**Aadam Naidoo**

**ST10280755**

**PROG6212 POE PART 1**

# ***Assignment: Contract Monthly Claim System Prototype***

[**Assignment: Contract Monthly Claim System Prototype 1**](#_ym1dd498b3ky)

[**Introduction 3**](#_dv0z7rx6vohv)

[**1. Documentation: Design Choices and Structure 3**](#_c7khb8ytpxe7)

[**2. UML Class Diagram for Databases 4**](#_yfhugrrlk2l9)

[**3. Project Plan 4**](#_or6mk4hi21dp)

[**4. GUI UI: Design and User-Friendliness 5**](#_ua7oc7c1rluv)

[**5. Version Control 6**](#_l7a90c7jcqqy)

[**Conclusion 6**](#_tgdwykrejrej)

# **Introduction**

The Contract Monthly Claim System (CMCS) prototype is designed to streamline the process of submitting, approving, and tracking claims for Independent Contractor (IC) lecturers. The system aims to facilitate efficient management of claims through an intuitive user interface developed using WPF. This prototype includes essential GUI components, a detailed project plan, and a UML class diagram, ensuring alignment with the rubric requirements. The design and implementation are focused on providing a clear, user-friendly experience while adhering to the project's constraints and assumptions.

# **1. Documentation: Design Choices and Structure**

Design Choices:

- MainWindow.xaml: Serves as the navigation hub of the application. It includes buttons for accessing other functionalities, such as claim submission, approval dashboard, and claim status tracking.

- Buttons: Submit Claim, Approval Dashboard, Status Tracker

- Labels: Title of the application

- ClaimSubmissionWindow.xaml: Allows lecturers to submit their claims. This window includes fields for entering claim details and uploading supporting documents. It is designed to ensure accurate claim submissions.

- Buttons: Submit, Cancel

- Labels: Claim Amount, Claim Date, Supporting Documents

- Text Boxes: Claim Amount, Claim Date

- Additional Components: File Upload Control

- ApprovalDashboardWindow.xaml: For Programme Coordinators and Academic Managers to review and approve or reject claims. This window provides a comprehensive view of pending claims, with options to approve, reject, or view details.

- Buttons: Approve, Reject, View Details

- Labels: Claims List

- Data Grid/List Box: Claims awaiting approval

- Search Box: Optional for filtering claims

- StatusTrackerWindow.xaml: Enables lecturers to track the status of their submitted claims. This window displays the current status of each claim, allowing lecturers to monitor progress and receive updates.

- Labels: Status List

- Data Grid/List Box: Claims status overview

- Search Box: Optional for filtering claims

- LoginWindow.xaml (Optional): Provides user authentication to access the system. It includes fields for entering user credentials to ensure secure access to the application.

- Buttons: Login, Cancel

- Labels: Username, Password

- Text Boxes: Username, Password

Structure of the Database:

- Tables: Lecturers, Claims, Approvers, Status, Documents

- Relationships: One-to-many between Lecturers and Claims, one-to-many between Claims and Documents, many-to-many between Approvers and Claims.

Assumptions and Constraints:

- Assumptions: Users will have basic familiarity with the system, and the application will be used in a secure network environment.

- Constraints: Prototype does not include real authentication or database integration.

# **2. UML Class Diagram for Databases**

Classes and Attributes:

- Lecturer: LecturerID, Name, Email

- Claim: ClaimID, LecturerID, Amount, Date, StatusID

- Approver: ApproverID, Name, Role

- Status: StatusID, Description

- Document: DocumentID, ClaimID, FilePath

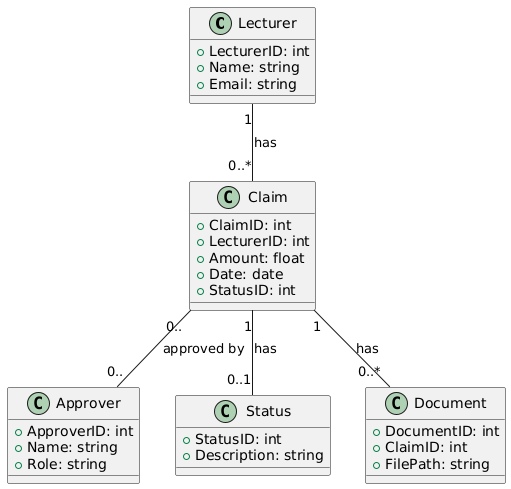
Relationships:

