

Overview

The Contract Monthly Claim System (CMCS) was created to help simplify and manage contract claims, approvals, and status monitoring for several positions within a business, including User/Lecturer, Manager/Coordinator, and HR. The system is designed utilizing the ASP.NET MVC architecture, which ensures a clear separation of data, business logic, and user interface. The user experience has been optimized for mobile devices, making it easy to access across several platforms.

1. Design Choices

1.1 MVC Architecture

The MVC (Model-View-Controller) architecture was chosen for its strength and flexibility in managing various user roles while keeping the application's logic modular and scalable. (Robert Sheldon 2023)

Model: Contains the data and business rules for the claims system. This involves maintaining lecturer information, claim data, and approvals. The approach protects data integrity by encapsulating information such as hourly rates, claim amounts, and approval statuses. (kaalel 2024)

View: This is the interface via which users interact (Lecturer, Manager, and HR). It offers several layouts and components depending on the user role. For example, a lecturer will see claim filing forms, but an HR officer will see claim approvals and invoice generation.

Controller: It processes user input and guides workflow depending on the role. For example, it oversees the filing of claims by Lecturers, the approval procedure for Managers, and the preparation of invoices by HR.

kaalel,08 Jul 2024, MVC Framework Introduction, Available on [https://www.geeksforgeeks.org/](https://www.geeksforgeeks.org/mvc-framework-introduction/) at: <https://www.geeksforgeeks.org/mvc-framework-introduction/>

Robert Sheldon, September 2023,model-view-controller (MVC), Available on [https://www.techtarget.com/](https://www.techtarget.com/whatis/definition/model-view-controller-MVC) at: <https://www.techtarget.com/whatis/definition/model-view-controller-MVC>

1.2 Language and Framework

Language: C# was chosen because of its reliability, compatibility with the .NET ecosystem, and strong support for MVC-based applications.

Framework: ASP.NET MVC was selected for its clean separation of concerns, allowing each role (User, Manager, HR) to have distinct functions with minimal code overlap.

Simplilearn,29 July 2024, Understand all about ASP.NET MVC, Available on [https://www.simplilearn.com/](https://www.simplilearn.com/tutorials/asp-dot-net-tutorial/what-is-asp-dot-net-mvc) at: <https://www.simplilearn.com/tutorials/asp-dot-net-tutorial/what-is-asp-dot-net-mvc>

1.3 User Interface (UI) Design

The UI was created with a mobile-first approach to guarantee that the system is accessible and user-friendly across all devices.

The design incorporates a navbar that changes to the user's job, linking to appropriate functionalities (e.g., claims, approval, and profile management).

Each job has a unique perspective, ensuring that users only see information that is relevant to them (for example, Managers may approve claims, whereas Lecturers can submit claims).



2. Database Structure

2.1 Tables

1. User (Base Table for all roles)

UserID (Primary Key): Auto-increment identifier for each user (common to Lecturer, Coordinator, Manager, HR).

Name (String): Full name of the user.

Email (String): Contact email for login and notifications.

UserType (Enum): Defines the type of user (Lecturer, Coordinator, Manager, HR).

2. Lecturer

Inherits **UserID** (Foreign Key) from the **User** table.

ContactNo (String): Contact number of the lecturer.

BankDetails (String): Secure bank details for payment processing.

HourlyRate (Decimal): Lecturer's hourly rate, used to calculate claims.

ProfilePicture (Blob): Profile picture for the Lecturer's profile view.

3. Coordinator

Inherits **UserID** (Foreign Key) from the **User** table.

4. Manager

Inherits **UserID** (Foreign Key) from the **User** table.

No additional fields are necessary, as all attributes and methods are inherited from the **User** class.

5. HR

Inherits **UserID** (Foreign Key) from the **User** table.

SalaryBudget (Decimal): The HR's budget allocation for salaries and payments.

