ACES Fund Requests

Project Proposal

May 25, 2024

Table of Contents

1. Introduction

1.1 Document Purpose

The primary objective of this document is to provide a comprehensive overview of the functionalities, specifications, and design considerations of a web-based application developed to modernize and streamline the fund management system for the Association of Computer Engineering Students (ACES). The proposed application seeks to automate the entire lifecycle of fund management, encompassing the generation, validation, and approval of fund requests. This automation aims to significantly enhance the efficiency, transparency, and accessibility of fund management processes for ACES members. The document serves as a detailed project proposal, outlining both functional and non-functional requirements to give developers, project managers, evaluators, and other stakeholders a thorough understanding of the system. It covers all aspects of the project, including user interface design, system architecture, data management, security protocols, and compliance with regulatory standards, ensuring that all parties have a clear and unified vision for the successful implementation and operation of the fund management application.

1.2 Product Scope

The ACES Fund Management System (AFMS) aims to modernize and streamline the fund management process for the Association of Computer Engineering Students (ACES) at the University of Peradeniya. By replacing the existing paper-based system with a web-based application, AFMS will enhance efficiency, transparency, and accessibility, creating a seamless and efficient fund management experience for ACES members. The admins can add approvers and requesters, the approvers can login, approve, reject requests and the requesters can submit requests and track the status of the requests. Furthermore, the Chatbot feature helps users to get quick and more accurate answers to FAQ questions.

1.3 Objective

The overarching goal of this project is to engineer a resilient, scalable, and intuitively navigable web application tailored to oversee the management of fund requests and approvals within the confines of the Association of Computer Engineering Students (ACES). With an unwavering focus on robustness and scalability, the proposed application is meticulously crafted to withstand evolving user demands and accommodate future growth seamlessly. At its core, the application is designed to deliver a streamlined, efficient, and transparent process, empowering all members of ACES to engage with confidence. By leveraging cutting-edge technology and user-centric design principles, the application promises to cultivate an environment where fund management becomes an effortless and transparent endeavor, fostering collaboration and innovation within the organization.

1.4 Intended Audience:

Admins

- Head of Computer Engineering Department

- Computer Engineering Academic Staff

Approvers

- ACES EXecutive members
- Project Supervisors

Users

- Computer Engineering Students

1.5 Key Benefits:

Efficiency and Time Savings

The integration of automated fund management processes represents a pivotal milestone in the optimization of resource utilization within the Association of Computer Engineering Students (ACES). By circumventing the arduous burden of manual tasks, the proposed system liberates students and administrators alike from the shackles of labor-intensive procedures. This liberation translates into a tangible boost in efficiency and time savings, affording ACES members the invaluable opportunity to redirect their focus towards more mission-critical endeavors. With mundane administrative tasks relegated to the realm of automation, individuals can channel their energies towards fostering innovation, nurturing collaboration, and driving the organization towards its overarching goals. Consequently, the newfound efficiency not only augments overall productivity but also cultivates an environment ripe for creativity, growth, and sustained success within ACES.

Transparency and Accountability

The implementation of a robust audit trail and real-time status tracking mechanism within the fund management system stands as a cornerstone in fostering transparency and accountability within the Association of Computer Engineering Students (ACES). By meticulously documenting every transaction and providing instant visibility into the status of fund requests, the system engenders a culture of openness and trust among its members. This transparency not only serves to instill confidence in the integrity of the fund management process but also ensures that all actions are subject to scrutiny and accountability. With a clear and accessible record of transactions at their disposal, stakeholders can navigate the system with confidence, knowing that every decision is traceable and every outcome is transparent. As a result, the system not only enhances operational efficiency but also cultivates a culture of accountability that is foundational to the long-term success and credibility of ACES.

Security and Data Integrity

The incorporation of stringent user authentication measures and fortified data management protocols underscores the unwavering commitment of the fund management system to safeguarding sensitive information within the Association of Computer Engineering Students (ACES). By implementing robust user authentication mechanisms, the system ensures that only authorized individuals gain entry to critical financial data, mitigating the risk of unauthorized access and potential data breaches. Moreover, the adoption of secure data management protocols further fortifies the system's defenses, erecting a formidable barrier against external threats and internal vulnerabilities. Through these proactive measures, the integrity of financial data remains inviolable, instilling confidence among ACES members and stakeholders in the system's ability to uphold data security standards and protect against potential risks. Thus, the system not only bolsters the organization's resilience against cyber threats but also reinforces trust in the confidentiality and integrity of financial operations within ACES.

Eco-Friendly Approach

The transition to a digitized fund management process heralds a transformative shift towards sustainability and environmental stewardship within the Association of Computer Engineering Students (ACES). By eschewing traditional paper-based methods in favor of digital alternatives, the system not only alleviates the organization's reliance on paper but also champions a more eco-conscious approach to resource utilization. This sustainable paradigm shift not only yields tangible cost savings by reducing expenditure on paper, printing, and storage but also resonates with broader initiatives to promote green practices and minimize ecological footprints. Furthermore, the digitization of fund management processes paves the way for greater operational efficiency and flexibility, as digital records are inherently easier to manage, update, and archive. Ultimately, by embracing digitalization, ACES not only realizes immediate benefits in terms of cost savings but also reaffirms its commitment to environmental sustainability, positioning itself as a forward-thinking and responsible steward of the planet's resources.