- Lecturer - Claims: One-to-many

- Claim - Documents: One-to-many

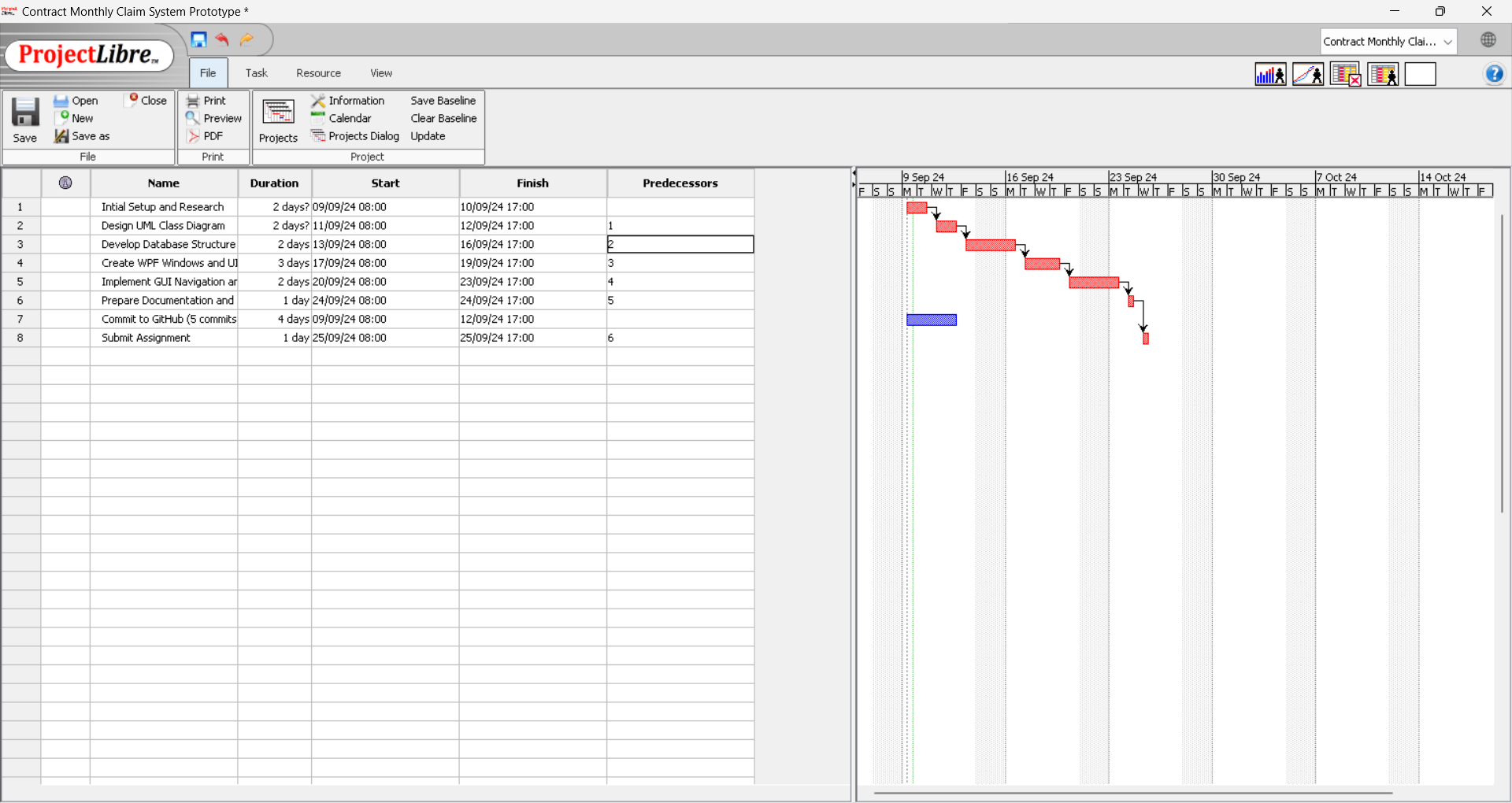
- Claim - Status: Many-to-one

- Approver - Claims: Many-to-many

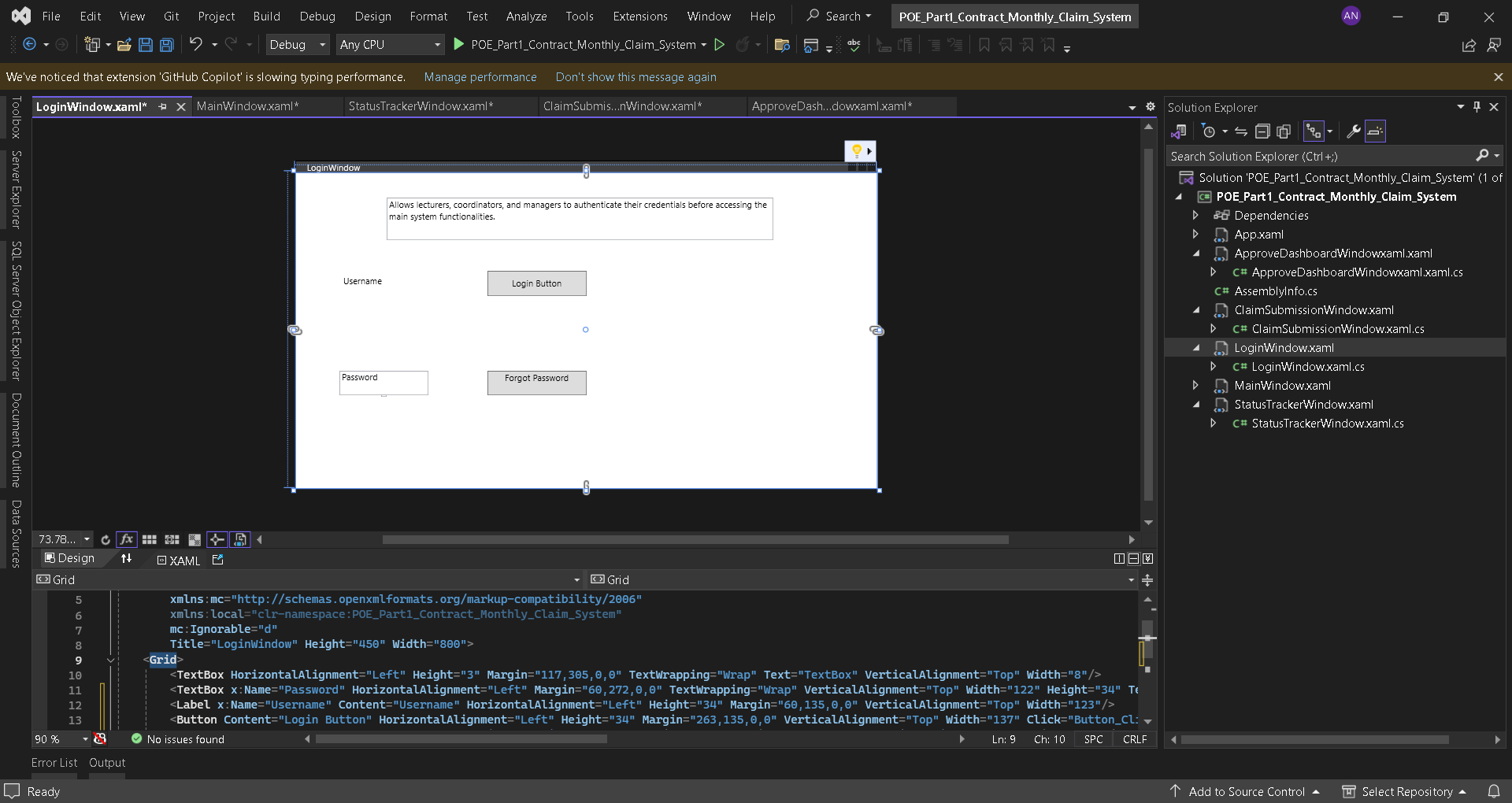


# **3. Project Plan**

| Task Name | Start Data | End Date | Predecessors |
| --- | --- | --- | --- |
| Initial Setup and Research | 2024-09-01 | 2024-09-02 |  |
| Design UML Class Diagram | 2024-09-02 | 2024-09-03 | 1 |
