Gorilla Claim: Insurance Claim Management System

With the Use of Visual Studio 2019 and Visio

Team Success: Oumou Toure, Audrey Marjamaa, Avery George, Eric Nofziger, Jed Martin, Joe Vainisi, Jesse Hamlin

Scrum Master- Ezri Braid-Grizzell

Introduction to Software Engineering: Dr. Panos Linos

Spring 2022

Table of Contents

Abstract:

Gorilla Claim (GC) was built as part of SE 361, Introduction to Software Engineering, with the help of Professor Panos Linos. We'd also like to express our gratitude to Kristen Palmer, our project stakeholder who coached and assisted our team throughout the project. We chose the insurance claim management system as our project this semester and were subsequently given the software requirements that we translated into user stories. Our team then employed the methodologies of Scrum to satisfy these user stories and finish the project in about 14 weeks, over the course of six Sprints. Gorilla Claim allows login as a client, claim manager, finance manager, and administrator. Clients are able to file claims, which are viewable to everyone and then approved by the claim and finance managers. The administrator is also able to change the privileges of certain users. Every type of user is able to edit their profile and also send and receive messages as part of the claim process. Object-oriented design (OOD) and the creation of a local database in Visual Studio was used to add all of this functionality, and we also created several UML diagrams, an Entity-Relationship diagram, and a series of detailed wireframes to help us complete our tasks.

Introduction:

Insurance companies are crucial for financial system stability because they are big investors in financial markets, there are developing links between insurers and banks, and insurers protect financial stability of families and businesses by insuring their risks. Now imagine not having a system that allows you to create claims and hear back about the status of your claims. If there wasn't a system in place for this, it would greatly extend the time it takes to submit a claim and have it processed, wasting valuable time and money for both the company and the consumer. This would also cause clients to go in person to insurance companies multiple times just to be able to file a claim instead of being able to do this from the comfort of their homes. This is why it is important for a system to be in place for insurance companies to be able to have a seamless flow of information between clients and employees and intercommunication within the company.

Using Visual Studio 2019, Visio, and other software, our team created an intranet application. Visual Studio 2019 is a Microsoft integrated development environment for creating computer programs, websites, online applications, and mobile apps. We concentrated on designing an application for reporting claims, sending messages, and reviewing claims for this application.

Product Description:

As a team we had three projects to choose from and we ended up selecting the Insurance Claim Management System. This is a web-based tool that covers a variety of insurance corporations and management operations. It is a seamless end-to-end system that offers important information

across the insurance company to assist successful decision making for clients, insurance administration, claim management, and financial accounting. The requirements for the program are stated below:

The Insurance Claims Management System should provide services for four types of users:

Clients:

1. Have a unique username and password and the ability to edit their profiles.
2. Create claims
3. View the messages sent by employees about claims and view the claim status.
4. Can apply, upload, and download all documents relating to the claim.

Administrators:

1. Have a unique username and password and the ability to edit their profiles.
2. Can view a certain user's details and details based on a particular date.
3. Grant privileges to users and register Client Manager and Finance Manager.
4. Can create a folder for each client and store their details.

Claim Managers:

1. Have a unique username and password and the ability to edit their profiles.
2. Can validate and transfer files to the Finance manager and download scanned documents.
3. Can view the messages and profiles of clients along with sending messages to them.

Finance Manager

1. Have a unique username and password and the ability to edit their profiles.
2. Can view the messages and profiles of clients along with sending messages to them.
3. Can provide information to the clients about claims by sending estimated amounts, validating claims, and transferring amount details to clients.

Project Management:

Using an Agile technique, this project was done in distinct time-boxed Sprints. We divided the project into five two-week sprints and one 4-week sprint using the Scrum Methodology, which will be detailed in the conclusion section. Our team would meet at the start of each Sprint to discuss the goals for the upcoming Sprint as well as who would be responsible for completing each goal and would also meet at the end of each Sprint to discuss what went well over the course of the Sprint and areas of improvement going forward.

Project Structure:

Our team worked with a Product Owner, Stakeholder, Scrum Master, and a team made up of the remaining members, all in accordance with the Scrum Methodology. Panos Linos was our Product Owner, Kristen Palmer was our Stakeholder, and Erzi Braid-Grizzell was our Scrum Master.

Our project was broken into user stories that went into our Product Backlog that comprised everything that needed to be done in a classic Agile format. Then, at the beginning of each two-week Sprint, our Scrum team met to decide which user stories were significant or relevant to that particular Sprint and added them to the list of goals to be completed within these two weeks. We selected as a group which individuals would work on which user stories based on their skill sets after we decided which user stories we were going to focus on. Each user story was assigned to individuals or groups of people based on their suitability or willingness to volunteer for the work.

i.) Roles and Contributions of each Team Member

Audrey Marjamaa: Back End

Avery George: Front End, GUI

Ezri Braid-Grizzell: Scrum Master, Front End

Eric Nofziger: Front end

Oumou Toure: Front End, GUI

Jed Martin: Front and Back End Integration

Jesse Hamlin: Back End

Joe Vainisi: Front End, GUI

ii.) Work Carried Out by Each Team Member

Audrey: Audrey contributed employee and client searches, which allowed an advanced search through back end implementation of class methods and database handlers that was then accessed via the user through accessible pages on the front end. She also contributed to the back end implementation of viewing and managing claims which allows user function based on user type and status, as well as the back end implementation of the messaging functionality for all application users.

