**Course Code:** MECS 0033

**Course Name:** Artificial Intelligence

**Section:** 52

**Lecturer’s Name:** Dr. Shafaatunnur binti Hasan

**Session/Semester:** 202420252

**Assignment Title:** UTM Campus Assistance Chatbot

**Group:** 3 (College Facilities)

**Member Details**:

|  |  |
| --- | --- |
| **Matrix No.** | **Name** |
| MEC245057 | Tan Zhuang Min |
| MEC245056 | Tan Seng Hooi |
| MEC245046 | Nurul ‘Aini Hamdan |
| MEC245041 | Teng Jiun Wah |

Table of Contents

Table of Contents 2

1.0 Introduction 3-4

2.0 Survey and Interview 5-11

3.0 User Interface Storyboard 12-14

4.0 Knowledge Base Design 15-24

4.1 Entity Recognition Knowledge Representation in First Order Logic 15

4.2 Intent Classification Knowledge Representation in First Order Logic 16-17

4.3 Issue Classification Knowledge Representation in First Order Logic 18

4.4 Resolution Refutation 19-23

4.5 Proving Tree 24

1. Introduction

This proposal presents the development of an AI-based Campus Assistant Chatbot aimed at improving the efficiency and effectiveness of reporting and resolving facility related issues. The document is structured to cover the knowledge representation techniques that underpin the chatbot’s development.

This study will begin with the Survey and Interview phase, which presents the findings of survey to assess the current state of existing system. This aimed to gain a comprehensive understanding of student’s experiences and perceptions regarding existing reporting mechanism. The survey focused on several key aspect, the frequency of students encounter facility related issues, the perceived effectiveness of current system, the clarity and reliability of existing platform, the student familiarity with the current reporting procedures, awareness of ongoing facility issues, the ease of tracking and following up issues, and overall satisfaction with campus facilities. Several questions were design to evaluate the necessity of chatbot-based reporting solution, and the feedback offered a valuable insight into desired chatbot features.

Following this, the User Interface Storyboard section visually demonstrate the chatbot’s intended interaction flow. It emphasizes an intuitive and user-friendly experience, showcasing how users will engage with the system.

At the core of the proposal is the Knowledge Base Design, which outlines how the chatbot interprets, processes, and resolves user reports. This section is subdivided into several critical components. The Entity Recognition will explain how the system identifies and categorizes essential elements. Follow by Intent Classification which describes the chatbot’s method for interpreting the user’s purpose or desired action through formal logic. The Issue Classification details how reported issues are systematically categorized into predefined classes using logical rules and the Resolution Refutation demonstrate how the system evaluates the validity of proposed solutions by applying logical reasoning. Lastly, the Proving Tree visualize the logical reasoning path to reach a conclusion or recommend a solution, ensuring systematic and consistent decision making.

1. Survey and Interview

To help us understand more about the current situation in UTM Johor Bahru regarding campus facilities, a survey form was distributed among the students on this campus. A total of **33 responses** were collected from the survey, and they were analyzed to gain insight and help us prioritize what the important features and functionalities are that the AI chatbot must possess.

