

CMPG 315

Phase 1

43054226 - Mariska Adriaanzen
45983690 - Marnus De Beer
38655349 - Christoffel Schoeman
38410737 - Ethan Sutherland
30054834 - Cecilia Theunissen

Table of Contents

| | |
|---|-----------|
| Reflections on online courses | 3 |
| Mariska's summary: | 3 |
| Christoffel Schoeman Summary: | 3 |
| Cecilia Theunissen Summary | 4 |
| Marnus de Beer Summary: | 4 |
| Ethan Sutherland Summary: | 5 |
| Work ethic | 5 |
| Description of Packet Tracer | 6 |
| 1.1 Machine Room | 6 |
| 1.2 Offices | 8 |
| 1.3 Technical Office | 8 |
| 1.4 Meeting Room | 8 |
| 1.5 Reception/Waiting Area | 9 |
| 1.6 Kitchen | 9 |
| 1.7 Open Floor Plan | 9 |
| 2.1 Server Topology | 10 |
| 2.2 Physical Topology | 10 |
| Remote Access | 10 |
| Budget | 10 |
| Budget based on Takealot: | 11 |
| Chat App Documentation | 13 |
| Server | 13 |
| Central Chat App with GUIs | 14 |
| Reflection on the messaging app development | 14 |
| Project management documentation | 15 |
| Group Roles and Responsibilities | 15 |
| Timeline Overview | 15 |
| Meeting outline | 16 |
| Attendance | 16 |
| Minutes of the meeting | 16 |
| Meeting 1 | 16 |
| Meeting 2 | 17 |
| Meeting 3 | 17 |
| Meeting 4 | 17 |
| Meeting 5 | 17 |
| Meeting 6 | 17 |
| Reflection of the project | 18 |
| Mariska | 18 |
| Cecilia | 18 |

| | |
|-------------|----|
| Christoffel | 18 |
| Marnus | 19 |
| Ethan | 19 |

Reflections on online courses

Mariska's summary:

Time management course: This course taught me about time management on a personal level. Although this course was good for personal development it didn't teach me some things I didn't know and when put into the context of a group work project it didn't specify how to manage time with a group which (in my opinion) would have been a better course to go through in my opinion.

Starting with git: I already know how to use GitHub on a basic level, so this course didn't teach me anything I didn't know.

Expert git: Once again, I knew most of what was said, and the Starting With Git was redundant here, but an interesting course.

Project management course: This course was a bit useless to me, because it felt like someone trying to teach me to “pappagaai” work with a definition rather than understanding the work. It felt like a lecturer using a class to read off slides that I could have done in the comfort of my house with a cup of coffee, and could have understood it better.

Packet tracer: The packet tracer course was very helpful since we will be using it in this assignment, and I have never used software like this before. It helped with the downloading process as well as getting the feel for the use of the application, like how to get the IP address and gateways, etc. The versions differ, so it took a while to get used to.

Christoffel Schoeman Summary:

The videos provided give information on time management, project management, introduction and fully explore GitHub and its properties.

While these time management and project resources did provide useful skills, such as utilising one's morning properly to use your time wisely, and a lot of the project management structures can be useful, but to me, it seems very much like a person-to-person preference.

The advice from both of these is sound, but often felt too vague for me, given the more specific circumstances in a network coding environment. As for the project management side of things, I do think the advice given is sound as well, but it seems more applicable for much larger groups (over 50 people or so) rather than smaller groups (like our 5-member group). For smaller groups, it is easier to communicate overall project goals and the components that make up a project, and then delegate components to the corresponding members.

As for GitHub and Packet Tracer, the Packet Tracer course helped greatly in setting up and getting started with the basics of Packet Tracer. While the same is true for GitHub, a whole course does seem like overkill, as it is not nearly as complex or “involved” as Packet Tracer.

Cecilia Theunissen Summary

The project management video was disappointing. The instructor only provided definitions in a heavy accent and did not explain anything in detail except for Microsoft Project and Oracle Primavera. It was not helpful since I had already covered most of these as module outcomes for CMPG 213.

The mini-course on time management was quite insightful. I was familiar with some of the details, but hearing them again reinforced the importance of having uninterrupted time to work on a project, focusing on one task at a time without multitasking or checking my phone.