Avery: Avery assisted in the graphical user interface design and implemented functions on the front end aspect of the project. He contributed work to setting the permission of each specific user that only allowed them access to certain abilities.

Ezri: As Scrum Master, Ezri set out tasks and goals for the teams to meet, as well as facilitated meetings and check-ins with the group. She also focused on code review and clean up, as well as building and cleaning up the front end for maximum user friendliness. Ezri helped finish creating and connecting the front end functionality of claims to the back end, and facilitated tracking and expansion of the application through the external tool Miro.

Eric: Eric created the base front-end designs of several pages and forms and helped work on keeping the wireframe up to date, along with bug checking and fixing. He added additional implementation on both the front and back end to the profile and messaging portions of the program and integrated the two. Eric also added functionality that allows finance and claim managers to assign and finalize claims and pass information off to the user.

Oumou: Oumou assisted in the design of the graphical user interface as well as other design projects. In addition, she was in charge of organizing the final report and also used outside technologies like Adobe Photoshop, which was then used to create the branding for the project. Oumou helped decide the background colors for the website and contributed the admin user stories which we implemented later in the project.

Jed: Jed was responsible for creation and maintenance of the wireframe using Miro. He also created some general forms in order to set the skeleton of the management system and contributed to general bug fixing, performance, and quality assurance improvements. Jed also created the UML diagram for the system.

Jesse: Jesse was responsible for back end development and management through the Azure database. Sketched out and began implementation of back end code organization and the connection between the database and the code. Built many of the classes that regulated client types and claims.

Joe: Finally, Joe was responsible for choosing the application branding scheme and implementing it across the front end of the application. He was the primary person responsible

for ensuring consistency in the look and branding across the entire application and also contributed to bug and error checking across the project.

Organization of Report:

This report contains thorough information on our team's methodology for developing our application. This contains things such as thorough sprint reports, team members' contributions, and insights we drew from this semester-long project.

Sprints for the Project:

Below is a breakdown for each of the steps in the process of finishing this project and how we utilized each step within the methodology of Scrum to finish this project.

a) Preparation for Meetings and Sprint Goals

After the end of each sprint presentation, our team would meet the next day to discuss our goals for the next two weeks for our new sprint. As a team, we would go over all of the goals from the last sprint, and anything we did not get done would be moved from the last sprint to this new sprint. We kept track of our progress through Microsoft Planner and also assigned tasks to individuals through this platform. We would also update our wireframe with the goals for this sprint and place them in the area where they go, and how they would be seen by the user. We next addressed who would be responsible for which goals and double-checked that everyone was on board with the Sprint's strategy.

b) Product and Sprint Backlog

Our group used Microsoft Planner to keep track of our objectives. We started by listing and developing all of the User Stories we planned to complete for the semester in a Google Doc, then highlighted the ones we completed in black and the ones we were presently working on in yellow. We entered all of the goals we wanted to achieve for that Sprint in a Sprint Backlog on Microsoft Planner at the start of each Sprint. While completing tasks, we recorded the items that we were able to finish in Microsoft Planner

as done for the current Sprint. This allowed us to quickly see which things we did during the Sprint, and if one of our tasks was not done, it was automatically put to the Sprint Backlog for the next Sprint.

c) Retrospective

We followed the questions that were prompted in Professor Linos' Retrospective Form when conducting our Retrospective meeting. Each team member would discuss what they thought worked and what didn't in the previous Sprint. We were able to identify what was consistently working or not working as we had more of these meetings throughout the semester. We were able to discover factors that assisted our team based on input from the team. Some of the few things that we agreed that worked well were setting time outside of class to meet because we learned early on only meeting during class caused a lot of issues to not be resolved. More tools that worked well for us as a team were Microsoft Planner and GroupMe. These applications allowed clear communications between team members. We were also able to figure out what was preventing us from working efficiently, such as communication issues and personal technology errors, and we were able to quickly address those issues in order to boost our team's productivity.

d) Daily Standup Meetings

At the beginning of each class session, we were given 5-15 minutes for our daily stand up meetings. Participants are asked to stand up and one by one respond to three questions: he What are you working on? What is preventing you from finishing any tasks? What are you planning to work on next? When we had a lot of targets and everyone was working on something different, this came in handy. This allowed anyone having difficulty with a task to seek advice from the group or seek assistance with their problem. It was also beneficial since everyone could get a sense of what was being produced for the application, even if it wasn't in an area they were currently concentrating on. It made us feel like we were working as a cohesive team.

e) Code Demos

We were able to regularly complete new functionality during each Sprint and demonstrated our work in the form of an application demonstration during the code review and corresponding presentation to the Product Owner, Stakeholders, and our peers, in which we were able to present our work and receive feedback on both our code and our design elements. Our demo used the most recent version of the application, which was navigated by several sorts of users to demonstrate full functionality.

f) Tools used during the Sprint

We used a number of different tools in order to complete this project. For communicating, we decided to use GroupMe, which helped us to better communicate with each other about what needed to be done and set up meeting times to meet outside of class. For branding, we used Adobe Photoshop, which is where we developed the Gorilla branding that is displayed throughout the project. As a means to be able to store all of the PowerPoints, final report, user stories, and other documents needed, we used Google Drive, which allowed everyone to be able to go in and edit the documents and have access to them. Throughout the project, we also updated a wire frame in order to see the updated version and flow of our project; to do so we used the online tool Miro. For assigning tasks and planning out each of the Sprints, we used Microsoft Planner so everyone was updated in real time on what everyone was working on and the tasks they had to accomplish. Lastly, to create our project and write code that was accessible for each person, we used Visual Studio, Github, and Visio.

