Final Presentation

Scrum Gang

Meet the Team

Josh Mangham: Scrum Master, Front End (GUI)
Brady Werling: Front End, Back End (Database)
Jack Sibbitt: Front End, Back End
Keegan Jackson: Back End, Information Coordinator

Cross-Functional

Project Overview

- Insurance Claim Management System
  - An integrated end-to-end insurance claim management system.

- Basic Modules:
  - Claim Application
  - Claim Feedback
  - Unique IDs
  - Register Users
  - Document Storage
  - Document Transfer

Functional Requirements

Client

- Login
- Apply for a Claim
- View Messages
- Upload Documents
- View/Edit Profile
- View Claim Status
- Download Claim Report

Claim Manager

- Login
- View Messages
- Download Documents
- View/Edit Profile
- View Client Profile
- Transfer Claim to Finance Manager

Finance Manager

- Login
- View Messages
- Estimate Claim Amounts
- View/Edit Profile
- View Client Profile
- Transfer Claim Amount to Client

Admin

- Login
- View all User Information
- View User Information by Date
- Assign Roles
- Modify User Details

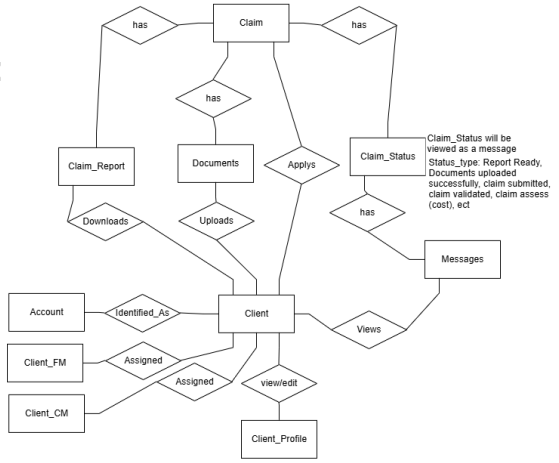
DBeaver

Initial Design

User Stories into the Database

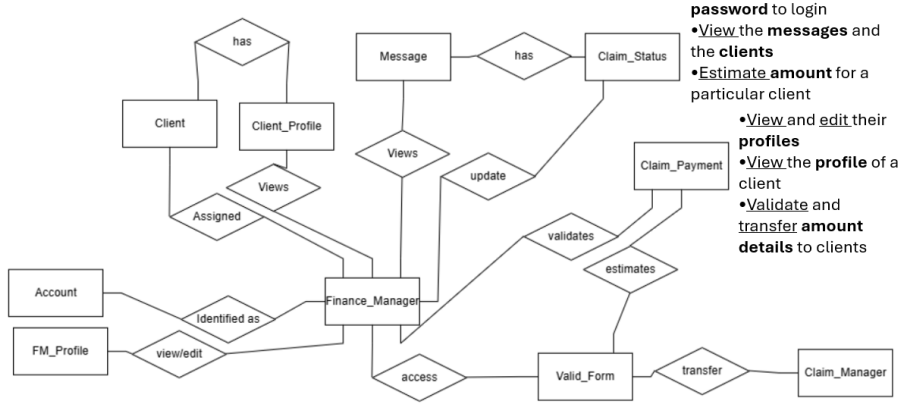
Functional Requirements: **Client**

- Apply for a **claim**
- Use their **username** and **password** to login
- View the **messages** sent by CM, FM, Admin
- Upload scanned **documents**
- View/edit their **profiles**
- View current **status**
- Download their claim **reports**

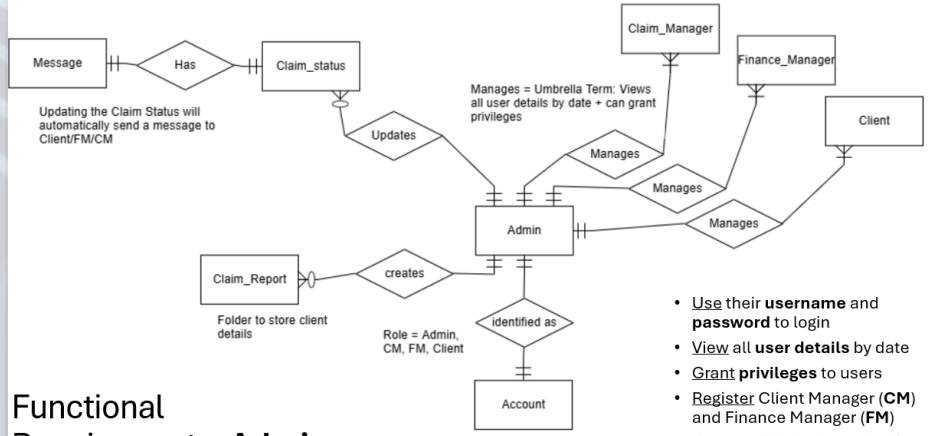


Functional Requirements: **Finance Manager (FM)**

- Use their **username** and **password** to login
- View the **messages** and the **clients**
- Estimate **amount** for a particular client
- View and edit their **profiles**
- View the **profile** of a client
- Validate and transfer **amount details** to clients



