

Dr. UML

March 20, 2025

CSIE IV 蕭耕宏110590005
CSIE IV 黃冠鈞110590028
CSIE IV 張庭瑋110590035
CSIE IV 吳宥駒110590066
Homework #2

1 Change History

1.1 HW1

- Add section Problem statement
- Add section Development Language

1.2 HW2

- Add section 3-8.
- Change section "Development Language" to "Software Environments" as demanded.
- Change the project name to "Dr. UML"

2 Problem statement

There are several tools available for creating UML diagrams on the internet but many of them come with paid subscriptions or limitations that make them less accessible. Moreover, they often resort to creating poorly formatted documents due to the lack of affordable, high-quality options. As the result, we propose Dr. UML.

Dr. UML is an innovative collaborative platform designed for software developers, system architects, and students who need to efficiently create and manage Unified Modeling Language (UML) diagrams.

The tool meets the pressing need for collaborative designing by allowing teams to work together simultaneously, regardless of their physical location. In addition, it offers real-time updates and integrated communication features. Dr. UML will be used primarily in design meetings, brainstorming sessions, and technical workshops where immediate visual feedback is essential. It is needed when precise and dynamic visual representation of complex systems is required to align team understanding and streamline development processes.

Dr. UML integrates a robust set of customizable UML elements with drag-and-drop functionality and real-time collaboration. This not only enhances the creative aspects of system design but also ensures that technical requirements are met with precision and clarity. The platform's intuitive design and collaborative capabilities make it an essential tool for modern software development teams, system architects, and students aiming to create high-quality UML diagrams efficiently.

3 Summary of System Features

- SF01: Create a UML diagram file.

- SF02: Edit a UML diagram(draw, edit Component properties, copy and paste Components)
- SF03: Save and load progress.
- SF04: Export UML diagram into image formats.
- SF05: Start a online Session, allowing other Users to join.
- SF06: Connect to online Sessions, edit UML with other users simultaneously.
- SF07: Real-time chatroom in a online Session.

4 Use Case Diagram

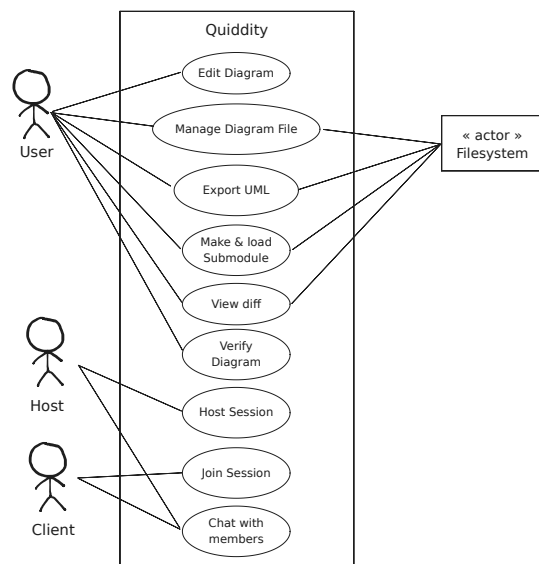


Figure 1: Use case diagram

5 Use Cases

5.1 UC01: Edit UML

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**
 - **User:** Wants to create and connect Components.

- **Preconditions:**

- System is up.

- **Success Guarantee:** UML is edited according to the User's specifications.

- **Main Success Scenario:**

1. User drags Gadgets from Toolbox.
2. User edits the Gadgets.
3. User establishes connections between Gadgets via Associations.
4. User modifies the Associations as needed.
5. Steps 1-4 are repeated in any sequence until diagram is complete.