6. Claims

ClaimID (Primary Key): Auto-increment identifier for each claim.

LecturerID (Foreign Key): Links to the Lecturer who made the claim.

HoursWorked (Float): Total number of hours worked by the lecturer.

HourlyRate (Decimal): The lecturer's hourly rate at the time of claim submission.

TotalPayment (Decimal): The total amount the lecturer is claiming, calculated from the hours worked and the hourly rate.

Status (String): The status of the claim (Pending, Approved, Rejected).

DateSubmitted (DateTime): The date the claim was submitted.

7. Documents

DocumentID (Primary Key): Auto-increment identifier for each document.

ClaimID (Foreign Key): Links to the claim the document is associated with.

FileName (String): The name of the document uploaded.

FilePath (String): The file path where the document is stored.

FileType (String): The type of document (PDF, Word, etc.).

FileSize (Float): Size of the file in megabytes (MB).

8. Approvals

ApprovalID (Primary Key): Auto-increment identifier for each approval.

ClaimID (Foreign Key): Links to the claim being approved or rejected.

ManagerID (Foreign Key): Links to the Manager who approved or rejected the claim.

Status (String): Status of the approval (Approved, Rejected).

DateApproved (DateTime): The date the claim was approved or rejected.

2.2 Relationships

1. User and Sub-classes (Lecturer, Coordinator, Manager, HR)

UserID is the common identifier for all user types. Each specific role (Lecturer, Coordinator, Manager, HR) inherits from the **User** table.

2. Lecturers and Claims

One-to-many relationship (One Lecturer can have multiple claims). The **LecturerID** foreign key links each claim to the lecturer who submitted it.

3. Claims and Documents

One-to-many relationship (One claim can have multiple supporting documents). Each document is linked to its respective claim through **ClaimID**.

4. Claims and Approvals

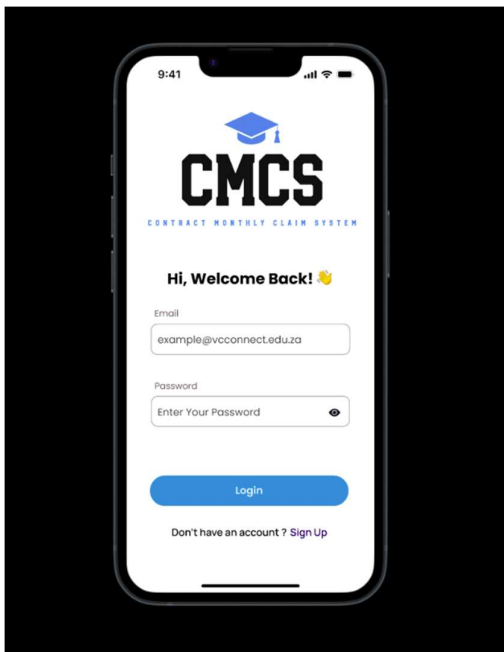
One-to-one relationship (Each claim has one approval decision). The **ClaimID** is linked to the **ApprovalID**, ensuring each claim has a corresponding approval or rejection.

Detailed Breakdown of CMCS Frames

The CMCS application is designed with three distinct roles: **User/Lecturer**, **Manager/Coordinator**, and **HR**. Each role has specific frames (screens) tailored to their functional requirements.

1. USER/LECTURER Frames

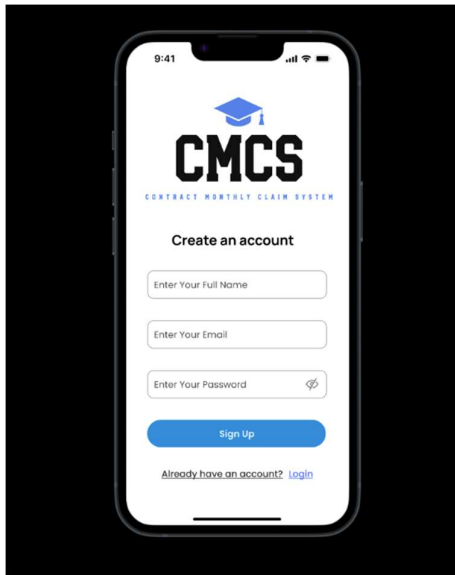
1. Login Page



Functionality: Common for all roles. Users enter their email and password to log into the system. There is also a link to sign up for new users.