2. Product Features :

2.1 User Authentication

The implementation of a secure authentication system leveraging Google OAuth 2.0 represents a sophisticated yet user-friendly approach to access control within the Association of Computer Engineering Students (ACES). By harnessing the robust capabilities of Google OAuth 2.0, the system ensures that only individuals with authorized @eng.pdn.ac.lk email addresses gain entry, thereby confining access exclusively to students and administrators affiliated with the Faculty of Engineering. This stringent access restriction not only safeguards sensitive financial data but also instills confidence in the integrity of the system among ACES members. Furthermore, by leveraging Google's widely trusted authentication framework, the system enhances user convenience and streamlines the login process, eliminating the need for separate credentials and reducing the risk of password-related vulnerabilities. Thus, through the amalgamation of cutting-edge security protocols and seamless user experience, the authentication system serves as a formidable barrier against unauthorized access while fostering a culture of trust and accountability within ACES.

2.2 Dashboard

User Authentication:

- **Secure login for both requesters and approvers:** The system ensures secure login functionality for both requesters and approvers, utilizing industry-standard encryption protocols to protect user credentials during the authentication process. This involves implementing strong password hashing techniques to safeguard against unauthorized access and brute-force attacks. Additionally, the system may incorporate measures such as multi-factor authentication (MFA) to add an extra layer of security, requiring users to verify their identity through a combination of factors such as passwords and one-time codes.
- **Secure registration for admins:** Admin registration is conducted through a secure process that verifies the identity and credentials of individuals seeking administrative privileges within the system. This may involve a multi-step verification process, where admins are required to provide valid identification and undergo background checks to ensure their eligibility for administrative roles. Furthermore, admin registration may be restricted to designated individuals within the organization, with strict controls in place to prevent unauthorized personnel from gaining administrative access.
- **Role-based access control (RBAC):** The system employs role-based access control mechanisms to manage user permissions effectively. Each user is assigned a specific role (e.g., requester, approver, admin), which determines their level of access to system functionalities and data. RBAC ensures that users only have access to the resources and actions necessary for their respective roles, minimizing the risk of unauthorized data exposure or manipulation.
- **Session management:** To prevent unauthorized access to user accounts, the system employs robust session management techniques. This includes generating unique session tokens upon successful login, encrypting session data to prevent tampering or interception, and implementing session timeouts to automatically log users out after a period of inactivity. Additionally, the system may incorporate features such as session revocation, allowing administrators to invalidate active sessions in the event of suspicious activity or user account compromise.
- **Audit logging:** The system logs all authentication-related activities, including login attempts, registration events, and changes to user roles or permissions. This audit trail provides administrators with visibility into user authentication activities, enabling them to monitor for suspicious behavior, track user access patterns, and investigate security incidents effectively. Audit logs are securely stored and may be accessed only by authorized personnel, ensuring the confidentiality and integrity of authentication data.
- **Compliance with security standards:** The authentication system adheres to established security standards and best practices, such as those outlined by OWASP (Open Web Application Security Project) and NIST (National Institute of Standards and Technology). This includes implementing secure coding practices, conducting regular security assessments and audits, and staying updated on emerging security threats and vulnerabilities. By aligning with industry-recognized security standards, the system demonstrates a commitment to maintaining the highest levels of security and reliability for user authentication processes.

Admins Dashboard :

- **View and manage submitted fund request forms:** The admin dashboard provides a comprehensive overview of all fund request forms submitted by users within the system. Admins can access a centralized dashboard displaying a list of pending, approved, and rejected fund requests, along with relevant details such as request ID, requester name, submission date, and requested amount. This allows admins to efficiently track the status of each request and prioritize their review based on factors such as urgency or funding availability.

- **Approve or reject fund requests with comments:** Admins have the authority to review and take action on submitted fund requests directly from the dashboard. They can approve or reject requests with a single click, selecting the appropriate action from predefined options available within the system. Additionally, admins have the option to provide comments or feedback when approving or rejecting requests, facilitating clear communication with requesters and ensuring transparency in the decision-making process. These comments are recorded and may be viewed by both requesters and other admins for reference purposes.
- **Detailed request information:** Upon selecting a specific fund request from the dashboard, admins can view detailed information pertaining to the request, including the purpose of the request, supporting documentation, and any additional comments provided by the requester. This comprehensive view enables admins to make informed decisions based on the merits of each request, ensuring alignment with organizational policies and budgetary constraints.
- **Actionable notifications:** The admin dashboard may include notifications or alerts to notify admins of new fund requests awaiting review, pending actions requiring attention, or updates on previously reviewed requests. These notifications help admins stay informed and proactive in managing fund requests, ensuring timely responses and minimizing delays in the approval process. Notifications delivered via email depending on the preferences of the admin and the configuration of the system.
- **Audit trail:** The admin dashboard maintains a detailed audit trail of all actions taken by admins regarding fund requests, including approvals, rejections, and comments. This audit trail provides a historical record of administrative activities, offering accountability and transparency in the fund management process. Admins can refer to the audit trail for compliance purposes, performance analysis, or retrospective review of decision-making processes.

