

# Title

Authors

*Department of Electrical Engineering and Computer Science*  
*Technische Hochschule Ostwestfalen-Lippe, University of Applied Sciences and Arts*  
Lemgo, Germany

akshay.chikhalkar@stud.th-owl.de, sean.nagel@stud.th-owl.de, kassabji.bassam@stud.th-owl.de

**Abstract**

## CONTENTS

<b>I</b>	<b>Introduction</b>	<b>3</b>
<b>II</b>	<b>Use cases and user stories</b>	<b>3</b>
<b>III</b>	<b>Using machine learning models as specification instruments</b>	<b>5</b>
<b>IV</b>	<b>Experiment and Results</b>	<b>6</b>
<b>V</b>	<b>Discussion</b>	<b>6</b>
<b>VI</b>	<b>Conclusion</b>	<b>6</b>
	<b>Appendix</b>	<b>7</b>
A	Conversation with ChatGPT . . . . .	7

## I. INTRODUCTION

## II. USE CASES AND USER STORIES

To further identify the desired functionality of the University Room Management System, two use cases and three user stories were identified and formulated. One use case and two user stories will be described in more detail in this section.

The identified use cases are *Searching and booking a room* and *Getting navigation instructions to a room*. For the use case *Searching and booking a room* a use case specification was created. The specification is shown in TABLE I. Some aspects of this use case specification will be further elaborated on.

Extension 1a describes searching for a room via room properties. Possible properties for searching and filtering rooms could be: type of the room (lecture room or seminar room), room capacity or the availability of projectors and blackboards.

In extension 5b the maximum allowed number of rooms is noted. This limit can be different depending on the user. Our current specification limits students to be allowed of booking only a single room at a given time. University staff is allowed to book multiple rooms at a time (e.g. a Professor booking multiple rooms for an exam). The specific limits for each group of staff should be selected according to the universities specific needs.

Extension 5c notes the possibility to book a room for future use according to a time schedule. This option shall not be available to students but only to university staff. For example a room can be booked for a specific time frame on a weekly basis. This stands in contrast to the general booking presented in the main success scenario. In that case the room is booked from the time of booking without specifying the length of the booking. The booking has to be terminated manually in the end. Alternatively the booking is also terminated, if the system registers that the room has not been used for a while or a previously scheduled booking is about to start. Please see the BPMN diagram above for this process.

The open issues of the use case note the booking of multiple rooms at once. The process for this will likely be very similar to booking a single room. Showing information for multiple rooms at once and managing the unavailability of a single room of the selection need to be further deliberated. Similarly to scheduled booking students should not be allowed to book multiple rooms.

Following three user stories were formulated:

- 1) As a Student, I want to book a room, so that I have a place to work.
- 2) As a Guest, I want to have navigational instructions, so that I am able to locate a room.
- 3) As a Professor, I want to book a room according to a schedule, so that I can give regular lectures in that room.

A good practice is to formulate user stories according to the INVEST principle. This principle describes that a user story should be independent, negotiable, valuable, estimable, small and testable. User stories 1 and 2 were elaborated further and analyzed with regard to the INVEST principle.

The first user story “As a Student, I want to book a room, so that I have a place to work.” is extended by following points:

- A student is only allowed to book at most one room at any given time.
- Students can only book rooms for the current time and are not able to reserve them for later.
- A student is identified by their university id.
- Bookings by students do not have an ending time specified.
- Students are able to terminate their bookings manually.
- Bookings by students are terminated automatically, if a booking by a university employee starts.

The user story fulfills the INVEST principle, as it does not depend on other user stories, leaves room for negotiation and presents a value to the students. A development team should be able to estimate the work required for realization. The story itself is small, as it only focuses on bookings for a single user group. The story can be tested, by students trying to book rooms.

The second user story “As a Guest, I want to have navigational instructions, so that I am able to locate a room.” is extended by following points:

- The user story is also valid for students and university employees.