Design Choice: The login screen is kept simple with minimal inputs to enhance ease of access. The call-to-action (Login button) is prominent. A welcoming message adds personalization ("Hi, Welcome Back!"). (Marsh & Marzán, 2022)

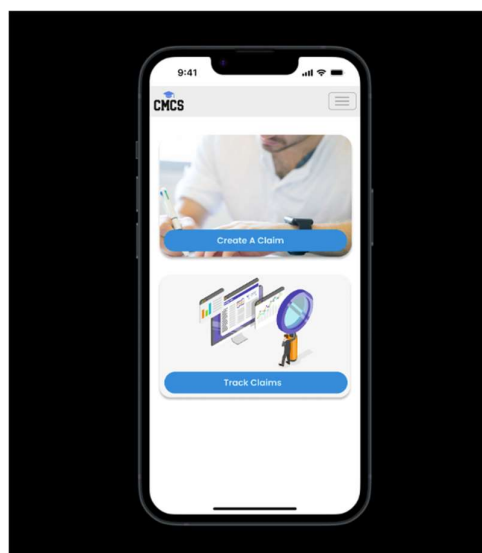
2. Registration Page



Functionality: New lecturers can create an account by entering their full name, email, and password.

Design Choice: The registration form mirrors the login screen's simplicity, providing only necessary inputs to prevent overwhelming new users. The form design follows a minimalistic approach to reduce confusion during sign-up. (Marsh & Marzán, 2022)

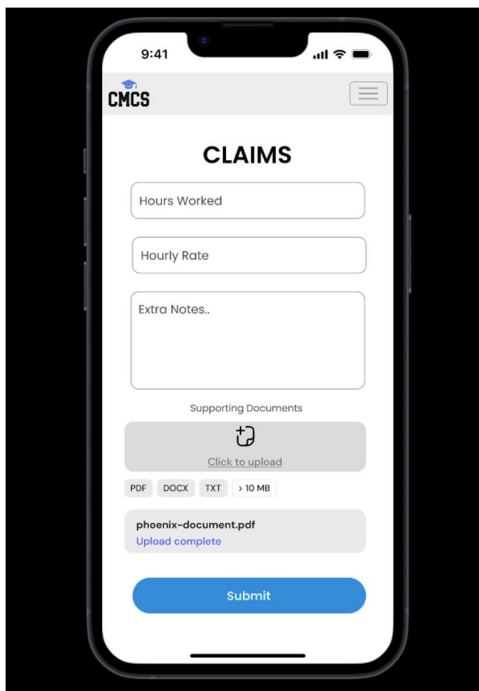
3. Homepage



Functionality: This page serves as the dashboard for lecturers. It contains buttons or sections to navigate to various actions such as submitting claims and viewing previous claims.

Design Choice: The layout is designed with large, clickable buttons for easy navigation on mobile devices. Each button leads to a core function: submitting claims or tracking claims. (Marsh & Marzán, 2022)