User Dashboard :

- **Submit the fund request form:** The user dashboard provides a user-friendly interface for submitting fund request forms. Users can access the dashboard and initiate the fund request process by filling out a structured form that captures relevant details such as the purpose of the request, requested amount, supporting documentation, and any additional comments or notes. The form may include validation checks to ensure that all required fields are filled out correctly before submission, reducing errors and streamlining the request process. Upon submission, users receive confirmation that their request has been successfully submitted and is now under review.
- **Track the status of submitted requests (PENDING, ACCEPTED, DENIED):** Users can track the status of their submitted fund requests directly from the dashboard. The dashboard displays a list of all requests submitted by the user, along with their current status (e.g., pending, accepted, denied). Users can easily identify the status of each request at a glance and monitor any changes in real-time. The status of each request is updated dynamically as it progresses through the review and approval process, providing users with timely visibility into the status of their requests and enabling them to plan accordingly.
- **View feedback on rejected requests:** In the event that a fund request is denied, users have the ability to view feedback or comments provided by administrators regarding the rejection. The user dashboard displays

detailed information about each rejected request, including the reason for the denial and any additional comments or suggestions provided by the reviewing admin. This feedback helps users understand the rationale behind the decision and provides valuable insights for improving future requests. Users can use this feedback to make adjustments to their requests or seek clarification from administrators as needed, fostering a collaborative and transparent communication process between users and administrators.

- **Edit or resubmit requests:** Depending on the system configuration, users may have the option to edit and resubmit their requests if they are denied or require modifications. The user dashboard provides functionality for users to review their previously submitted requests, make necessary edits or revisions based on feedback received, and resubmit the revised request for review. This iterative process allows users to address any concerns raised by administrators and increase the likelihood of approval for their fund requests. Additionally, the dashboard may display a history of all previous requests submitted by the user, providing a comprehensive overview of their fund management activities over time.

2.3 AI Chatbot

- **Generating answers for user questions:** The AI chatbot is equipped with natural language processing (NLP) capabilities that enable it to analyze and interpret user queries. Using advanced algorithms, the chatbot generates accurate and contextually relevant responses to a wide range of questions posed by users. These responses may include providing information, answering inquiries, offering assistance with tasks, or directing users to relevant resources within the system. The chatbot continuously learns from user interactions and feedback, refining its responses over time to improve accuracy and effectiveness.
- **Greeting capabilities:** The chatbot is programmed with greeting capabilities to initiate and maintain interactive conversations with users in a friendly and engaging manner. Upon user interaction, the chatbot greets the user with a personalized message or salutation, creating a welcoming atmosphere for the conversation. Greeting capabilities may include recognizing different greetings such as "hello," "hi," or "good morning," and responding appropriately based on the context of the interaction. This helps to establish rapport and facilitate smoother communication between the user and the chatbot.
- **Interactive chatbot sessions:** The chatbot offers interactive chat sessions that allow users to engage in dynamic conversations and receive real-time responses to their queries. Through a user-friendly chat interface, users can interact with the chatbot in a conversational manner, typing or speaking their questions and receiving instant replies. The chatbot employs conversational AI techniques to maintain context and coherence throughout the conversation, enabling users to seamlessly navigate through different topics and tasks. Interactive chatbot sessions may involve asking follow-up questions, providing clarifications, or guiding users through step-by-step processes to accomplish their objectives.
- **Personalization and customization:** The chatbot is designed to personalize interactions and tailor responses based on user preferences, history, and behavior. By analyzing user data and past interactions, the chatbot can adapt its responses to match the user's individual needs and preferences. Personalization features may include remembering user preferences, recommending relevant content or services, and adjusting conversation style or tone based on user feedback. Additionally, users may have the option to customize the chatbot experience by setting preferences for language, topic preferences, or interaction preferences, allowing for a more personalized and user-centric experience.

- **Multi-turn conversations:** The chatbot supports multi-turn conversations, allowing users to engage in complex dialogues spanning multiple interactions. Users can ask follow-up questions, provide additional context, or navigate through different branches of conversation to achieve their objectives. The chatbot maintains context and memory of previous interactions, ensuring continuity and coherence throughout the conversation. Multi-turn conversations enable users to explore topics in depth, address multiple inquiries within a single session, and accomplish complex tasks with ease.

3. Product Functionalities :

3.1 Fund Request and Approval Workflow

The system streamlines the fund request and approval process by automating the generation, submission, review, and decision-making stages, reducing manual intervention and accelerating the overall workflow. Through seamless integration of user-friendly interfaces and backend algorithms, the system ensures efficient handling of fund requests while maintaining transparency, accountability, and compliance with organizational policies:

Audit Trails:

- **User activities:** The system maintains detailed audit trails of user activities, capturing a comprehensive record of actions performed by individual users within the system. This includes logging user login/logout events, form submissions, fund request approvals/rejections, profile updates, and any other interactions with system functionalities. Each logged event is timestamped and attributed to the corresponding user, providing visibility into user behavior and facilitating accountability.
- **System changes:** Audit trails track system changes, documenting any modifications, updates, or configurations made to the system settings or infrastructure. This encompasses changes to system parameters, security settings, access controls, software updates, and any other modifications that impact system functionality or behavior. By recording system changes, the audit trail enables administrators to monitor system health, track performance optimizations, and identify potential security vulnerabilities or configuration errors.
- **Data access:** Audit trails monitor data access activities, recording instances of data retrieval, modification, or deletion by authorized users. This includes tracking access to sensitive data repositories, database queries, file downloads/uploads, and data exports. Audit logs capture details such as the user accessing the data, the type of data accessed, the timestamp of access, and the purpose of access. By monitoring data access, the audit trail helps ensure compliance with data privacy regulations, detect unauthorized access attempts, and facilitate forensic investigations in the event of security incidents or data breaches.
- **Granular event logging:** The audit trail provides granular event logging capabilities, allowing administrators to customize the level of detail captured for each type of activity. Administrators can configure audit trail settings to specify which events are logged, the level of verbosity for each event type, and the retention period for audit logs. This flexibility enables organizations to tailor audit trail settings to their specific security and compliance requirements, balancing the need for detailed event logging with considerations such as storage space and performance impact.

