Human-Computer Interaction

Project Report - 2024

Mitsakis Nikos - p3210122
Pantelidis Ippokratis - p3210150
Velissaridis Yorgos - p3210255

Sign Stage Theatre for All

Contents

Introduction:

Project Overview 2

Methodology 2-3

Cycle I: 4-9

Target Group 4

Functions 4

DCG Grammars 5-6

Draw.io Mockups 6-8

Evaluation Method I - Questionnaire 9

Evaluation Method II - Interview 10

Cycle II:

Target Group 11

Functions 11

Figma Mockups 11-14

Evaluation Method I - Cognitive Walkthrough 15

Evaluation Method II - Think-Aloud Protocol 15

Evaluation Method III - Usability Inspection 15-16

Cycle III:

Target Group 17

Functions 17

User Interface 17-20

Evaluation Method I - Questionnaire 20-21

Evaluation Method II - Cognitive Walkthrough 21-22

Evaluation Method III - Think-Aloud Protocol 23

Evaluation Method IV - Usability Inspection 23-24

Extra: LLM chatbot explanation 25-26

Conclusion 27

You can find a demo video here: YT_Demo_Video

Introduction

The advancement of mobile technology has revolutionized the way we interact with services, offering convenience and accessibility at our fingertips. This project focuses on developing a user interface for a text-based chat application, enabling users to seamlessly communicate with the box office of a hypothetical theater. The goal is to provide a platform where theater-goers can effortlessly obtain information about shows, book or cancel tickets, and seek support, all through their preferred messaging apps like Viber or WhatsApp.

Project Overview

Our hypothetical theater features two auditoriums, each showcasing a unique production with two daily performances: a matinee and an evening show. The need for an efficient and user-friendly system to manage ticketing and information inquiries is paramount in ensuring a satisfying customer experience.

Methodology

The development of this user interface will be conducted in three iterative cycles, adhering to the spiral model of software development. This approach allows for continuous refinement through iterative prototyping and evaluation, ensuring that the final product is both functional and user-centric.

1. **Cycle 1: Preliminary Prototyping and User Research**
    - Focus on identifying user needs and creating simple prototypes for initial feedback.
2. **Cycle 2: Advanced Prototyping and Usability Testing**
    - Develop detailed digital prototypes and conduct usability testing to refine the design.
3. **Cycle 3: Final Implementation and Comprehensive Evaluation**
    - Implement the interface on mobile devices or emulators, incorporating extensive user feedback to ensure a seamless experience.

By following the spiral model, we aim to iteratively improve our design, ensuring that each cycle builds on the feedback and findings of the previous one. This structured yet flexible approach will allow us to deliver a robust and user-friendly chat-based interface for theater ticketing and information services.

Ctrl+

Cycle I: Initial Assessment

Target Group

In the first cycle of our interface development using the spiral model, we decided to target two user categories:

1. **Individuals who own smartphones and are already familiar with their use.**
2. **Individuals with hearing impairments (partially or completely deaf).**

The needs of the first category are adequately addressed by adhering to general user interface development principles, which we followed closely. To accommodate the second category, we ensured that the functionality of each available option in the interface is clear from its visual design alone, without relying on any auditory cues.

Functions

The functions of the prototypes in the first cycle are as follows:

1. Information about the theater plays
2. Ticket booking
3. Ticket cancellation
4. Communication with a human employee

We started with these specific functions because we considered them the most important for the application and those that, through the evaluation methods, would provide us with the most information for improving the interface in the next development cycle.

DCG Grammars

In the initial cycle of our interface development, we focused on establishing the fundamental linguistic framework for the chatbot by developing Definite Clause Grammars (DCGs) in Prolog. These grammars, crafted in both English and Greek, form the basis for the chatbot's ability to understand and process user queries in a structured and systematic manner.

In our project, DCGs serve a crucial role in the preliminary design of the chatbot's understanding capabilities. They provide a simple yet robust way to check if user queries conform to the expected patterns of the language, ensuring that the chatbot can recognize and process valid inputs.

We implemented the DCGs in SWI Prolog, creating rules that define the basic structure of sentences that the chatbot should understand. These rules encompass common phrases and commands related to theater ticketing, such as asking for show times, booking tickets, or canceling reservations.

For example:

```
?- phrase(s, [i, want, to, book, a, ticket]).
true .

?- phrase(s, [i, want, to, get, information, about, the, theater]).
true .

?- phrase(s, [i, want, to, book, seats, for, the, seagull, by, chekhov]).
true .

?- phrase(s, [i, want, to, cancel, a, reservation]).
true .

?- phrase(s, [i, want, to, leave, a, complaint]).
true .

?- phrase(s, [i, wish, to, speak, with, an, employee]).
true .

?- phrase(s, [i, wish, to, learn, details, about, hamlet, by, shakespeare]).
false.
```

By defining these grammars, we created a foundational layer for the chatbot to parse and understand user inputs. When a user inputs a sentence, the DCG checks if it matches the defined patterns. If the sentence is understood, the system returns `true`, indicating a valid query, otherwise, it returns `false`.

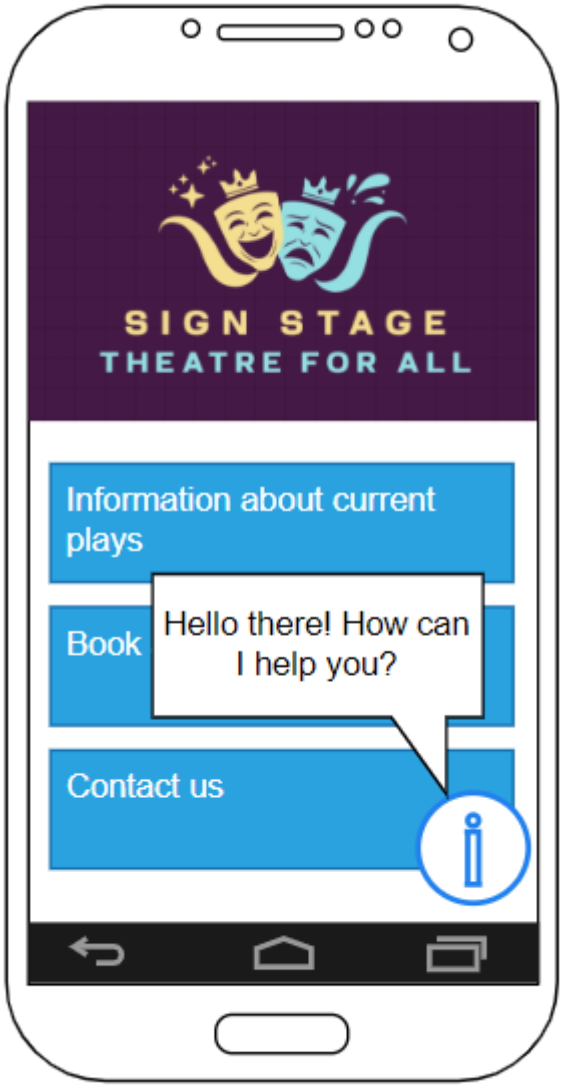
While the DCGs provide a simplified mechanism for initial understanding, the final chatbot will integrate a large language model (LLM) to generate responses. The role of the DCGs in this cycle is to act as a preliminary filter, ensuring that user inputs are within the scope of the chatbot's capabilities.