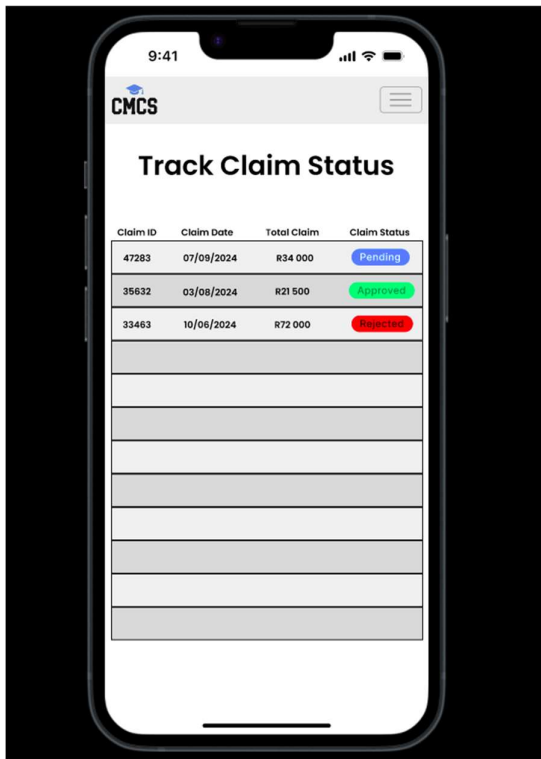
4. Monthly Claim Submission

A screenshot of a mobile application interface for submitting claims. The app is titled 'CMCS' with a graduation cap icon. The screen is titled 'CLAIMS'. It features three input fields: 'Hours Worked', 'Hourly Rate', and 'Extra Notes..'. Below these is a 'Supporting Documents' section with a 'Click to upload' button and a file list showing 'phoenix-document.pdf' with the status 'Upload complete'. At the bottom is a large blue 'Submit' button. The status bar at the top shows the time as 9:41 and signal/battery icons.

Functionality: Lecturers can enter the number of hours worked, the hourly rate, and the total claim amount. Once filled out, they can submit the claim with functionality for uploading supporting documents for approval.

Design Choice: The form is structured with clear labels and input fields, optimized for mobile. Auto-calculation of the total claim amount (based on hours worked and hourly rate) provides immediate feedback to users, making the process quick and intuitive. (Marsh & Marzán, 2022)

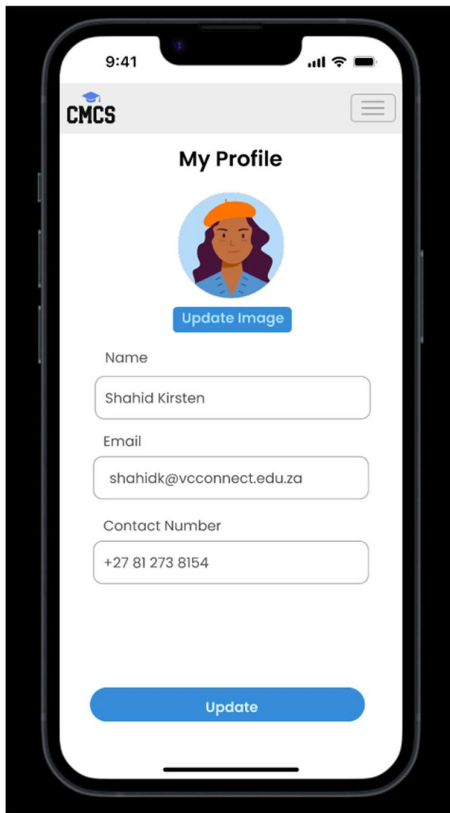
5. Track Claim Status



Functionality: Displays a list of all submitted claims with their status (Pending, Approved, or Rejected). Each claim can be clicked to view further details.

Design Choice: A color-coded table (green for approved, yellow for pending, red for rejected) ensures that users can quickly scan through their claims. The status column is the most prominent feature in the layout, making it easy to track progress at a glance. (Marsh & Marzán, 2022)

6. Profile Page



Functionality: Displays the user's personal information such as name, email, and contact number. Users can update their details if needed.