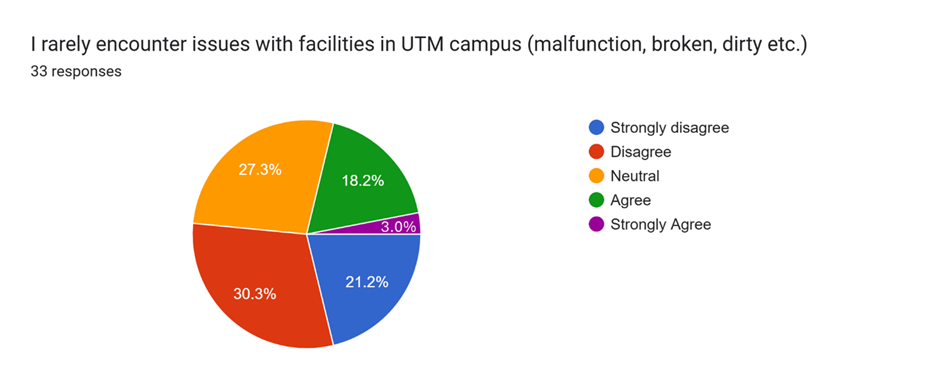
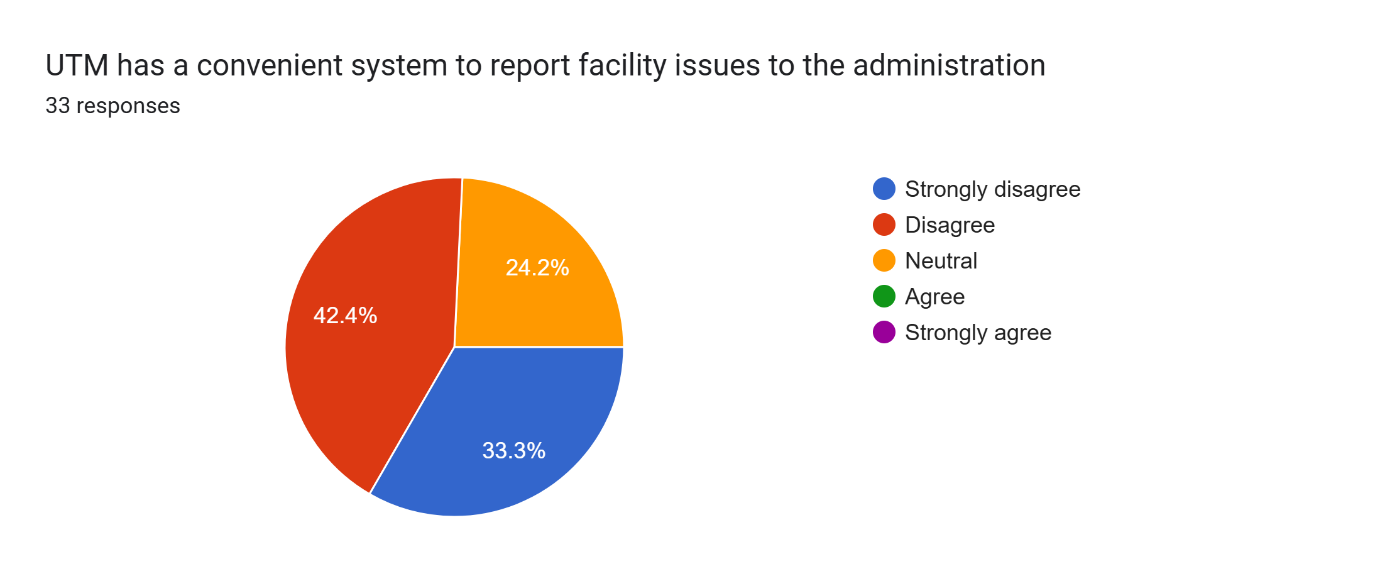
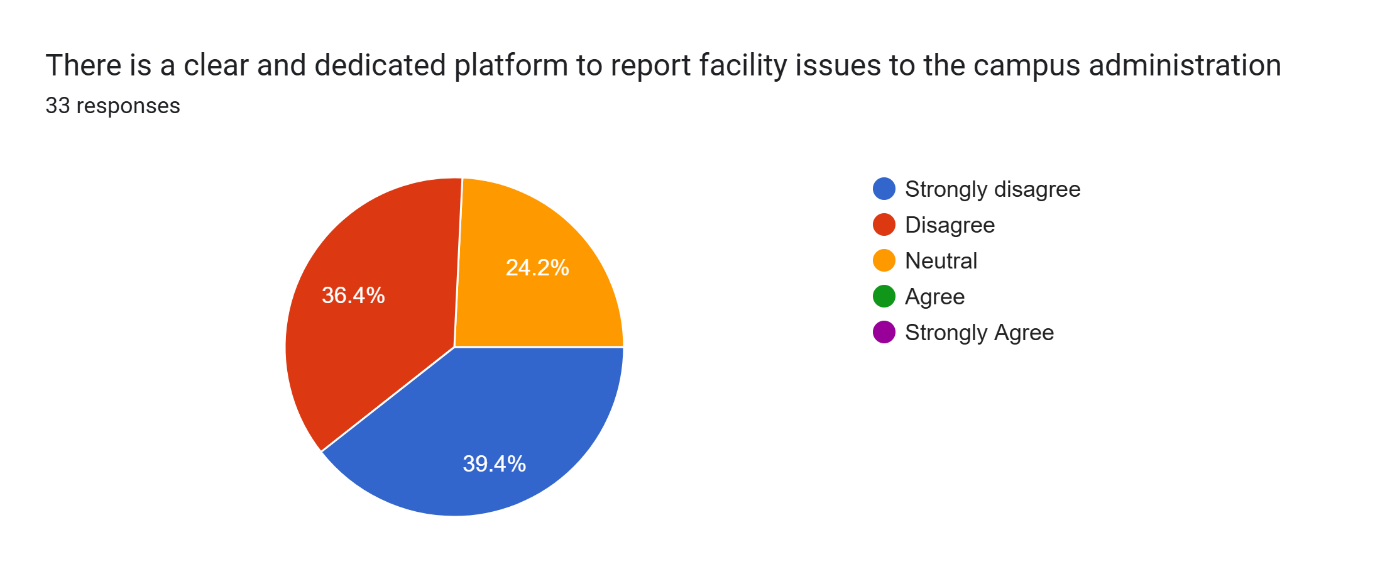
Question 1

Figure 1. Analysis of Respondent Data for Question 1

Among the 33 respondents, more than half of them (51.5%) lean toward disagreeing that they rarely encounter facility issues on the campus. This question was designed to tell us about the current situation of the campus facilities, and the result has shown that it is very common for students to encounter problems with facilities on campus, such as faulty or dirty facilities. This highlights the potential value of bringing in an AI chatbot to help with current UTM facility issues.

Question 2 & 3

Figure 2. Analysis of Respondent Data for Question 2

Figure 3. Analysis of Respondent Data for Question 3

These two questions were asked so that we would know whether it is necessary for the AI chatbot to be able to help students report facility issues. Question 2 responses have shown that most of the students (75.7%) were struggling to report facility issues to the administration. This result is then explained by the responses in Question 3, where most respondents (75.8%) disagree that there is a clear and dedicated platform for this purpose. This reveals the true cause of the inconvenience and urges us to put more emphasis on the issue reporting function of the AI chatbot.