You can find our DCG Grammars here : dcg_grammars

Draw.io Mockups

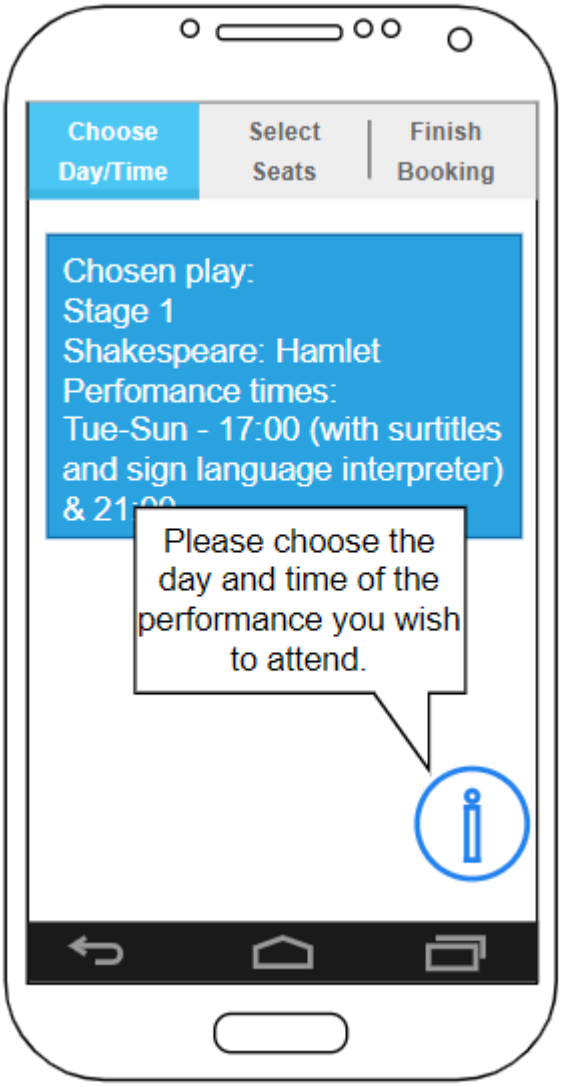
Instead of creating hand-drawn prototypes, we opted to develop very simple prototypes using Draw.io. These prototypes illustrate our initial vision for the user interface, providing a visual representation of the key screens and interactions. This approach allows us to quickly and effectively communicate our design ideas and gather early feedback.

Home Screen

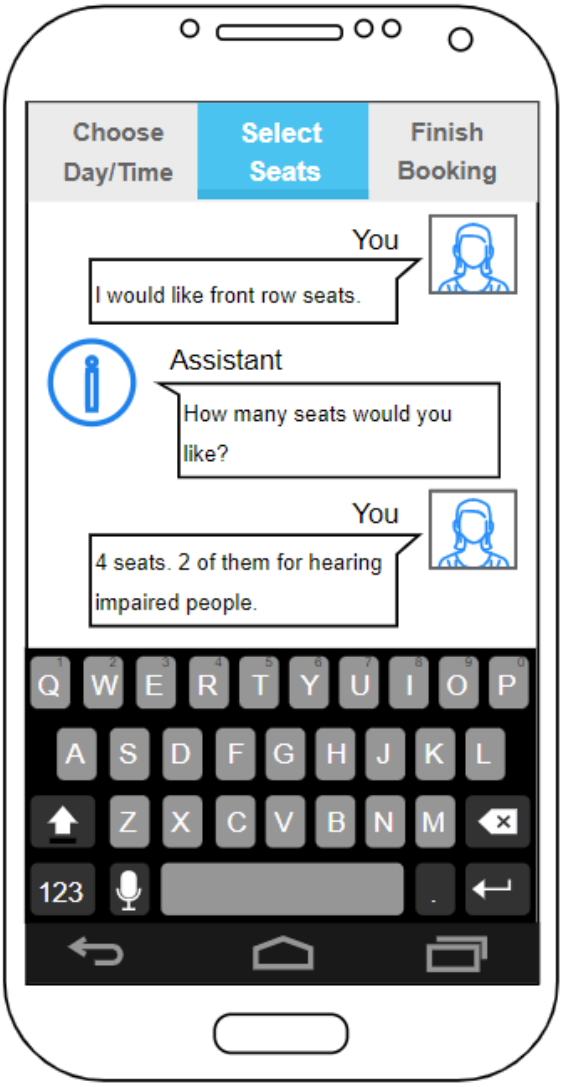


Info Screen

Booking Screen (cycle 1)



Booking Screen (cycle 2)



You can find all our Draw.io prototypes here : drawio_prototypes

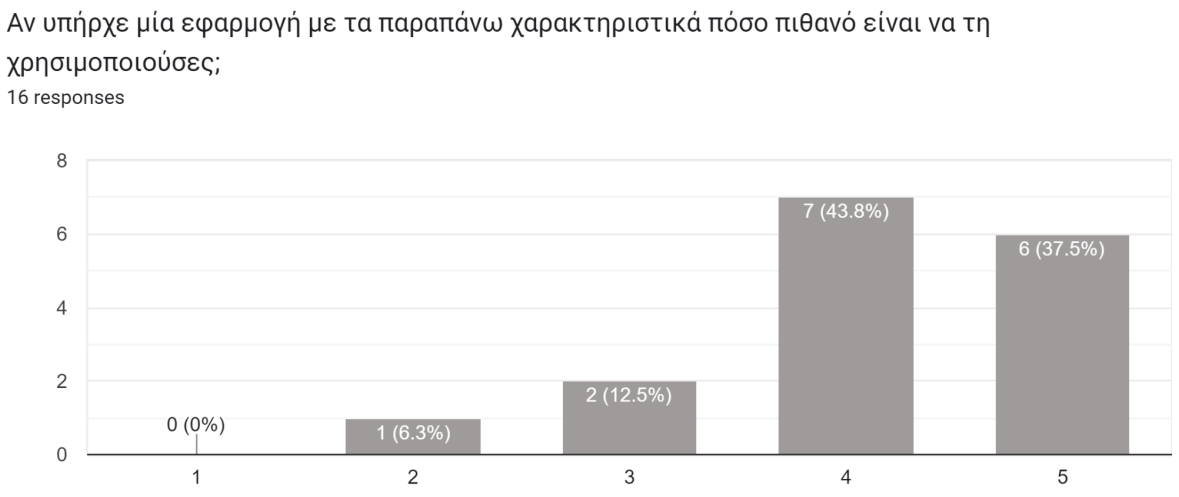
Evaluation Method I

Questionnaire

We created a questionnaire in Greek, with both closed and open type questions to assess the impression the app in general and our mockups in particular had on various people. The questionnaire was answered by 24 people overall, 16 of which attended theater plays and would be willing to use a mobile app that would replace the theater ticket booth.