Design Choice: The design is user-centric, displaying information in large, readable fonts with clearly labelled fields for updating profile details. This page is kept simple, emphasizing ease of use. (Marsh & Marzán, 2022)

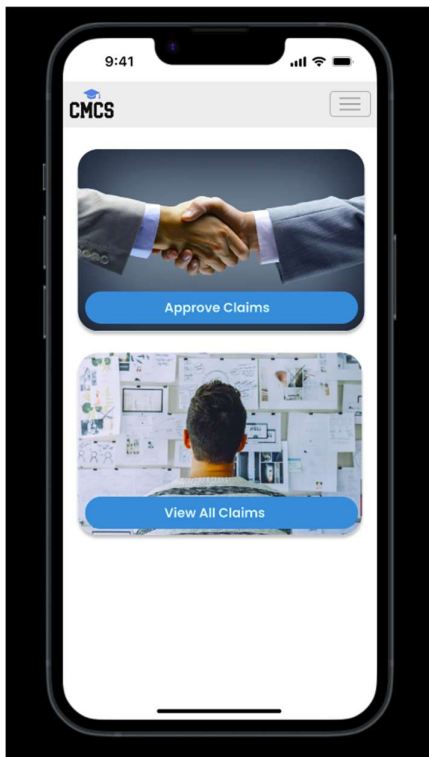
2. MANAGER/COORDINATOR Frames

1. Login Page

Functionality: Common login page for all roles.

Design Choice: Same as explained under the User/Lecturer section.

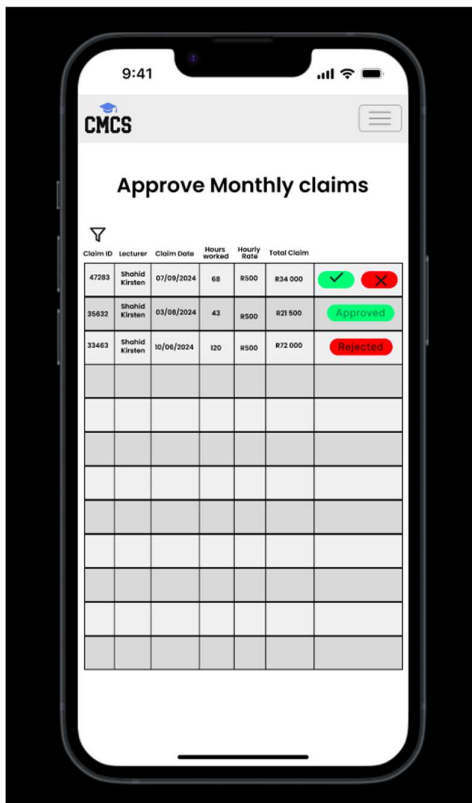
2. Homepage



Functionality: The homepage for Managers/Coordinators includes buttons for viewing claims, approving claims, and managing claims from lecturers.

Design Choice: The layout is designed with action buttons arranged in a hierarchical manner. Claims management actions are highlighted as these are the most frequently used features for this role. (Marsh & Marzán, 2022)

3. Approve Monthly Claims



Functionality: Displays a table of all claims submitted by lecturers, along with the option to approve or reject each claim. Each row in the table shows the lecturer's name, hours worked, the total amount claimed, and the claim's current status.

Design Choice: The table is laid out with clear headers for each column. Action buttons (Approve/Reject) are color-coded (green for approve, red for reject), ensuring quick and easy decision-making. Managers can approve or reject multiple claims from a single screen. (Marsh & Marzán, 2022)

4. View Pending Claims



Functionality: Displays only the claims that are awaiting approval from the Manager/Coordinator. The functionality allows the Manager to review these claims before making a decision.

Design Choice: Pending claims are displayed in a streamlined table format, making it easy to focus on the necessary actions. (Marsh & Marzán, 2022)

5. View Approved Claims



Functionality: Displays all the claims that have been approved, allowing Managers/Coordinators to keep track of approved claims at any time.