Question 4

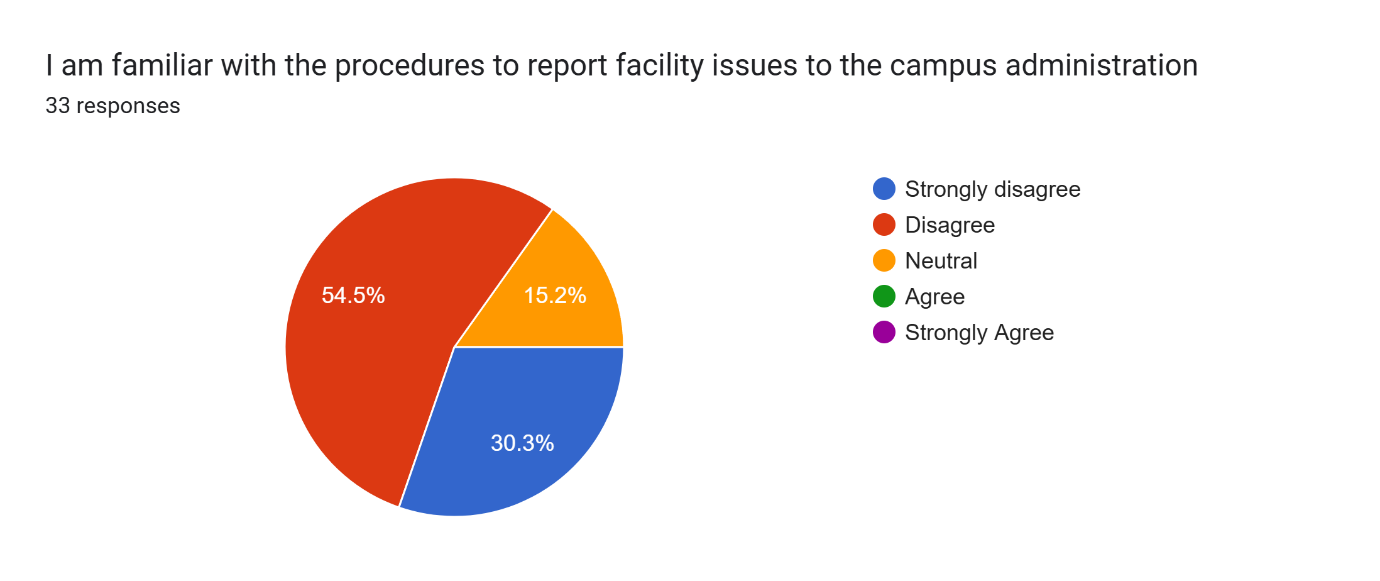


Figure 4. Analysis of Respondent Data for Question 4

In case the responses from Questions 2 and 3 indicated that the campus already had a dedicated platform to report facility issues, this question would tell us how familiar UTM students are in using this platform for issue reporting. Then, from here we would decide whether the AI chatbot should help to guide students to navigate through the platform to report facility issues. However, since the results from Questions 2 and 3 have shown that such a platform does not yet exist, the responses to this question will not be utilized in this assignment.

Question 5

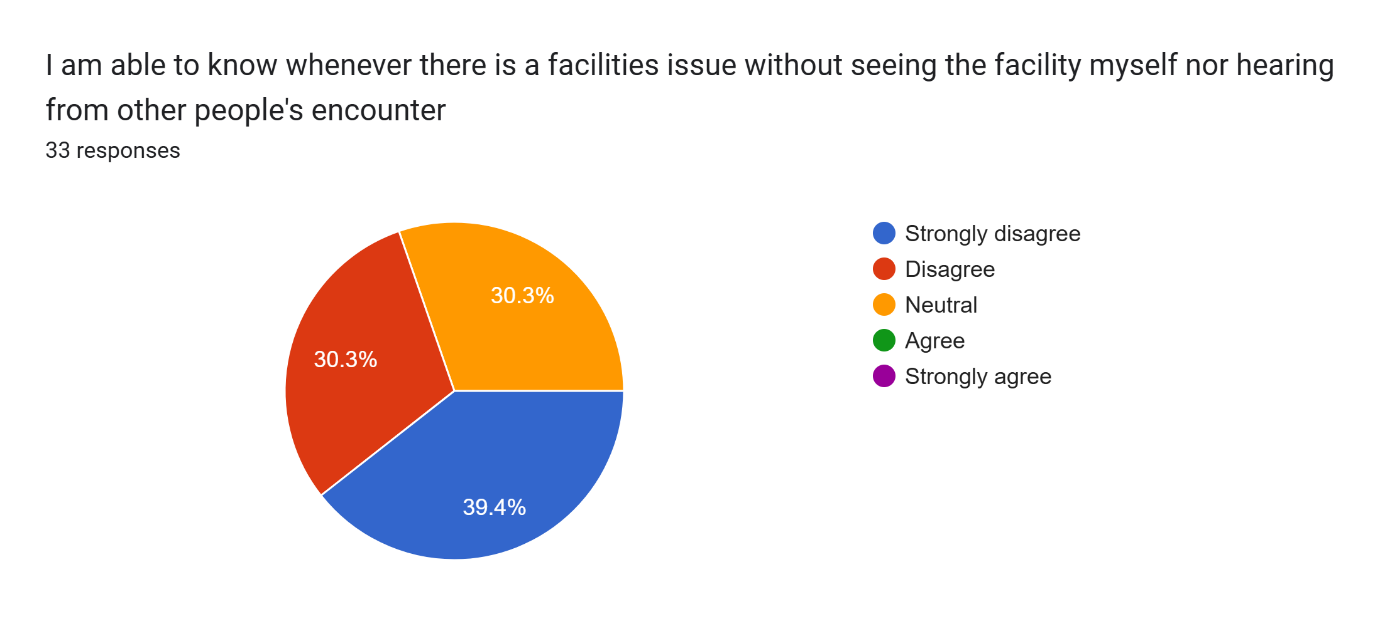
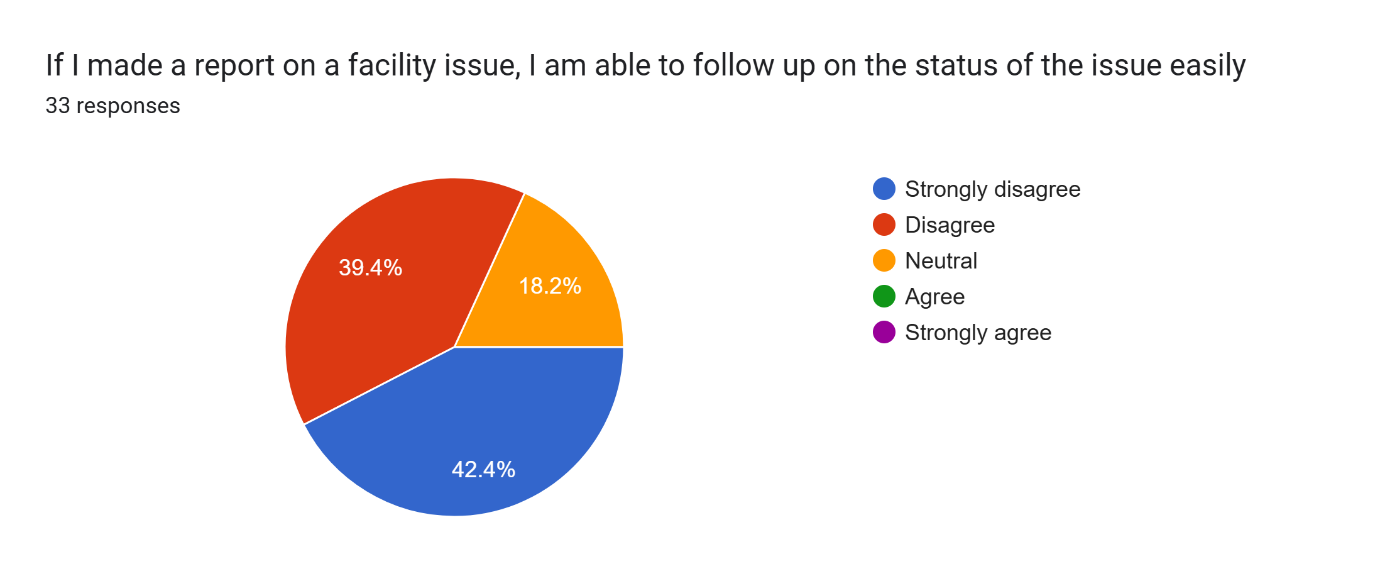