More than a thirds of these 16 would prefer it if the app language was English. Almost all of them appreciated the features related to hearing-impaired accessibility, and would recommend the app to friends either with or without hearing problems.

Final feedback question: "If there was an app with all the above features, how is it likely that you would use it?"



You can find all individual results here: questionnaire_results(cycle1)

Evaluation Method II

Interviews

We performed interviews with three participants, one 57 year old civil engineer, one 26 year old upcoming theater director and one 56 year old retiree where we tested our mockups in conjunction with our DCG grammars in SWI Prolog in real-time.

Our main findings across participants were:

- The app would be much preferable to a queue at the theater
- It is appreciated that one can use the app manually, without utilizing the chatbot
- Speech recognition and playback would be useful features
- It is better that the chatbot carry out functions, instead of giving instructions to the user so that he/she may carry them out manually (expect for payments)

Cycle II: Prototyping

Target Group

In the second cycle of our interface development using the spiral model, we decided to add a target user category to our previous two:

3. **Individuals with visual impairments (partially or completely blind).**

The needs of this 3rd category are attended to by adding the functionality of speech recognition and speech production, while using the chatbot.

Functions

The functions of the prototypes of the second cycle are the same as those of the first, plus:

5. Login
6. Pay (performed during ticket booking)
7. View e-tickets
8. Settings
9. Contact
10. Make complaints

The addition of these functions enhances usability by providing secure access, streamlined booking, easy ticket management, personalization options, support contact, and efficient feedback mechanisms, making the app more comprehensive and user-friendly.

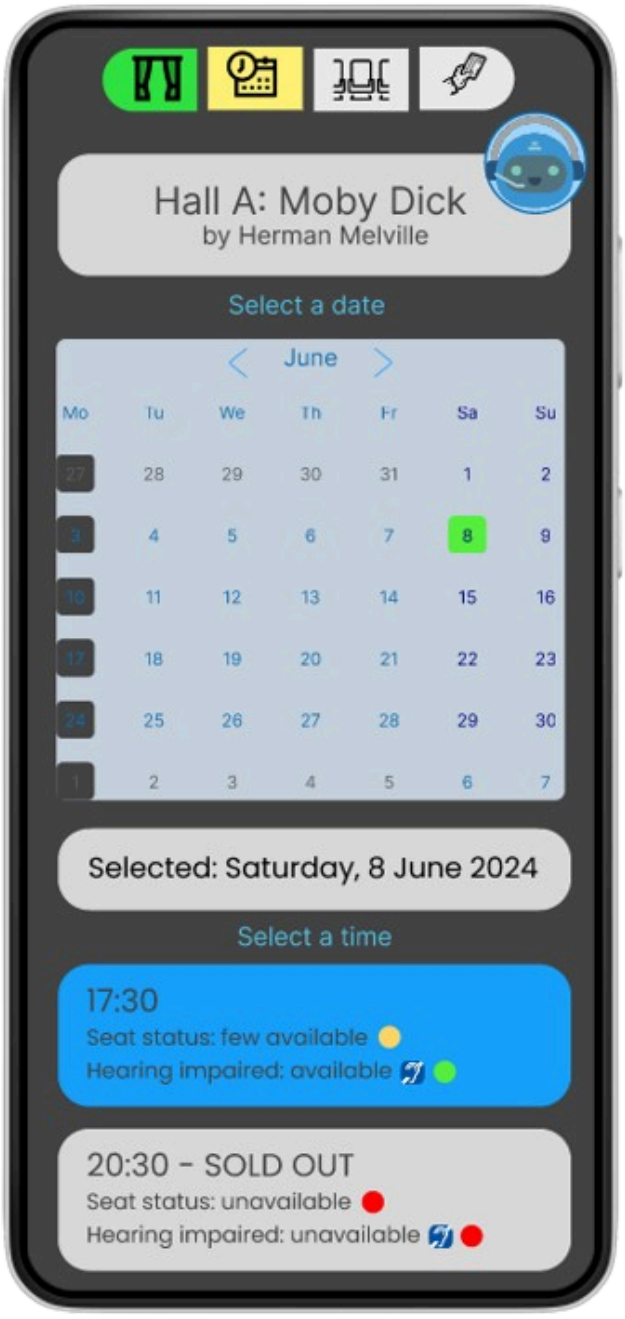
Figma Mockups

We are transitioning from Draw.io to Figma to improve our prototypes with detailed designs and interactive features. Figma allows us to create clickable prototypes that better simulate the user experience. This will help us refine design elements,

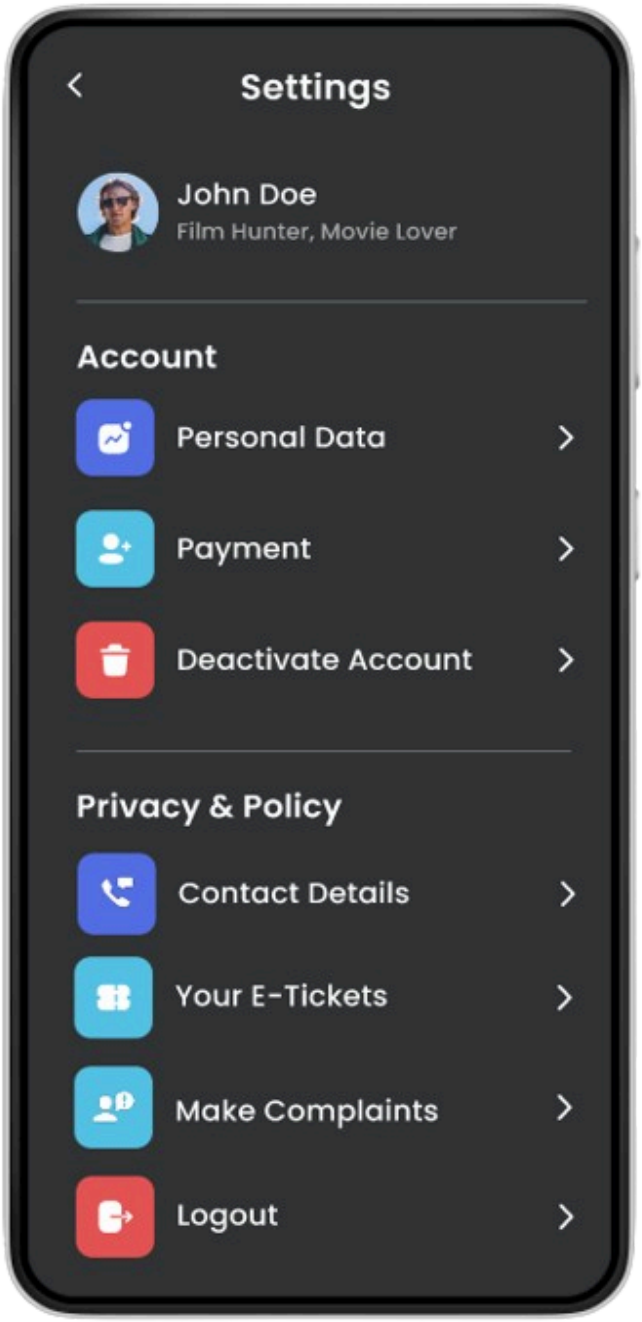
ensure accessibility, and create intuitive user flows for a more engaging and user-friendly interface.