Design Choice: Approved claims are displayed in a clean, organized table. Each approved claim is marked in green, making it visually distinct from pending or rejected claims. (Marsh & Marzán, 2022)

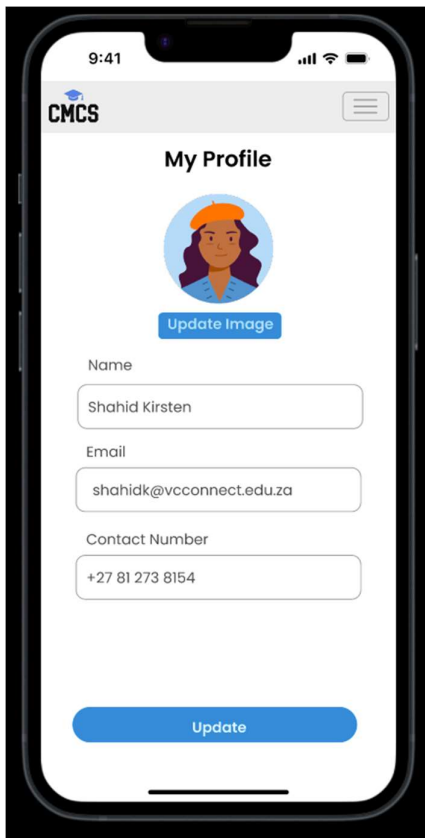
6. View Rejected Claims

[illegible]

Functionality: Displays all the claims that have been rejected by the Manager/Coordinator.

Design Choice: Rejected claims are highlighted in red for quick recognition. The layout includes a comment or reason section next to each rejected claim, explaining why it was denied. (Marsh & Marzán, 2022)

7. Profile Page



Functionality: Similar to the Lecturer profile page but designed for Managers. They can update their personal information such as name, email, and contact details.

Design Choice: The page is kept straightforward, with editable text fields for personal information and a button to save any updates. The profile picture option is available as well. (Marsh & Marzán, 2022)

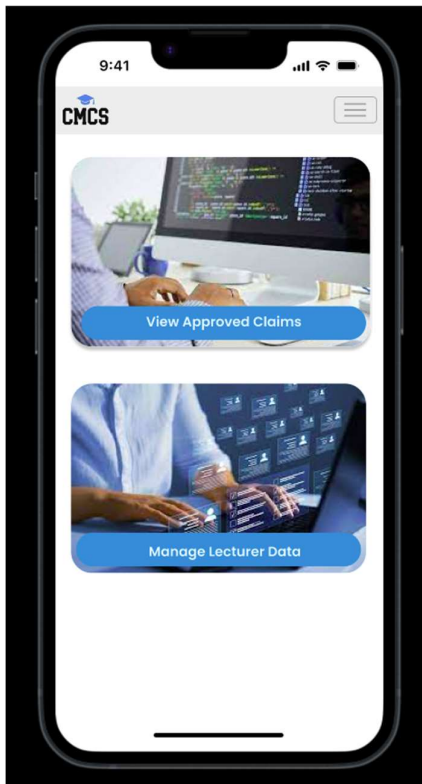
3. HR Frames

1. Login Page

Functionality: Common login page for all roles.

Design Choice: Same as explained under the User/Lecturer section.

2. Homepage



Functionality: The HR homepage includes options to manage lecturer profiles, view approved claims, and generate invoices.

Design Choice: The layout prioritizes claims management, with prominent buttons that guide HR to key tasks such as invoice generation and lecturer data management. (Marsh & Marzán, 2022)