The GitHub basic video serves as a good refresher for those who have not used Git in a while. It would be a great advantage for anyone who does not know where to start when using GitHub. The 4-hour GitHub video provided a lot of detail, but it makes sense as it helps a person become an expert in GitHub. The first section is just a repetition of the other, shorter video that was also given. This is a nice course, assuming that the person taking it does not know Git. Overall, it is a good but lengthy course.

The packet tracing course was an excellent start for someone like me who has no prior experience with these technologies. It offers learning activities to ensure that you know how to use the Packet Tracer on the downloadable files.

Marnus de Beer Summary:

The time management course was one of the most beneficial for me. I loved that certain methods were given so that you can start to implement these methods yourself and have a plan.

It was nice to have a recap of git and GitHub since my group and I from last year were already familiarised with it. A whole course for it was maybe not needed, and I did not like that everything was done in the command line(personal preference).

The resources that you can download for the project management course are nice. The folders contain useful information that can be easily revisited, however, it was hard for me to understand the instructor. It was not so insightful since we already covered a lot of this work in CMPG213.

The packet tracer course was greatly appreciated since I had no prior knowledge about packet tracing or how to set it up(generally all the basics about packet tracer).

Ethan Sutherland Summary:

Completing the online classes provided some useful techniques for the future for improving both my technical knowledge and soft skills. The time management course emphasised the significance of proper planning and work prioritisation, which is great for me because I do have a very chaotic lifestyle. As a result, I will try and take a more step-by-step approach to time management to reduce stress in the future.

The Git and GitHub classes were a great recap for me because I have worked with and used them before. But it was a bit of an information overload, I especially liked the terminal section because I never went into so much detail with this before.

The project management training was a bit hard to follow, but I have worked on group projects before, and it was just good to have a recap on the different methods, such as the PERT Chart and the Critical Path Method.

Although the Cisco Packet Tracer course was more technical, it was by far the most useful for me, and I am excited to explore more about this network simulation tool, especially for gaining more knowledge on cybersecurity.

Overall, looking back, the classes were helpful, and it makes me feel more at ease about working on another group project.

Work ethic

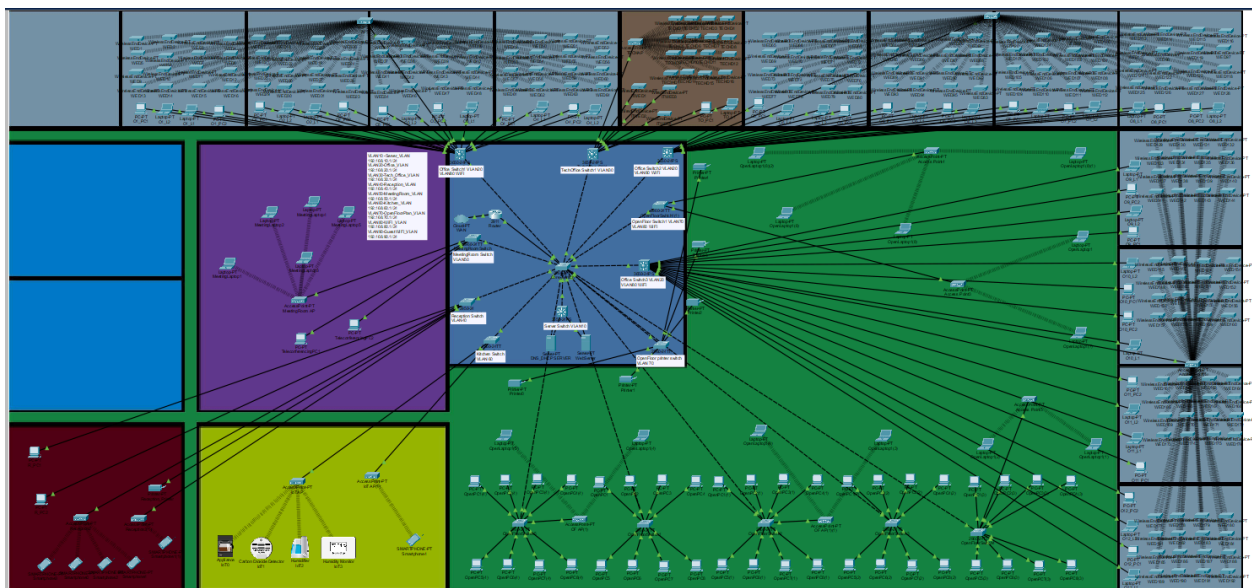
Our group established a clear and collaborative work ethic to ensure the project progressed smoothly and fairly:

- **Punctuality and Commitment:** All members respected scheduled meeting times and were expected to attend prepared. If someone could not attend, prior notice was required.
- **Mutual Respect:** Every group member was treated with respect, and all ideas and opinions were welcomed and discussed openly. Constructive feedback was encouraged to maintain a positive and inclusive working environment.
- **Equal Work Distribution:** Tasks were divided fairly, taking into consideration each member's availability and strengths. This helped balance the workload and ensure that no single person was overwhelmed.
- **Shared Responsibility:** Both the coding and documentation components of the project were shared equally among members. Everyone contributed to technical and written

tasks to promote learning and accountability.

- **Collaboration:** Our group used a combination of tools to facilitate effective collaboration and maintain organisation throughout the project. GitHub served as our central repository for sharing and managing code, enabling version control, real-time collaboration, and clear visibility of each member's contributions. For documentation, we relied primarily on Google Docs, which allowed for seamless, collaborative editing and easy access across devices. The network was designed and simulated using Cisco Packet Tracer as per the project requirements, while all programming components were developed using the Python language.
- **Accountability:** Each member was responsible for completing their assigned tasks on time. Missed deadlines were followed up with immediate communication and support if needed.

Description of Packet Tracer



1.1 Machine Room

The machine room serves as the core of the office devices and serves to assign IP addresses, through DHCP, to all the devices within the office. It also serves to store important files for the company.

The machine room consists of the following components:

- 10 Switches of the following names:

- Core switch (2960- 24tt)
- Office switch 1 (3650- 24ps)
- Office switch 2 (3650- 24ps)
- Office switch 3 (3650- 24ps)
- TechOffice switch 1 (3650- 24ps)
- Meeting Room switch (2960- 24tt)
- Reception switch (2950-24)
- Kitchen switch (2960- 24tt)
- Open Floorwitch (12960-24TT)
- Open Floor printer switch 1 (2960-24TT)
- Server switch(3650-24PS)
- 2 Servers
- Router (2911)
- Cloud-PT

Office Switches

These switches all operate together to connect all the devices from all the offices into the machine room to allow access to the servers and router Wi-Fi connection, and DHCP IP address allocation. All 3 of these switches use the same VLAN (19192.168.20.x which was made possible through trunking.

All 3 of the switches are connected to the core switch, which is then connected to the router. Trunking allows all 3 switches to make use of the 1 connection made to the router from the core switch.

Technical Office Switch

This switch works very similarly to the 3 office switches, but only 1 is needed as there are fewer services in the tech office than in all the offices combined. Note that this switch operates under the VLAN 19192.168.30.x.

It is again similar in setup to the other devices in that trunking is used through the core switch to enable sharing of one connection through the core switch to the router.

Router

The router is used as a firewall between the Internet and the office setup. The router provides internet access for internal VLANS using PAT (Port Address Translation). NAT enables multiple VLAN devices to share a single public IP. All VLAN subinterfaces use `ip helper-address 192.168.10.2` to forward DHCP broadcast requests to the server. This ensures dynamic IP assignment across all VLANS from centralised pools. The DHCP server must be configured to serve all VLAN subnets. This eliminates the need to run separate DHCP servers in each VLAN.

1.2 Offices

Each office room consists of the following components:

- 16 WirelessEndDevices(per room)
- 2 Personal Computers(per room)
- 2 Laptops(per room)
- One Access point(for every 4 rooms)

For all the office rooms, each has 16 WirelessEndDevices which connect wirelessly to an access point. For every 4 rooms, one access point is added. One access point has a total of 64 wireless connections (WirelessEndDevices). E.g. Office room 1 - 4, 5 - 8 and 9 -12. Each office room has 2 personal computers and 2 laptops. These 4 devices and the access point connect to an office switch. For the 12 total office rooms, there are 3 Office Switches, which are located in the machine room. E.g. Office room 1 - 4 connects to Office_Sw2, 5 - 8 connects to Office_Sw1, and 9 - 12 connects to Office_Sw3. These 3 Office Switches connect to the core switch.