| Develop Database Structure | 2024-09-03 | 2024-09-04 | 2 |
| Create WPF Windows and UI Design | 2024-09-04 | 2024-09-06 | 3 |
| Implement GUI  Navigation | 2024-09-07 | 2024-09-08 | 4 |
| Prepare Document and Report | 2024-09-09 | 2024-09-09 | 5 |
| Commit to GitHub(5 commits) | 2024-09-06 | 2024-09-09 |  |
| Submit Assignment | 2024-09-09 | 2024-09-09 | 6 |
|  |  |  |  |



# **4. GUI UI: Design and User-Friendliness**

1. LoginWindow

Purpose: Allows users (lecturers) to log into the system.

Content and Controls:

- TextBox for Username: To input the lecturer’s username.

- PasswordBox for Password: To input the lecturer’s password securely.

- Button for Login: Submits the login credentials.

- Label : Display error messages or instructions related to login (e.g., "Invalid username or password").

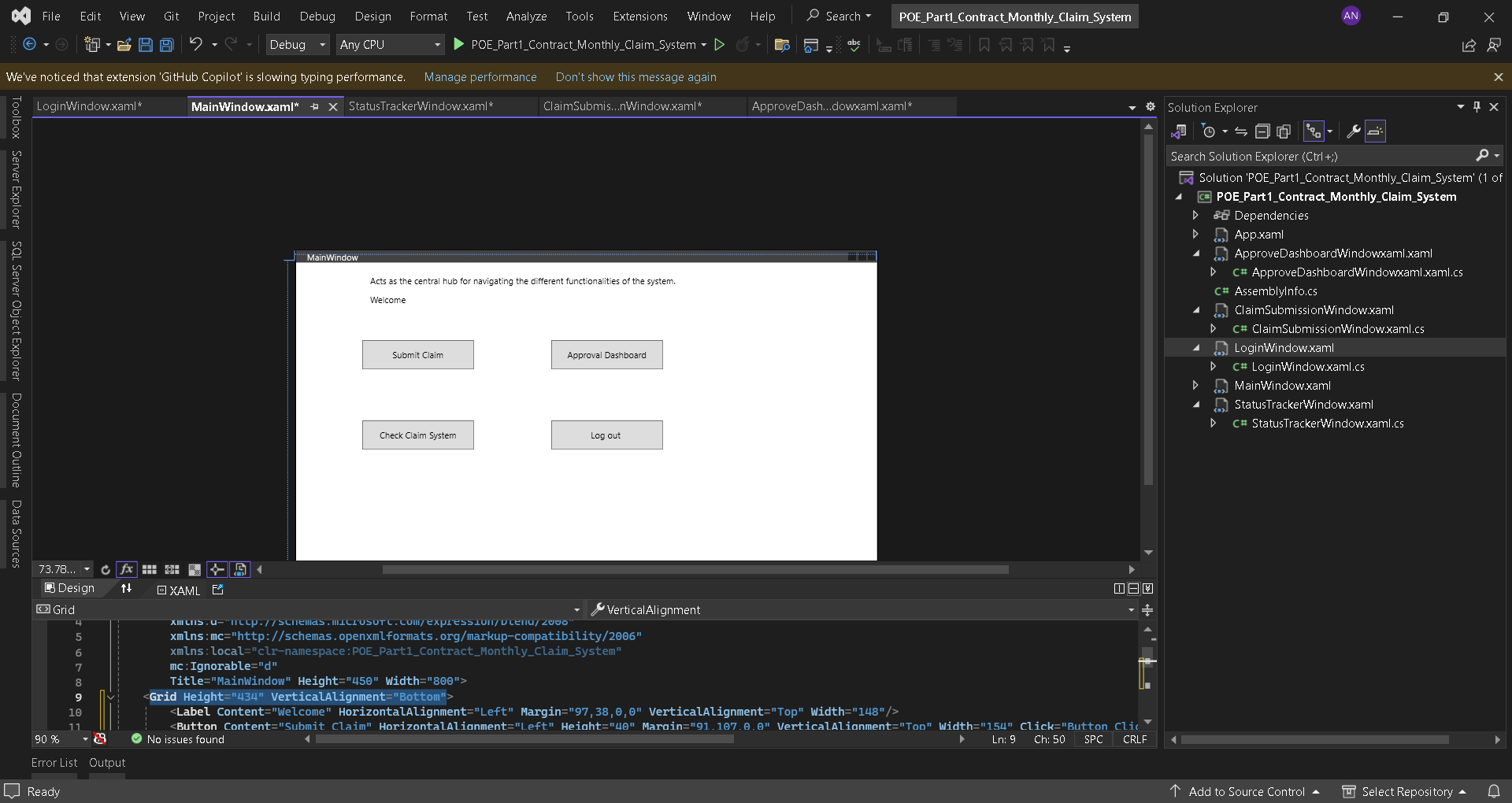
Functionality:

- The user enters their username and password.

- Upon clicking the Login button, the system should validate the credentials.

- If valid, it redirects the user to the MainWindow.

- If invalid, an error message should be displayed.



2. MainWindow

Purpose: The main interface after logging in, providing access to other functionalities.

Content and Controls :

- Button: Navigate to ClaimSubmissionWindow to create a new claim.

- Button : Navigate to StatusTrackerWindow to view claim status.

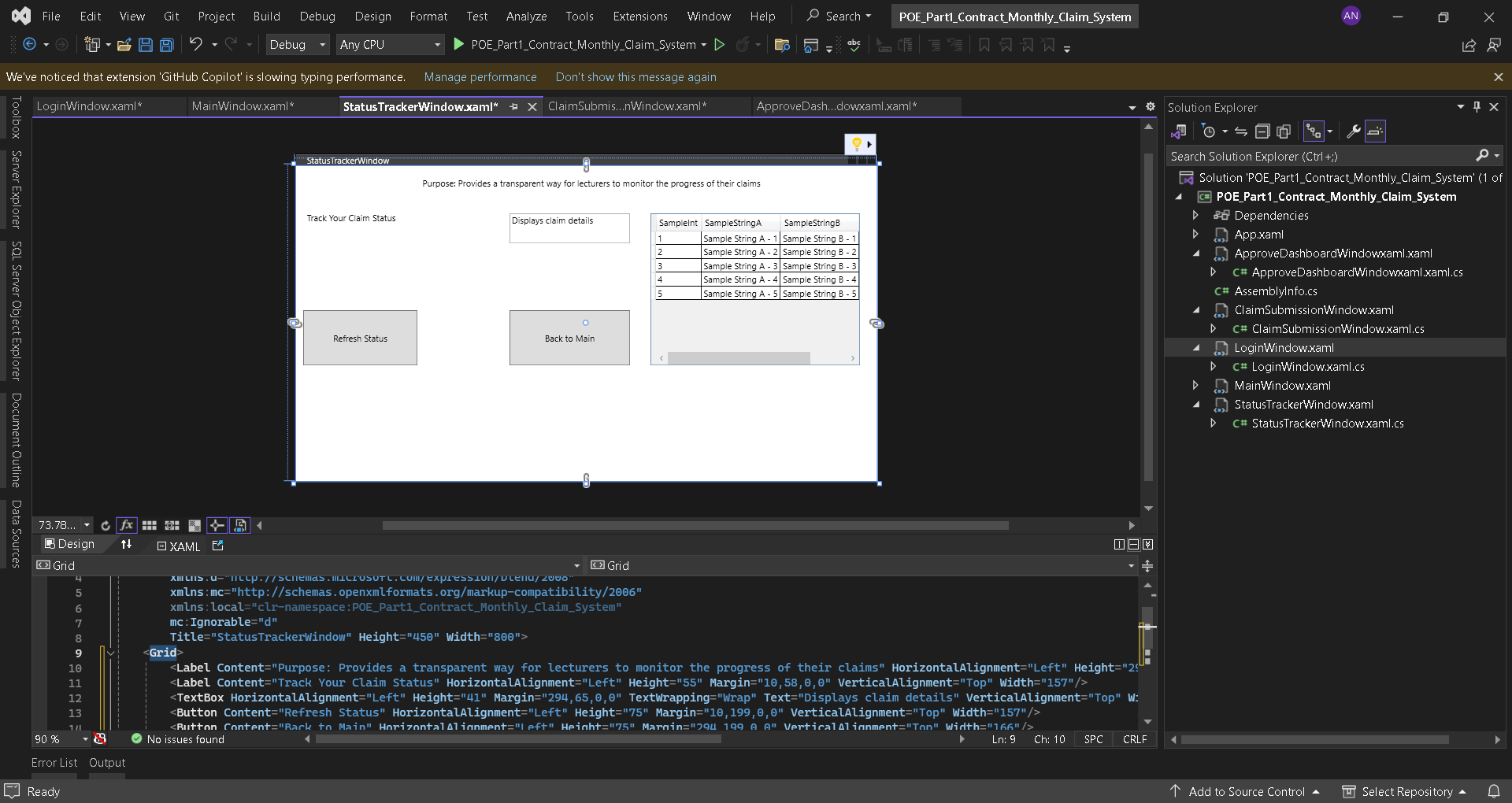
- Button: Navigate to ApproveDashboard (for users with appropriate roles).