Here are some of the mockups we created using Figma:

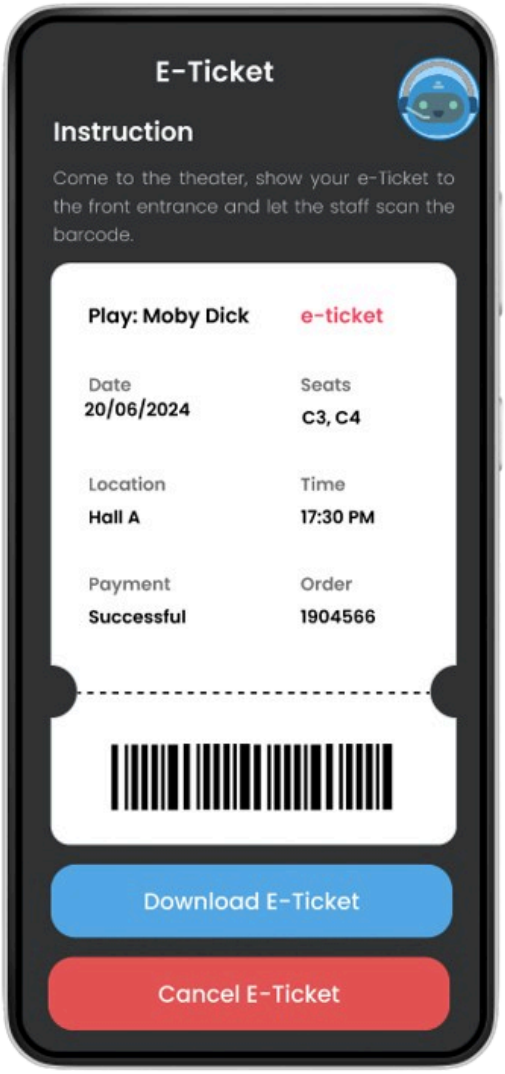
Booking Screen



Settings Screen

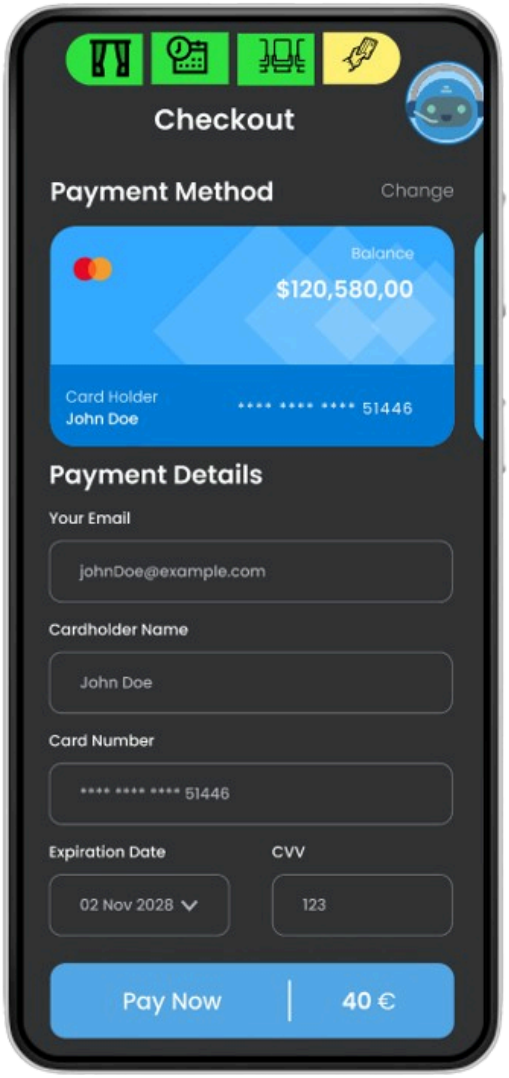


E-Tickets Screen



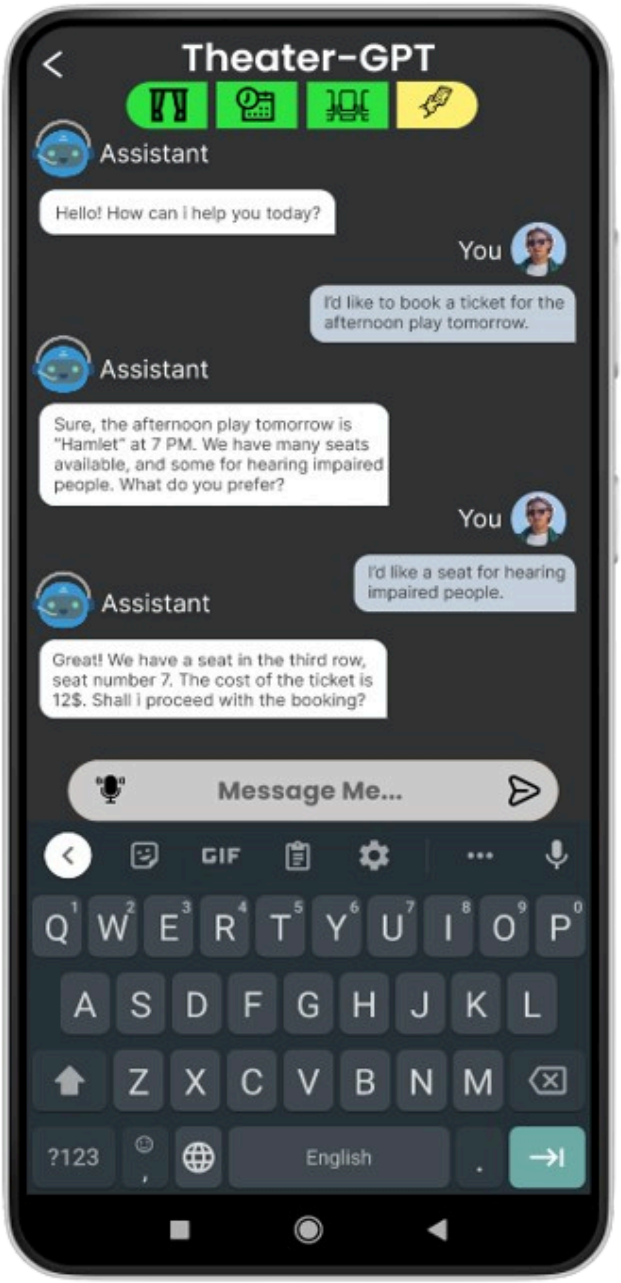
Seat Selection Screen

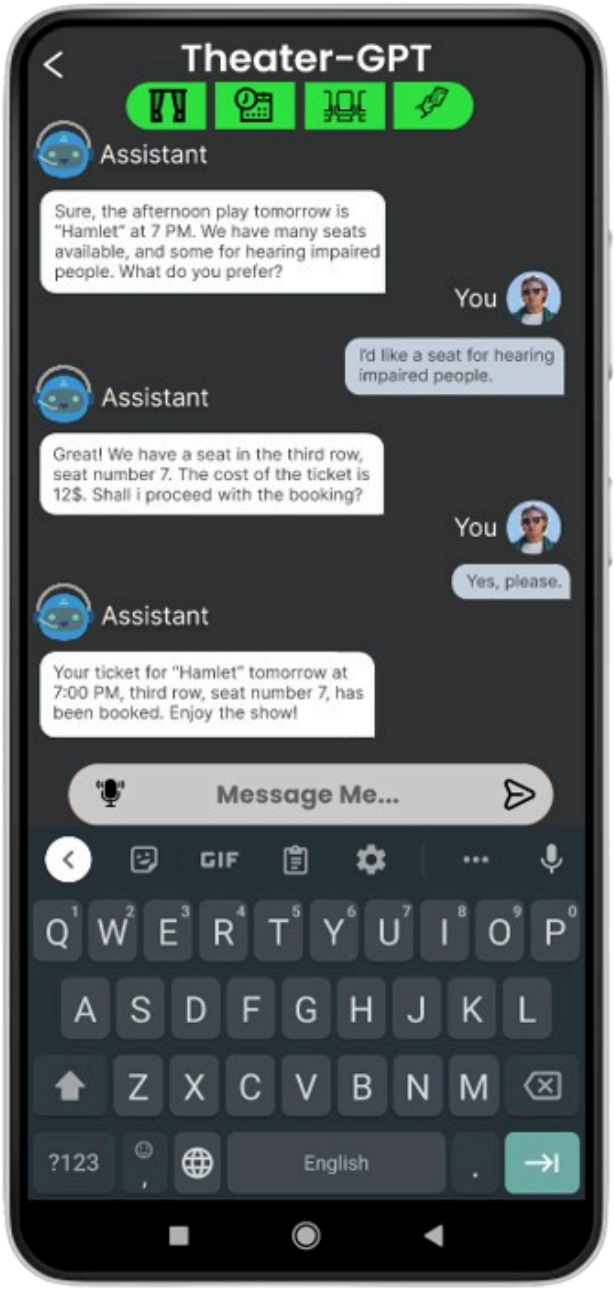
Payment Screen



Our Figma interfaces showcase several standout UI elements that enhance user experience. Features like the assistant bubble, reminiscent of messaging apps, offer intuitive interaction. Additionally, the sleek credit card and e-ticket elements provide a seamless transaction experience. Not to mention, the colorful and user-friendly seat selection map simplifies navigation and selection processes. These elements combine to create interfaces that are both functional and visually engaging, ensuring a smooth and enjoyable user journey.

Booking Scenario through assistant:





You can find all our mockups here: [HCI Assignment – Figma]

Evaluation Method I

Cognitive Walkthrough

Scenario: I want to make a booking for a play

Necessary actions:
1) Home screen - press "book" on the play I want to make a booking for (small gulf of evaluation)
2) Performance selection screen - press on the exact calendar date I want to see the play on and then on the afternoon or night performance.
3) Seat selection screen - press on the seats I want to book, that are available (not red)
4) Checkout screen - fill in credit card details and press the "pay now" button

Overall, the gulf of execution is quite low for this scenario (the most complicated of our application), indicating that users can easily determine the actions they need to perform and how to execute those actions with minimal confusion or need for trial and error. This suggests that the interface is intuitive and aligns well with the users' mental models, effectively supporting their task completion without significant cognitive effort.

Evaluation Method II

Think-Aloud Protocol

Additionally, we employed the Think-Aloud Protocol, facilitated by one of our classmates, to gather insights into user interactions with our prototypes. The participant was asked to verbalize their thoughts and actions while navigating the interface, providing valuable feedback on usability, clarity of instructions, and overall user experience. This method allowed us to identify pain points and areas for improvement in real-time, ensuring that our interface design aligns closely with user expectations and needs.