1.3 Technical Office

The Technical room consists of the following components:

- 16 WirelessEndDevices
- 4 Wired End Devices
- 1 Personal Computer
- 1 Laptop
- 1 Access point

Like the office rooms, there are 16 wireless end devices that connect to the access point and the Technical room also as a TechOffice_Sw1 which is located in the machine room. The 4 WiredEndDevices, 1 personal computer, 1 laptop and access point is connected to the TechOffice_Sw1 via wired. The TechOffice_Sw1 connects to the Core Switch through a wire.

1.4 Meeting Room

The Meeting room consists of the following components:

- 5 Laptops
- 2 Personal Computers
- 1 Access Point
- 1 Switch

The 5 laptops are connected wirelessly to the access point. The 2 personal computers and the 1 access point are connected to the switch via a wired connection. Lastly, the switch is connected to the Core Switch in the machine room.

1.5 Reception/Waiting Area

The Reception/waiting area consists of the following components:

- 1 Switch
- 2 Personal Computers
- 1 Printer
- 1 Access Point
- 4 Mobile Devices

Staff can have between 2 and 4 devices on the Reception wifi, where guests will be limited. These devices connect wirelessly to the access point. 2 Personal computers, a printer and the access point are connected via a wired to the switch. This switch also connects to the Core switch with a cable.

1.6 Kitchen

The Kitchen room consists of the following components:

- 1 switch
- 1 access point
- 4 IoT
 - 1 Appliance
 - 1 Carbon Dioxide Detector
 - 1 Humidifier
 - 1 Humidity Monitor

The 4 IoT connects to the Kitchen Access Point via wireless. The Access Point is connected to the Kitchen Switch with a wired connection. The Kitchen Switch is connected to the Core Switch, which enables the Internet connection for the IoT devices.

1.7 Open Floor Plan

The open floor plan consists of the following components:

- 5 Access Points
- 5 Printers
- 40 Personal Computers
- 12 Laptops

5 switches, each with 8 personal computers connected via wire, in a star topology that are placed next to each other horizontally. Only the 1st, 3rd, and 5th (last switch in the row) have additional access points connected via wired, where each access point has 2 laptops connected wirelessly. E.g. the first switch, OpenFloorSwitch6 (left-hand side), has an access point connected with a cable (OF AP(1)), which has two laptops connected wirelessly. The third

switch, OpenFloorSwitch2 (middle), is an access point connected with a cable (OF AP(1)(1)), which has two laptops connected wirelessly. The fifth switch, OpenFloorSwitch(right-hand side), has an access point connected with a cable (OF AP), which has two laptops connected wirelessly. Next, we have a switch (OpenFloorSwitch(1)) that has 2 wired access points connected (AccessPoint 4 and AccessPoint3). There are 4 laptops connected wirelessly to AccessPoint4, and there are 2 laptops connected wirelessly to AccessPoint3. We have an OpenFloorPrinterSwitch, which has 5 printers connected via a wired connection. Lastly, all the switches in the open floor plan connect to the core switch via wire.

2.1 Server Topology

The typology of the network will be described in two ways. The first concerns the physical layout of the topology, and the second concerns the data flow of the topology (logical topology).

2.2 Physical Topology

This network makes use of a star topology, with the centre of the topology being the machine room that houses crucial devices, such as servers, routers, etc.

All the end devices, such as personal computers, laptops, printers, etc, are connected (with copper straight-through cables) to switches with distinct VLAN setups. These switches are then connected to a single core switch in the centre that then connects to the router, which allows for internet connection.

Through the means of trunking, individual cable connections to the different switches can be used to allow communication of all devices in the network with one another.

Remote Access


The proposed solution for remote action is to use remote access applications like TeamViewer or AnyDesk. Any of these applications can be used to remotely access the computers to assist staff with errors or with troubleshooting. The best way to access computers remotely is to use a VPN or a VPN connection. This was not set up in Packet Tracer, but this can be implemented in the future as the system of the client grows.