- **Extensions:**

- *a In the event of System failure, User restarts the system. UML will revert to the last successfully saved state.
- *b When editing text, User may select the following text styling options:
 1. Text color
 2. Text size
 3. Text font
 4. Bold, Italic, Underline
- *c User may select Gadgets within the canvas.
- *d When User drags Gadget with multiple Associations, System will automatically arrange them.
- *e User can determine the layering order when Gadgets overlap.
- *f User may copy and paste Components.
 1. Copying or pasting Associations will also include connected Gadgets.
- *g User may undo or redo actions.
 - 1.a Edit fails if Gadget is dragged to an invalid location.
 - 2.a Different Gadgets will have distinct Fields available for editing.
 - 2.b Edited Gadgets will automatically scale to fit the changes.
 - 3.a If User connects Gadget to nothing, a default Gadget will be generated automatically.
 - 3.b Deleting a Gadget will also remove the associated connections.
 - 3.c Self-Associations are allowed.
 - 4.a User may change the type of Association.
 - 4.b User may add or remove text fields in Association.
 - 4.c When multiple Associations are created between two Gadgets, System will automatically distinguish the different paths to prevent overlap.

- 4.d User can modify the path of an Association.
- **Special Requirements:**
 - When multiple users are editing, no unintended behavior should occur.
- **Frequency of Occurrence:** Often
- **Open Issues:** None specified

5.2 UC02: Manage UML File

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**
 - **User:** Wants to load existing UML and save in the Filesystem.
 - **Filesystem:** Requires the file to be read/saved properly.
- **Preconditions:**
 - System is up.
- **Success Guarantee:**
 - UML file is saved in the Filesystem.
 - UML file is loaded into System.
- **Main Success Scenario:**
 1. User views all the available UML files from Filesystem.
 2. User selects a file and loads it into the System.
 3. User edits UML as described in UC1.
 4. User saves UML to Filesystem.
- **Extensions:**
 - 1-2.a User may choose to create a new, empty UML instead of loading one from Filesystem.
 - 2.a If System fails loading the file, User will be prompted to either retry the operation or abort.
 - 4.a If System fails saving the file, User will be prompted to either retry the operation or abort.
 - 4.b When saving the file, User may select one of the supported UML formats (for future verification purposes).
- **Frequency of Occurrence:** Occasionally
- **Open Issues:** None specified

5.3 UC03: Export UML

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**
 - **User:** Wants to export UML project to image formats with desired name and extension.
 - **Filesystem:** Requires the exported image to be saved properly.
- **Preconditions:**
 - System is up.
 - User has opened a UML project.
- **Success Guarantee:** Exported image is saved to Filesystem.
- **Main Success Scenario:**
 1. User starts the exporting current UML.
 2. User selects a supported image format and specifies a filename.
 3. System saves the exported image to Filesystem.
- **Extensions:**
 - 2.a Supported formats:
 - * JPEG
 - * PNG
 - * SVG
 - * WebP
 - 3.a If Filesystem fails to save the exported image, User will be prompted to either retry the operation or abort.
- **Frequency of Occurrence:** Occasionally
- **Open Issues:** None specified

5.4 UC04: Manage Submodule

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**

- **User:** Wants to export and import Submodule.
- **Filesystem:** Wants to save the Submodule file.
- **Preconditions:**
 - System is up.
 - User has opened a UML project.
- **Success Guarantee:**
 - Submodule is exported on Filesystem.
 - Submodule is imported on canvas.
- **Main Success Scenario:**
 1. User imports Submodule as template.
 2. User edits UML as described in UC1.
 3. User selects Components and exports them as a Submodule.
 4. The Submodule file is saved to Filesystem.
- **Extensions:**
 - 1.a If Filesystem fails to import Submodule, User will be prompted to either retry the operation or abort.
 - 3.a If nothing is selected, User cannot export it as Submodule.
 - 3.b Optionally, the saved Submodule may have empty Fields.
 - 4.a If Filesystem fails to save Submodule file, User will be prompted to either retry the operation or abort.
- **Frequency of Occurrence:** Sometimes
- **Open Issues:** None specified