- Label: Displays welcome message or user information.

Functionality:

- Provides navigation options to access different parts of the system based on the user's role.

- This is the central hub from which users can access the ClaimSubmissionWindow and StatusTrackerWindow.

3. ClaimSubmissionWindow

Purpose : Allows lecturers to submit new claims.

Content and Controls:

- TextBox : For entering the claim description.

- TextBox: For entering the claim amount.

- DatePicker: To select the date of the claim.

- Button: Submit button to send the claim.

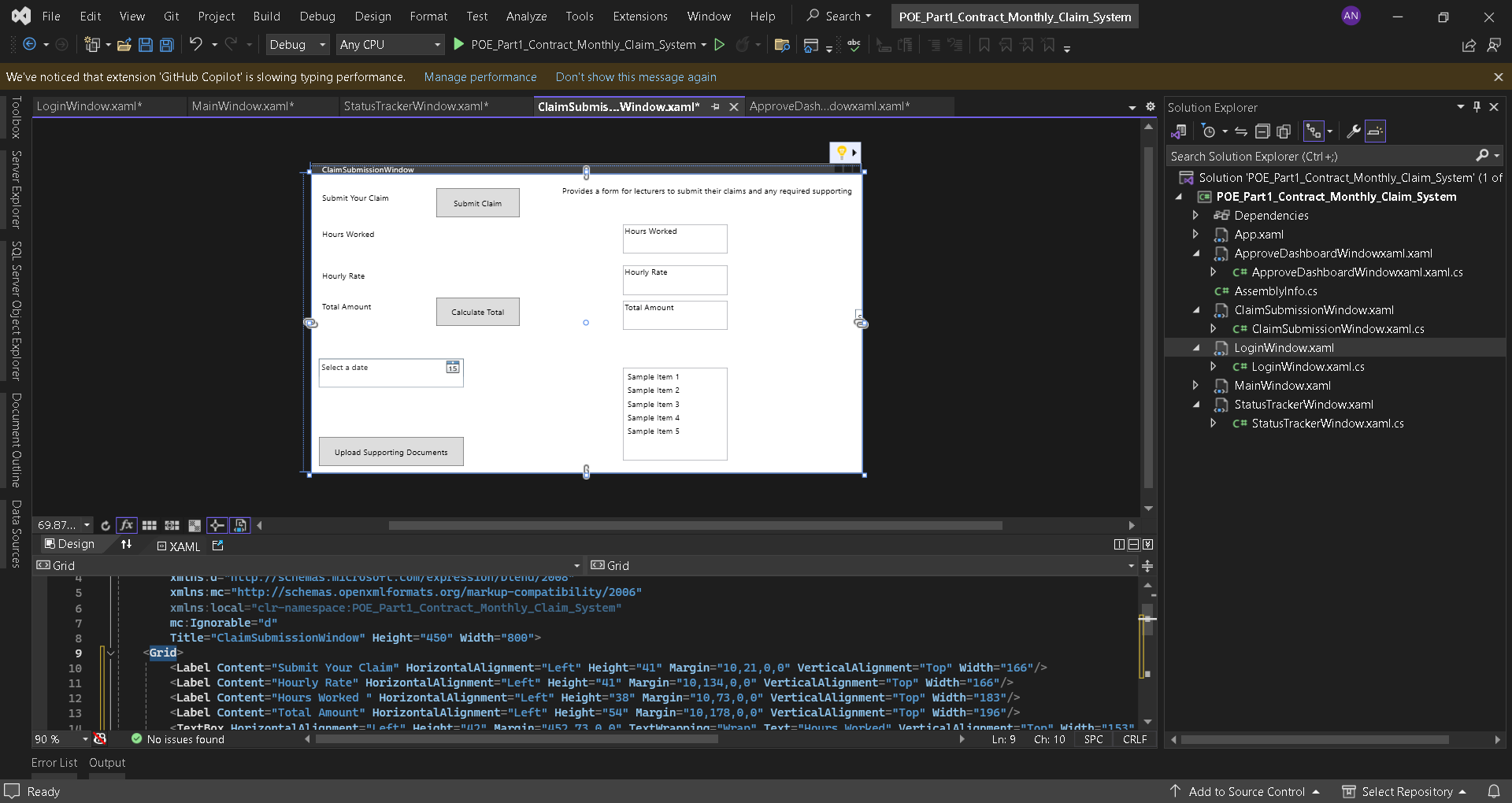
- Label : To provide instructions or validation messages (e.g., "Please enter a valid amount").