Budget

| Devices | Quantity | Price (R) | Total (R) |
|--------------------------------------|----------|-----------|-------------------|
| Switches (33650- 24ps) | 12 | 2,000.00 | 24,000.00 |
| Router (2911) | 1 | 1,500.00 | 1,500.00 |
| Copper Straight Through Cable (100m) | 6 | 300.00 | 1,800.00 |
| Access Point | 15 | 2,500.00 | 37,500.00 |
| Labour | 126 | 450.00 | 56,700.00 |
| AnyDesk | 1 | 1244.48 | 1,244.48 |
| Total | | | 121,500.00 |

Budget based on Takealot:

Switches:



Sponsored

TP-LINK 24-Port Gigabit Easy Smart Switch,...


TP-Link

R 1,999 R 2,000

In stock CPT JHB

★ 4.2 (4)

+ Add to Cart



Sponsored

TP-LINK 8-Port Gigabit Desktop Switch (LS108G)


TP-Link

R 429 R 500

In stock CPT JHB

★ 4.4 (10)

+ Add to Cart




Sponsored

TP-LINK 10-Port Gigabit Desktop Switch

TP-Link

R 1,599 R 2,000

In stock CPT JHB



Sponsored









TP-LINK 28 Port POE GB Smart Switch

TP-Link


R 5,499 R 8,000

In stock CPT JHB

Routers:

| | | | |
|---|--|--|--|
|  <p>Sponsored</p> <p>TP-LINK Archer MR200 Wireless Dual Band 4G LTE Router</p> <p>TP-Link</p> <p>R 1,699 <small>R 2,499</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.7 (17)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>TP-Link MR6400 300Mbps Wireless N 4G LTE Router</p> <p>TP-Link</p> <p>R 1,499 <small>R 1,899</small></p> <p>In stock CPT DBN JHB</p> <p>★ 5 (2)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>Tenda F3 300Mbps Wireless Router</p> <p>Tenda</p> <p>R 275</p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>WIRELESS-N Wifi Extender / Repeater - up to 300Mbps</p> <p>R 160</p> <p>In stock CPT JHB</p> <p>★ 4.3 (98)</p> <p>+ Add to Cart</p> |
|  <p>Mercusys MW302R 300Mbps Multi-Mode Wireless Router</p> <p>Mercusys</p> <p>R 1,199 <small>R 1,599</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>MTN M30s Mi-Fi Router - LTE SS - Black (NL)</p> <p>MTN</p> <p>R 249</p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>TP-Link MR6400 300Mbps Wireless N 4G LTE Router</p> <p>TP-Link</p> <p>R 1,399 <small>R 1,899</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>Predator Connect W6x Gaming Router High-Speed Wireless</p> <p>Acer</p> <p>R 1,799 <small>R 1,999</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |

Copper-Straight Through (LAN cable prices used):



61% OFF

High Quality CAT6 LAN Network Cable 100M

ZATECH









R 272 R 700

In stock CPT DBN JHB

★ 4 (87)

+ Add to Cart

Access Points:

| | | | |
|---|--|---|--|
|  <p>Sponsored</p> <p>TP-LINK Archer MR200 Wireless Dual Band 4G LTE Router</p> <p>TP-Link</p> <p>R 1,699 <small>R 2,499</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.7 (17)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>TP-Link MR6400 300Mbps Wireless N 4G LTE Router</p> <p>TP-Link</p> <p>R 1,499 <small>R 1,899</small></p> <p>In stock CPT DBN JHB</p> <p>★ 5 (2)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>TP-Link TL-EAP110 300Mbps Wireless N Ceiling Mount Access Point</p> <p>TP-Link</p> <p>R 699 <small>R 1,099</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (70)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>Cudy AC1200 Dual-Band Access Point</p> <p>Cudy</p> <p>R 737 <small>R 899</small></p> <p>In stock JHB</p> <p>★ 2.8 (5)</p> <p>+ Add to Cart</p> |
|  <p>Ubiquiti U6+ Access Point</p> <p>Ubiquiti</p> <p>From R 2,616</p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>EAP110 N300 Wireless N Ceiling Mount Access Point</p> <p>TP-Link</p> <p>R 699 <small>R 1,099</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>TP-Link TL-MR6500V 300Mbps Wireless N 4G LTE Router + Vlog</p> <p>TP-Link</p> <p>R 1,099 <small>R 2,799</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |  <p>Sponsored</p> <p>TP-Link Archer BE24000 Quad-Band W6-Fi 7 Router</p> <p>TP-Link</p> <p>R 14,999 <small>R 22,999</small></p> <p>In stock CPT DBN JHB</p> <p>★ 4.5 (163)</p> <p>+ Add to Cart</p> |

Chat App Documentation

Python was the chosen language for this chat app. Within Python, the following packages were used:

- Flask
- Flask-SocketIO
- Eventlet
- Render Cloud (server hosting site)

Flask is a micro web framework designed specifically for Python. It typically allows for web development and web application design.

