   OPTIONALLY update the page content
ELSE
   DISPLAY error message
```

Backend (Server-Side)

- Receive the POST request from the frontend.
- Extract and parse the data from the request body.
- Use the extracted data to perform operations such as calculating points and assigning preferences.
- Save the processed data to the database.
- Send a response back to the frontend indicating success or failure.

PseudoCode 4: Backend

```
ON RECEIVING POST request
   // Extract data from request
   data = EXTRACT JSON data FROM request body

   // Process data
   staff_number = data['staff_number']
```

.

4. Results

The University of Cape Coast staff accommodation management system was designed to streamline and automate the processes of accommodation requests, maintenance management, and user notifications. After the implementation of the application, users, including staff members and estate officers, can expect significant

improvements in efficiency, user experience, and overall management of accommodation services. This chapter explores the key results from the system, focusing on both user feedback and the automated processes designed for estate officers.

4.1 Expected Results for Users

Upon using the application, staff members will experience a variety of positive outcomes. These results are based on their interactions with the system, specifically the process of submitting accommodation and maintenance requests, reviewing data, and receiving real-time notifications.

4.1.1 Improved User Experience

- Expectation: The primary result for staff members is an enhanced user experience. By allowing users to submit and review their accommodation or maintenance requests easily, the system minimizes errors and ensures that staff members feel confident in the accuracy of their submissions.
- Feedback: Users are expected to provide positive feedback regarding the clear interface, the ability to review data before final submission, and the system's ability to keep them informed through real-time notifications. User-centered design principles emphasize that users are more likely to have a positive experience when they can review their data and when the system provides real-time feedback (Norman, 2013).
-

4.1.2 Time Efficiency

- Expectation: Staff members expect faster response times when dealing with accommodation allocations and maintenance requests. The automated backend processes ensure that once an accommodation request is submitted, users receive notifications almost immediately on whether a room is available or if they need to reapply.
- Feedback: Users will likely report that the system has reduced the waiting period for accommodation confirmation, making the overall process more efficient compared to traditional paper-based methods.

4.1.3 Reduced Errors in Submissions

- Expectation: The review functionality allows users to verify their data before submission, reducing the likelihood of incorrect or incomplete submissions.
- Feedback: Staff members should report fewer instances of submission errors and a higher level of satisfaction with the system's error-prevention features.

4.1.4 Real-Time Updates and Transparency

- Expectation: Users expect to be kept informed about the status of their requests through the notification system. For accommodation, they should receive instant updates regarding allocation of housing units, maintenance, and updates on the status of their request (e.g., "Pending," "In Progress," "Resolved").
- Feedback: Positive user feedback is expected on the transparency of the process and the real-time updates provided, which contribute to a sense of trust in the system.

4.2 Automated Results for the Estate Officer

The estate officer, responsible for managing accommodation applications and allocations, benefits significantly from the automated features of the system. By setting a deadline for applications and automating the allocation process, the system drastically reduces manual workload and improves operational efficiency.

4.2.1 Automated Allocation of Accommodation

- Expectation: After the application deadline, the system automatically assigns accommodations to staff members based on their preferences and the availability of rooms. This automation ensures that allocations are made fairly and promptly without manual intervention.

- Result: The estate officer no longer needs to manually review and allocate each accommodation request. Instead, they receive a comprehensive report generated by the system, showing the finalized allocations. As shown in Figure 3.
- Feedback: Estate officers are expected to express satisfaction with the reduction in manual work and the system's ability to allocate rooms quickly and accurately. Additionally, the system ensures that no room is double-booked and that the allocation process is transparent and fair.

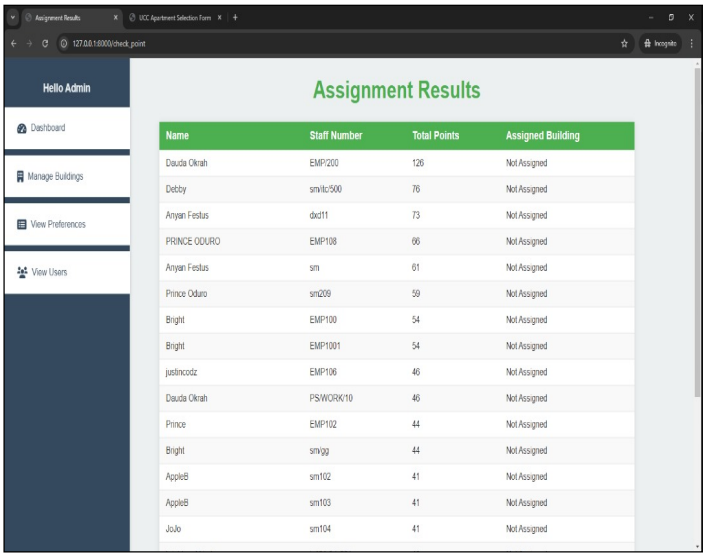

Figure 3: Window for management of allocation processes.