Project Sprints:

Below is a complete breakdown of each Sprint that discusses our goals for the Sprint and what specifically we accomplished.

Sprint 0:

During this sprint we had many goals and we were able to achieve all of them. The first few goals for this sprint were to select a team name and a scrum master. We then reviewed Scrum to make sure we understood the process and were able to arrange the team how it was provided that we should. We had our first Standup meeting where the only issue that was brought up involved downloading the software, but we were quickly able to resolve this. The next thing that we did was select the project for the semester. For the selection of the project we all read the three options and all unanimously decided that we wanted to pick the Insurance Claim Management System project to complete during the semester. We also set up a means for communication between the team and settled on GroupMe, and for a way to assign tasks to individuals we picked Microsoft Planner. Additionally, we developed Scrum user stories which caused us to break down each of the requirements into the 4 types of users. We then assigned two people to each user type to create a more detailed user story. To do this we used a Google Doc in our shared SE 361 Drive. We also discussed the strengths and interests of each team member and assigned each of them to either Database back end or UI front end. Next, as a group we wanted to create a GitHub repository in order to work on the project together. Finally, we made sure to download all of the necessary applications on all of the team members computers. These applications were Parallels for the Mac users since we can't get Visual Studio directly on Macbooks, Visual Studio, and finally Visio.

Sprint 1:

During Sprint 1 our goals were more oriented towards developing the project now that we had the necessary tools to begin. This sprint we were not able to finish all of our goals. The goals we were able to achieve was creating a database for login information, a login screen with 4 levels of permission, and a landing page for each user. Things we weren't able to get done was

meeting with the product owner about branding, so we were not able to start anything with the GUI in this sprint. The other goal we did not get done was give admin permissions to register Claim Manager and Finance Manager. During this Sprint, we decided to use a wireframe to better see and understand the flow of our project and the way and which users would see the pages as they work their way through the application. We had some technical difficulties with the database servers, and we weren't always quick to respond to GroupMe messages. Because of our problems, we agreed to schedule meetings outside of class to discuss our progress and areas where we could use assistance. During this time we also had issues delegating tasks due to the fact that we didn't know each other that well.

Sprint 2:

Sprint 2 was the time when our team made significant progress and our work began to resemble an application. To this end, we were able to define our definition of done, which is that it satisfies the requirements of the associated user story, the program must run without any errors, and everyone on the team must agree that it's done correctly. During this Sprint we were able to successfully separate front end and back end tasks. We also were able to start implementing branding for our product which we ran past our product owner. Some goals that we completed this Sprint from the product backlog at this point was the ability for the Claim and Finance Managers to be able to create profiles for clients and for admin to have the permission to register Claim and Finance managers. During this Sprint the wireframe really became important to help track the application because we started to increase implementation and add more pages and classes.

Sprint 3:

During Sprint 3 we were able to implement more object-oriented code, which was one of our biggest goals for the entirety of the project at this point. Meeting outside of class helped us greatly due to our schedule, but we did not meet as much as planned or as much as we had in previous Sprints. Due to the amount of goals we weren't able to implement, we decided to designate two days outside of class to meet. One goal that we did achieve was to give clients the ability to file a claim, and we were also able to refactor code in order to clean it up and in turn

help us work faster. Finally, we were able to implement our Gorilla branding throughout the entire application, and we created a timeline for goals that we wanted to accomplish with the last two sprints.

Sprint 4:

During this Sprint we were able to get almost all of our goals accomplished. One of these goals was to give Finance and Claim Managers the ability to access and update client claims. We also gave the admin the ability to view client claims. Additionally, clients were able to upload files and store them as part of the claim because of our work in this Sprint. This Sprint is the Sprint so far that we most consistently met on additional days outside of class. Finally, we began working on the final components of our application's data validation. We didn't have any issues during this Sprint. Our group decided to continue using Microsoft Planner and GroupMe because we had found our groove and were getting a lot done outside of class.

Sprint 5:

Our group concentrated on tiny cleanup chores for our final Sprint of the semester, including finishing messaging functionality, cleaning up the User Interface, and populating data tables. Checking for errors and drafting the final report/presentation were also on the list of objectives. The team continued to utilize Microsoft Planner to organize our activities for the final demo and report so that we knew exactly what we needed to do to complete those major jobs and could store all of our last Sprint assignments in one place. Throughout this Sprint we met a lot more than other Sprints just to make sure that everyone knew what tasks were assigned to them and what was accomplished so far in the Sprint. These additional meetings occurred partly because this was our longest Sprint throughout the project, and we had about a month to finish this stretch of the project. We spent the last few days before our presentation correcting bugs and wrapping up loose ends, as well as going to the Speaker’s Lab in order to practice our final presentation.