General Insights:

- The overall layout was intuitive and easy to navigate. Main features were accessible, making it simple for the user to move through different sections of the interface.
- Instructions for basic tasks, like booking tickets or viewing purchased tickets, were generally clear and easy to follow.
- The participant appreciated the aesthetic design and use of visual elements, noting that they contributed to a pleasing and engaging user experience.

Main Points:

- The user found that some texts resembled buttons, causing confusion about what was clickable and what was not.
- More careful seat coloring was needed. The user had difficulty distinguishing between available and unavailable seats due to poor color differentiation.
- The user felt that the progress bar icons could be better designed to convey the stages of the ticket booking process more clearly and intuitively.

Evaluation Method III

Usability Inspection

The usability inspection was conducted by another team of three students attending the HCI course at our university. This inspection provided valuable insights on how to implement our mockups in code to enhance the user interface's usability. Key points from the inspection include:

1. Ensure consistency with the back button, either include it everywhere or nowhere.
2. After each function performed by the chatbot, the user should be navigated to the corresponding app screen (not back to the chatbot).
3. Clarify the steps for actions like payment or reservation.
4. Increase font size, especially on the home page, and use larger icons.
5. Allow forward-backward navigation in the booking process.
6. Display a chatbot message when the app is opened.

Cycle III: Final Implementation

Target Group

The target group of the third cycle of development of our user interface is the same as the target group of the second stage, to ensure continuity in design improvements, maintain focus on accessibility features, and effectively address the specific needs of these users, enhancing overall usability and satisfaction.

Functions

No new functions were added in this cycle, but the previous functions were implemented in code and improved upon with regards to the needs of our target group.