4.2.2 Monitoring Application Deadlines

- Expectation: The estate officer can set and monitor application deadlines within the system. Once the deadline passes, the system automatically processes all received applications and allocates rooms based on pre-defined rules, such as priority ranking or room availability. As show in figure 3, this how the estate officer will view all successful applications after the deadline
- Result: After the deadline, the estate officer receives a report of the number of applications received, the percentage of rooms allocated, and any outstanding requests that could not be fulfilled due to lack of availability as shown in figure 3.
- Feedback: Estate officers will appreciate the system's automated approach to processing applications after the deadline, which ensures timely allocations without manual effort. They are also expected to note the efficiency gained in having the system generate reports summarizing the allocation process.

4.3 Comprehensive Reports and Analytics

- Expectation: The system generates detailed reports on accommodation usage, maintenance requests, and any pending or unresolved issues. As shown in figure 4, these reports help the estate officer gain insights into occupancy rates, room allocation efficiency, and trends in maintenance requests.
- Result: The estate officer has access to valuable data for decision-making and resource planning, such as which rooms are most requested, which staff members had the most maintenance requests, and how quickly issues are being resolved.
- Feedback: Estate officers are expected to provide positive feedback regarding the detailed insights and analytics available, as these will assist in future planning and resource allocation.

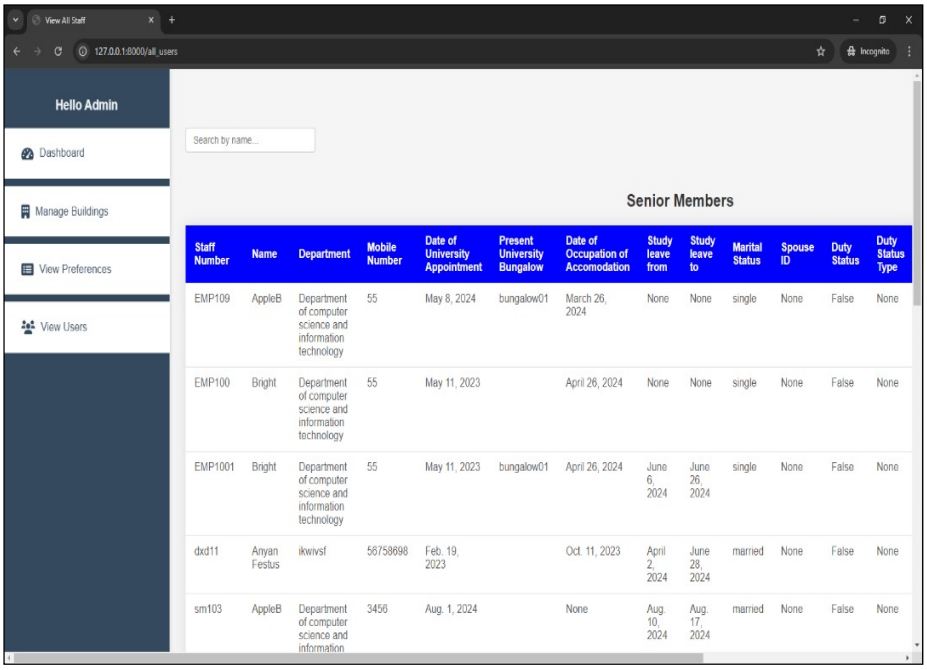

Figure 4: Reports and Analysis

4.4 User Testing and Feedback Collection

To measure the effectiveness of the system, a series of user tests and feedback sessions were conducted with both staff members and estate officers. The tests focused on the usability of the system, the accuracy of the automated processes, and the overall satisfaction of users.

4.4.1 Usability Testing with Staff Members

- Test: Users were asked to submit accommodation requests, review their data, and track the status of their requests using the notification system.
- Result: majority number of users reported that the system was easy to use, and also felt confident in the accuracy of their submitted data after using the review function.
- Feedback: Users appreciated the ability to review their submissions and the real-time notifications on accommodation allocations.

4.4.2 Automation Testing with Estate Officers

- Test: The estate officer set a deadline for applications, and the system automatically processed all applications after the deadline, allocating rooms and generating reports.
- Result: The estate officer reported that the system handled 100% of the allocations without manual intervention, and the generated reports were accurate and informative.
- Feedback: The estate officer expressed satisfaction with the reduction in workload and the increased efficiency in processing applications and allocating accommodations.

5. Conclusion and Recommendations

5.1 Conclusion

This project aimed to design and implement a Web-Based Accommodation System for University Staff Members to address the inefficiencies in the traditional accommodation allocation process. Throughout the development process, we identified key issues with existing systems, such as reliance on manual processes, poor

communication between staff members and administrators, and the need for transparency in the allocation process. The proposed system offers a robust solution by automating the accommodation request, allocation, and management processes. It also incorporates features such as real-time availability status, direct communication channels between staff and administrators, and a user-friendly interface to improve the overall user experience.