Name	Searching and booking a room
Scope	University Room Management System
Level	User goal
Primary Actor	Student, Professor or other university staff
Stakeholders / interests	Other members of the university (students, professors, other staff) University / university management
Preconditions	User is in possession of a valid university card and has a user account
Postconditions	Status of the rooms in the room database is updated
Main success scenario	<ol style="list-style-type: none"> <li>1. User searches a room via a room number</li> <li>2. System returns a list of rooms fulfilling the search query</li> <li>3. User selects a room from the list</li> <li>4. System shows detailed information about the selected room</li> <li>5. User books the room</li> <li>6. System updates booking status of the room</li> <li>7. System shows booking information and confirmation</li> </ol>
Extensions	<ol style="list-style-type: none"> <li>1a. User searches room via room properties <ol style="list-style-type: none"> <li>1. User opens page to filter specific properties</li> <li>2. User selects room requirements and confirms</li> </ol> </li> <li>2a. No room fulfilling the search criteria exists <ol style="list-style-type: none"> <li>1. System shows message: "No rooms found"</li> <li>2. System presents rooms similar to the search query</li> </ol> </li> <li>5a. Room is already booked <ol style="list-style-type: none"> <li>1. System shows message: "Room is already booked"</li> <li>2. System suggests a list of similar rooms</li> </ol> </li> <li>5b. User already booked the allowed maximum amount of rooms <ol style="list-style-type: none"> <li>1. System shows message: "You cannot book another room"</li> <li>2. System shows list of rooms already booked by the user</li> </ol> </li> <li>5c. User books room according to a schedule (for future use) <ol style="list-style-type: none"> <li>1. System presents a day and time picker</li> <li>2. User selects day of week and time</li> <li>3. System asks for the frequency of the booking</li> <li>4. User selects how often booking should be repeated</li> <li>5. System books room for all selected time slots</li> <li>5a. Room is not available in all time slots <ol style="list-style-type: none"> <li>1. System shows message: "Room is not available at &lt;DAY / TIME&gt;"</li> <li>2. System provides alternative time slots and rooms</li> <li>3. User selects alternatives</li> </ol> </li> </ol> </li> </ol>
Special requirements	Stable database Simple UI design Fast search
Technology / data variation	Available via a terminal at the university or via a web app
Frequency of occurrence	Very regularly, up to multiple requests per minute at most active times Long pauses without interaction also likely
Open issues	Booking of multiple rooms at once

TABLE I: Use case specification for the use case *Searching and booking a room*

- Navigation is realized through building schematics, as well as indoor and outdoor maps.
- Other points of interest (e.g. parking spots, water fountains) are out of scope and might be part of standalone user stories.
- Users do not need to be verified for navigation.

This user story is independent of other user stories and still leaves room for negotiation. The value of better navigation at the university campus is provided. The use case is small and the required amount of work can be estimated. Tests might include the navigation between rooms or buildings of the university by external people. Thus, this use case also fulfills the INVEST principle.

### III. USING MACHINE LEARNING MODELS AS SPECIFICATION INSTRUMENTS

One use case and one user story from the previous section were intentionally not further specified. Instead, they were used to analyze the ability of language models like ChatGPT<sup>1</sup> to generate parts of a system specification. In particular, we used the aforementioned ChatGPT by OpenAI to generate a use case specification and elaborate on a user story according to the INVEST principle.

The conversation with ChatGPT can be found in appendix A.

The initial prompt included a short description of the system and asked to write a use case specification for the specific use case. ChatGPT returned a specification including a name, ID, primary actor, preconditions, postconditions, trigger, main flow, alternate flow, exceptional flow and notes. To simplify comparisons ChatGPT was asked to provide the specification with the same aspects as those of the specification in TABLE I.

On first sight, the generated response seems to be good. Some points even include additional explanations. But there are other interesting things to note:

The use case's level was identified as being a user goal and a sub-function. The level should be either one or the other, but not both. By investigating the specification it can be seen, that it should probably be a user goal instead of a sub-function.

The three stakeholders students, professors and guests are identical to the primary actors previously noted in the specification. The only stakeholder, which is not a primary actor is the university management. In this case the primary actors should not be listed in the stakeholders section.

The generated preconditions are relatively simple and are not necessarily needed for the use case specification. Additionally, the third precondition, that the user needs to be logged in, directly contradicts the main success scenario, in which the user is logging in during step 2. Either the precondition or the following scenario is wrong.

The main success scenario starts with the user opening the system application. This is different to our approach, where the scenario started with the application being ready for user interaction. The approach chosen by ChatGPT is not wrong, but just different. Step 7 and 8 describe how the scenario continues depending on whether a room was found or not. The main success scenario should only contain the successful case of the room being found. The other case should be included as an alternative flow. Interestingly enough, the single alternative flow of the specification handles this case. Thus, step 8 of the main flow is unnecessary.

For the point frequency of occurrence the generated specification is defined as "frequently throughout the academic year". This is a very non-descriptive specification, as it is unclear if the use cases will occur multiple times per hour, per day or per week.