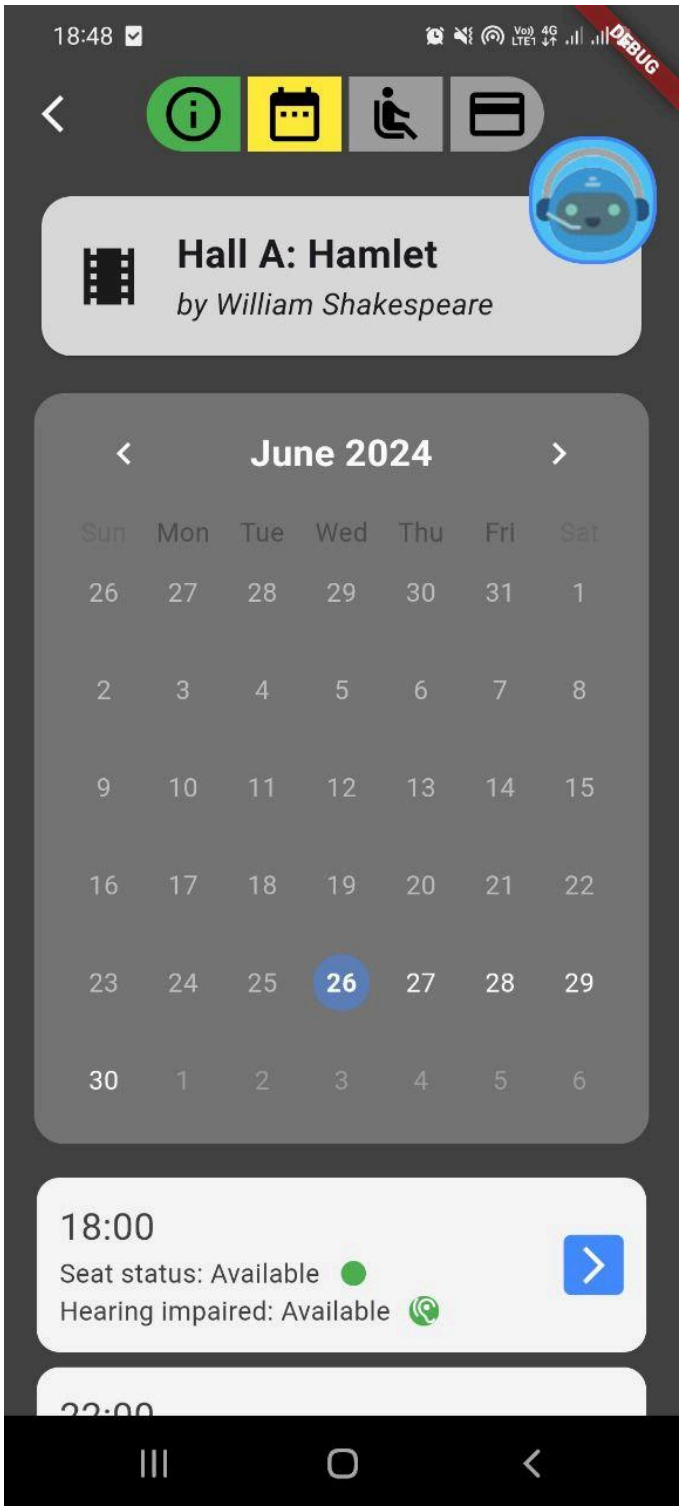
For example, the assistant bubble we had presented in our second cycle prototypes was given the property of being movable on the screen, similar to other famous chat applications. This can be useful for the user, who both has the assistant bubble at their disposal and can move it so it doesn't obscure any screen content.

User Interface

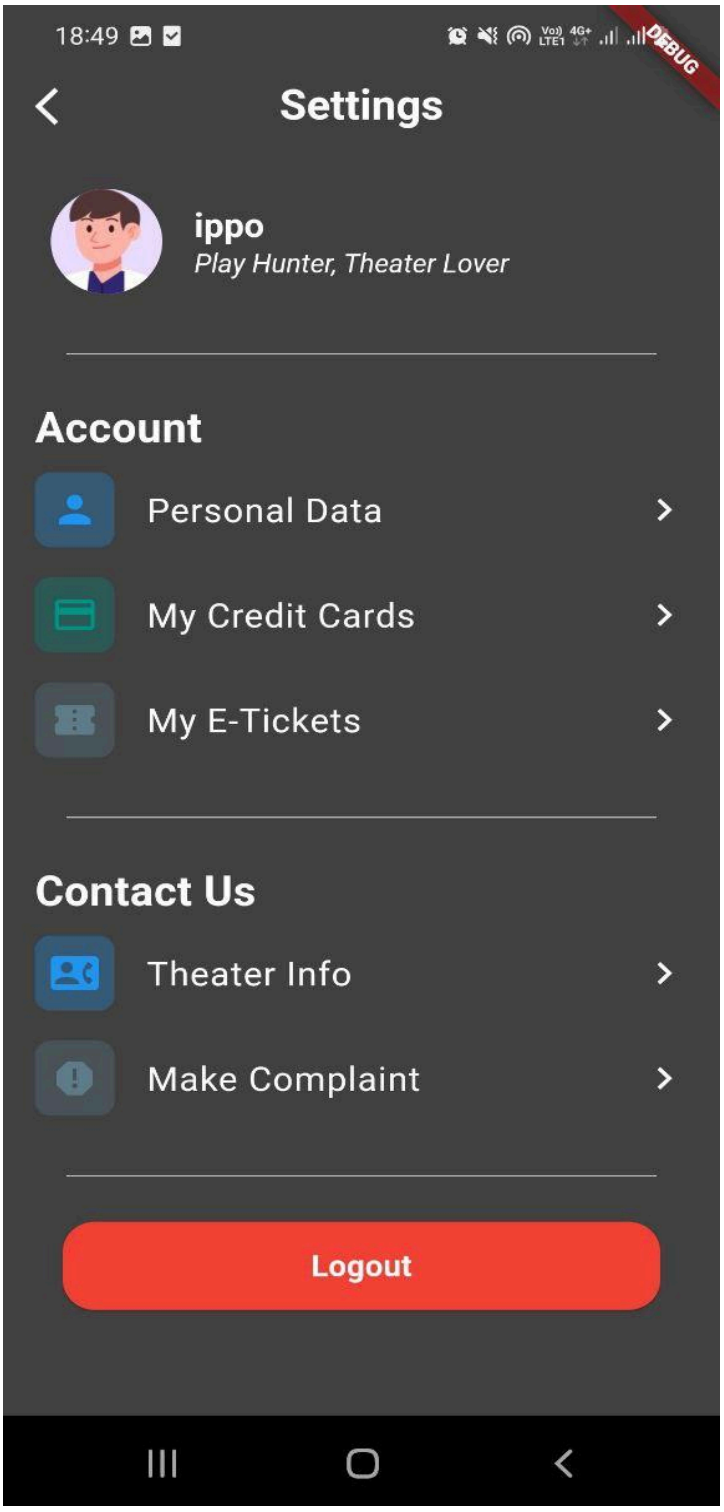
In the final cycle of our user interface development, we implemented the insights gathered from previous evaluations to refine our design further. We visualized the detailed Figma mockups into code, ensuring that the interface was not only visually appealing but also functional and intuitive in real-world usage scenarios. By focusing on enhancing accessibility features and addressing usability issues identified in earlier cycles, such as improving button visibility and clarifying navigation prompts, we aimed to deliver a seamless and satisfying user experience.