5.5 UC05: Manage Submodule

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**
 - **User:** Wants to export and import Submodule.
 - **Filesystem:** Wants to save the Submodule file.
- **Preconditions:**

- System is up.
 - User has opened a UML project.
- **Success Guarantee:**
 - Submodule is exported on Filesystem.
 - Submodule is imported on canvas.
- **Main Success Scenario:**
 1. User imports Submodule as template.
 2. User edits UML as described in UC1.
 3. User selects Components and exports them as a Submodule.
 4. The Submodule file is saved to Filesystem.
- **Extensions:**
 - 1.a If Filesystem fails to import Submodule, User will be prompted to either retry the operation or abort.
 - 3.a If nothing is selected, user cannot export it as Submodule.
 - 3.b Optionally, the saved Submodule has empty Fields.
 - 4.a If Filesystem fails to save Submodule file, User will be prompted to either retry the operation or abort.
- **Frequency of Occurrence:** Sometimes
- **Open Issues:** None specified

5.6 UC06: Verify UML

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**
 - **User:** Wants to verify the correctness of UML.
- **Preconditions:**
 - System is up.
 - A UML is available.
- **Success Guarantee:** System verifies and displays the correctness of the UML diagram.
- **Main Success Scenario:**

1. User opens a UML project.
2. User instructs System to verify the UML.
3. System checks the correctness of the UML.
4. System displays the verification results.

- **Extensions:**

- 1.a If System fails to open the UML file, User will be prompted to either retry the operation or abort.
- 3.a If System fails to verify the UML, User will be prompted to either retry the operation or abort.
- 3.b System verifies the UML by diagram type.
 1. If the UML type verification is unavailable, inform the User.
- 4.a If System fails to display result, User will be prompted to either retry the operation or abort.
- 4.b System informs User of verification results, which may include:
 1. All clear, UML is correct in terms of diagram type.
 2. Warnings, UML is mostly free of syntax errors, but contains some bad smells™.
 3. Invalid, UML contains critical errors.

- **Special Requirements:**

- Error messages should be user-friendly and provide actionable insights.

- **Technology:**

- **Frequency of Occurrence:** Sometimes

- **Open Issues:** None specified

5.7 UC07: Host Session

- **Scope:** Dr. UML

- **Level:** User goal

- **Primary Actor:** User

- **Stakeholders and Interests:**

- **User:** Wants to start a Session.

- **Preconditions:**

- System is up.
- User has opened a UML project.

- **Success Guarantee:** User opened a Session, enabling clients to join.
- **Main Success Scenario:**
 1. User opens a session for a project.
 2. User assumes the role of Host. Host can accept connections from clients.
- **Extensions:**
 - 1.a If errors occur during establishment of session, notify User of the issue encountered.
 - 2.a If a connection error occurs, notify User of the issue encountered.
- **Frequency of Occurrence:** Sometimes
- **Open Issues:** None specified

5.8 UC08: Join Other's Session

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** Client
- **Stakeholders and Interests:**
 - **Host:** Wants Client to join current opened project.
 - **Client:** Wants to connect to an existing project.
- **Preconditions:**
 - System is up.
 - A stable connection exists between Host and Client.
 - Host opened a UML project.
- **Success Guarantee:** Connection is established and maintained, UML is synced, and no conflicts occur.
- **Main Success Scenario:**
 1. Host starts a Session.
 2. Client connects to the Session.
 3. Both Host and Client edit the UML as described in UC1.
 4. Step 3 is repeated until UML is complete.
 5. Host ends the Session.
- **Extensions:**

- *a At any time, a Component can only be edited by exactly one User.
 - 2.a If a connection error occurs, notify Client of the issue encountered.
 - 3-4.a Host can remove any Client from the Session.
 - 3-4.b If an action fails to send, the System retries the action.
 1. If retry limit is reached, System will remove that Client from current Session.
 - 3-4.c Clients may redo and undo their own edits.
 - 5.a Session ends if Host closes it, whether intentionally or unintentionally.
- **Frequency of Occurrence:** Sometimes
 - **Open Issues:** Undo/redo conflicts