Flask-SocketIO is critical, as this package allows for server-client bidirectional communication.

Eventlet is also a crucial package that allows for the ease of use of network communication.

Render Cloud is the web server hosting site used for this chat app. The server file was deployed onto this site, and through this, communication across different networks is possible.

This documentation will now be divided into two subsections: the [server.py](#) file and the [ChatApp.py](#) file. [Server](#) allows for communication with the Render Cloud services, and ChatApp forms important Graphical User Interfaces (GUIs for the application).

Server

The server handles critical web connections and socket handling. Several functions are important for the messaging app to function.

- Register
- Handle group messages
- Handle private messages
- Disconnect
- Start server

Register:

Handles the registration by associating a username and a socket with each other. It then also broadcasts that someone has joined all other users.

Group Message

This method takes “data” as an argument, which has two components. A “from” and a “message”. Sender is assigned to from, and message is assigned to message. It then prints the sender name and the message and also broadcasts it to all members.

Private messages

Similar to the group message, it takes an argument “data” but with an additional argument of “to”. So it has a from, a to and a message. The sender is assigned to “from”, the recipient is assigned to “to”, and the message is assigned to message. This message will *not* be broadcasted and will only be printed for the recipient.

To find the recipient, a for-loop is used to cycle through the connected users list to find where the name=recipient and recipient_sid = sid. Once it's found, it only emits a message to the recipient.

Disconnect function

Disconnects a user from the messaging room. The pop() function in the dictionary is used, which returns the username and makes the request.sid null. It then broadcasts the disconnect to all other users.

Start server:

A simple “main” function that prints that the server is indeed running, what port is used on the server. It also begins the flask app for bidirectional server-client communication.

Central Chat App with GUIs

The Graphical User Interface (GUI) of the messaging app is built using Python’s **tkinter** library, with additional support from **ttk** for styled widgets and **PIL** for image handling. The main components and their purpose are:

- **Main Window:** The application launches a Tk window with a dark theme, custom dimensions (600x600), and a logo at the top.
- **Username Entry Screen:** Initially, the user is presented with a frame to enter their username. This frame includes:
 - A label prompting for the username.
 - A text entry field.
 - A "Join Chat!" button which, when clicked, hides this frame and displays the chat interface.
- **Chat Interface (after login):**
 - **Tabbed Chat View:** Implemented using **ttk.Notebook**, allowing:
 - A **Group Chat** tab.

- Dynamic **Private Chat** tabs, one per user when a private chat is initiated.
- **Sidebar for Online Users:** Displays a list of connected users (excluding yourself) in a Listbox. Double-clicking a user opens a private chat tab.
- **Group Chat Tab:** Contains a **ScrolledText** widget to display messages and an input section with an Entry and a Send button.
- **Private Chat Tabs:** Created dynamically when a private conversation starts. Each tab includes its own message display area and input field.
- **Message Handling:**
 - Users can send group messages with Enter key or the Send button.
 - Private messages are sent similarly, and conversations appear in their respective tabs.
 - Incoming messages are displayed in a read-only format and auto-scroll to the bottom.
- **Username Submission:** When a user submits a valid username, the GUI switches from the entry view to the full chat interface, and the client attempts to connect to the server.
- **Responsive Feedback:** Tabs for private messages get a visual indicator (e.g., "username") when new messages arrive in inactive tabs.

This GUI design separates the concerns of user login, message display, and chat controls, making the application intuitive and user-friendly.

Reflection on the messaging app development

Making the messaging app was a rewarding but intense development process. For context, the group primarily focused on finishing the Packet Tracer component, and then addresses the messaging app afterwards. So when we got to the messaging app, there was not as much time left to address it.

One of the biggest challenges of the messaging app was implementing the server that allows for across-network communication. Initially we just worked on local network communication using the IP addresses of devices on the same network, and the program worked well in that regard. But then a problem we encountered was that integrating some sort of server system (using Flask, for example) proved very difficult. So difficult in fact that we had to scrap the entire program and start fresh with not a lot of time left. Once a good web-server host (in this case, Render) was found and implemented into the new program, making a server file and client file was not as difficult as was expected and a basic app could be quickly crafted.