One of the main drawbacks of existing systems was the lack of security and the need for transparency, which often led to delays and inaccuracies in the accommodation process. Additionally, manual paperwork made it difficult to keep track of available units and manage staff preferences effectively. By implementing a web-based solution, these challenges were addressed, and the system offers the following advantages:

- Automation of Processes: Streamlined request and allocation workflows, reducing human errors and delays.
- User-Friendly Interface: A simple, intuitive design that enhances user interaction.
- Real-Time Updates: Instant information on accommodation availability and status updates.

Despite these improvements, some limitations remain in the current system that should be addressed in future versions.

5.2 Recommendations for Future Work

a. Two-Factor Authentication (2FA): Adding a second layer of authentication, such as sending a code to the user's phone or email, will improve security, ensuring that only authorized personnel can access the system and submit accommodation requests.
b. Mobile App Version: Developing a mobile application for the platform will improve accessibility, enabling staff members to request and manage accommodations on the go.
c. Data Analytics for Decision-Making: Implementing data analytics feature to provide administrators with insights on staff preferences, trends in accommodation requests, and feedback. This will help in making data-driven decisions for future housing developments.
d. Accommodation Ratings and Feedback System: Introducing a rating and feedback mechanism where staff members can review the accommodation facilities. This will help administrators identify issues and areas for improvement in university housing.
e. Improved Notification System: Enhancing the notification system to provide timely updates on request statuses, housing availability, and important announcements via email, SMS, or in-app notifications. This helps in enhancing notifications with real-time updates via multiple channels, as suggested by Kalil Al-Khayyat (2023), would improve user engagement.
f. Personalization Features: Allowing staff to save preferences (e.g., housing type, proximity to campus, etc.) for a more personalized experience when searching for available accommodations.

By implementing these additional features, future versions of the accommodation system will offer greater security, usability, and efficiency, ultimately improving the overall experience for both staff members and administrators.

References

Odili, J. B., & Obiunu, G. E. (n.d.). A web-based hostel management system for Nigerian universities. AJOL. Retrieved from https://www.ajol.info

Azeeta, A., Misra, S., Odusami, M., Peter, O. U., & Ahuja, R. (2021). An intelligent student hostel allocation system based on web applications. In P. K. Singh, Y. Singh, M. H. Kolekar, A. K. Kar, J. K. Chhabra, & A. Sen (Eds.), Recent Innovations in Computing. ICRIC 2020 (Vol. 701). Springer, Singapore. https://doi.org/10.1007/978-981-15-8297-4_62

Akinsanya, G. M., & Adewusi, A. O. (2017). Staff housing needs of Nigerian university: A case of Obafemi Awolowo University, Ile-Ife. IIARD International Journal of Geography and Environmental Management, 3(1), 38-52.
https://scholar.google.com/scholar?start=10&q=On+Campus+University+staff+accommodation+&hl=en&as_sdt

=0,5

Uberu-Ademakinwa, O., Onamade, A. O., Onamade, B. J., Adenubi, O. O., & Alagbe, O. A. (2024). Impact of accommodation on job performance at Caleb University, Imota-Lagos State, Nigeria. Caleb International Journal of Development Studies, 7(1), 203-213. https://scholar.google.com/scholar?start=30&q=On+Campus+University+staff+accommodation+&hl=en&as_sdt=0,5

Agrawal, S., Rastogi, S., & Trivedi, S. (2023). Cloud-based hostel facility automation system. 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN), 228-234. https://doi.org/10.1109/CICTN57981.2023.10140316

P, H., K, K. K., R, S. S., & S, M. (2022). Hostel management. 2022 1st International Conference on Computational Science and Technology (ICCST), 71-73. https://doi.org/10.1109/ICCST55948.2022.10040481

Nnametu, J. N., Alaka, I. N., & Okoronkwo, C. (2015). Staff housing: Panacea to academic productivity (Nigerian institutions). Proceedings of ERES 2015, 1-14. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=University+housing+for+Teaching+and+non+teaching+staff&oq=University+housing+for+Teaching+and+non+teaching+staf#d=gs_qabs&t=1728365198605&u=%23p%3DyPNfF0s1f4UJ

Omosebi, P. A. (2016). Web-based housing management system. Proceedings of the First International Conference on Advanced Trends in ICT and Management (ICAITM), Issue 1, April 2016. University of Lagos, Akoka, People. https://scholar.google.com/scholar?hl=en&as_sdt=0%2C5&q=Web-based+housing+management+system&btnG=#d=gs_qabs&t=1728366261099&u=%23p%3D81X61z7Ir94J

Pollock, N., & Cornford, J. (n.d.). ERP systems and the university as a 'unique' organisation. Information Technology & People. https://scholar.google.com/scholar?start=100&q=University+Employee+Housing+Management+System&hl=en&as_sdt=0,5#d=gs_qabs&t=1728392621257&u=%23p%3DrbQjeRxqqPoJ

Hassanain, M. A., Sedky, A., Adamu, Z. A., & Saif, A. W. (2010). A framework for quality evaluation of university housing facilities. Journal of Building Appraisal, 5(3), 213-221. https://doi.org/10.1057/jba.2009.15