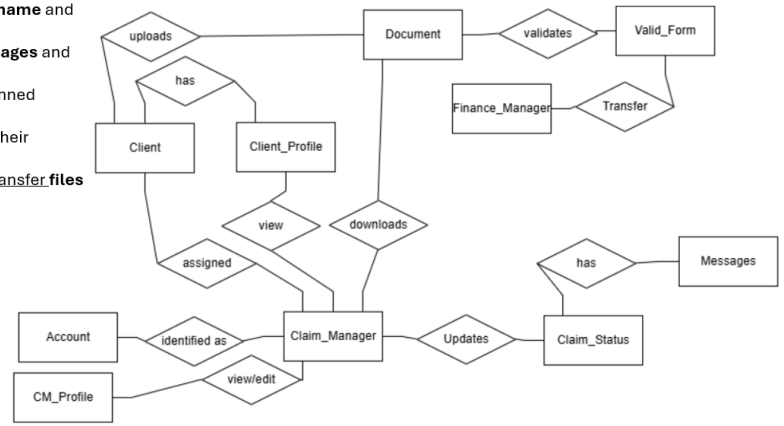
Functional Requirements: **Admin**



- Use their **username** and **password** to login
- View all **user details** by date
- Grant **privileges** to users
- Register Client Manager (**CM**) and Finance Manager (**FM**)
- Create a **Folder** for each client to store their details

Functional Requirements:**Claim Manager (CM)**

- Use their **username** and **password** login
- View the **messages** and the **clients**
- Download scanned **documents**
- View and edit their **profiles**
- Validate and transfer **files** to FM



Project Timeline

Scrum Gang

Project Timeline

Sprint #0

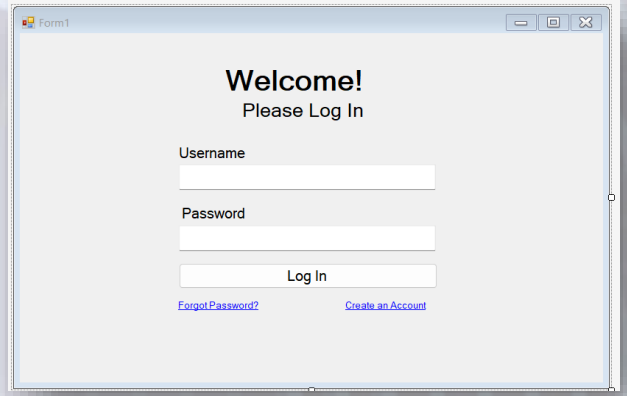
- Project Selection
- Role Assignment
- User Stories
- Login Prototype
- Trello & GitHub

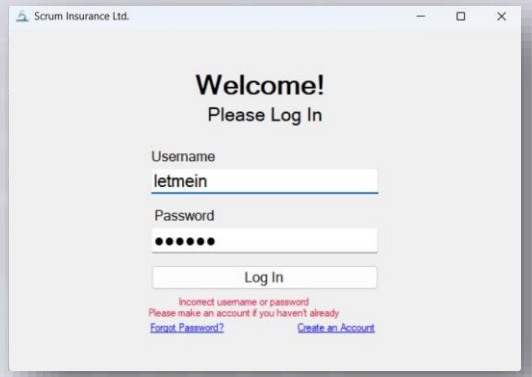
Sprint #1

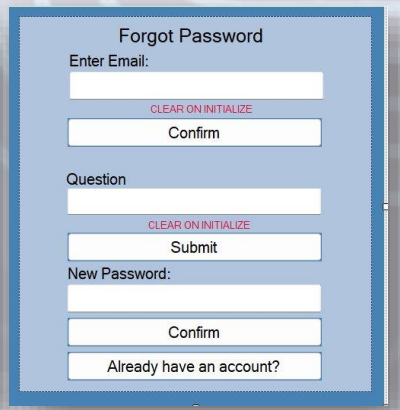
- Login Function
- Create Account
- Forgot Password
- Error Messages
- DBeaver Setup

Sprint #2

- Styling Updated
- Database Design
- Insert & Select
- Landing Page
- Session Class