Figure 5. Analysis of Respondent Data for Question 5

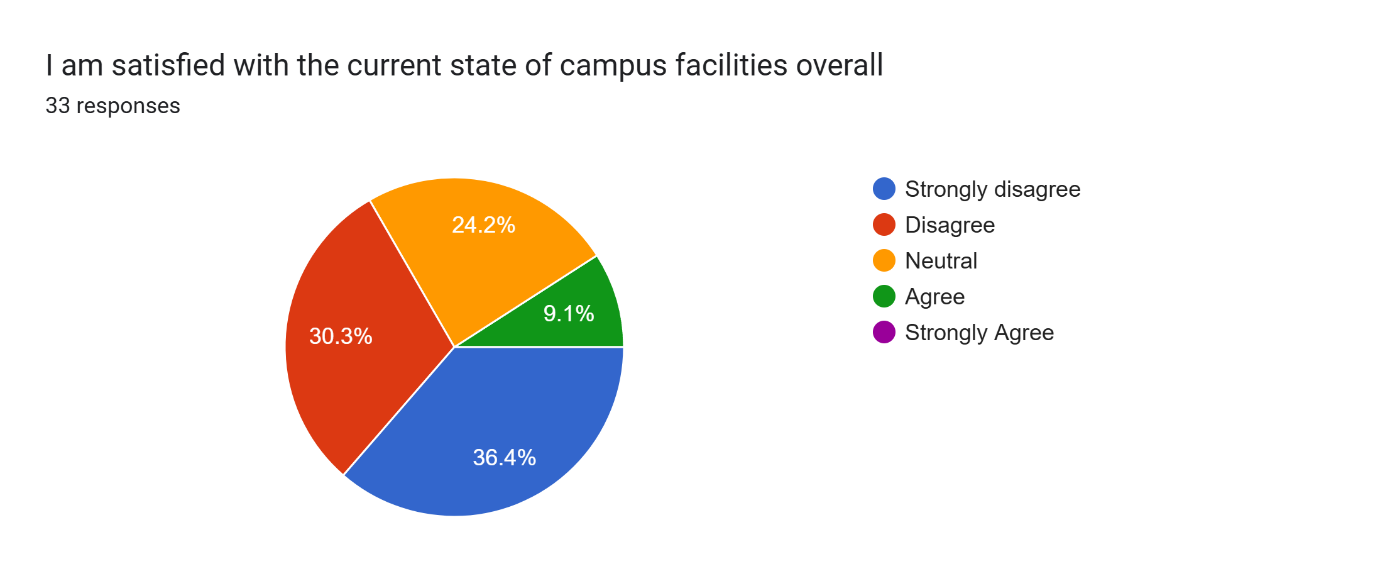
The goal of this question is to find out whether the campus administration informs students if any facility is down, for example through email announcements. 30.3% of the respondents disagree and 39.4% of the respondents strongly disagree that they are able to know whenever a facility has an issue without having to personally check on it or hear from somebody else. This could mean that either the campus administration does not actively communicate with students about facility-related issues, or the current method of communication is less effective. This outcome leads us to one important takeaway: our AI chatbot needs to allow students to easily check on the status of any facility on the campus.

Question 6

  
Figure 6. Analysis of Respondent Data for Question 6

This question will help us understand how systematically the campus handles reported cases regarding facility issues. With most respondents (81.8%) disagreeing that they can easily follow up on issues, it can be seen that there is very likely no system to track all the cases submitted by students. At the same time, it is also possible that the facility administration is also struggling to manage and keep track of all the issues due to the lack of proper tools for this task. What we can conclude from this question is that it is crucial to establish a list of cases that is transparent to both staff and students, so that staff can always have an overview of the progress of all the cases, while students can see their submitted cases in the list, knowing that the issue will be taken care of.