The biggest challenge was thus integrating server functionality as we did not take that into consideration first.

Project management documentation

To coordinate tasks and monitor progress, we utilised GitHub's Project Board to outline the different phases of the project, assign responsibilities, and track deadlines. While we didn't always adhere strictly to the prescribed timelines; the board served as a valuable reference point throughout the development process.

In addition to the GitHub board, we held weekly check-in meetings via Discord (excluding the recess week). These meetings were used to assess progress, reassign or adjust tasks if needed, and set new goals for the upcoming week. This helped maintain accountability and ensured that all members were aligned on project priorities.

Group Roles and Responsibilities

- **Mariska – Group Leader**
Responsibilities: Oversight of the project and coordination
Packet Tracer Sections: *Open Floor Space* and *Meeting/Board Room*
- **Lia – Packet Tracer Expert**
Responsibilities: Troubleshooting, design guidance, and documentation support
Packet Tracer Sections: *13 Offices* and *Technicians' Room*
- **Christoffel – Code Buff**
Responsibilities: Back-end server programming and integration
Packet Tracer Sections: *Machine/Server Room*
- **Ethan – GUI Mastermind**
Responsibilities: GUI design and integration
Packet Tracer Sections: *Machine/Server Room*
Coding: *Server-side logic using Python*
- **Marnus – Administrator**
Responsibilities: Documentation organization, version control support
Packet Tracer Sections: *Reception/Waiting Area* and *Kitchen*

Timeline Overview

- 28/03/2025 - 08/04/2025 : Reflection of online courses
- 08/04/2025 - 16/04/2025 : Network design (using packet tracer)
- 18/04/2025 - 16/05/2025 : Text Message App
- 07/05/2025 - 16/05/2025 : Documentation

This can also be viewed on our GitHub Project Board: [Roadmap · CMPG 315 Project - Group 2](#)

Meeting outline

Attendance

| | In-Person | Discord | Discord | In-Person | In-Person | Discord |
|-------------|------------------|------------------|------------------|------------------|------------------|------------------|
| | 2025/04/01 | 2025/04/24 | 2025/05/08 | 2025/05/10 | 2025/05/11 | 2025/05/13 |
| | 13:00 - 14:00 | 11:00 - 12:30 | 14:30 - 16:00 | 10:00 - 19:00 | 10:00 - 14:00 | 17:30 - 19:00 |
| Mariska | X | X | X | X | X | X |
| Lia | X | X | X | X | Online | X |
| Ethan | X | X | X | X | X | X |
| Marnus | X | X | X | Online | – | X |
| Christoffel | X | X | X | X | X | X |

Minutes of the meeting

Meeting 1

- Mariska is appointed as the leader
- What we are expected to do is discussed
- We got to know each other
- We appointed the work to do (due 8 April)
- We also made a Discord server for future meetings since the semester is close
- We made a Google Drive document for our first phase
- Everyone familiarised themselves with the GitHub project/repository Mariska made beforehand

Meeting 2

- Looked at the Packet Tracer requirements
- Gave all members a chunk of the work to be done by Sunday, 4 May
- The work was distributed as follows:
 - Mariska – Meeting room and open floor space
 - Lia – 13 Offices and technician's office
 - Marnus – Reception and Kitchen
 - Ethan and Christoffel – Machine room

Meeting 3

- Discussed what technologies to use for the chatApp
- Decided on using Python
- Chatted about all outstanding documentation to be done
- Discussed merging all the documentation already done into one document
- Discussed making a PowerPoint presentation for the demo day
- Discussed having a "work day/s" on the weekend
- Planned a workday on Saturday between 10 and 5

Meeting 4

- Today was set out to work without interruptions
- Everyone picked a different part of the project to work on:
 - Lia - Packet Tracer
 - Mariska - Documentation + Presentation
 - Ethan and Christoffel - Chatapp
 - Marnus - Helped with Packet tracer documentation online
- At the end of the day some of us agreed to get together the next day as well and others were excused for mothers day plans and hard work

Meeting 5

- Mariska, Christoffel and Ethan came together in person and worked on the chat app as well as the github readme
- Lia worked on packet tracer at home