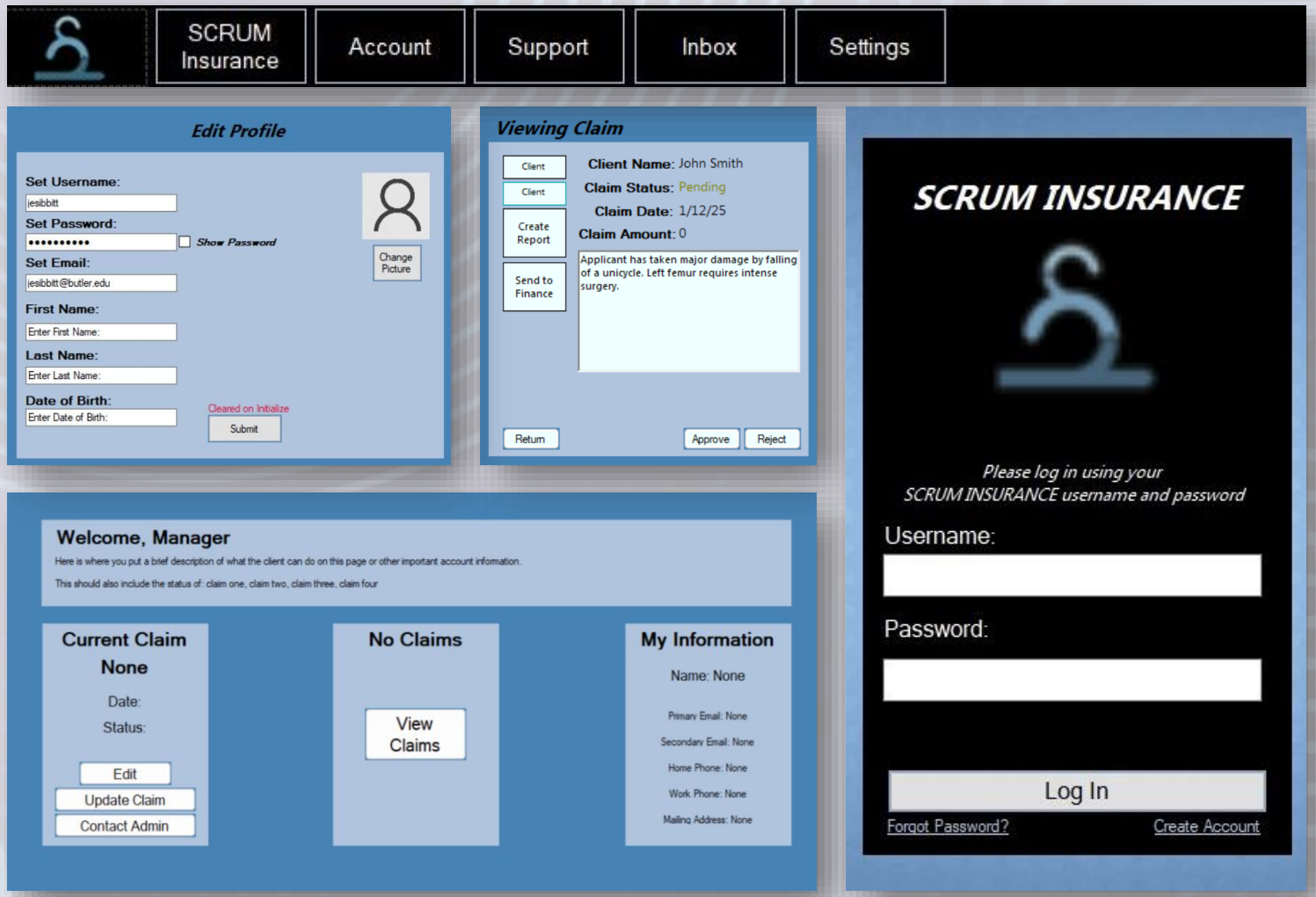


Sprint #3

Styling Updates
- Login Page
- Landing Page

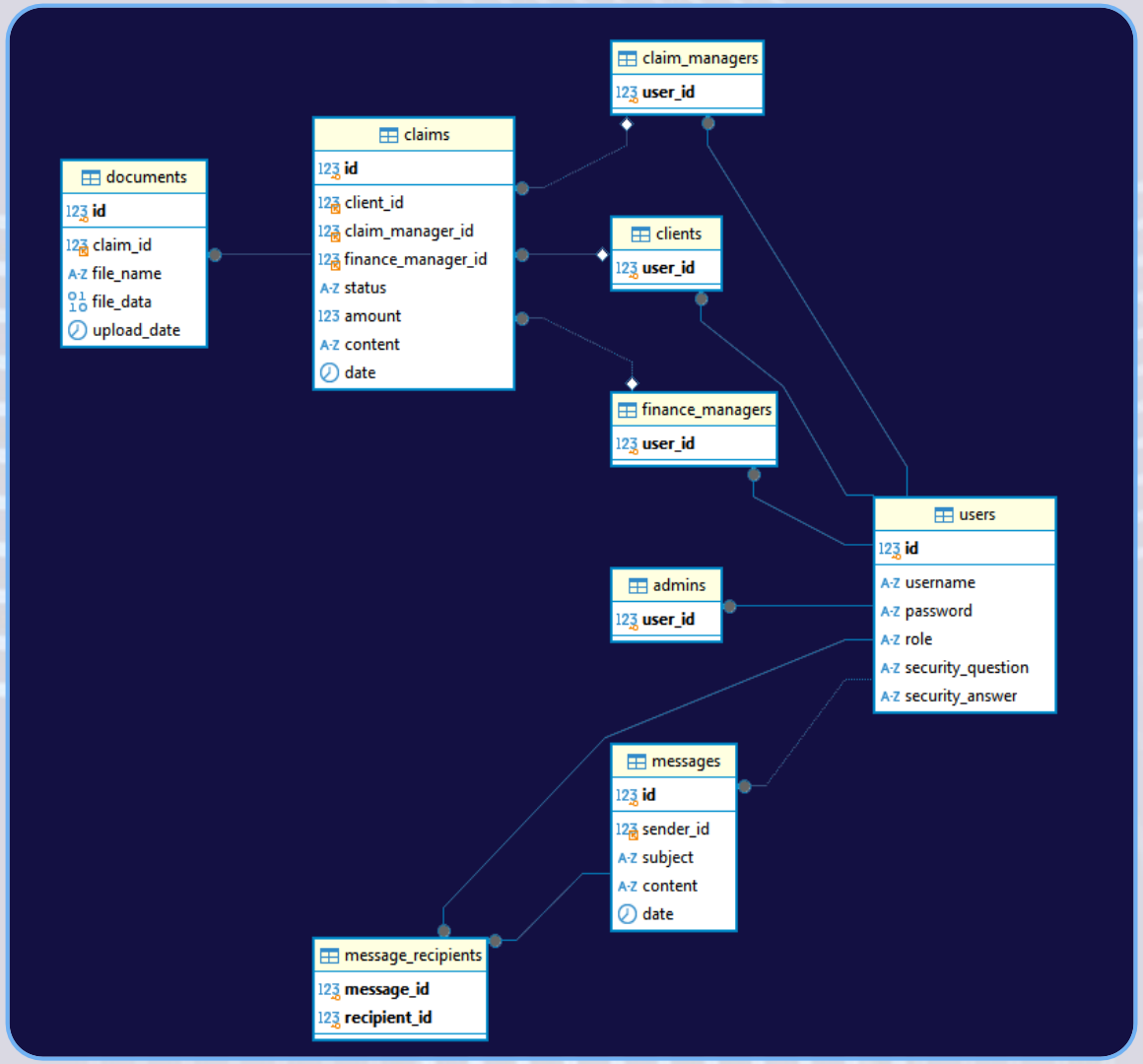
New User Controls
- Dashboard
- Edit Profile
- Claim Viewer