- **Integrity and tamper-evidence:** Audit trails are designed to maintain data integrity and tamper-evidence, ensuring the reliability and trustworthiness of logged events. This involves implementing cryptographic techniques such as hashing and digital signatures to protect audit log data from unauthorized tampering or manipulation. Additionally, audit logs may be stored in secure, tamper-evident repositories with restricted access controls to prevent unauthorized modifications or deletions. By preserving the integrity of audit trail data, organizations can rely on audit logs as credible sources of evidence for compliance audits, legal proceedings, and internal investigations.

Fund Request Submission:

- **Online form with necessary details:** The fund request submission process begins with students accessing an online form specifically designed for submitting fund requests. The form is structured to capture essential details required for processing the request, such as the purpose of the funding, requested amount, date needed, and any additional comments or explanations. The form may include dropdown menus, text fields, checkboxes, and other input options to accommodate various types of information. Mandatory fields are indicated to ensure that all required information is provided by the student before submission.
- **Attachment of supporting documents:** In addition to filling out the form, students have the option to attach supporting documents relevant to their fund request. This may include receipts, invoices, quotes, project proposals, or any other documentation necessary to substantiate the request and provide context to reviewers. The system supports file attachments in various formats, allowing students to upload documents directly from their devices. Uploaded files are securely stored and associated with the corresponding fund request for easy reference and review by administrators during the approval process.
- **Storage of submitted data as individual objects:** Upon submission, the data entered by the student, along with any attached supporting documents, is stored in the system's database as individual objects or records. Each fund request is represented as a distinct database entry containing all relevant information provided by the student, such as request details, attached files, submission timestamp, and unique identifier. This organized data structure enables efficient retrieval, processing, and management of fund requests by administrators, ensuring that each request is accurately documented and easily accessible for review and decision-making.
- **Validation and error handling:** The fund request submission process includes validation checks to ensure the accuracy and completeness of the information provided by students. The system performs real-time validation of form inputs to detect any errors or inconsistencies, such as invalid data formats or missing required fields. Error messages are displayed to the student indicating areas that require correction, allowing them to revise their submission before finalizing it. Additionally, the system may incorporate client-side and server-side validation mechanisms to prevent submission of incomplete or erroneous requests and maintain data integrity within the database.

Notification to Authorities:

- **Automated email notifications:** Upon submission of a fund request by a student, the system triggers automated email notifications to designated authorities, including the Project Supervisor, Head of Department (HOD), and representatives from ACES. These notifications are generated instantaneously and sent to the respective email addresses configured within the system. The emails contain pertinent details about the submitted request, such as the requester's name, request ID, purpose of the request, requested amount, and any attached supporting documents. The notifications serve as alerts to inform authorities about new fund requests requiring their review and approval.

- **Direct link to review request details:** In addition to receiving email notifications, authorities are provided with a direct link or access to the system's interface where they can review the details of the submitted fund request. The link embedded within the email redirects authorities to a designated portal or dashboard within the system, specifically tailored for reviewing and managing fund requests. Upon accessing the portal, authorities can view comprehensive information about the request, including the form submissions, attached documents, submission timestamp, and any additional comments provided by the student. This direct access streamlines the review process and facilitates prompt action by the authorities.
- **Customizable notification preferences:** The system offers customizable notification preferences, allowing authorities to configure their preferred method of receiving notifications (e.g., email). Authorities can specify their preferred notification channels and frequency of alerts based on their availability and preferences. Additionally, authorities may have the option to opt-in or opt-out of specific notification types or categories, ensuring that they receive only relevant and actionable alerts pertaining to fund requests requiring their attention.
- **Escalation mechanisms:** In cases where timely action is critical, the system may incorporate escalation mechanisms to ensure that fund requests are reviewed promptly by designated authorities. If a request remains pending for an extended period without action from the assigned authorities, the system may automatically escalate the request to higher-level authorities or trigger reminders/alerts to notify the responsible individuals about the pending review. Escalation mechanisms help prevent bottlenecks in the approval process and ensure that fund requests are processed in a timely manner, minimizing delays and enhancing overall efficiency.

Review and Decision:

- **Approvers can add comments before giving decisions:** Approvers are provided with the capability to add comments or feedback before making a decision on fund requests. This feature allows approvers to provide contextual information, clarification, or rationale behind their decision, which can be viewed by the requesters. By enabling communication between approvers and requesters through comments, the system promotes transparency and fosters a collaborative decision-making process. Requesters gain insights into the reasoning behind the decisions, enabling them to understand the outcome and make informed adjustments to future requests if needed.
- **Multi-level approval process:** The system facilitates a multi-level approval process where authorities review fund requests sequentially and provide decisions ("Yes" or "No") based on their assessment. Each request undergoes review by designated authorities in a predetermined hierarchy, such as Project Supervisor, Head of Department (HOD), and ACES representatives, before reaching a final decision. The multi-level approval process ensures thorough scrutiny of fund requests and allows for input from multiple stakeholders with varying levels of authority and expertise. Decisions are aggregated at each level, with subsequent approvers building upon the feedback provided by their predecessors.