Question 7

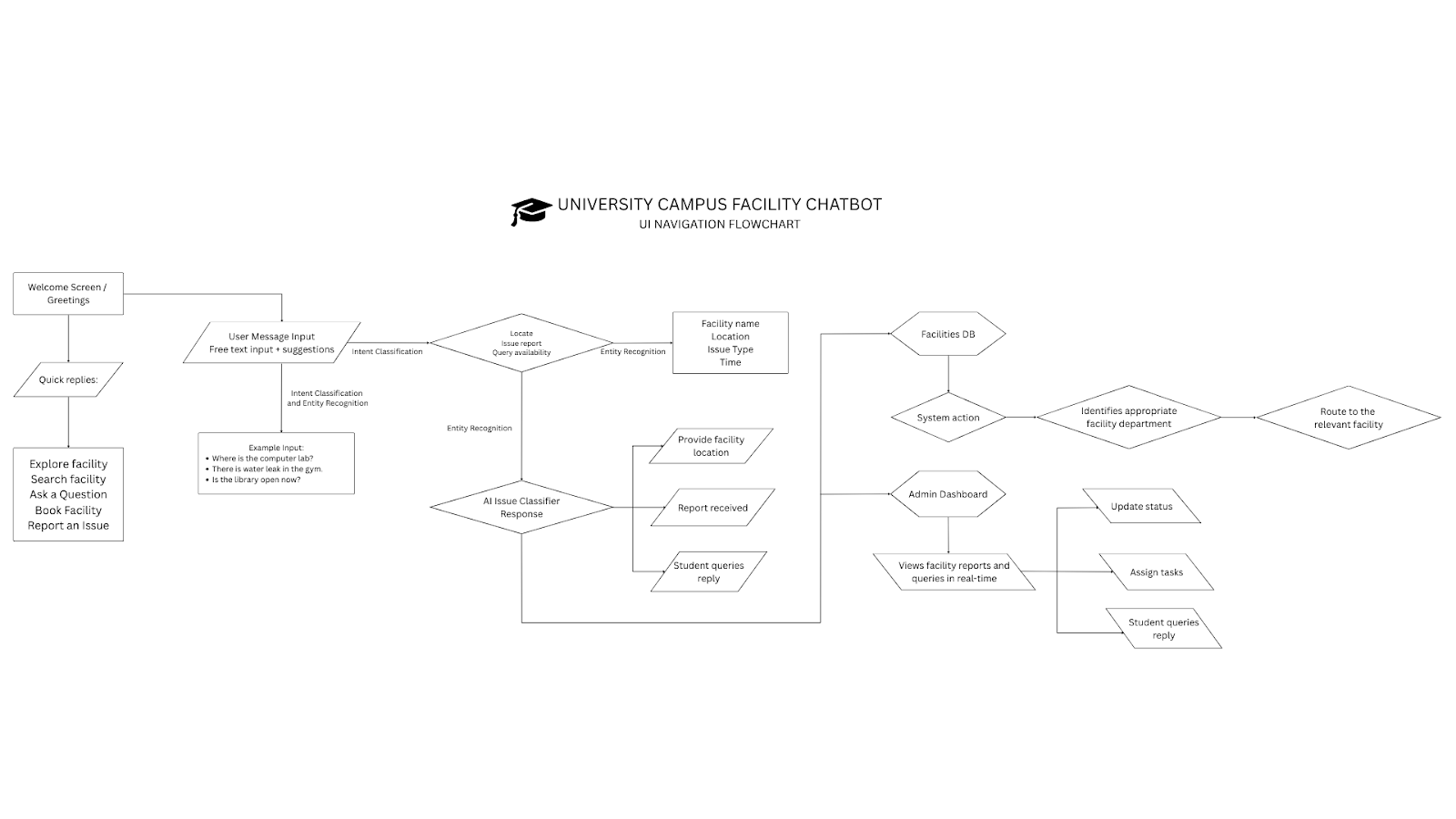
  
Figure 7. Analysis of Respondent Data for Question 7

The last question checks the satisfaction level of students towards the state of campus facilities and how they are being managed. Although only 9.1% of the total respondents felt satisfied overall, we can see that the majority of the respondents do not, hence there is still a lot of room for improvement. Therefore, bringing in an AI chatbot to alleviate the issues we understood from this survey will be significant.

Stakeholder Interview

An interview was conducted with one of the stakeholders, who is a final year student at UTM Johor Bahru. His opinion towards the 7 questions above falls into the majority category. But in addition to that, he also highlighted a few important functionalities that will make a good AI chatbot:

1. It is best if the reporting function not only accepts issues but also accepts students’ suggestions for improvement.
2. The chatbot should also allow students to book any kind of facility directly online.
3. The chatbot should be able to give students general information about the facilities, such as the location. This is especially useful for first-year students as they haven't yet learned their way around the campus.
4. User Interface Storyboard

Figure 8. UI Navigation Flowchart on University Campus Facility Chatbot  
Link: <https://www.canva.com/design/DAGnz48o-wM/_YNrVf7QaQv-WUGqITxbKw/edit>