Database
- Full Redesign
- Naming Rules



Final Implementation

Database



Claims Table

Column Name	#	Data Type	Not Null	Auto Increment
id	1	int(10) unsigned	[v]	[v]
client_id	2	int(10) unsigned	[ ]	[ ]
claim_manager_id	3	int(10) unsigned	[ ]	[ ]
finance_manager_id	4	int(10) unsigned	[ ]	[ ]
status	5	enum('Pending','Financing','Approved','Rejected')	[ ]	[ ]
amount	6	int(10) unsigned	[ ]	[ ]
content	7	text	[ ]	[ ]
date	8	datetime	[v]	[ ]

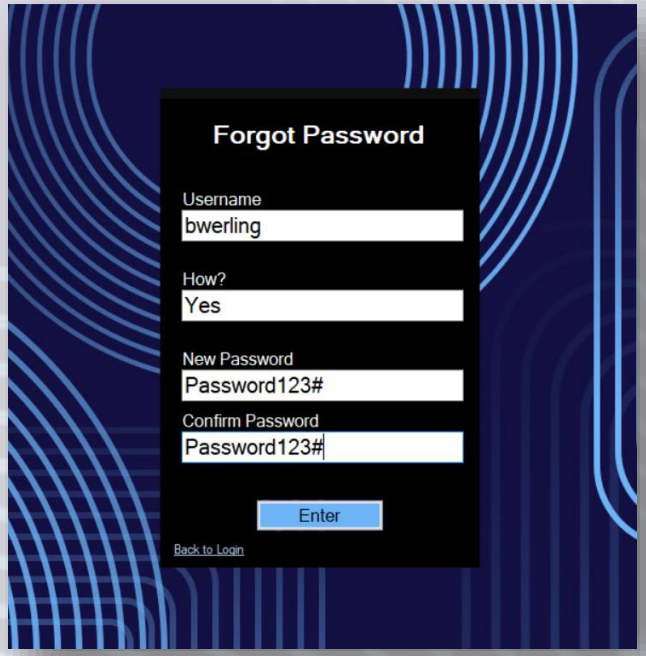
Documents Table

Column Name	#	Data Type	Not Null	Auto Increment
id	1	int(10) unsigned	[v]	[v]
claim_id	2	int(10) unsigned	[v]	[ ]
file_name	3	varchar(260)	[v]	[ ]
file_data	4	longblob	[v]	[ ]
upload_date	5	datetime	[v]	[ ]

User Table

Column Name	#	Data Type	Not Null	Auto Increment
id	1	int(10) unsigned	[v]	[v]
username	2	varchar(20)	[v]	[ ]
password	3	varchar(20)	[v]	[ ]
role	4	varchar(50)	[v]	[ ]
security_question	5	varchar(255)	[v]	[ ]
security_answer	6	varchar(255)	[v]	[ ]