- **Secure storage of decisions in the database:** Decisions made by authorities are securely stored in the system's database, ensuring the integrity and confidentiality of sensitive information. Each decision is recorded as a structured data object containing details such as the request ID, decision outcome ("Yes" or "No"), comments provided by the approver, timestamp of decision, and identity of the approving authority. Decisions are stored in an encrypted format to prevent unauthorized access or tampering, and access controls are implemented to restrict visibility to authorized personnel only.
- **Notifications to users:** Upon completion of the review and decision-making process, notifications are sent to users informing them of the outcome of their fund request. If the request is approved, users receive a notification confirming the approval along with any relevant instructions or next steps. Conversely, if the request is rejected, users are notified of the decision and may optionally receive feedback or comments provided by the approver. Notifications are delivered via email based on the user's preferred communication channel, ensuring timely and transparent communication of decision outcomes to requesters.

3.2 Request Process Tracking

The system tracks the status of fund requests:

- **States:** "PENDING," "ACCEPTED," "DENIED": The system categorizes each fund request into one of three distinct states to clearly indicate its current status:
  - PENDING: This status is assigned to newly submitted requests that are awaiting review and decision by the authorities. It signifies that the request is in progress and has not yet been finalized.
  - ACCEPTED: This status is assigned to requests that have been reviewed and approved by the necessary authorities. An "ACCEPTED" status indicates that the requested funds will be disbursed according to the established procedures.
  - DENIED: This status is assigned to requests that have been reviewed but not approved by the authorities. A "DENIED" status indicates that the request did not meet the criteria for approval, and no funds will be disbursed.
- **Dynamic Updates:** Real-time status changes based on authorities' decisions: The system provides real-time updates to the status of fund requests, dynamically reflecting the decisions made by authorities. As each approver reviews and acts on a request, the status is updated immediately within the system. This ensures that requesters and relevant stakeholders have access to the most current information regarding the progress and outcome of each request. Users can log in to their dashboards at any time to see the live status of their submissions, providing transparency and up-to-date tracking throughout the approval process.
- **Re-application:** Students can reapply for denied requests with necessary adjustments: For requests that receive a "DENIED" status, the system offers the functionality for students to reapply with necessary adjustments. This feature enables students to review the feedback provided by authorities, make the required changes or improvements to their request, and resubmit it for consideration. The re-application process is streamlined to allow students to update the original request form, attach additional or revised supporting documents, and provide any new information that addresses the reasons for the initial denial.

This iterative process helps students refine their requests, increasing the likelihood of approval upon resubmission and fostering continuous improvement in fund request quality.

- **Detailed status history and audit log:** The system maintains a detailed history of status changes and decisions for each fund request. This audit log includes timestamps, the identity of the authorities who reviewed the request, the specific actions taken (e.g., approved, denied, comments added), and the progression of the request through the different approval stages. Users can access this history to understand the complete lifecycle of their fund requests, from submission to final decision. This comprehensive tracking ensures accountability and provides a transparent record for both requesters and administrators to reference as needed.

Security Measures:

- **User Authentication Filtering:** Access is restricted to authorized users: The system employs robust user authentication mechanisms to ensure that only authorized users can access and interact with the fund management application. This includes implementing secure login protocols such as Google OAuth 2.0, which restricts access to users with verified @eng.pdn.ac.lk email addresses. During the authentication process, user credentials are encrypted to protect sensitive information from being intercepted or compromised. Additionally, multi-factor authentication (MFA) may be employed to add an extra layer of security, requiring users to verify their identity through a secondary method such as a mobile app or SMS code.
- **Admin Access Control:** Controlled entry for administrators: Access control measures are in place to regulate and monitor the entry of administrators into the system. Administrators are granted specific roles and permissions based on their responsibilities, ensuring that they can only access and perform actions relevant to their duties. The system uses role-based access control (RBAC) to define and enforce these permissions, preventing unauthorized access to sensitive areas of the system. Admin activity is logged and audited, providing a traceable record of actions taken within the system to ensure accountability and compliance with security policies.
- **Secure Approval Process:** Text-based confirmation to prevent accidental actions: To enhance the security and integrity of the approval process, the system incorporates a text-based confirmation mechanism. Before finalizing any approval or rejection of fund requests, administrators are required to enter a confirmation code sent to their registered device (e.g., via email or SMS). This additional step ensures that approvals and rejections are intentional and prevents accidental or unauthorized actions. The confirmation process is designed to be user-friendly while providing a robust safeguard against inadvertent decisions. Furthermore, the system can include time-limited tokens for these confirmations, adding an extra layer of security by ensuring that the codes expire after a short period, reducing the risk of misuse.
- **Data encryption and secure storage:** All sensitive data within the system, including user credentials, fund request details, and decision logs, is encrypted both in transit and at rest. The system utilizes strong encryption standards such as TLS (Transport Layer Security) for data transmission and AES (Advanced Encryption Standard) for data storage. This ensures that even if data is intercepted or accessed without authorization, it remains unreadable and secure. Additionally, the system employs secure storage solutions with regular security audits and compliance checks to maintain data integrity and confidentiality.