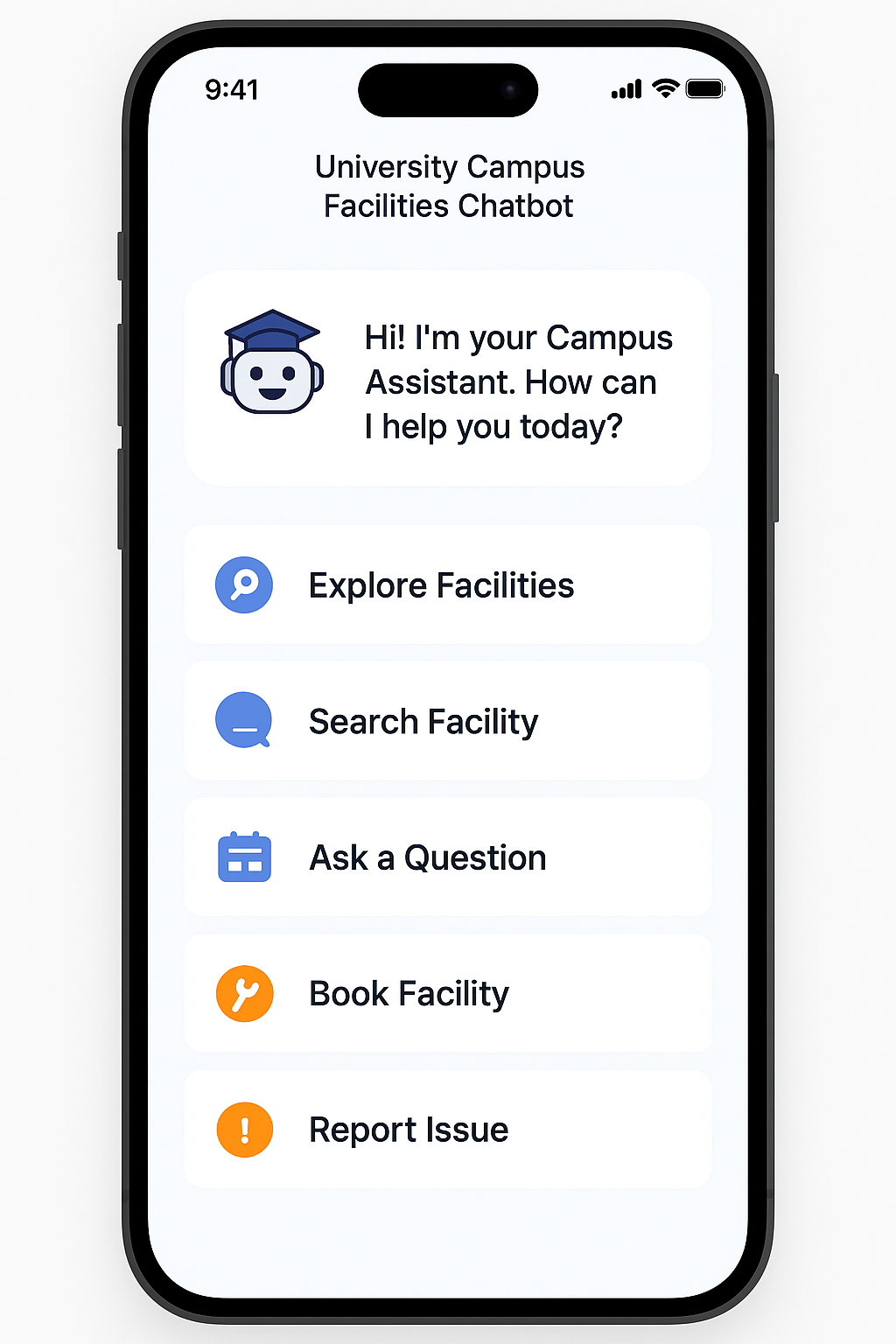


Figure 9. Example of University Campus Facility Chatbot UI

University Campus Facility Chatbot – UI Navigation Flow

1. Welcome Screen

Text Displayed:

* "Welcome to the University Facilities Chatbot"
* "How can I help you today?"

Buttons/Quick Replies:

* Explore Facilities
* Search Facilities
* Ask a Question
* Book Facility
* Report an Issue

1. **User Message Input**

UI Element:

* Free text input + suggestions

Example Input:

* "Where is the computer lab?"
* "There is a water leak in the gym."
* "Is the library open now?"

1. Intent Classification & Entity Recognition (Backend Process)

**Invisible to user** but the chatbot:

* Understands the intent: locate, report issue, query availability
* Extracts entities: facility name (e.g., "gym"), location, issue type, time

1. AI Issue Classifier Response

System identifies appropriate facility department and next action.

Bot Example Response:

* "You're looking for the Computer Lab. Here's the nearest one:"
* "I’ve logged your issue with the gym. Can you provide a photo or describe it more?"
* "The Library is open today from 8 AM to 8 PM."

1. Routing to Facilities DB

System action:

* The issue or query is routed to the right facility department through an internal DB/API.

1. Admin Dashboard

Backend Access:

* Admin views facility reports and queries in real-time
* Option to update status, assign tasks, reply to student queries

4.0 Knowledge Base Design

The knowledge base for the UTM Campus Assistant Chatbot is designed with three focus, intent classification, entity recognition, and issue classification.

4.1 Entity Recognition Knowledge Representation in First Order Logic

|  |  |  |
| --- | --- | --- |
| **No** | **Natural Language** | **First-Order Logic (FOL)** |
| 1 | If the query mentions “library”, the chatbot identifies “library” as a facility | ∀q, (ContainsKeyword(q, “library”) => ContainsEntity(q, Facility)) |
| 2 | If the query mentions “lab”, the chatbot identifies “Lab” as a bookable facility | ∀q, (ContainsKeyword(q, “lab”) => ContainsEntity(q, BookableFacility)) |
| 3 | If the query mentions “air conditioning unit”, the chatbot identifies the air conditioning unit as a component | ∀q, (ContainsKeyword(q, “air conditioning unit”) => ContainsEntity(q, Component)) |
| 4 | If the query mentions “male hostel”, the chatbot identifies the query to refer to the facility category | ∀q, (ContainsKeyword(q, “male”) ˄ ContainsKeyword(q, “hostel”) => ContainsEntity(q, FacilityCategory)) |
| 5 | If the query mentions “at lab 3”, the chatbot identifies the query to refer to a specific location | ∀q, (ContainsKeyword(q, “at lab 3”) => ContainsEntity(q, Location)) |

4.2 Intent Classification Knowledge Representation in First Order Logic

|  |  |  |
| --- | --- | --- |
| **No** | **Natural Language** | **First-Order Logic (FOL)** |
| 1 | If a student asks “where is robotics lab”, or “find robotics lab”, their intent is to search | ∀u, (ContainsKeyword(u, “where is”) ˅ ContainsKeyword(u, “find”) => Intent (u, Search)) |
| 2 | If student asks for “general info about hostel”, their intent is to explore by category | ∀u, (ContainsKeyword(u, “general info”) => Intent(u, ExploreByCategory)) |
| 3 | If student asks “how to book lab slot”, or “how to reserve lab slot”, their intent is to book a facility | ∀u, (ContainsKeyword(u, “how to book”) ˅ ContainsKeyword(u, “reserve”) => Intent(u, Reserve) |
| 4 | If student asks “how to report an issue”, their intent is to report issue | ∀u, (ContainsKeyword(u, “report an issue”) => Intent(u, ReportIssue)) |
| 5 | If a student asks “how to check status of issue” or “track status of issue”, their intent is to check the status of reported issue | ∀u, (ContainsKeyword(u, “check status”) ˅ ContainsKeyword(u, “track status”) => Intent(u, CheckStatus)) |
| 6 | If a staff member wants to “set availability” or “block dates” for a facility, their intent is to manage facility schedules | ∀s, (ContainsKeyword(s, “set availability”) ˅ ContainsKeyword(s, “block dates”)=> Intent(s, ManageSchedules)) |
| 7 | If a staff member needs to “approve pending bookings” or “decline a request”, their intent is to view bookings/requests | ∀s, (ContainsKeyword(s, “approve bookings”) ˅ ContainsKeyword(s, “decline request”) => Intent(s, ViewBookingsRequests)) |
| 8 | If a staff member wants to “view reported issues” or “update status of report”, their intent is to respond to reports | ∀s, (ContainsKeyword(s, “view reported issues”) ˅ ContainsKeyword(s, “update status report”) => Intent(s, RespondToReports)) |
| 9 | If a staff member asks for “usage frequency” or “booking trends”, their intent is to view facility insights | ∀s, (ContainsKeyword(s, “usage frequency”) ˅ ContainsKeyword(s, “booking trends”) => Intent(s, FacilityInsights)) |
| 10 | If a staff member wants to “chat with students” or “answer live queries”, their intent is to chat with students | ∀s, (ContainsKeyword(s, “chat with students”) ˅ ContainsKeyword(s, “answer live queries”) => Intent(s, ChatWithStudents)) |