Sprint #4

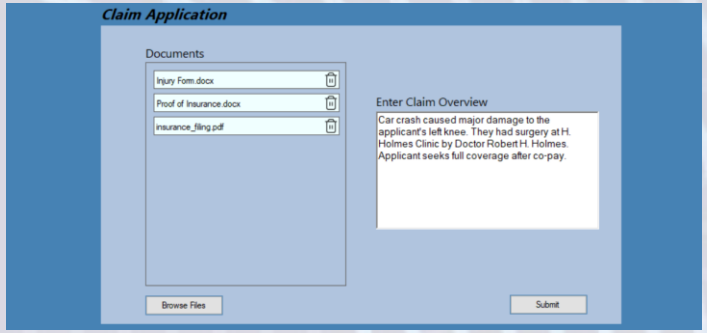


Restyling

- Vector Art Background
- Recolored and resized

Database

- Security (config.json file)
- User Supertype Table
- Custom C# Classes



New Functionality

- Claim Application
- Document Upload
- Message Viewer
- Claim Viewer
- Password Requirements

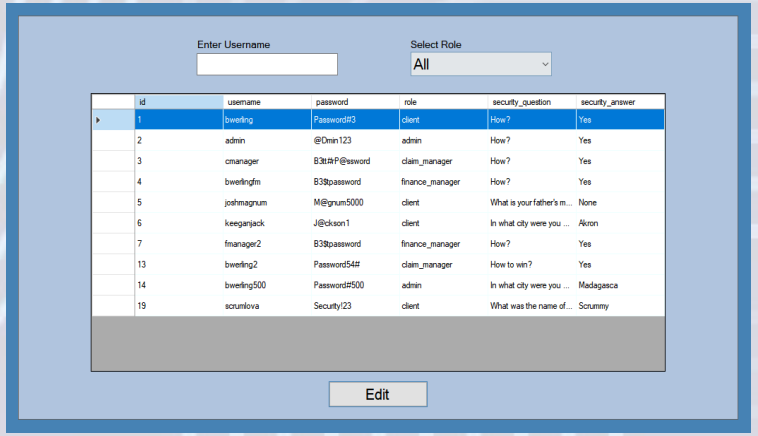
Queries

- Abstract Query Class
- Select and Insert

Sprint #5

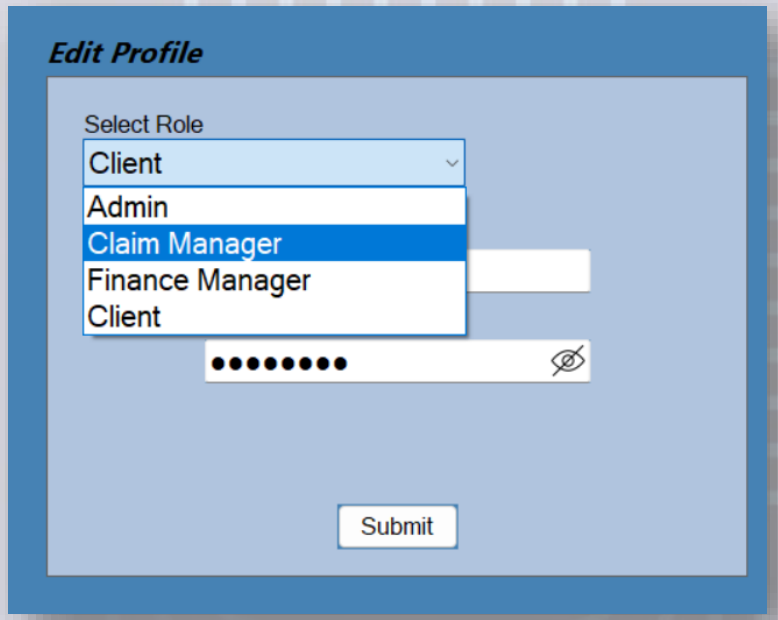
Refactoring
- Dynamic Panels
- Dashboard/Footer

Queries
- Last Inserted ID
- Nullable Return
- Minimize Data
- Object Lists
- Custom Delete

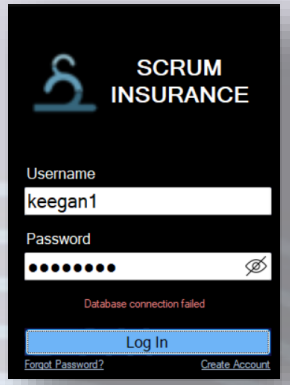


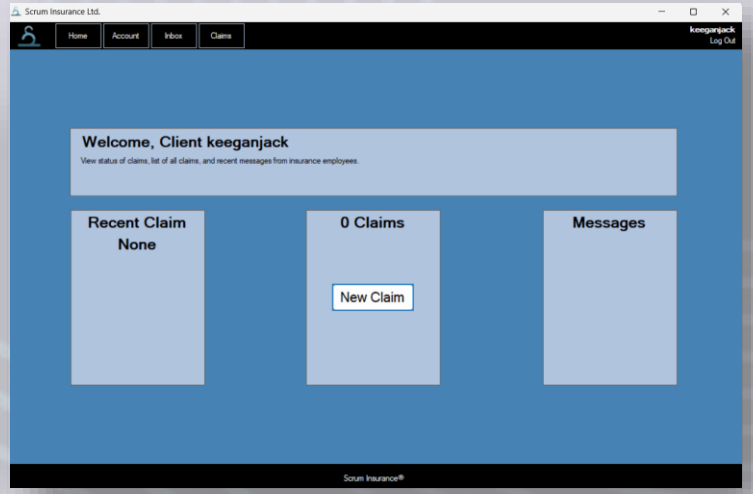
New Functionality
- Form Download
- Send Message
- View User Grid
- Edit Roles
- Claim Transfer
- Homepage Info

