AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय

Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

Department of Computer Engineering

"OOMD Miniproject"

Submitted in partial fulfillment of the requirements for the degree of

BACHELOR OF ENGINEERING

In

COMPUTER ENGINEERING

Submitted By

Name of the Student: Piyusha Rajendra Supe

Roll No: 23CO315

Under the Guidance of

Mr. S. R. Bhise

ALL INDIA SHRI SHIVAJI MEMORIAL SOCIETY'S COLLEGE OF ENGINEERING PUNE-411001

Academic Year: 2025-26 (Term-I)

Savitribai Phule Pune University

AISSMS
COLLEGE OF ENGINEERING
ज्ञानम् सकलजनहिताय

Approved by AICTE, New Delhi, Recognized by Government of Maharashtra
Affiliated to Savitribai Phule Pune University and recognized 2(f) and 12(B) by UGC
(Id.No. PU/PN/Engg./093 (1992)
Accredited by NAAC with "A+" Grade | NBA - 7 UG Programmes

Department of Computer Engineering

CERTIFICATE

This is to certify that **Piyusha Rajendra Supe** from **Fourth Year Computer Engineering** has successfully completed her work titled "**Object Oriented Modeling and Design Mini-project**" at AISSMS College of Engineering, Pune in the partial fulfillment of the Bachelor's Degree in Computer Engineering.

Mr. S. R. Bhise	**Dr. S. V. Athawale**	**Dr. D. S. Bormane**
(Faculty In-charge)	(Head of Department)	(Principal)
Computer Engineering	Computer Engineering	AISSMSCOE, Pune

ACKNOWLEDGEMENT

It is with profound gratitude and deep respect that I take this opportunity to acknowledge the invaluable support and guidance I have received throughout the course of my OOMD project. This journey has been both intellectually enriching and personally fulfilling, significantly enhancing my understanding of UML tools, technologies, and their real-world application. First and foremost, I would like to express my heartfelt thanks to **Mr. S. R. Bhise** for his expert guidance, unwavering support, and constructive feedback. His mentorship and encouragement were instrumental in shaping the direction, depth, and quality of this project. I also extend my sincere appreciation to the Head of the Department for their visionary leadership and for fostering an environment that encourages academic growth and innovation. The resources and opportunities provided under their guidance played a crucial role in the successful completion of this project. A special note of thanks goes to the supporting staff, whose timely assistance and cooperation ensured that the process was smooth and efficient. Their commitment and dedication were invaluable during every phase of this project. Finally, I would like to express my deepest gratitude to my parents, whose constant love, patience, and unwavering support have been my source of strength throughout this journey. Their belief in my abilities and their encouragement played a crucial role in keeping me motivated and focused. This project has not only strengthened my technical skills but has also taught me the importance of perseverance and continuous learning. I remain sincerely thankful to everyone who contributed to the successful completion of this project.

Academic Year: 2025-2026

Piyusha Rajendra Supe (23CO315)

TABLE OF CONTENTS

Sr. No	Title	Page No.
1	Acknowledgement……	3
2	Abstract …….....	5
3	Introduction…...……...	6
4	System Requirements …….	7
5	Methodology……	8
6	UML Diagrams……. ……..…….....	9-12
7	Conclusion…….……	13
8	References…….	14

ABSTRACT

This mini project focuses on the systematic development of a complete set of Unified Modelling Language (UML) diagrams to represent the design and functionality of a software system. The primary objective of this work is to practice and demonstrate the application of UML as a standardized modelling language for analysing, designing, and documenting software requirements and architectures. In this project, all major UML diagrams required for a comprehensive representation of a system have been created, including Use Case Diagrams, Class Diagrams, Activity Diagrams, Sequence Diagrams, Collaboration Diagrams, State Transition Diagrams, Component Diagrams, and Deployment Diagrams. Each diagram highlights a different perspective of the system, from functional requirements and user interactions to internal structure, dynamic behaviour, and implementation view. The diagrams were developed iteratively to ensure consistency, clarity, and completeness. Special attention was given to capturing actors and their interactions, system modules, relationships among classes, message flows, state changes, and physical deployment aspects. By covering both structural and behavioural views, the project demonstrates how UML can effectively bridge the gap between user requirements and implementation details. This mini project not only enhances understanding of software modelling but also improves the ability to communicate complex system designs visually. The resulting set of UML diagrams provides a blueprint of the target system, which can serve as a guide for future implementation or as a reference for documentation and quality assurance purposes.