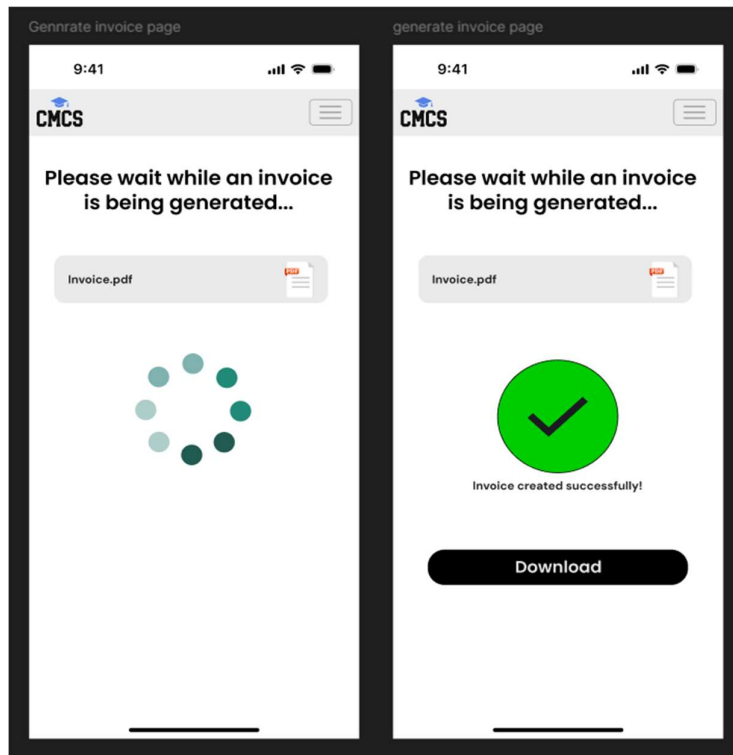
3. View Approved Claims



Functionality: Displays all claims that have been approved across the system. HR can use this view to verify that all claims are ready for invoice generation.

Design Choice: The table is similar to the one seen by Managers, showing detailed information about each claim, but with a specific focus on claims that are approved. Color-coding (green) is used to ensure that only finalized claims are visible. Additionally, a simple generate invoice button has been added for simplicity. (Marsh & Marzán, 2022)

4. Generate Invoice



Functionality: Once the generate invoice button is clicked it takes the HR to these frames. This frame allows HR to generate invoices based on the approved claims. After selecting the relevant claims, the system processes and generates a downloadable invoice.

Design Choice: A progress indicator is shown while the invoice is being generated, followed by a confirmation message once the invoice is ready. A prominent “Download” button is displayed once the invoice is available. The overall design is clean and minimal, focusing on simplicity and efficiency. (Marsh & Marzán, 2022)

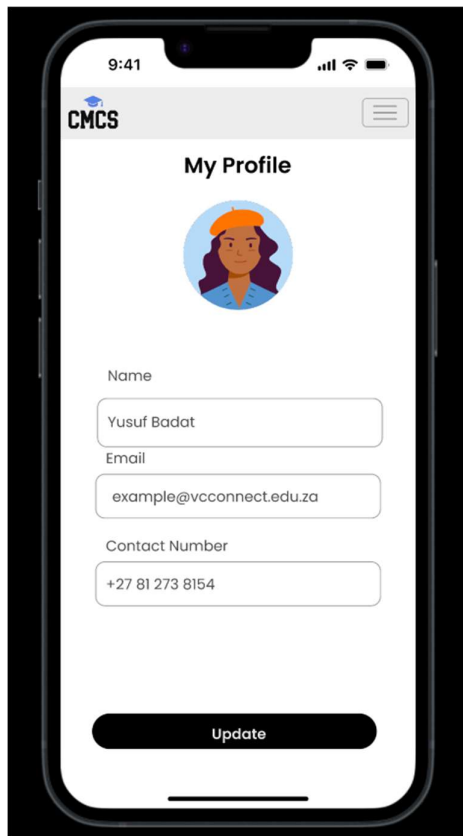
5. Manage Lecturer Data



Functionality: Displays a table of all lecturers in the system, along with options to manage or remove their data. HR can add new lecturers or update existing profiles.

Design Choice: The table is organized with editable fields and action buttons (manage) next to each row. The design emphasizes user-friendliness, with input forms for adding new lecturers kept simple and straightforward. (Marsh & Marzán, 2022)

6. HR Profile Page



Functionality: Similar to the lecturer and manager profile page. Displays the information of the HR. HR can update this information if necessary.