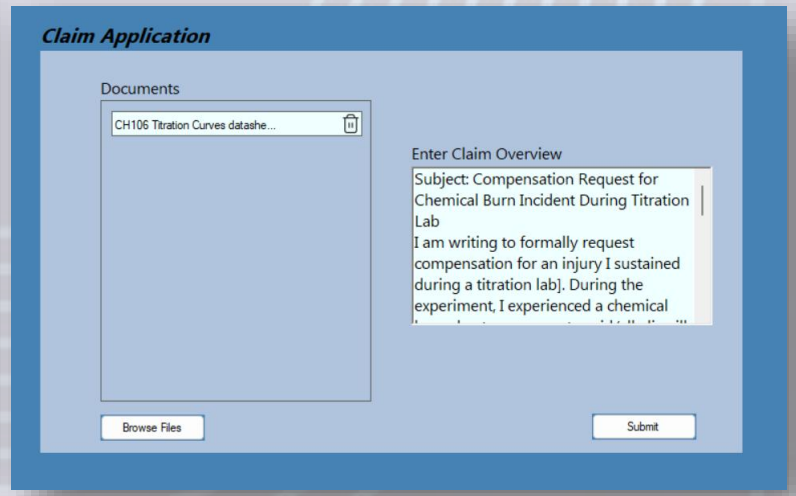
Feedback
- Response Text
- Show DB Errors