INTRODUCTION

Effective handling of user complaints and service issues is a critical component of any organization's customer relationship strategy. Manual processes for tracking, routing, and resolving complaints often lead to delays, inefficiencies, and poor customer satisfaction. To address these challenges, software-driven Complaint Management Systems (CMS) are widely implemented to streamline the recording, categorization, assignment, and resolution of complaints.

In this project, the Complaint Management System is envisioned as an application that allows customers to register complaints online, while administrators and support staff can categorize, assign, track, and resolve these complaints. The system ensures that every complaint is logged, prioritized, and handled efficiently, while providing updates and feedback to the user.

To fully capture the system, the project produces all major UML diagrams, each offering a distinct perspective:

- **Use Case Diagram** – shows how different actors (e.g., Customer, Admin, Support Staff) interact with the system.
- **Class Diagram** – models the main entities like Complaint, User, Department, and their attributes/relationships.
- **Activity Diagram** – depicts the workflow of processes such as complaint registration, escalation, and resolution.
- **State Diagram** – illustrates how a complaint transitions between states (New → In Progress → Resolved → Closed).
- **Sequence Diagram** – details the sequence of interactions when a complaint is submitted and processed.
- **Component Diagram** – shows how the application's modules (user interface, complaint module, notification module, database) interact.

By preparing these diagrams, the project demonstrates how UML can be practically applied to design a real-world software solution. This modelling process enables early validation of requirements, identification of potential issues, and clear documentation for implementation.

Ultimately, this project emphasizes the power of UML in creating a well-structured blueprint for a Complaint Management System. It bridges the gap between abstract requirements and actual implementation, ensuring a systematic, efficient, and scalable design before development begins.

SYSTEM REQUIREMENTS

1. Hardware Requirements

Component	Minimum Specification	Recommended Specification
Processor	Intel i3 / AMD equivalent	Intel i5 or above
RAM	4 GB	8 GB or above
Storage	500 MB free for UML tools & diagrams	2 GB free
Display	1366×768 resolution	Full HD 1920×1080 or higher
Input Devices	Keyboard, Mouse / Trackpad	Keyboard, Mouse / Trackpad
Internet	Basic broadband (for cloud-based UML tools)	Stable broadband (for real-time collaboration)

2. Software Requirements

Category	Options (Free)	Options (Paid)
Operating System	Windows 10+, Ubuntu 22.04 LTS+, macOS 12+	Same with Enterprise Support
UML Modelling Tools	PlantUML, StarUML (community edition), Visual Paradigm Community, Lucidchart (free tier), draw.io (diagrams.net)	Enterprise StarUML, Visual Paradigm Pro, Enterprise Architect
Text Editor (if code-based)	VS Code, Notepad++, Sublime Text	JetBrains IDEs (IntelliJ, PyCharm)

3. Additional Requirements

- **Internet connection** if you're using cloud tools like Lucidchart or draw.io.
- **Image editing software** (like GIMP or Paint.NET) if you want to refine exported diagrams.
- **PDF/Word software** (MS Word, LibreOffice, Google Docs) to embed diagrams in your project report.

METHODOLOGY

Methodology for UML Diagrams: Complaint Management System

1. Requirement Gathering

- **Functional:** Users submit complaints; Admin assigns and resolves complaints; Employees update status; Notifications and reports.
- **Non-functional:** Secure, role-based, web-based, fast response.
- **Actors:** Customer, Admin, Employee.

2. System Analysis

- Identify key interactions and entities: User, Complaint, Notification, Report.
- Map relationships between actors and system functionalities.

3. UML Diagram Selection

4. Diagram Design Approach

- **Use Case:** Draw actors, use cases, associations (include, extend).
- **Class:** Identify classes, define attributes/methods, depict associations and aggregation.
- **Sequence:** Show lifelines and message flow for scenarios like complaint submission or resolution.
- **Activity:** Represent workflow: Submit → Admin Review → Assign → Employee Update → Resolved → Feedback.
- **State:** Model complaint states: New → Assigned → In Progress → Resolved → Closed → Reopen.