- **Regular security audits and updates:** The system undergoes regular security audits and updates to identify and mitigate potential vulnerabilities. These audits are conducted by security experts who evaluate the system's defenses, penetration test its components, and recommend improvements. The system is also regularly updated with the latest security patches and enhancements to protect against emerging threats. By maintaining a proactive security posture, the system ensures ongoing protection of user data and operational integrity.

Syncing:

- **Set limits per user:** The system enforces limits on the amount and frequency of fund requests that each user can submit, based on predefined criteria. These limits can be configured by administrators and may vary depending on the user's role, department, or previous request history. For instance, students may have a cap on the total amount they can request per semester, while project leaders may have higher thresholds. By setting these limits, the system helps manage resources effectively and prevents misuse or overuse of funds. Additionally, users are notified of their remaining available limits when submitting new requests, ensuring transparency and informed decision-making.
- **Sync approved requests with accounting or financial systems:** Once fund requests are approved, the system automatically syncs the details with the institution's accounting or financial management systems. This integration ensures that financial records are updated in real-time, reducing manual data entry and minimizing the risk of errors. The synchronization process involves transferring relevant data such as the request ID, approved amount, requester details, and purpose of the fund to the financial system. This seamless integration facilitates accurate and efficient financial reporting, budgeting, and auditing processes.
- **Update Budget Balances:** The system continuously updates budget balances to reflect the latest financial activities. When a fund request is approved, the allocated amount is deducted from the relevant budget category. Conversely, if a request is denied or canceled, the system restores the allocated amount to the budget. This dynamic updating ensures that budget balances are always current and accurately reflect the organization's financial status. Users with appropriate permissions can view real-time budget summaries and detailed transaction histories, enhancing financial transparency and oversight.
- **Ensure requests align with available budget:** Before approving any fund requests, the system checks to ensure that the requested amount aligns with the available budget. This involves comparing the requested funds against the remaining budget balance for the specific category or project. If the requested amount exceeds the available funds, the system flags the request and may either prevent submission or notify the approvers of the discrepancy. This budget alignment check helps prevent overspending and ensures that financial resources are allocated responsibly and within the organization's budgetary constraints.
- **Audit and compliance tracking:** The system maintains detailed records of all synchronized transactions and budget updates to support audit and compliance requirements. These records include timestamps, user identities, request details, and financial adjustments. Audit trails provide a comprehensive view of financial activities, enabling administrators to trace and verify transactions during audits or reviews. Compliance tracking ensures that all financial operations adhere to institutional policies and regulatory standards, promoting accountability and governance in fund management.

- **User notifications for budget status:** Users are notified of the budget status and availability when submitting fund requests. The system provides real-time feedback on whether the request can be accommodated within the available budget. If the budget is insufficient, users receive alerts and may be prompted to adjust their request accordingly. This proactive notification system helps users make informed decisions and align their requests with the organization's financial capabilities.

Administration Purposes

- **Generate reports on fund utilization:** The system provides comprehensive reporting tools that enable administrators to generate detailed reports on fund utilization. These reports can include various metrics such as total funds requested, approved, and disbursed over specific periods. Administrators can filter and sort data by different criteria, such as department, project, requester, or fund category. Reports may also include visual representations, such as graphs and charts, to facilitate easy analysis and interpretation of fund utilization patterns. This capability helps administrators monitor spending, identify trends, and make informed decisions about future budget allocations.
- **Generate reports on pending requests:** Administrators can generate reports specifically focused on pending fund requests. These reports provide insights into the number of requests awaiting approval, the total amount of funds pending, and the time duration each request has been in the pending state. The system can group pending requests by various parameters, such as requester, department, or priority level, and highlight any bottlenecks in the approval process. By having a clear overview of pending requests, administrators can take proactive measures to expedite reviews and ensure timely decision-making.
- **Generate reports on historical data:** The system allows administrators to access and generate reports based on historical data, providing a long-term view of fund management activities. Historical reports can include data from previous fiscal years, allowing for year-over-year comparisons and trend analysis. These reports may cover various aspects such as total funds requested and approved, frequency of fund requests, and historical approval rates. Historical data analysis helps administrators understand past funding patterns, evaluate the effectiveness of previous budget allocations, and plan more accurately for future financial needs.
- **Customizable report parameters:** The reporting tools offer customization options, allowing administrators to define specific parameters and criteria for the reports. For example, administrators can set custom date ranges, select specific departments or projects, and choose particular metrics to be included in the report. Customizable reports enable administrators to focus on specific areas of interest and obtain tailored insights that are relevant to their needs. Additionally, the system supports exporting reports in various formats (e.g., PDF, Excel, CSV) for further analysis or sharing with stakeholders.
- **Automated report scheduling:** The system can automate the generation and distribution of reports based on predefined schedules. Administrators can set up regular intervals (e.g., weekly, monthly, quarterly) for reports to be generated and sent to specific recipients via email or made available in the system's dashboard. Automated scheduling ensures that stakeholders receive timely updates on fund management activities without the need for manual intervention, improving efficiency and consistency in reporting.
- **Real-time data access and dashboards:** In addition to generating static reports, the system provides real-time access to fund management data through interactive dashboards. These dashboards offer a

dynamic view of key metrics and performance indicators, updating automatically as new data is entered into the system. Administrators can customize their dashboards to display the most relevant information, such as current budget balances, recent fund requests, and approval timelines. Real-time data access enables administrators to make informed decisions quickly and respond to financial developments as they occur.