4.3 Issue Classification Knowledge Representation in First Order Logic

|  |  |  |
| --- | --- | --- |
| No | Natural Language | First-order Logic (FOL) |
| 1 | If the issue mentions a facility and “no power”, it is an electrical issue | ∀i, (Intent(i, ReportIssue) ˄ ContainsEntity(i, Facility) ˄ ContainsKeyword(i, “no power”) => IssueType(i, ElectricalIssue)) |
| 2 | If the issue mentions a facility and “unclean”, it is a hygiene issue | ∀i, (Intent(i, ReportIssue) ˄ ContainsEntity(i, Facility) ˄ ContainsKeyword(i, “unclean”) => IssueType(i, HygieneIssue)) |
| 3 | If the issue mentions a location and “water leak”, it is a plumbing issue | ∀i, (Intent(i, ReportIssue) ˄ ContainsEntity(i, Location) ˄ ContainsKeyword(i, “water leak”) => IssueType(i, PlumbingIssue)) |
| 4 | If the issue mentions a location and “damaged floor”, it is a structural issue | ∀i, (Intent(i, ReportIssue) ˄ ContainsEntity(i, Location) ˄ ContainsKeyword(i, “damaged floor”) => IssueType(i, StructuralIssue)) |
| 5 | If the issue mentions a component and “not working”, it is an equipment issue | ∀i, (Intent(i, ReportIssue) ˄ ContainsEntity(i, Component) ˄ ContainsKeyword(i, “not working”) => IssueType(i, EquipmentIssue)) |

4.4 Resolution Refutation

This section outlines a typical scenario at UTM: a student discovers a malfunctioning air conditioning unit. Unsure of how to report the issue due to a lack of clear, accessible mechanisms, the student turns to the UTM Campus Assistant Chatbot for an intuitive reporting solution.

Below is an example resolution refutation proving tree for this scenario based on the First-Order Logic (FOL) rules provided in 4.1, 4.2, and 4.3. Due to the architecture of the chatbot, the student’s query will travel through three stages, Entity Recognition, Intent Classification, and Issue Classification.

4.4.1 Scenario

**Scenario:** A student opens the UTM Campus Assistant Chatbot mobile application, and types the following text to the chatbot: “I would like to report an issue about an air conditioning unit that is not working”.

**Student Query (chatbot’s input):** “I would like to report an issue about an air conditioning unit that is not working”

**Clauses:**

1. ContainsKeyword(student\_query, “report an issue”)
2. ContainsKeyword(student\_query, “air conditioning unit”)
3. ContainsKeyword(student\_query, “not working”)

4.4.2 Entity Recognition Stage

**Goal:** To prove that the entity “air conditioning unit” is recognized as a “Component”

ContainsEntity(q, Component)

***Unifying q with student\_query,***

ContainsEntity(student\_query, Component)

**Negated Goal:**

1. ¬ContainsEntity(student\_query, Component)

**Proof:**

***Add clause from Entity Recognition Knowledge Base,***

∀q, (ContainsKeyword(q, “air conditioning unit”) => ContainsEntity(q, Component))

***Dropping universal quantifier and unifying q with student\_query,***

ContainsKeyword(student\_query, “air conditioning unit”) => ContainsEntity(student\_query, Component)

***Converting to Conjunctive Normal form,***

1. ¬ContainsKeyword(student\_query, “air conditioning unit”) ˅ ContainsEntity(student\_query, Component)

***Resolving 5 with 2,***

1. ContainsEntity(student\_query, Component)

***Resolving 6 with 4,***

NIL

**Conclusion:**

The derivation of NIL proves that the chatbot correctly infers that the student query contains an entity called “Component”. This proven fact will be used in the next stage.

4.4.3 Intent Classification Stage

**Goal:** To prove that the student’s intent was to report an issue

Intent(u, ReportIssue)

***Unifying u with student\_query,***

Intent(student\_query, ReportIssue)

**Negated Goal:**

1. ¬ Intent(student\_query, ReportIssue)

**Proof:**

***Add clause from Intent Classification Knowledge Base,***

∀u, (ContainsKeyword(u, “report an issue”) => Intent(u, ReportIssue))

***Dropping universal quantifier and unifying u with student\_query,***

ContainsKeyword(student\_query, “report an issue”) => Intent(student\_query, ReportIssue)

***Converting to Conjunctive Normal form,***

1. ¬ ContainsKeyword(student\_query, “report an issue”) ˅ Intent(student\_query, ReportIssue)

***Resolving 8 with 1,***

1. Intent(student\_query, ReportIssue)

***Resolving 9 with 7,***

NIL

**Conclusion:**

The derivation of NIL proves that the chatbot correctly infers that the student’s intent was to report an issue. This proven fact will be used in the next stage.