5. Tools

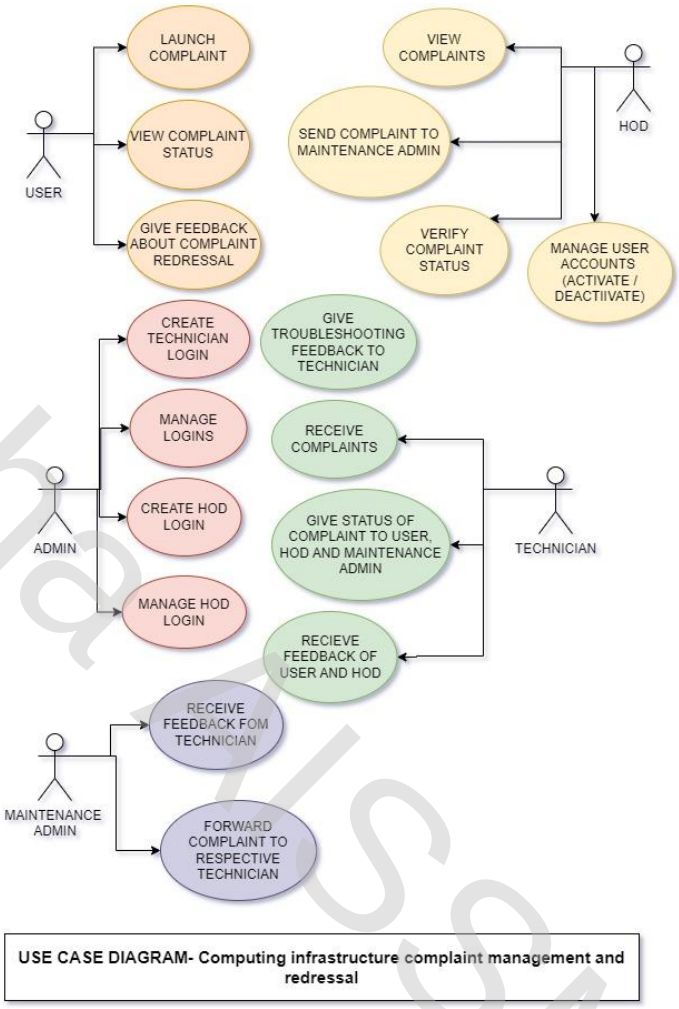
- Online: Draw.io, Lucidchart, Creately
- IDE: StarUML, Visual Paradigm
- Text-based: PlantUML

6. Validation & Documentation

- Ensure all requirements, actors, and flows are represented.
- Include diagrams in the report with titles and brief explanations.