5.9 UC09: Chat with Members

- **Scope:** Dr. UML
- **Level:** User goal
- **Primary Actor:** User
- **Stakeholders and Interests:**
 - **User:** Wants to communicate with other users in Session via text messages.
- **Preconditions:**
 - System is up.
 - Users have joined a Session.
- **Success Guarantee:**
 - Users in Session can communicate with each other with text messages.
 - Users in Session can view chat history.
- **Main Success Scenario:**
 1. User opens the chatroom for current Session.
 2. User views chat history.
 3. User types and sends messages.
- **Extensions:**
 - 1.a If a new Session is created, System creates a new chatroom with no messages.
 - 2.a If System fails to load the chatroom, it will attempt to retry.

1. If retry limit is reached, notify User of the issue encountered. User will not be able to view the previous messages.
 - 3.a If System fails to send User's message, it prompts the User to either remove or resend the message.
- **Frequency of Occurrence:** Sometimes
 - **Open Issues:** None specified

6 Non-functional Requirements and Constraints

6.1 Performance Requirements

- **NFR1: Response Time:** Operations (e.g., dragging components, editing text, collaboration) should respond within **1s**.
- **NFR2: Concurrent Users:** The system should support at least **4 users** editing a UML diagram in real-time.
- **NFR3: File Handling:** Loading or saving UML files should take **less than 3 seconds** (for diagrams with 100+ elements).
- **NFR4: Export Speed:** UML diagrams should be converted to PNG, JPEG, SVG, or WebP formats within **5 seconds**.
- **NFR5: Network Efficiency:** Collaboration mode should minimize data traffic and prioritize critical updates to reduce bandwidth usage.

6.2 Usability Requirements

- **UR1: User-friendly Interface:** Users should be able to understand basic operations within **20 minutes**.
- **UR2: Undo/Redo:** The system should support **at least 50 levels** of undo and redo history.
- **UR3: Accessibility:** Keyboard shortcuts should be provided to enhance usability.
- **UR4: Collaboration Features:** Users should see real-time updates and be able to communicate via chat or annotations.

6.3 Reliability & Availability Requirements

- **RAR1: Uptime:** The system should maintain **99.9% availability**.
- **RAR2: Autosave:** Progress should be automatically saved every **30 seconds**.
- **RAR3: Error Handling:** The system should handle network failures gracefully, allowing users to reconnect without data loss.

- **RAR4: Data Consistency:** All users should see the same UML diagram state in collaborative mode.

7 Glossary

- Submodule: a part of UML diagram, it can be imported into other UML diagram
- Gadget: a block contains text Fields
- Toolbox: A bar containing Components, allowing Users to add them to the canvas.
- Association: connection between two Gadgets
- Field: the place where User can insert text
- Session: A shared project allowing other Users to collaborate.
- Component: Gadget or Association o the canvas.
- Host: A User who has started a Session.
- Client: A User who has joined a Session.
- User: A general term that refers to any participant, including both the Host and Client.

8 Software Environments

Golang.

9 Measurement

蕭耕宏	張庭瑋	黃冠鈞	吳宥駒
25/03/06 13:00 - 14:00	—————	25/03/06 13:00 - 14:00	25/03/06 13:00 - 14:00
25/03/06 15:00 - 16:00	—————	25/03/06 15:00 - 16:00	25/03/06 15:00 - 16:00
25/03/08 19:00 - 23:00	25/03/08 19:00 - 23:00	25/03/08 19:00 - 23:00	25/03/08 19:00 - 23:00
25/03/09 20:30 - 23:30	25/03/09 20:30 - 23:30	25/03/09 20:30 - 23:30	25/03/09 20:30 - 23:30
9 hours	7 hours	9 hours	9 hours

References