Conclusions and Lessons Learned

a. Overall Experience with Agile

We were all new to the Agile development approach when we started this project. Our group was also inexperienced with the field of software engineering, which did admittedly make things harder at the onset of the project. We discovered that software engineering entails far more than just writing programs. It took some time to get the hang of the Scrum methodology, but once we did, we realized how successful this strategy is. We discovered that working in iterative Sprint periods was enjoyable because they enabled us to stay focused for two weeks on a realistic set of goals, achieve them, and show our Product Owner and peers our progress. The two-week Sprints were well-structured, beginning with a Sprint Planning meeting and concluding with a Sprint Retrospective. It was great to have a Sprint Planning meeting at the start of each Sprint to get us focused and figure out what targets we needed to achieve. We discovered that planning for two weeks is easier and more successful than planning for the entire project at once. Additionally, having a Sprint Retrospective meeting at the end of each Sprint provided us a great opportunity to discuss our progress not only with the program, but as a team. Having a set time to talk through our habits and determine if they were working for us was great, and if we found something that wasn't working for us, we could reevaluate the situation and move forward as a better team.

Our team developed a strong working relationship with one another throughout the course of the semester, and by the end, we were a highly functional group that communicated effectively. Our team found the Scrum technique to be quite beneficial, and we learned a great deal from it as well as from each other. At the conclusion of our project, our team has gained fresh perspectives on software engineering. It was formerly merely a subject to us, with little meaning attached to the word. Now we all know what it entails and how beneficial it is to the overall growth of a project. An important skill that we as a group took away from this experience was how to operate as a team utilizing the Agile technique, which is something that we can all apply to our future careers. We all value and appreciate having a comparable real world experience.

b. Views on an Object-Oriented Approach

We just constructed the User Interface components first, then began filling in the background code for what interacting with each item on the screen would accomplish. By the second Sprint, we had worked out that this was an extremely inefficient way of working that was dooming our project. Naturally, we started looking at the code and noticed that some of the mechanisms behind it were similar. We then began to consider how the classes and methods contained within them could be reused in order to make work easier. The frequency with which we were pulling up or utilizing information about the individual in the database who was now logged in to the application spurred this novel method. This prompted us to develop our first class, the user class, which had all of the information available on the user login credentials data table as well as methods for retrieving it, and thus our Object Oriented approach was born. The Appendices contain more information on the various classes, especially the user class.

This Object-Oriented approach allowed us to keep our code clean and tidy while increasing the functionality. Before we implemented the databases and classes we put them in the wireframe as a group so that we could decide together where we thought each page would look the best. Our program was able to be built quickly as a result of this. We began by developing the classes we would utilize, then developed the methods for inserting data from those classes into the database, and then used a Graphical User Interface to display the data in the database tables. This Three-Tier Architecture considerably improved our ability to add functionality while also making it easy to replace things in the future if they need to be updated. Furthermore, a well-designed software will make it simple for any future programmers who may join the project to understand the code without difficulty.

We had one specific Object-Oriented adjustment we wanted to make if we had additional time to work on this project. When a new claim is created, we have a method in our SQL Connection class that emails the database admins. We wanted to make this method more broad by including two String parameters. The first input would be the email address of the person to whom the email would be sent, and the second parameter would be the desired message to be sent in the email. We were quite proud of our ability to employ Object Oriented Design, and it greatly aided the success of our project.

c. Team Obstacles

For us, the most difficult aspect of our undertaking was being completely self-sufficient. We are normally coached through projects in class, but for this assignment, we were given a list of criteria and instructed to get to work. It was first difficult to change to the mindset required for this training, but we all quickly adjusted. We were ready to begin after we had established our communication base.

The next big obstacle was figuring out who would take each role in the project. Our Scrum Master devised the plan for each of us to go around in a circle and decide which part of the project we would be contributing to. If a team member had no clue the group helped make suggestions until they were able to come up with where they wanted to do work. After this the greatest obstacle, which was a recurring theme throughout the project, was with Github. It was apparent early that Github would keep some code from one person and delete code from others and so therefore we could never merge without conflicts. Another issue was when people would try to push into the database at the same time. We fixed these issues by talking to one of our Stakeholders and communicating to one another when someone was going to push something on to Github.

We confronted human challenges in addition to technical challenges. People not replying in our GroupMe was the most typical issue with our communication. Members were not required to type a response, but if they liked the message, the sender would at least know who had seen it or who would be at the set meeting times.. Setting up times to meet for Sprint Planning and Retrospectives was another challenge we faced. The amount of time allocated in class alone was insufficient in giving us the time to achieve all our goals. We found a solution by conducting a WhenToMeet poll to determine when the majority of individuals were available. From there we scheduled recurring meetings on Wednesday and Fridays. We would only meet on other days in addition to those if we really needed to communicate something or something really needed to get done.

Aside from any technical or Scrum-related goals, getting to know each other and bonding as a team was a struggle. Pleasant, positive chemistry between team members makes work more enjoyable, allows for more work to be done, and fosters trust, which is critical for a collaborative

endeavor. Connecting with new people takes time, but our team eventually got there. We found ourselves asking how each other's weekends went or what everyone was up to in their lives as we had more meetings during the semester, just to learn more about one another as individuals outside of the academic atmosphere. This was excellent for our team chemistry, and we ended up working well together which caused us to feel as though we were a part of a team and helped us bond as we conquered the challenges we faced.