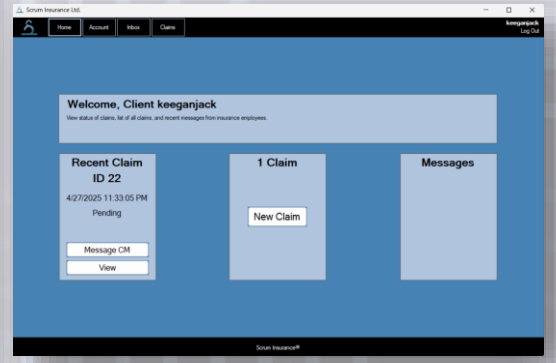


Sprint #5

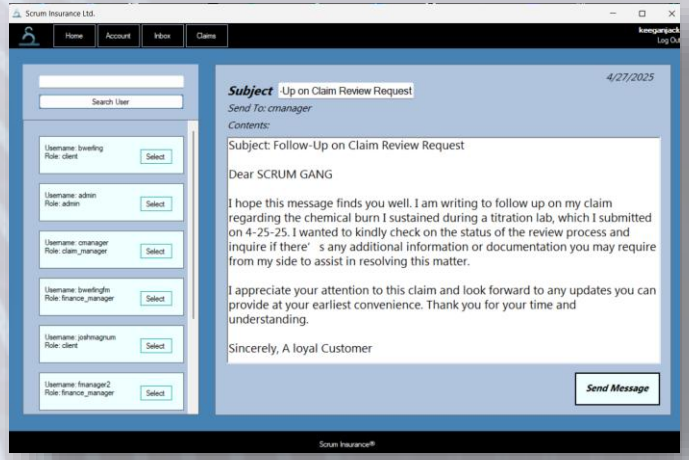


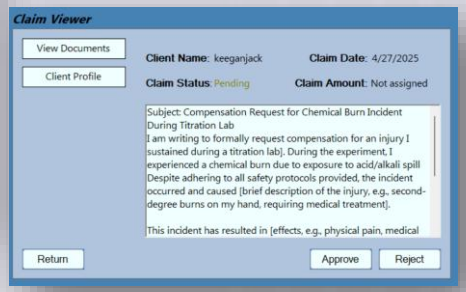


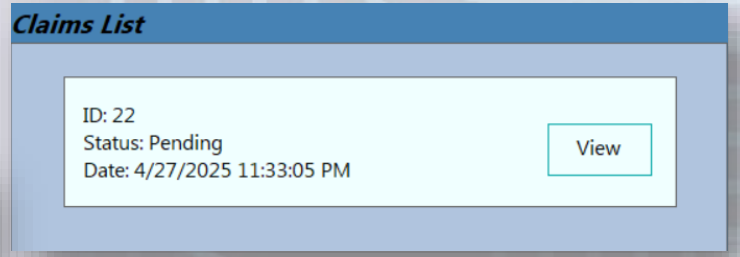




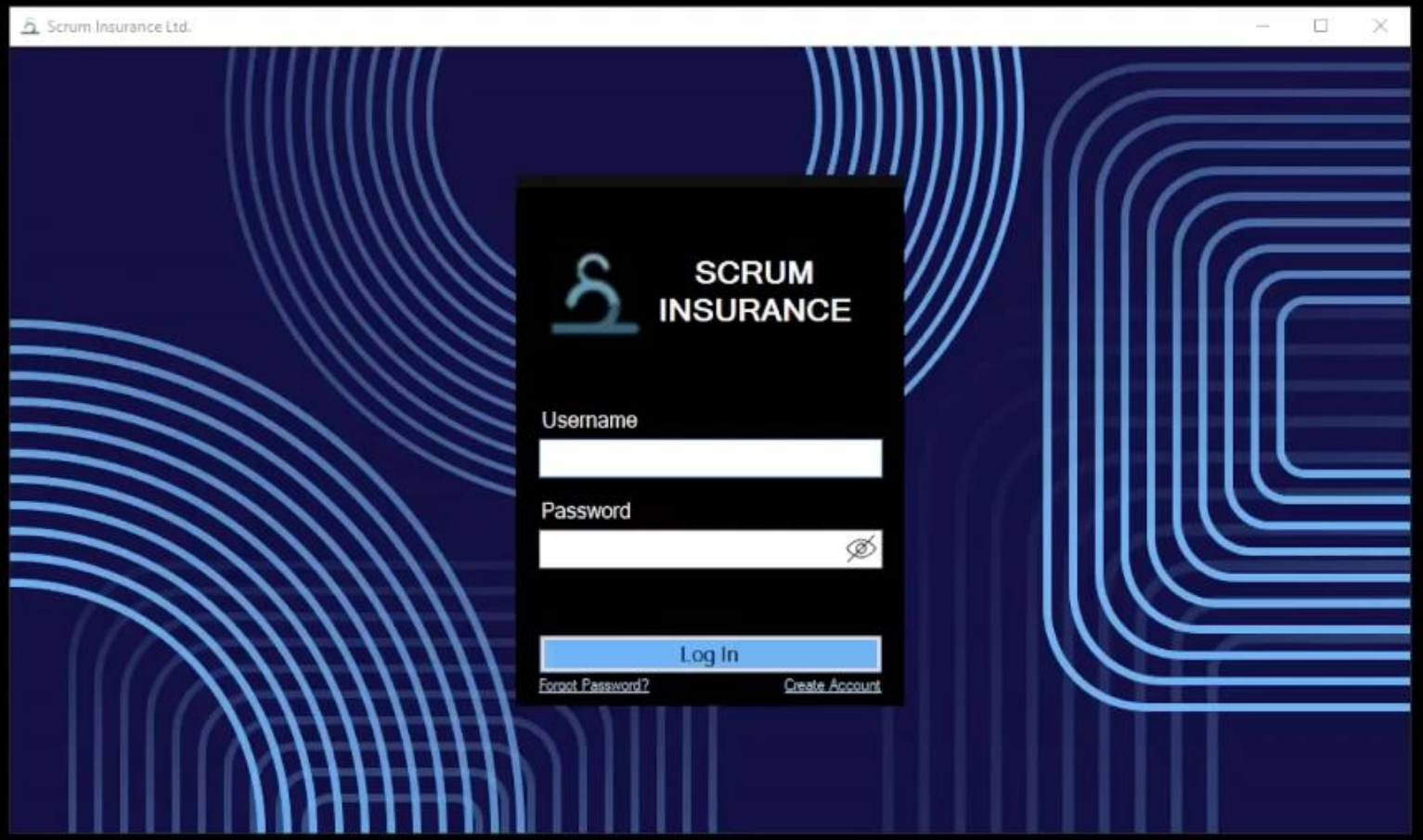
5	22	6	3	[NULL]	Pending	[NULL]	Subject: Compensation Request for Chemical Burn Incide	2025-04-27 23:33:05