Here are the user interfaces of our mobile application, reflecting the designs previously presented in Figma:

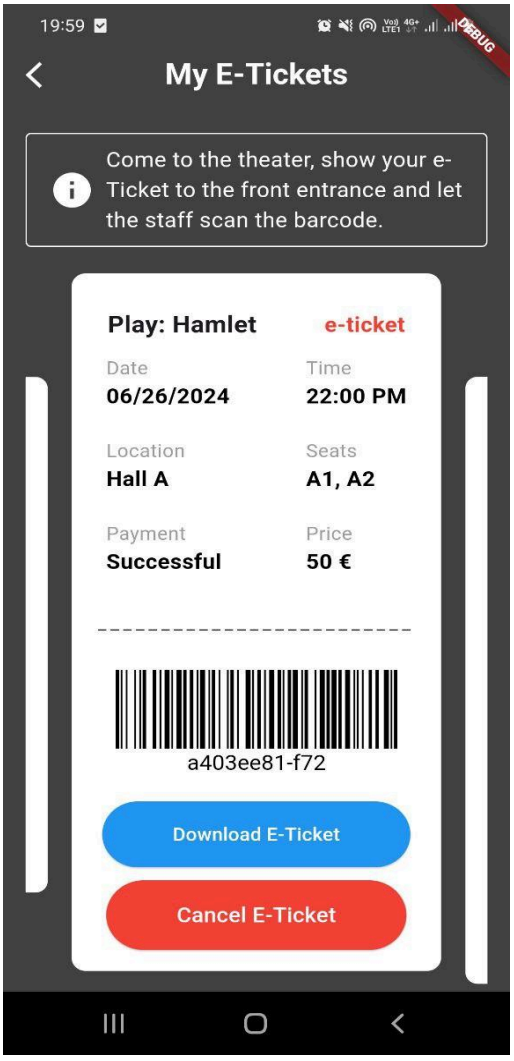
Booking Screen



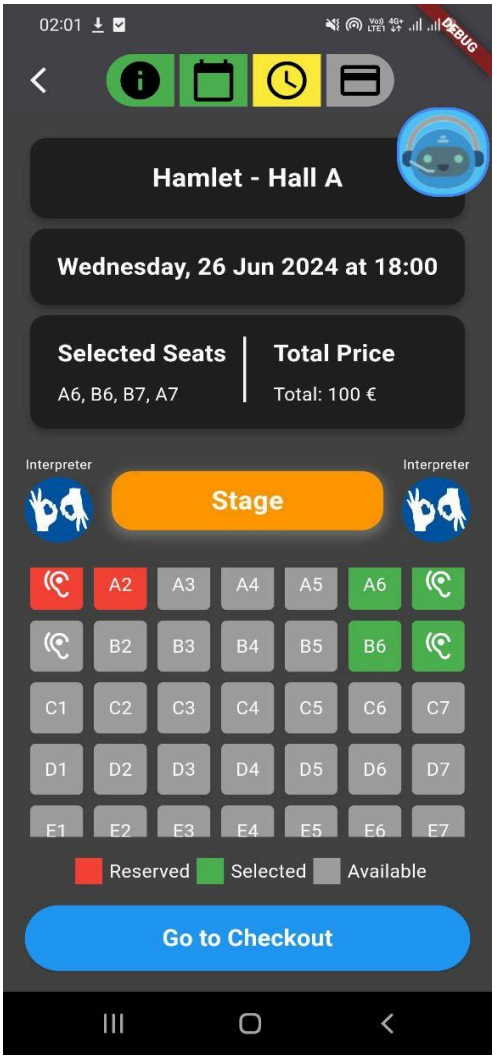
Settings Screen



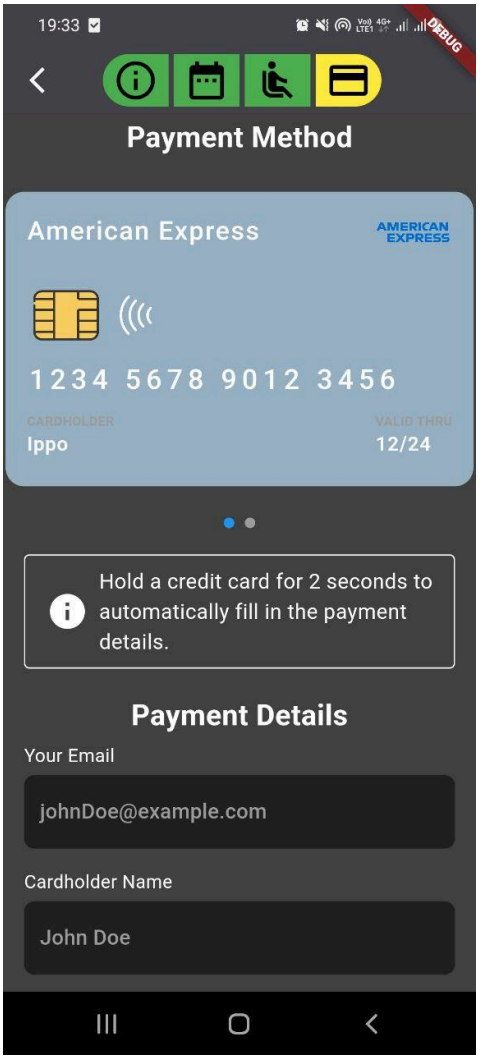
E-Tickets Screen



Seat Selection Screen

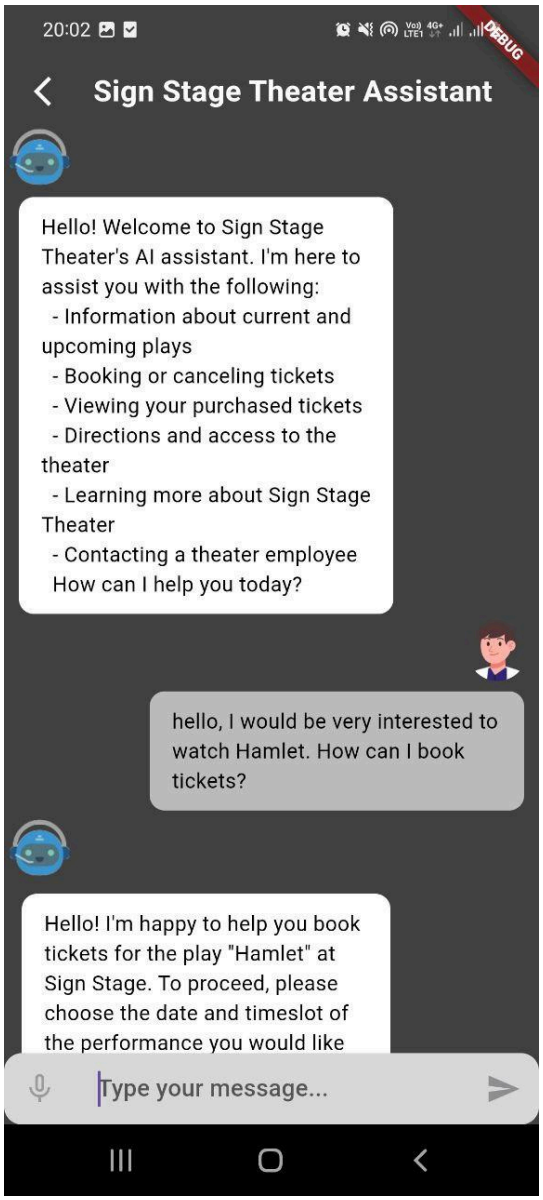


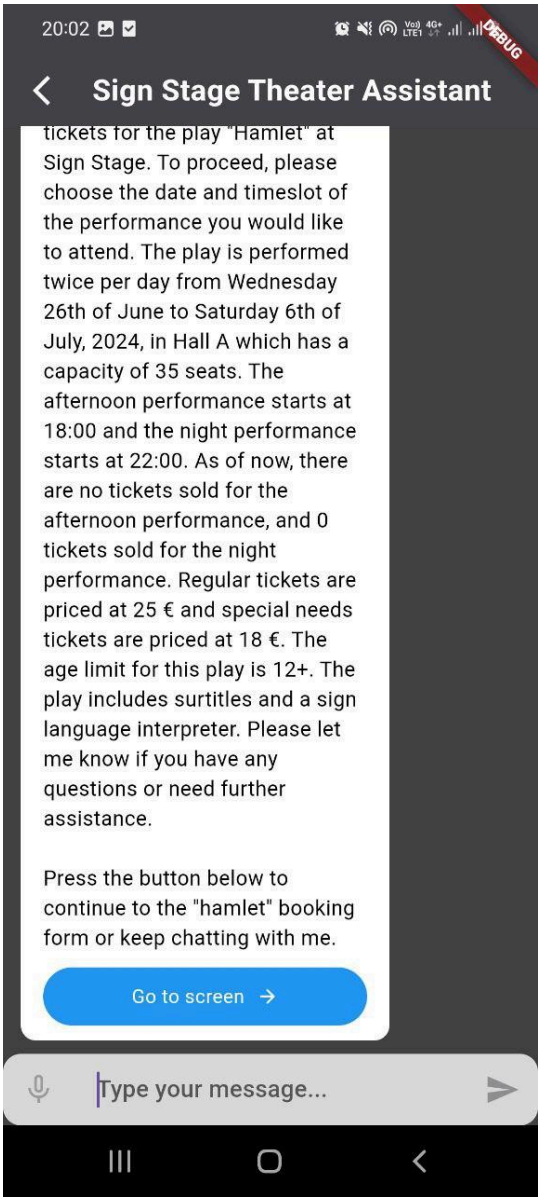
Payment Screen



We've clearly used all the feedback from the second cycle evaluations to shape our user interfaces. Every detail, from layout to function, reflects what users have told us they need. Our goal is to create intuitive designs that make using our products effortless and enjoyable, based directly on user input.

Booking Scenario through assistant:





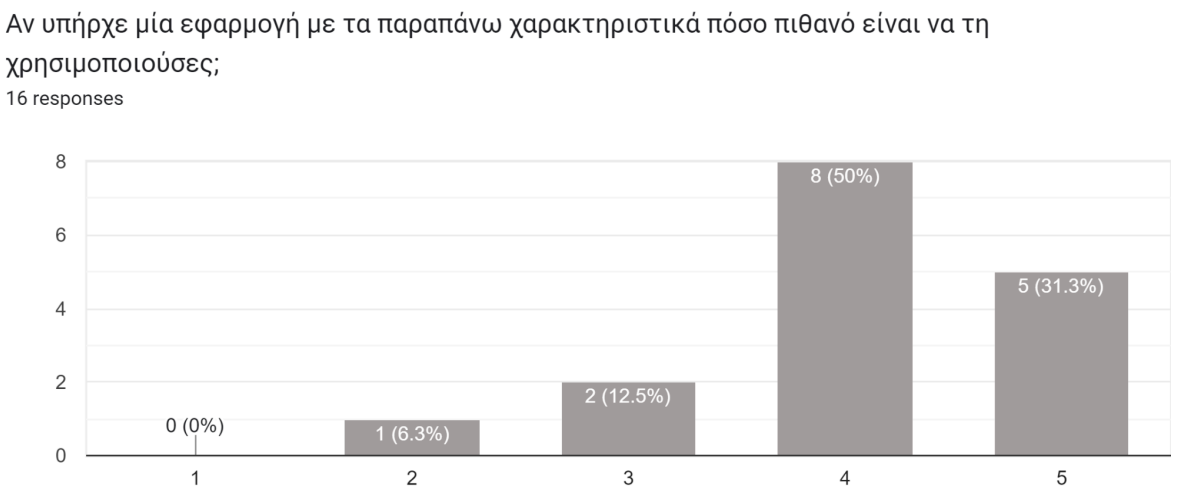
Evaluation Method I

Questionnaire

We created another questionnaire in Greek, similar to the first one, but with a focus on the interface of the app, rather than its general appeal. The questionnaire was answered by 19 people overall, 16 of which attended theater plays and would be willing to use a mobile app that would replace the theater ticket booth.

Most of the displayed screens were completely understandable to the participants, with only minor suggestions given. All the participants stated that they would suggest the app to friends of theirs who enjoy the theater.

Final feedback question: "If there was an app with all the above features, how is it likely that you would use it?"



You can find all individual results here: questionnaire_results(cycle3)

Evaluation Method II

Cognitive Walkthrough

Scenario: I want to make a complaint using the chatbot

Necessary actions:
1) Home screen - I press the assistant bubble to open the chatbot interface.

2) Chatbot message interface - I communicate through text or speech with the chatbot to express that I have a complaint to make.

a) If the chatbot fails to understand three consecutive times, it suggests contacting a theater employee and provides a button to do that takes me to the theater info screen. I press the button.
b) If it understands, it provides a button to redirect me to the make complaint screen. I press the button.

3a) Phone screen - I click on the phone icon to make the call to the theater employee.