Functionality:

- The lecturer enters details for a new claim, including a description, amount, and date.

- Clicking the Submit button sends the claim to the database.

- The claim is then displayed in the StatusTrackerWindow with a status of "Pending".



4. StatusTrackerWindow

Purpose : Allows users to track the status of their submitted claims.

Content and Controls:

- DataGrid or ListView : Displays a list of claims with their current status (e.g., Pending, Approved, Rejected).

- Button : Refresh button to update the status list.

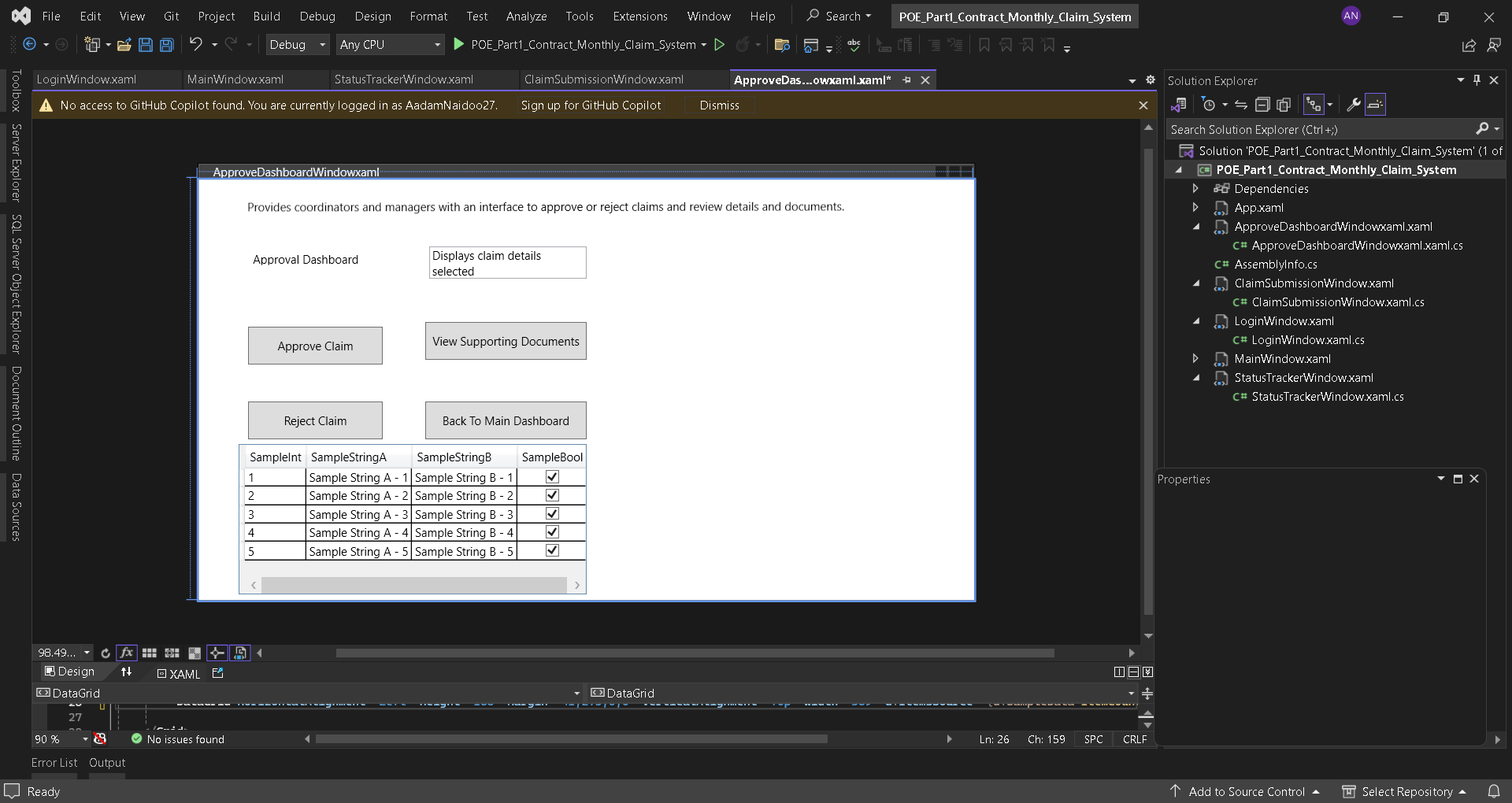
- Label : To indicate the section title or instructions.

Functionality:

- Displays a list of all claims submitted by the lecturer.

- Each claim entry should include details like claim ID, submission date, and status.

- The Refresh button updates the list with the latest claim status.

5. ApproveDashboard

Purpose: Used by Programme Coordinators and Academic Managers to review and approve/reject claims.

Content and Controls:

- DataGrid or ListView : Shows claims awaiting approval, including details like claim ID, lecturer name, and claim amount.

- Button : Approve and Reject buttons for each claim.

- Label : To provide instructions or feedback about claim processing.

Functionality:

- Displays claims that are pending approval.

- Allows Programme Coordinators and Academic Managers to review the details and either approve or reject each claim.

- Changes to claim status are updated in the system and reflected in the StatusTrackerWindow.

# **5. Version Control**

- Commit Dates:

- First Commit: 2024-09-06

- Second Commit: 2024-09-07

- Third Commit: 2024-09-08

- Fourth Commit: 2024-09-09

- Fifth Commit: 2024-09-09

- Commit Messages:

- Initial Setup: Added project files and initial setup.

- Database Structure: Designed database schema and added initial tables.

- UI Components: Created WPF windows and basic UI components.

- Functionality: Implemented navigation and basic functionality.

- Final Adjustments: Refined UI design and added final touches.

# **Conclusion**

The Contract Monthly Claim System prototype aims to provide a seamless experience for lecturers and administrators through a well-designed interface and structured workflow. The WPF-based application includes crucial windows and components for claim submission, approval, and status tracking. By following the provided project plan and using version control practices, the prototype aligns with rubric requirements, ensuring comprehensive coverage of all necessary aspects for a successful submission. This approach not only facilitates effective claim management but also sets the foundation for further development and enhancements.

<https://github.com/AadamNaidoo27/PROG_POE_PART1_ST10280755_AADAM_NAIDOO>