Despite the fact that our team had a very straightforward path to completion, we have a few suggestions based on our experience that could make the journey for a fresh team just getting started considerably smoother. First and foremost, it is critical to form early bonds with your teammates. It facilitates the rapid development of positive relationships and fosters an environment in which team members are not reluctant to seek support or advice from one another. Secondly, we'd recommend that each team gets told that they will need to meet multiple times outside of class or getting this project done will not be possible. It might be helpful to set a goal during Sprint 0 that the team needs to come up with meeting times that worked for everyone. Ultimately, our team would make few changes to our procedure since we are happy with the system we were able to develop and the ties we formed as a group.

Detailed Contributions:

These are stated up above in Chapter 2. Our roles stayed the same throughout the project.

Appendices:

Sprint Goals:

During each of our Sprints we used a 1-5 scale to rate how easy or hard we thought our goals would be to implement.

Difficulty Scale	
1	Very Simple
2	Simple
3	Moderate
4	Difficult
5	Very Difficult

Sprint 0:

- Select Scrum Master and Team Name (1)
- Review some brief introduction to Scrum (1)
- Select project for the semester (2)
- Setting up a system project for project tracking (2)
- Set up a system and norms for team communication (2)
- Identify/discuss team members strengths/interests for various roles in the project (1)
- Setup software for all team members on their own devices (3)
- Carefully review requirements for selected project (2)
- Develop user stories (2)

Sprint 1:

- Ensure everyone can connect to the database (2)
- Create a database for user login information (3)
- Login screen with 4 levels of permission (2)
- Loading page for each type of user (2)

- Meet with product owner about branding (2)
- Give admin permissions to register CM and FM (3)
- Research password encryption (2)

Sprint 2:

- Create profiles for clients, Claim Managers, Finance Manager (3)
- Track application design through wireframe (2)
- Give admin permissions to register Claim Manager and Finance Manager (4)
- Begin implementation of visual branding (4)
- Give Claims Manager, Finance Manager, and admin ability to search for users(4)
- Implement try and catch on login screen (3)

Sprint 3:

- Implement visual branding (3)
- Give client the ability to file a claim (3)
- Refactored code to work cleaner and faster (2)
- Crate claim status levels and make them updateable (3) In Progress
- Give Claim and Financial Managers ability to update claims (4)
- Give admin ability to view and access claim history (4)

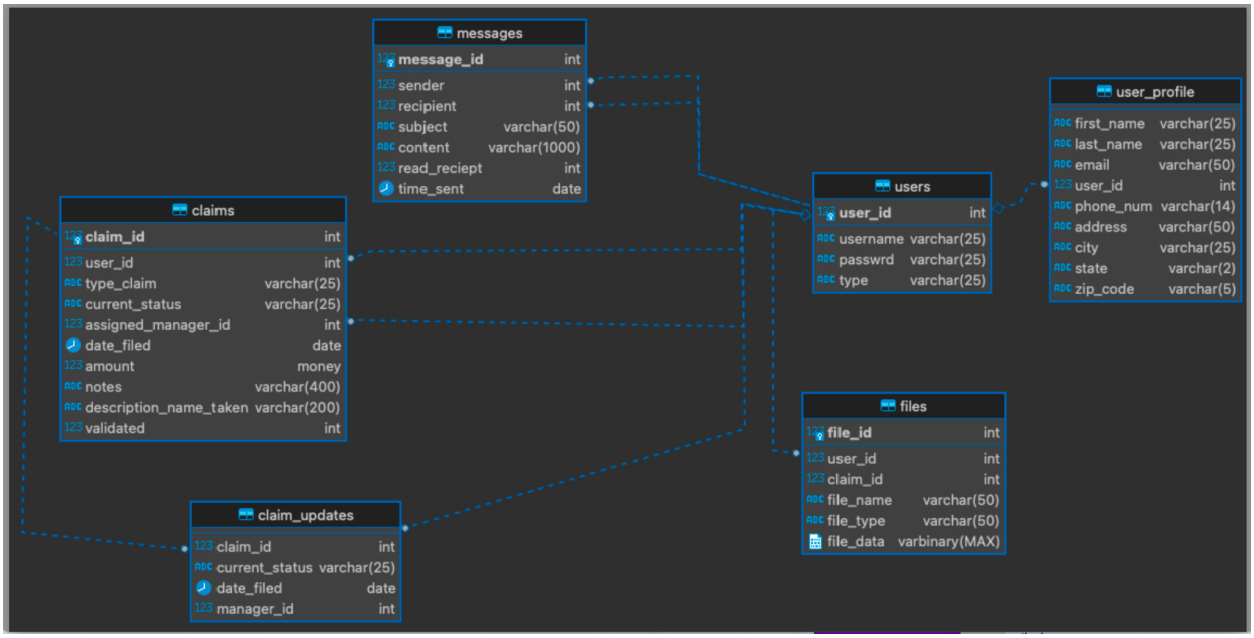
Sprint 4:

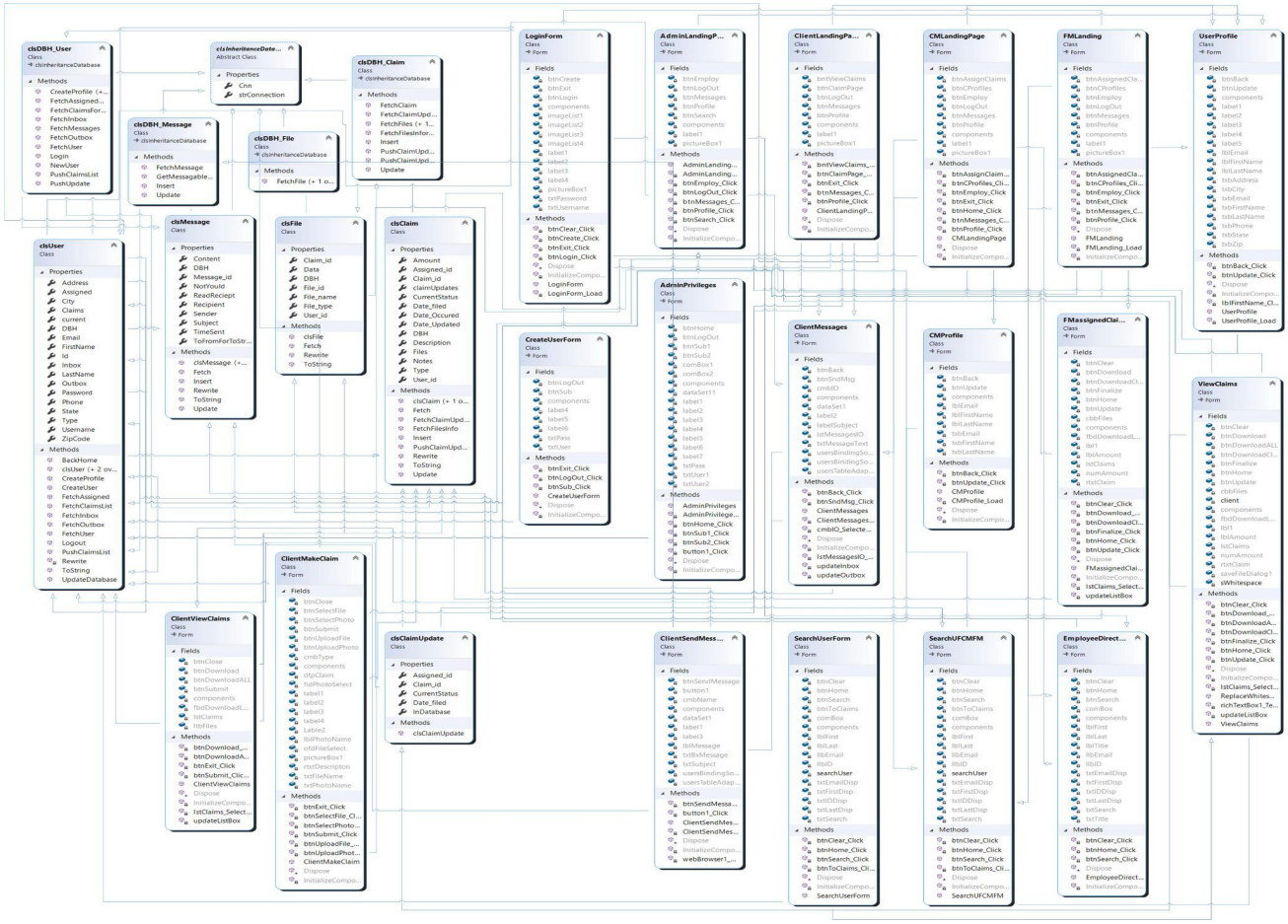
- Give Finance Managers ability to access and update client claims (4)
- Give Claim Managers ability to access and update client claims (4)
- Give client ability to upload files and store as part of claim (3)
- Give admin ability to view client claims (3)
- Test product reliability and quality assurance (2)
- Implement a way to convert claims into downloadable files (4)
- Create UML diagram of application (3)

Sprint 5:

- Give claim manager ability to select and validate claims (3)
- Give claim manager ability to assign claim to finance manager (3)
- Give finance manager ability to look at assigned claims (3)
- Implement ability to download claim (5)
- Create message inbox (4)
- Create ability to write messages (4)

UML Diagrams and Classes:



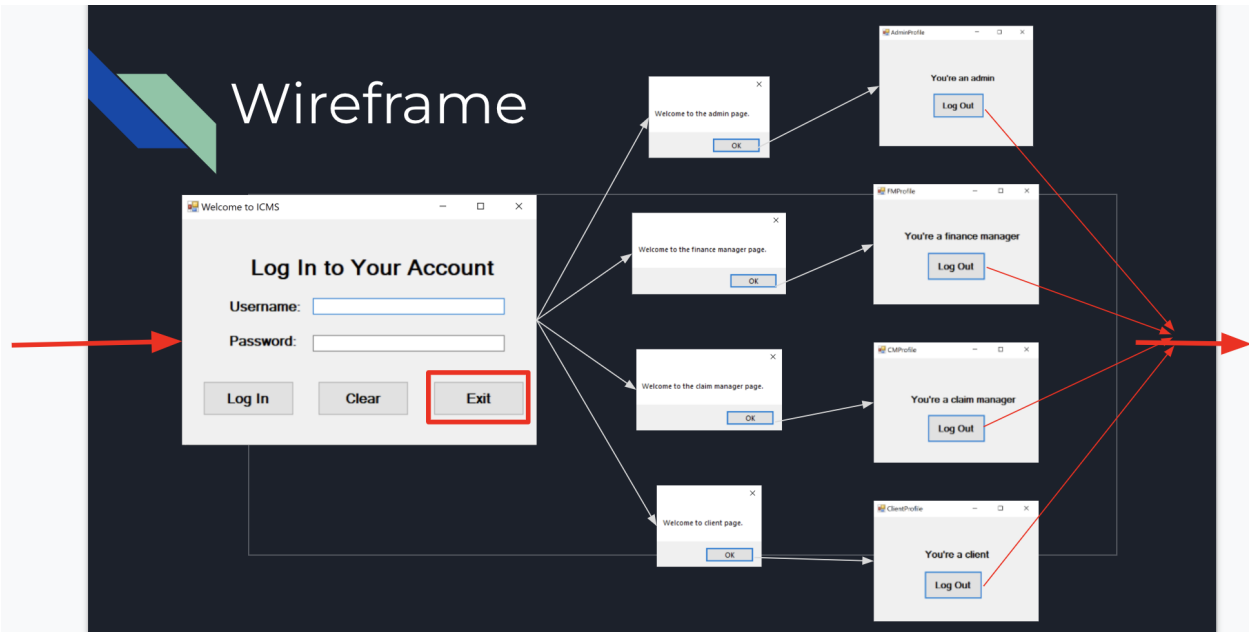


Wireframes:

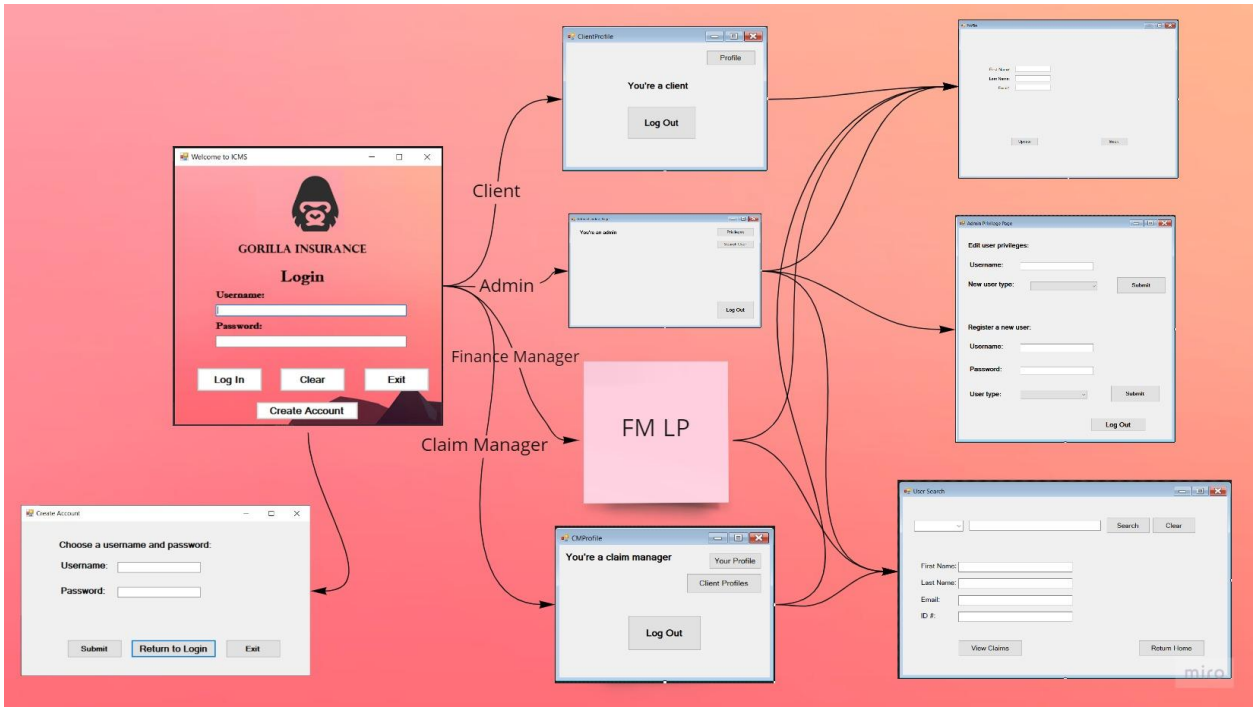
Sprint 0:

There is no wireframe for this sprint.