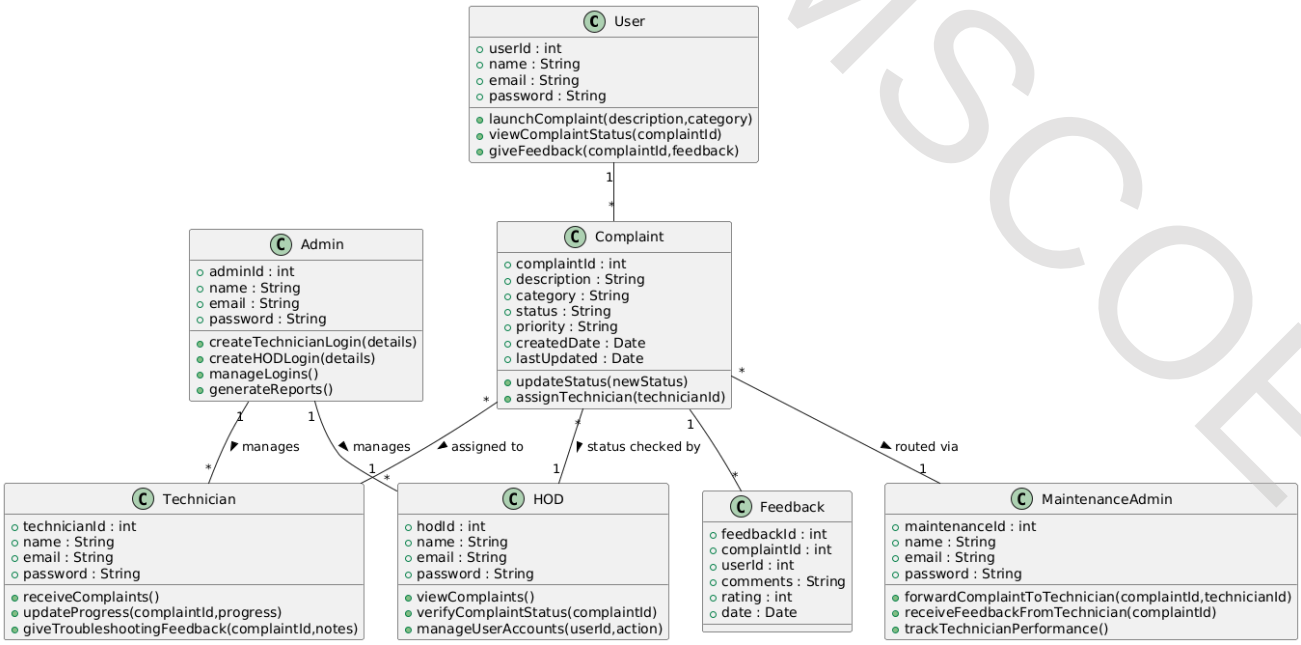
IMPLEMENTATION

[1] Use case diagram

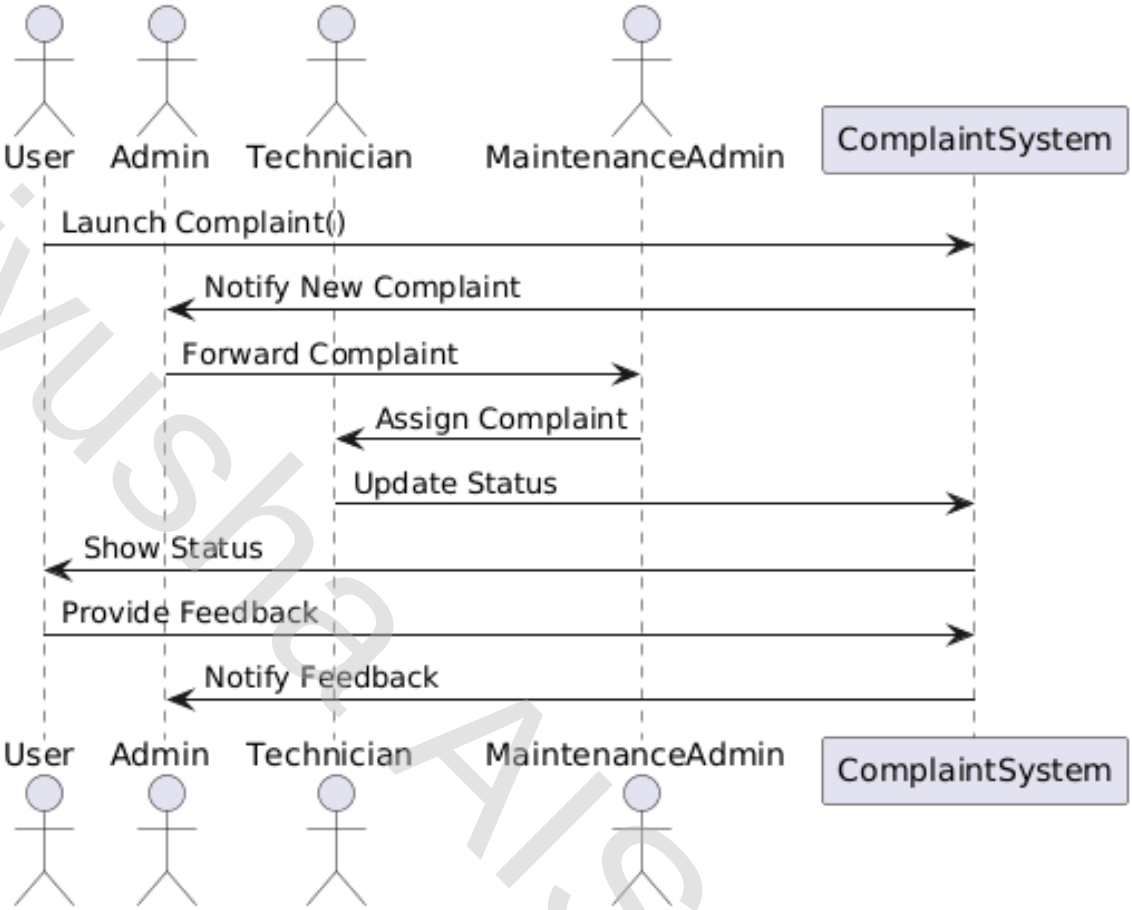


USE CASE DIAGRAM- Computing infrastructure complaint management and redressal

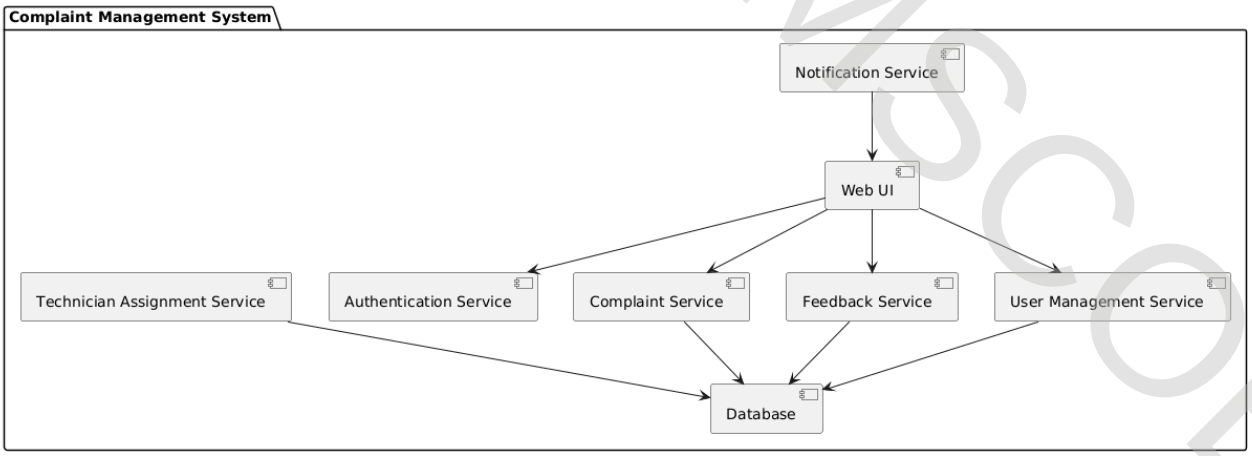
[2] Class diagram



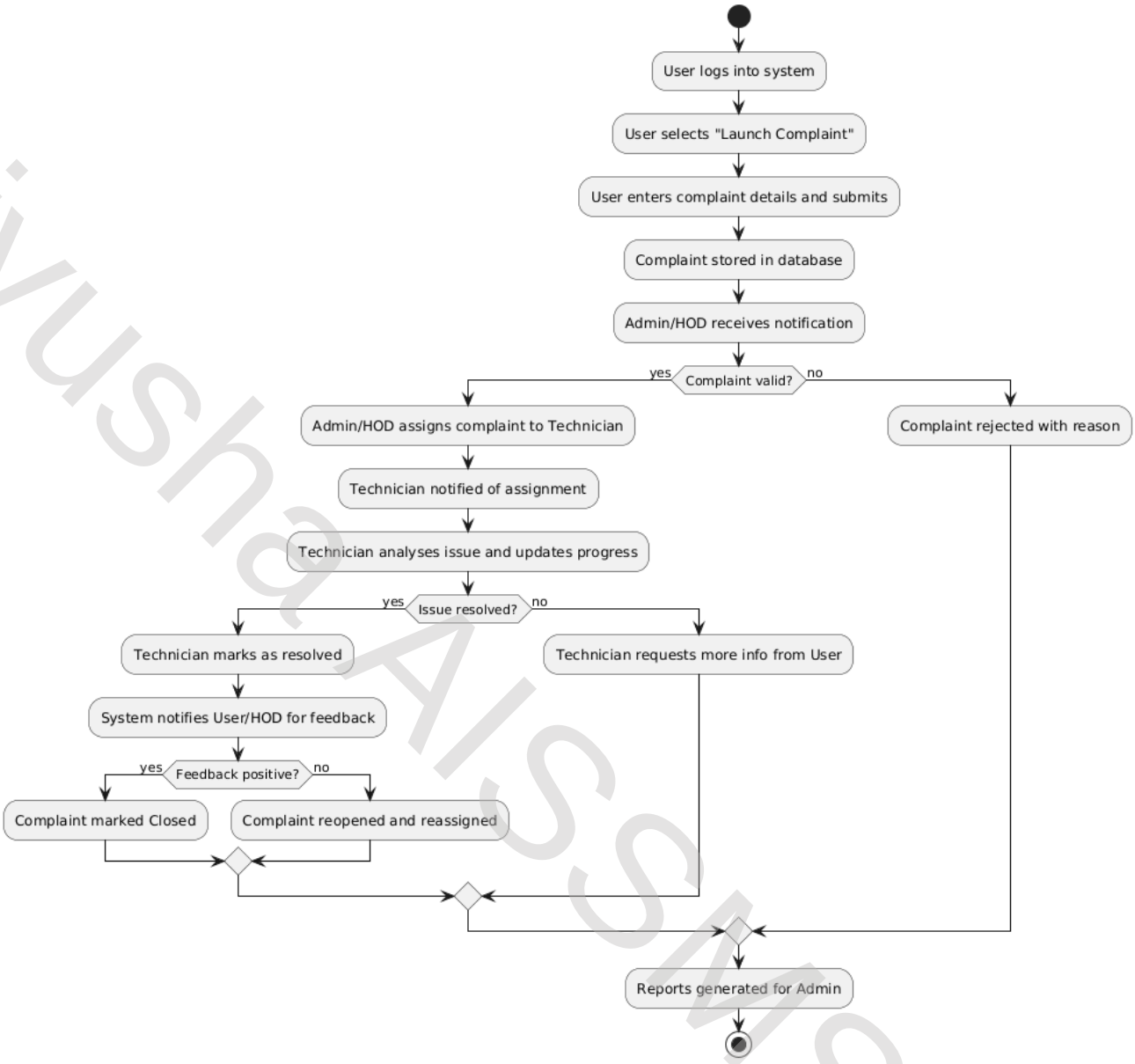
[3] Sequence diagram



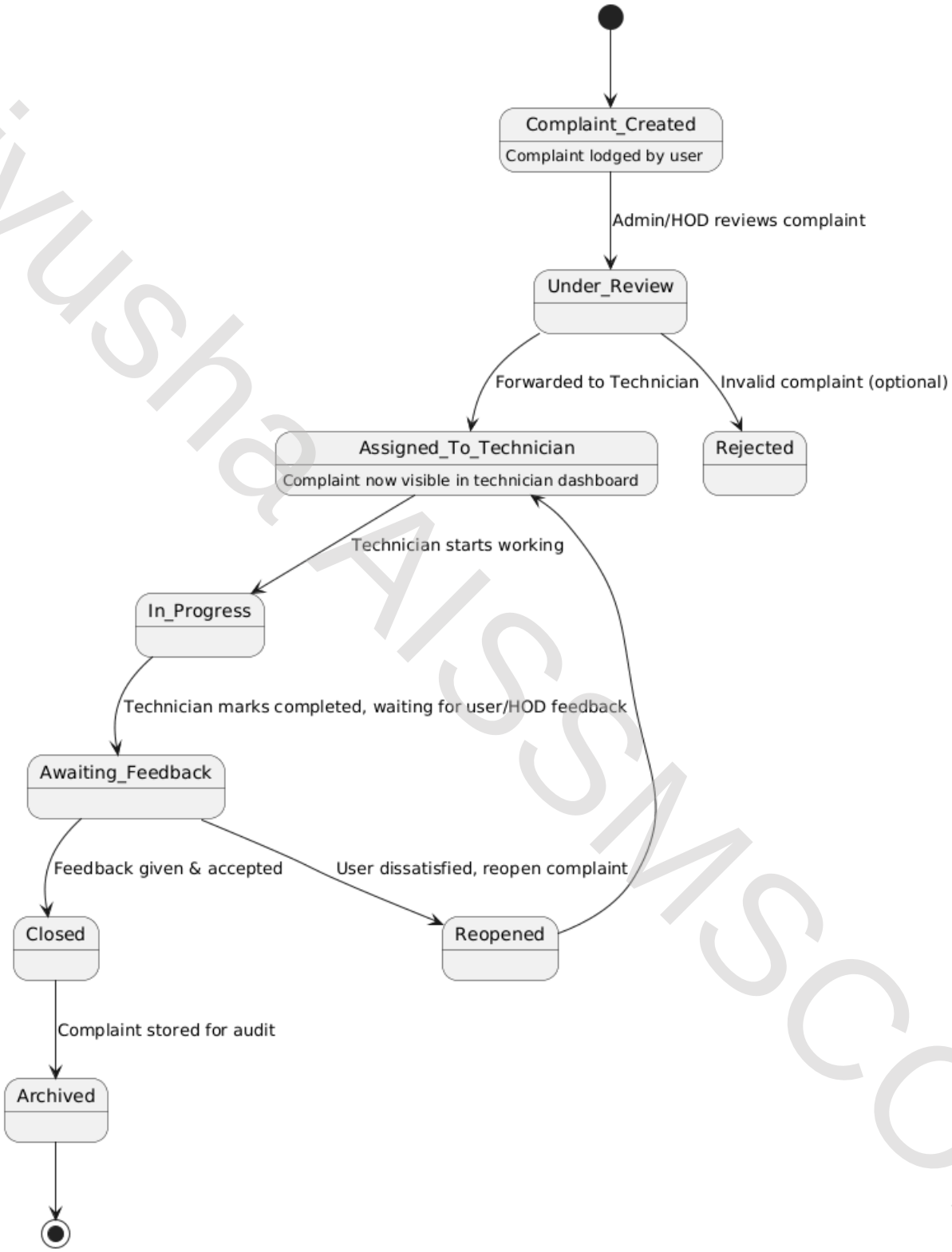
[6] Component Diagram



[4] Activity diagram



[5] State diagram



CONCLUSION

The UML modeling of the Complaint Management System has successfully provided a comprehensive and systematic view of the system's architecture and functionalities. Through the Use Case diagrams, we