- **Compliance and audit readiness:** The reporting tools support compliance and audit readiness by maintaining detailed records of all financial transactions and fund management activities. Reports can be generated to demonstrate adherence to organizational policies, financial regulations, and audit requirements. Comprehensive documentation of fund requests, approvals, and expenditures ensures transparency and accountability, facilitating smooth audit processes and reinforcing trust in the system's integrity.

3.2 Non - Functional Requirements

The system is meticulously engineered to manage a substantial volume of requests and user interactions simultaneously, ensuring optimal performance without significant slowdowns. It employs advanced load balancing and efficient database management techniques to maintain swift response times even under peak usage. Security is paramount, with robust encryption protocols safeguarding user and financial data during storage and transmission, and stringent access controls enforced based on user roles to prevent unauthorized access. Reliability is achieved through a high uptime architecture that incorporates failover mechanisms and automated recovery processes, enabling quick restoration from any disruptions. The user interface is designed with usability in mind, featuring a clear, intuitive layout that simplifies navigation and enhances the user experience. Furthermore, the system is built with scalability at its core, capable of seamlessly expanding to accommodate growing numbers of users and increasing data volumes without compromising performance or reliability, ensuring it remains a viable solution as the organization evolves.

3.3 Domain Requirements

The system is designed to ensure full compliance with all pertinent financial regulations and standards, incorporating regulatory requirements into its operational framework to guarantee legal adherence and financial integrity. It offers comprehensive auditability by maintaining a transparent and detailed audit trail for all approvals and financial transactions, capturing crucial data points such as timestamps, user identities, actions taken, and decision rationales. This meticulous record-keeping facilitates both internal reviews and external audits. Budget alignment is seamlessly integrated into the system's workflow, with automated checks that compare each fund request against available budget allocations, ensuring fiscal responsibility and preventing overspending. Furthermore, the system's robust reporting capabilities enable administrators to generate detailed reports on various aspects of fund management, including real-time fund utilization, status of pending requests, and historical financial data. These reports are customizable and can be scheduled for regular delivery, providing critical insights and supporting informed decision-making processes.

4. Technical Specifications

4.1 Technological Stack

The project leverages the MERN stack (MongoDB, Express.js, React, Node.js), which offers a comprehensive solution for developing modern web applications.

MongoDB

- Description: A NoSQL database known for its flexibility and scalability.
- Role: Stores all data related to fund requests, user authentication, and system logs.
- Benefits: Schema-less nature allows for easy modification of data structures as the project evolves.

Express.js

- Description: A minimal and flexible Node.js web application framework.
- Role: Handles the server-side logic, routes, and API endpoints.
- Benefits: Simplifies the development process with robust middleware options and easy integration with other modules.

React

- Description: A JavaScript library for building user interfaces.
- Role: Creates dynamic and responsive frontend components for the web application.
- Benefits: Allows for reusable components, efficient state management, and a virtual DOM for fast updates.

Node.js

- Description: A JavaScript runtime built on Chrome's V8 JavaScript engine.
- Role: Executes JavaScript code server-side, enabling the backend of the application.
- Benefits: Handles asynchronous operations efficiently, making it ideal for scalable network applications.

4.2 Integration

The AFMS will incorporate API integration to facilitate seamless fund management operations. This will allow for efficient data exchange between various systems, enhancing the overall functionality of the project fund management. Future integration possibilities include connecting with the ESCAL inventory system to ensure comprehensive resource management across ACES activities, and streamlining both financial and material assets tracking.

4.3 Scalability and Performance

The architecture of AFMS is designed with scalability in mind, employing techniques such as load balancing and horizontal scaling to manage increasing data volumes and user demands efficiently. By utilizing microservices, the system can be scaled independently, ensuring continuous performance optimization. Additionally, caching mechanisms will be implemented to enhance data retrieval speeds and reduce server load, ensuring real-time updates and a responsive user experience.

4.4 User Experience

AFMS will feature an intuitive, user-friendly interface designed to cater to various devices, including desktops, tablets, and mobile phones. The user experience is central to the application, ensuring ease of navigation and accessibility. By employing modern UI/UX design principles, the system aims to provide a seamless and efficient interaction for ACES members, facilitating quick fund requests and approvals.

4.5 Compliance

The development of AFMS will adhere to all relevant financial regulations and data protection laws to ensure the secure handling of sensitive information. This includes compliance with standards such as GDPR for data privacy and any specific financial regulations pertinent to the University's operations. Robust security measures, including secure authentication and encrypted data storage, will be implemented to protect user data and maintain integrity throughout the fund management process.

4.6 Diagrams

Use Case Diagram 🧷

A Use Case Diagram is a visual representation that outlines the interactions between users (actors) and the system to achieve a goal. In the context of the ACES Fund Management System (AFMS), the Use Case Diagram will illustrate various user roles (e.g., students, project supervisors, administrators) and their interactions with the system, such as submitting fund requests, approving requests, and tracking request statuses.

Activity Diagram 🧷

An Activity Diagram shows the dynamic aspects of the system by depicting the workflow of activities. For AFMS, the Activity Diagram will detail the sequence of actions involved in the fund request process, from submission by students to review and approval by authorities, and notification of outcomes. It highlights the flow of control from one activity to another and any decision points in the process.