Design Choice: The profile page is designed to provide a clear, comprehensive view of data, with editable fields for HR to make quick adjustments. The layout is similar to the profile page seen by Lecturers and Managers, ensuring consistency across the system.

Marsh, J. and Marzán, J. (2022) *UX for beginners: A crash course in 100 short lessons* Joel Marsh ; illustrations by Jose Marzan, JR. Sebastopol: O'Reilly.

4. Assumptions & Constraints

4.1 Assumptions

1. Data Integrity

It is believed that all users' input data (lecturers, coordinators, managers, and human resources) would be adequately checked before being saved in the database. This contains user information, claim amounts, hours spent, and submitted documents. Lecturers should provide correct claims.

2. Performance

The system is intended for modest use, with a certain number of lecturers and claims processed at any given time. Data storage techniques are designed with performance in mind.

3. User Behaviour

It is believed that users would behave responsibly inside the system, with Lecturers submitting claims on time, Coordinators examining and confirming claims, and Managers doing the required approval steps. HR is supposed to manage the pay budget and lecturer data in a safe and effective manner.

4. Security

Users are expected to secure their login credentials and prevent illegal access. Role-based access restrictions are in place to guarantee that only suitable individuals have access to specific functions (for example, HR handling pay budgets).

5. Technology Stack

It is expected that the system's technologies (ASP.NET Core MVC, Entity Framework, and SQL Server) would be stable and maintained throughout the development and deployment phases. There will be no significant modifications or upgrades to these technologies over the project's duration.

6. User Technical Literacy

It is assumed that users (Lecturers, Coordinators, Managers, HR) have basic technical proficiency and will be able to navigate the web interface, submit claims, and track status without extensive training.

Indeed Editorial Team, 16 August 2024, What Are Project Assumptions? (With Examples), Available on <https://www.indeed.com/> at: <https://www.indeed.com/career-advice/career-development/what-are-project-assumptions-examples>

4.2 Constraints

1. **File Storage:** Profile pictures for lecturers are stored as blobs in the database. Large images might affect performance, so size limitations are enforced.
2. **Database Performance:** If the number of claims grows significantly, there might be a need for optimizing query performance and indexing.
3. **Scalability:** The system is designed to handle a moderate volume of claims. For significantly larger data volumes, architectural changes may be required.

5. Conclusion

This CMCS design guarantees flexibility, scalability, and role-specific functionality while maintaining a clear separation of responsibilities using the MVC framework. The system is designed for simplicity of use, with Lecturers submitting claims, Managers approving or rejecting them, and HR creating invoices and managing lecturer data. The database design facilitates seamless data management, and the user interface is organized for clarity and ease of navigation between roles.

REFERENCE LIST

Indeed Editorial Team, 16 August 2024, What Are Project Assumptions? (With Examples), Available on <https://www.indeed.com/> at: <https://www.indeed.com/career-advice/career-development/what-are-project-assumptions-examples>

kaalel, 08 Jul 2024, MVC Framework Introduction, Available on <https://www.geeksforgeeks.org/> at: <https://www.geeksforgeeks.org/mvc-framework-introduction/>

Marsh, J. and Marzán, J. (2022) UX for beginners: A crash course in 100 short lessons Joel Marsh ; illustrations by Jose Marzan, JR. Sebastopol: O'Reilly.

Robert Sheldon, September 2023, model-view-controller (MVC), Available on <https://www.techtarget.com/> at: <https://www.techtarget.com/whatis/definition/model-view-controller-MVC>

Simplilearn, 29 July 2024, Understand all about ASP.NET MVC, Available on <https://www.simplilearn.com/> at: <https://www.simplilearn.com/tutorials/asp-dot-net-tutorial/what-is-asp-dot-net-mvc>