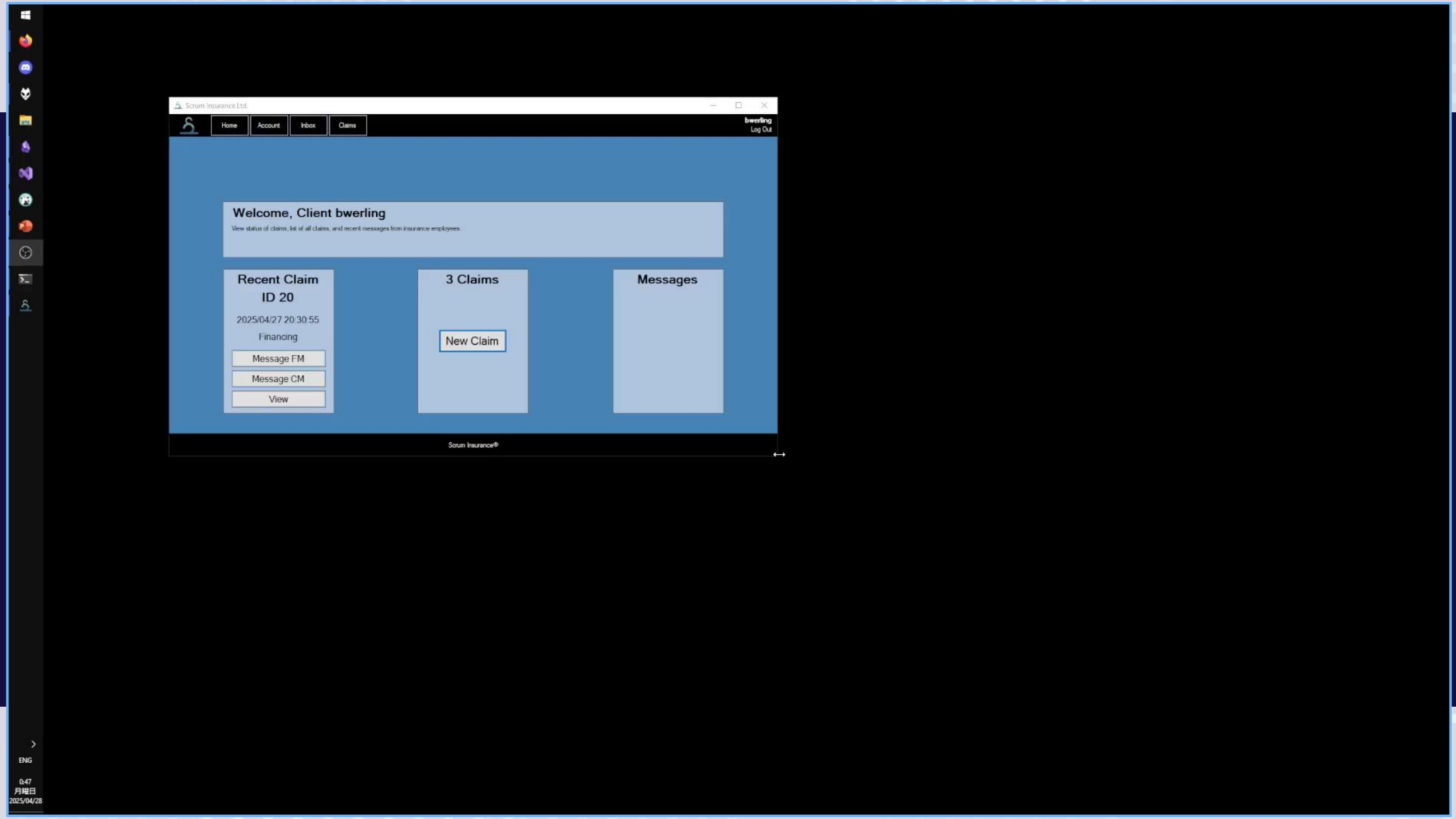






DEMO


Scrum Insurance Ltd.
SCRUM
INSURANCE
Username
Password
Log In
Forgot Password?
Create Account


Scrum Insurance Ltd.
Home
Account
Inbox
Claims
bwerling
Log Out
Welcome, Client bwerling
View status of claims, list of all claims, and recent messages from insurance employees.
Recent Claim
ID 20
2025/04/27 20:30:55
Financing
Message FM
Message CM
View
3 Claims
New Claim
Messages
Scrum Insurance®
ENG
0:47
月曜日
2025/04/28

Conclusion

Scrum Gang

Conclusion

Did we meet our goals?

Client:

- [x] Apply for a claim
- [x] User their username and password to login
- [x] View the messages sent by CM, FM, and Admin
- [x] Upload scanned documents
- [x] View/Edit Profiles
- [x] View current Status
- [ ] Download their claim reports

Administrator:

- [x] Use their username and password to login
- [x] View all user details by date
- [x] Grant privileges to users
- [x] Register Client Manager & Finance Manager
- [ ] Create a claim report for each client to store their details

Finance Manager:

- [x] Use their username and password to login
- [x] View /Send messages
- [x] Estimate claim amount for a particular client
- [x] View and edit their profile
- [x] View the profile of a client
- [x] Validate and transfer amount details to clients

Claim Manager

- [x] Use their username and password to login
- [x] View /Send messages
- [x] Download scanned documents
- [x] View and edit their profile
- [x] View the profile of a client
- [x] Validate and transfer files to FM

Conclusion

How did the project go?

- Overall Evaluation: **Good**
- The team gained significant knowledge and experience throughout the project
- Many of us had little no experience with Git, C#, Visual Studio, and the agile methodology
- Exposure to a semi-realistic SE working environment
- Both individual coding skills and collaborative teamwork abilities improved
- A substantial portion of user stories were accomplished, leading to a functional product

Conclusion

Scrum Thoughts?

- Scrum facilitated communication of the work process
- Scrum meetings helped us learn each other's strengths
- Sprint retrospectives were the only part of Scrum that we felt unsure about
- Trello board notes felt great to put into the 'Done' pile
- The iterative process lends itself to increasing team productivity

Conclusion

What would we do differently?

- Established a better communication system at the start of the project
- Dedicate more time outside of class for collaborative efforts towards the project
- Spend less time on Retrospectives
- Improve communication in sharing individual work progress

Future Directions

- Generating Claim Reports for Clients
- Add A Message Sent Notification
- Flesh out the program
  - Add more user and claim details
  - Improve GUI
- Continue to Refactor and Clean Code

Questions?

Scrum Gang