Meeting 6

- Finalized the chat App GUI
- Finalized the Documentation
- Chatted about what everyone is going to present in the demo

Reflection of the project

Mariska

This project was a rollercoaster. I was appointed the group leader, thus putting me in charge of managing the meetings, distributing work and also naturally helping with work wherever necessary. Firstly, Packet Tracer was very difficult to use and understand. I assigned a section for everyone to do and ultimately appointed Lia to refine it for us, but this proved very difficult

since no one knew how to use packet tracer and eventually we also realised everyone used a different version of Packet Tracer making it hard to do and understand. Lia also said that she really struggled with the refinement and had to start from scratch more than once, so all in all, I do not feel confident that I will ever use Packet Tracer again, but it was a good learning opportunity.

The ChatApp was also very challenging but gratifying when it eventually worked. Christoffel and Ethan mostly worked on the ChatApp using Python, and it was frustrating to figure out since it was unmarked territory, but we learned a lot from it, like using sockets and hosting a server to be connected to the internet.

Lastly, the documentation was luckily a breeze since we kept track of meetings, how Packet Tracer was used, etc , throughout the project.

Cecilia

This project was challenging and intuitive. My main objective was to complete Packet Tracer and to assist with documentation. Whilst working with Packet Tracer, I encountered a lot of bugs and settings that had to be added up to three or four times because of the slow application that had to be used. After everyone added their parts to the GitHub repository, I added everything together and applied the VLANs and configurations. I struggled with the configuration for at least 10 hours, and after a lot of time spent on Packet Tracer, it finally worked, and I couldn't be more glad. There were no group conflicts that I encountered. The scheduling was a problem, but our team leader managed to create great times for everyone to meet and made sure that there were no conflicts.

Christoffel

This project was an interesting but challenging task. A lot was learnt from both sections of this project and applying to these tasks together as a group. For the packet tracer, it probably taught me more of how network components interact ,more so than the theory, especially concepts like VLAN, DHCP, DNS, and other crucial components. It was challenging actually learning how to set up even basic components and even just making different computers or servers ping was exciting. Getting together as a group to discuss who would do what, and how we would merge it was also a fun but challenging time. Especially with merging files, where instead of making one large packet tracer file we all could work on together, we instead made separate files. This was in fact not the best idea and we quickly pivoted back to one large file we all contributed to.

For the messaging app, it was quite an intense experience as we had to develop it under heavy time constraints, and then also had to redo it after realising we made a critical error in trying to attach a server to the end product instead of developing with server-client communication from the ground up. But once the app was up and running properly for the first time, then it was very satisfying! It was also nice to learn about how serving hosting works, with services like Render.

Overall, the project was a good challenge and the group I worked with were fantastic! There were definitely moments of hard crunch, but we managed to overcome these obstacles.

Marnus

This project was very challenging and different because of Packet Tracer. My role included working on the kitchen and reception room, and to do the documentation of packet tracer. Since I have never touched cisco packet tracer there was a deadline made and I needed to learn it from scratch in order to meet these deadlines. Thus there was a lot of self study involved and trial and error. Thankfully, I am part of a great team/group that would always help and respond if unsure. For instance, I got lost on GitHub, but my group helped me. I feel the time and effort made at first to get to know everyone in the group had a big impact on why everyone met their deadlines and the work could be divided fairly. We all had different schedules but still managed to meet in person and in discord. The communication was very good which made the documentation easier to do for the whole project.

Ethan

This project was quite a challenging and but a rewarding experience. The work with Packet Tracer was quite new for me, and so designing the machine room in addition to connecting all of the needed servers certainly took some getting used to. The setup of DHCP, DNS, as well as routing between different parts of the network especially taught me the actual network component interactions beyond just the theory. I felt locally building it then connecting it to the internet would be easier on the server side. That method failed to perform so a revision occurred. Concerning the messaging app, building the back end was familiar, but creating a GUI in Python was new to me. That was a valuable part of the project for me, but also a steep learning curve.

Having the team was an outstanding aspect of it. Even though there were some technical challenges, they did not obstruct that thing. Each of us found a place within the project, and everyone gave it their all. Each member excelled in different areas, so we all grew a lot closer during the process. Surely, I would work with them again indeed.