After generating the discussed use case specification ChatGPT was prompted to write a user story using the INVEST principle for user story 3: "As a Professor, I want to book a room according to a schedule, so that I can give regular lectures in that room."

The answer starts by listing the six facets of INVEST. It is declared that the given user story fulfills all aspects of the INVEST principle. Additionally, acceptance criteria for the use case are defined by the language model. The acceptance criteria seem sensible to the use case. Although the first five acceptance criteria are structured very similar to a use case's main success scenario. The answer also includes some additional considerations, which also seem sensible in the context of the system.

While reading ChatGPT's answer the question arose, how the model would handle a user story, which does not fulfill all aspects of INVEST. Thus, the model was prompted to write another user story according to the INVEST principle. The user

<sup>1</sup><https://openai.com/chatgpt>

story requested in this prompt is: “As a University member, I want to use the Room Management System, because it will make life easier.” This user story is very broad and unspecific, such that it is neither estimable nor small. The user story also is not independent, as it focuses on the entire room management system at once.

The answer of ChatGPT is disappointing, as it is argued that the given user story fulfills all aspects of INVEST. Especially statements like “The user story has a focused scope”, or “It focuses on the specific need [...] to utilize the Room Management System” show that the problems of the user story are not detected by ChatGPT. The task to write the use case done according to the INVEST principle is done, but there is no plausibility check performed.

Although the answers by ChatGPT are not completely correct, the amount of sensible output by the language model are quite impressive, especially if you consider the small amount of information given in the prompts. The answers of ChatGPT need to be checked for plausibility and cannot be used as is. But the generated use case specifications and user story elaborations can be used as a good starting point, which requires some manual fine-tuning by human designers. Using language models for initial specification drafts can probably decrease the required time for a system specification.

It should also be noted that the used ChatGPT is a general purpose language model. More specified models might yield even better results. Currently, there are developments on models with the specific task to generate such specification and even write some initial code. One example that should be noted in this context is “MetaGPT”<sup>2</sup>, which according to its development page is able to generate user stories, requirements, APIs and more.

#### IV. EXPERIMENT AND RESULTS

#### V. DISCUSSION

#### VI. CONCLUSION

#### ACKNOWLEDGMENT

<sup>2</sup><https://github.com/geekan/MetaGPT>

## APPENDIX

### A. Conversation with ChatGPT

The conversation with ChatGPT described in section III:

Prompt:

A University Manager system, which is a system where students can book lecture rooms at the university for study and professors can book seminar rooms to give lectures. In addition the University Manager system, give the possibility for students, professor and guests to have navigation instruction for a seminar room they search for.

Write me a use-case specification using the proper and formal structure for that.

A use case specification for the use case called (getting navigation instructions to a seminar room in the university) that is implemented in the University Manager system.

Answer:

#### **Use Case Specification: Getting Navigation Instructions to a Seminar Room in the University**

**Use Case Name:** Get Navigation Instructions to a Seminar Room

**ID:** UC-NAV-01

**Description:** This use case allows students, professors, and guests using the University Manager system to obtain navigation instructions to reach a specific seminar room within the university campus. Users can search for a seminar room by its room number or name, and the system will provide step-by-step directions to help them reach the desired location.

**Primary Actor:** User (Student, Professor, or Guest)

#### **Preconditions:**

- The University Manager system is up and running.
- The user is logged into the system (for registered students and professors).

#### **Postconditions:**

- The user receives navigation instructions to the requested seminar room.

**Trigger:** User initiates the request to get navigation instructions to a seminar room.

#### **Main Flow:**

- 1) The user opens the University Manager system's application or website.
- 2) If required, the user logs into the system using their credentials. (Skipped for guests)
- 3) The user selects the option to "Get Navigation Instructions" from the main menu.
- 4) The system presents the user with a search bar to enter the seminar room number or name they wish to navigate to.
- 5) The user enters the seminar room number or name in the search bar.
- 6) The system validates the entered information and searches for the requested seminar room in the database.
- 7) If the seminar room is found:
  - a. The system calculates the best route to reach the seminar room from the user's current location or a specified starting point within the campus.
  - b. The system generates step-by-step navigation instructions, including directional cues (e.g., left, right, straight), landmarks, and approximate distances.
  - c. The navigation instructions are displayed on the user's device screen.
- 8) If the seminar room is not found:
  - a. The system displays an error message indicating that the seminar room could not be located.
  - b. The use case terminates.

#### **Alternate Flow:**

- 6a. If the entered seminar room number or name is not valid or does not exist:
  - i. The system displays an error message indicating that the seminar room could not be found.