Class Diagram🧷

A class diagram is a type of static structure diagram in the Unified Modeling Language (UML) that describes the structure of a system by showing its classes, their attributes, operations (or methods), and the relationships among objects. It is a crucial part of object-oriented modeling and is used to visually represent and design the system architecture.

5. Implementation Plan

5.1 Resource Allocation

- **Project Manager** - Ms. Yasodha Vimukthi
- **Team Lead** - E/19/111 M.D. Galappaththi
- **Software Developing Lead** - E/19/227 Jeewajith Madusanka, E/19/091 P. A. M. Dissanayake
- **QA, ML, and AI Lead** - E/19/166 W. W. K. Jayathunga
- **UI/UX Designing Lead** - E/19/017 W. S. Amanda
- **Deployment and Operations Lead** - E/19/111 M.D. Galappaththi
- **Marketing Specialist** - E/17/083 Mahela Ekanayake
- **Maintenance and Support Specialist** - E/19/074 - B.A.M.I.E Dharmarathne
- **Documentation Specialist** - E/19/304 - R.M.S.P.Pushpakumara

5.2 Development Plan

- **UI/UX Design:** Initial design and planning.
- **Frontend Development:** Start and continue development.
- **Backend Development:** Build and integrate server-side functionalities.
- **Database Integration:** Connect the backend with MongoDB.
- **Testing and Feature Addition:** Comprehensive testing and adding extra features.
- **Documentation:** Track the progress, Api Documentation and Manuals
- **Final Touches:** Ensure system readiness for deployment and monitor software performance and make necessary adjustments.
- **Marketing:** Develop and implement marketing strategies, conduct market research and analyze trends and Promote the project to potential users and stakeholders

5.3 Deployment Plan

A detailed deployment plan ensures a smooth transition of the AFMS from development to production. This plan includes pre-deployment, deployment, and post-deployment phases, with a focus on ensuring minimal disruption, user readiness, and system stability.

Pre-Deployment Phase

1. Environment Setup

- Development Environment: Ensure a stable development environment mirroring the production setup.

- Staging Environment: Set up a staging environment identical to production to test the entire application.

2. Code Review and Testing

- Code Review: Conduct thorough code reviews to ensure compliance with coding standards and best practices.
- Unit Testing: Validate individual components through unit testing.
- Integration Testing: Ensure integrated modules work seamlessly together.
- User Acceptance Testing (UAT): Engage ACES members, including admins, approvers, and requesters, to test the application and provide feedback.

3. Data Migration and Preparation

- Data Backup: Securely back up existing data from the current system.
- Data Cleansing: Cleanse the data to eliminate inconsistencies.
- Data Migration: Migrate the data to the new system, ensuring data integrity and accuracy.

4. Security and Compliance Checks

- Vulnerability Assessment: Conduct a security assessment to identify and resolve vulnerabilities.
- Compliance Verification: Ensure the system complies with relevant regulations and university policies.

5. Documentation Preparation

- Technical Documentation: Create comprehensive technical documentation for developers and administrators.
- User Manuals: Develop user manuals and guides for different user roles within ACES.
- Training Materials: Prepare training materials, including video tutorials.

Deployment Phase

1. Deployment Planning

- Deployment Schedule: Plan the deployment during off-peak hours to minimize disruption to ACES activities.
- Rollback Plan: Prepare a rollback plan to revert changes if any issues arise during deployment.

2. Deployment Execution

- Server Configuration: Configure production servers, including load balancers, databases, and application servers.
- Code Deployment: Deploy the application code to the production environment using CI/CD tools.
- Database Update: Apply necessary database schema changes and verify data migration accuracy.
- Configuration Management: Update configuration files and environment variables to match the production setup.

3. Verification and Testing

- Smoke Testing: Perform smoke tests to verify core functionalities post-deployment.
- Performance Testing: Conduct performance tests to ensure the system handles the expected load.

Post-Deployment Phase

1. Monitoring and Support

- Monitoring: Implement monitoring tools (e.g., New Relic, Datadog) to track application performance, error logs, and user activities.
- Incident Management: Establish an incident management process to handle and resolve issues promptly.

2. User Training

- Training Sessions: Conduct training sessions for ACES admins, approvers, and requesters to familiarize them with the new system.
- Interactive Workshops: Organize hands-on workshops to provide practical experience.

3. Feedback and Iteration

- Feedback Collection: Collect feedback from users regarding their experience and any issues encountered.
- Issue Resolution: Prioritize and address reported issues promptly.
- Iterative Improvements: Implement improvements based on user feedback and performance data.

4. Documentation and Knowledge Base

- Knowledge Base: Set up an online knowledge base with tutorials, troubleshooting guides, and FAQs for user self-service.

- Support Channels: Provide multiple support channels (email, chat, phone) for user assistance.

5. Performance Review

- Review Meetings: Conduct regular review meetings to assess the system's performance and identify areas for enhancement.
- Report Generation: Generate reports on system usage, performance metrics, and user satisfaction to inform future updates.

6. Conclusion

The AFR project aims to revolutionize fund management for ACES by introducing a modern, efficient, and secure web application. Leveraging the MERN stack, this system will enhance data security, user experience, and operational transparency, setting a strong foundation for future improvements in financial management within the organization.