identified all actors and their interactions, clarifying system requirements and user roles. The Class diagrams captured the structure of core entities, their attributes, methods, and relationships, ensuring a clear blueprint for implementation. Sequence and Activity diagrams illustrated the dynamic behaviour and workflows of complaint handling, enabling developers to understand process flows and interactions step by step. The State diagrams highlighted the life cycle of complaints, ensuring that all possible states and transitions are considered for accurate status tracking.

By following this UML-based methodology, the system design becomes more organized, maintainable, and scalable. It also facilitates better communication among developers, testers, and stakeholders, reducing ambiguity and potential errors during development. Overall, UML diagrams have proven to be an invaluable tool for visualizing, analysing, and designing the Complaint Management System, ensuring that it meets functional and non-functional requirements efficiently. The structured modeling approach lays a strong foundation for successful implementation, future enhancements, and effective project documentation.

REFERENCES

1. https://www.tutorialspoint.com/uml/index.htm
2. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/uml-class-diagram-tutorial/
3. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-class-diagrams/
4. https://creately.com/guides/sequence-diagram-tutorial/
5. https://developer.ibm.com/articles/an-introduction-to-uml/
6. https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-activity-diagram/
7. https://www.lucidchart.com/pages/tutorial/uml-use-case-diagram
8. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-sequence-diagrams/
9. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-activity-diagrams/
10. https://www.geeksforgeeks.org/system-design/unified-modeling-language-uml-state-diagrams/