- ii. The user is given the option to re-enter the room number or name.
- iii. The user may choose to cancel the navigation request.
- iv. The use case continues from step 4.

**Exceptional Flow:**

- 1) If the University Manager system is not operational due to maintenance or technical issues:
  - a. The user is informed about the unavailability of the system.
  - b. The user may be asked to try again later or contact support for assistance.
  - c. The use case terminates.

**Notes:**

- The University Manager system may integrate with campus maps or GPS services to provide accurate navigation instructions.
- The system could offer additional features, such as displaying accessibility information or estimated time of arrival to the seminar room.
- For security reasons, certain seminar rooms (e.g., faculty offices) may have restricted access for navigation instructions. In such cases, the system should inform the user accordingly.

Prompt:

Use Case Specification: Getting Navigation Instructions to a Seminar Room in the University in terms of :  
Name  
Scope  
Level  
Primary actor  
Stakeholders  
preconditions  
postconditions  
Main success scenario  
Extensions  
Special requirements:  
Technology / data variation:  
Frequency of occurrence:  
Open issues:

Answer:

**Name:** Getting Navigation Instructions to a Seminar Room in the University

**Scope:** University Manager System

**Level:** User Goal (Primary), Subfunction (Secondary)

**Primary Actor:** User (Student, Professor, or Guest)

**Stakeholders:**

- Students: Require navigation instructions to find seminar rooms for academic purposes.
- Professors: Need navigation guidance to reach seminar rooms for conducting lectures or meetings.
- Guests: Visitors to the university who may require navigation assistance to attend seminars or events.
- University Management: Ensures the system enhances campus navigation and user experience.

**Preconditions:**

- The University Manager system is operational and accessible to the primary actor.
- The user has a compatible device with access to the University Manager system.
- For registered students and professors, the user is logged into the system.

**Postconditions:**



- The user receives accurate navigation instructions to the requested seminar room within the university campus.

#### **Main Success Scenario:**

- 1) The user opens the University Manager system's application or website.
- 2) If required, the user logs into the system using their credentials. (Skipped for guests)
- 3) The user selects the option to "Get Navigation Instructions" from the main menu.
- 4) The system presents the user with a search bar to enter the seminar room number or name they wish to navigate to.
- 5) The user enters the seminar room number or name in the search bar.
- 6) The system validates the entered information and searches for the requested seminar room in the database.
- 7) If the seminar room is found:
  - a. The system calculates the best route to reach the seminar room from the user's current location or a specified starting point within the campus.
  - b. The system generates step-by-step navigation instructions, including directional cues (e.g., left, right, straight), landmarks, and approximate distances.
  - c. The navigation instructions are displayed on the user's device screen.
- 8) If the seminar room is not found:
  - a. The system displays an error message indicating that the seminar room could not be located.
  - b. The user is given the option to re-enter the room number or name.
  - c. The user may choose to cancel the navigation request.

#### **Extensions:**

- 6a. If the entered seminar room number or name is not valid or does not exist:
- i. The system displays an error message indicating that the seminar room could not be found.
  - ii. The user is prompted to re-enter the room number or name.
  - iii. The user may choose to cancel the navigation request.
  - iv. The use case continues from step 4.

#### **Special Requirements:**

- The University Manager system should integrate with campus maps or GPS services for accurate navigation.
- The system may provide additional features such as displaying accessibility information or estimated time of arrival to the seminar room.
- Certain seminar rooms may have restricted access, and the system should handle these cases appropriately.

#### **Technology / Data Variation:**

- The system should be accessible via various platforms (web, mobile apps) to accommodate different user devices.
- The availability of campus map data and GPS accuracy may vary depending on the university's infrastructure.

#### **Frequency of Occurrence:**

- The use case is expected to occur frequently throughout the academic year as students attend lectures, professors conduct classes, and events are held.

#### **Open Issues:**

- How will the system handle navigation to seminar rooms located in different buildings or floors within the campus?
- Should the system provide additional options such as displaying alternative routes or real-time updates for room availability?
- How will the system ensure privacy and security of users' location data while providing navigation instructions?

Prompt:

write me a user story using the INVEST principal for the following user story:

user story: As a Professor, I want to book a room according to a schedule so that I can give regular lectures in that room.

Answer:

**User Story:** As a Professor, I want to book a room according to a schedule so that I can give regular lectures in that room.