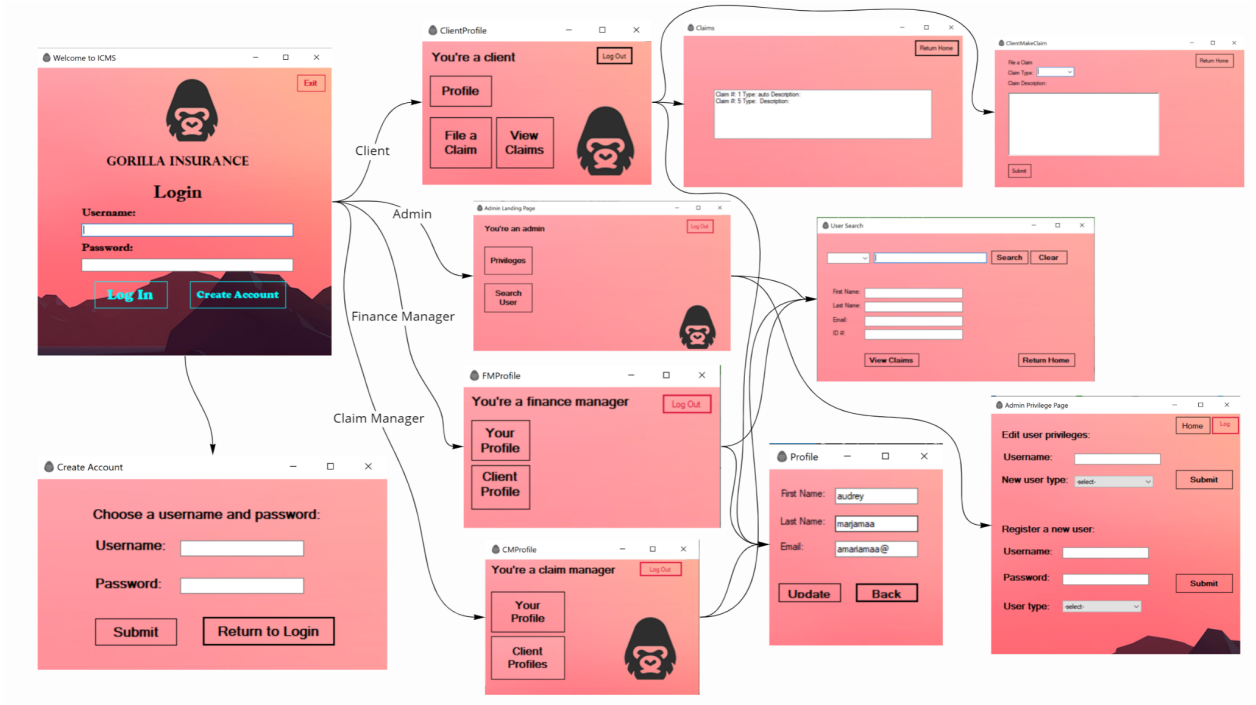
Sprint 1:



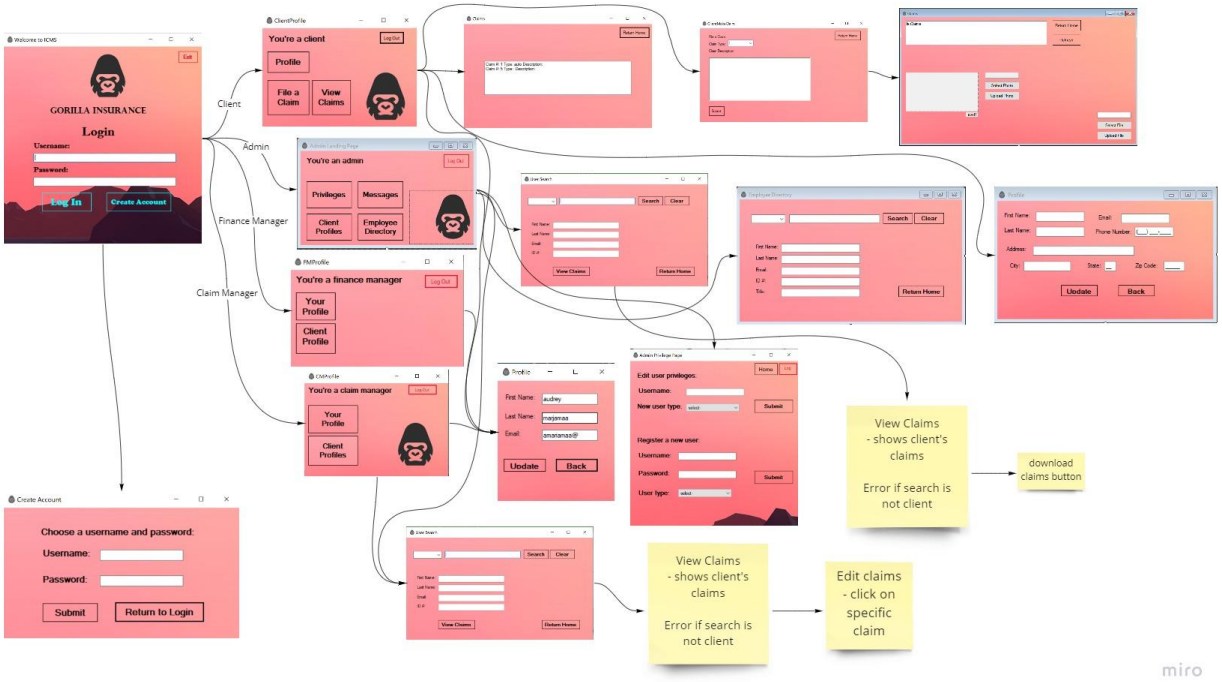
Sprint 2:



Sprint 3:



Sprint 4:



Sprint Presentations:

[Sprint 0 Presentation]

[Sprint 1 Presentation]

[Sprint 2 Presentation]

[Sprint 3 Presentation]

[Sprint 4 Presentation]

[Final Presentation]

User Stories:

[User Stories]

After each Sprint the goals that we got finished were highlighted black and the ones that we wanted to achieve during the next sprint or were still in progress were highlighted in yellow.

Peer evaluations:

Team C Sprint 0 Review Feedback

Team C Sprint 1 Review Feedback

Team C Sprint 2 Review Feedback

Team C Sprint 3 Review Feedback

Team C Sprint 4 Review Feedback