4.4.4 Issue Classification Stage

**Goal:** To prove that the reported issue is an “Equipment Issue”

IssueType(i, EquipmentIssue)

***Unifying i with student\_query,***

IssueType(student\_query, EquipmentIssue**)**

**Negated Goal:**

1. ¬ IssueType(student\_query, EquipmentIssue)

**Proof:**

***Add clause from Issue Classification Knowledge Base,***

∀i, (Intent(i, ReportIssue) ˄ ContainsEntity(i, Component) ˄ ContainsKeyword(i, “not working”) => IssueType(i, EquipmentIssue))

***Dropping universal quantifier and unifying i with student\_query,***

Intent(student\_query, ReportIssue) ˄ ContainsEntity(student\_query, Component) ˄ ContainsKeyword(student\_query, “not working”) => IssueType(student\_query, EquipmentIssue)

***Converting to Conjunctive Normal form,***

1. ¬ Intent(student\_query, ReportIssue) ˅ ¬ ContainsEntity(student\_query, Component) ˅ ¬ContainsKeyword(student\_query, “not working”) ˅ IssueType(student\_query, EquipmentIssue)

***Add clause from Entity Recognition Stage,***

1. ContainsEntity(student\_query, Component)

***Add clause from Intent Classification Stage,***

1. Intent(student\_query, ReportIssue)

***Resolving 11 with 3,***

1. ¬ Intent(student\_query, ReportIssue) ˅ ¬ ContainsEntity(student\_query, Component) ˅ IssueType(student\_query, EquipmentIssue)

***Resolving 14 with 12,***

1. ¬ Intent(student\_query, ReportIssue) ˅ IssueType(student\_query, EquipmentIssue)

**Resolving 15 with 13,**

1. IssueType(student\_query, EquipmentIssue)

**Resolving 16 with 10,**

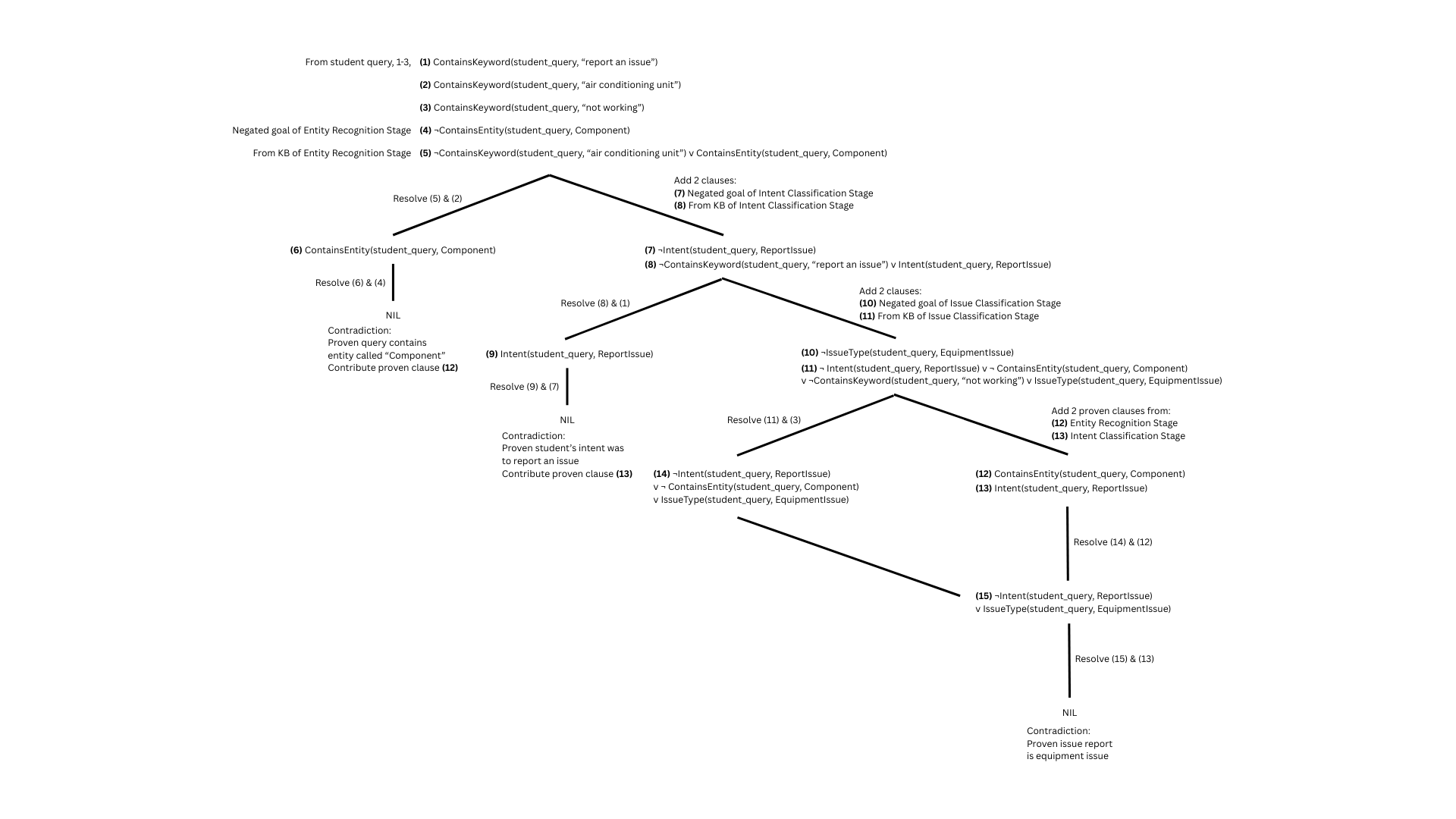
NIL

**Conclusion:**

The derivation of NIL proves that the chatbot correctly infers that the issue reported is an equipment issue.

4.5 Proving Tree

link:   
<https://www.canva.com/design/DAGn5SSXHdI/b8VcX6WJyd1-TJPaWt1iyQ/edit?utm_content=DAGn5SSXHdI&utm_campaign=designshare&utm_medium=link2&utm_source=sharebutton>

**Figure 10. Proving Tree**