**INVEST Principle:**

- **Independent:** The user story is self-contained and can be implemented without dependencies on other user stories. It focuses on the specific need of the Professor to book a room for regular lectures.
- **Negotiable:** The details of the implementation are open to negotiation between the Professor (user) and the development team. The team can collaborate with the Professor to determine the specific room booking requirements and scheduling preferences.
- **Valuable:** The user story provides clear value to the Professor by enabling them to secure a designated room for their regular lectures, ensuring consistent teaching arrangements and an organized learning environment.
- **Estimable:** The user story is well-defined and has a clear scope, making it estimable for the development team. The effort required to implement the room booking functionality can be reasonably assessed.
- **Small:** The user story has a focused and concise scope, which allows the development team to implement it within a single development iteration.
- **Testable:** The user story's acceptance criteria are specific and measurable, allowing for effective testing. It is possible to verify the successful implementation of the room booking feature based on the defined criteria.

**Acceptance Criteria:**

- The Professor can access the University Manager system and navigate to the "Room Booking" section.
- The system presents a form with options to select the desired room, date, start time, and end time for the lecture.
- The Professor can enter the required details and submit the booking request.
- The system validates the availability of the selected room for the specified date and time.
- If the room is available, the system confirms the booking and sends a confirmation email to the Professor.
- The booked room should be reserved in the system's schedule for the specified date and time.
- The Professor can view, modify, or cancel their booked lectures within the system as needed.
- In case of any conflicts or unavailability, the system provides alternative room options or available time slots for the Professor's lectures.
- The system ensures that only authorized Professors can access the room booking functionality.

**Additional Considerations:**

- The system should have user authentication to ensure that only authorized Professors can access the room booking feature.
- The system may send reminders or notifications to the Professor before the scheduled lecture time.
- The room booking feature should be user-friendly and accessible via multiple platforms, such as web and mobile applications.
- The system should handle and display room availability in real-time to avoid double bookings.
- The user interface should be intuitive and provide clear feedback on the status of the room booking process.

Note: The user story may be further elaborated during backlog grooming or refinement sessions to capture more detailed requirements and user expectations.

**Prompt:**

write me a user story using the INVEST principal for the following user story:

user story: As a University member, I want to use the Room Management System, because it will make life easier.

**Answer:**

**User Story:** As a University member, I want to use the Room Management System because it will make life easier.

**INVEST Principle:**

- **Independent:** The user story is self-contained and can be implemented independently without relying on other user stories. It focuses on the specific need of the University member to utilize the Room Management System for simplifying room-related tasks.
- **Negotiable:** The user story allows for negotiation between the University member (user) and the development team to identify the specific features and functionalities that will enhance their experience with the Room Management System.
- **Valuable:** The user story holds clear value for the University member by offering a convenient and efficient solution for managing room-related activities, streamlining processes, and saving time.

- **Estimable:** The user story is well-defined and has a clear scope, making it estimable for the development team. The effort required to implement the Room Management System features can be reasonably estimated.
- **Small:** The user story has a focused scope, enabling the development team to implement it within a single development iteration.
- **Testable:** The user story's acceptance criteria are specific and measurable, enabling effective testing. The success of the Room Management System can be assessed based on the defined criteria.

#### **Acceptance Criteria:**

- The Room Management System should be accessible to all University members through a user-friendly web application or mobile interface.
- The system should have user authentication to ensure that only authorized University members can access and use the Room Management features.
- University members should be able to log in using their credentials or university ID.
- The Room Management System should provide a clear and intuitive user interface with navigational menus and visual cues.
- University members should be able to view available rooms, their schedules, and occupancy status in real-time.
- The system should offer a simple and efficient process for booking or reserving rooms for various purposes (e.g., lectures, study sessions, meetings).
- Room availability should be displayed in a visual calendar format, with filtering options based on date, time, location, or room amenities.
- The system should send notifications or reminders to users about upcoming room reservations or changes in room status.
- University members should be able to easily modify or cancel their room reservations within the system.
- The Room Management System should maintain accurate and up-to-date records of room bookings and usage.
- The system should have a responsive design to ensure usability across different devices and screen sizes.

#### **Additional Considerations:**

- The Room Management System could incorporate features like room occupancy limits, equipment availability, and room-specific details.
- To enhance user adoption, the system could provide user tutorials or help documentation for first-time users.
- The system should prioritize data security and privacy, ensuring that only authorized personnel can access sensitive room-related information.
- Regular user feedback and iterative improvements can be incorporated to enhance the Room Management System based on user preferences and needs.

Note: The user story may be further elaborated during backlog grooming or refinement sessions to capture more detailed requirements and user expectations