3b) Make complaint screen - I provide the requested information though the text fields, specify my complaint using the options in the drop down menu. If my complaint does not fit into one of the predefined categories, I create a custom complaint headline and fill in the complaint description text field.

The cognitive walkthrough reveals that even first-time users can easily find the next action required in this scenario. The interface design clearly indicates the next steps, minimizing user confusion. The steps are logically ordered and intuitive, making it straightforward to follow through with the complaint process.

The gulf of execution is very low, as users can easily determine the actions they need to perform at each step. Pressing buttons and entering information into text fields are simple, clear tasks that align with typical user expectations.

Similarly, the gulf of evaluation is estimated to be very low because the system provides clear feedback at each stage, ensuring users understand the result of their actions. For instance, the chatbot explicitly suggests calling an employee if it cannot process the complaint, and it provides direct navigation to the appropriate screen if it understands the complaint. This effective feedback loop helps users gauge the success of their actions effortlessly.

Overall, the interface supports task completion with minimal cognitive effort, confirming the usability and user-friendliness of the application.

Evaluation Method III

Think-Aloud Protocol

To further refine our interface, we conducted a second round of the Think-Aloud Protocol with a 20-year-old student and theater enthusiast. During this session, the participant vocalized their thoughts and actions while using our application, providing new insights into usability, instruction clarity, and the overall user experience. This approach enabled us to gather real-time feedback, ensuring that our interface design remains aligned with user expectations and needs.

Main Points:

- The progress bar appeared clickable, causing confusion about its functionality.
- The progress bar should be removed from the chat screen to avoid distraction.
- Certain areas in the seat selection section looked clickable, potentially leading to navigation issues.
- The color of the time slots should be changed from blue to a very light gray for better differentiation.
- The "hold to autofill" message was not noticeable and often skipped by the user.
- The navigation suggestions related to chat requests need improvement for better clarity and usability.

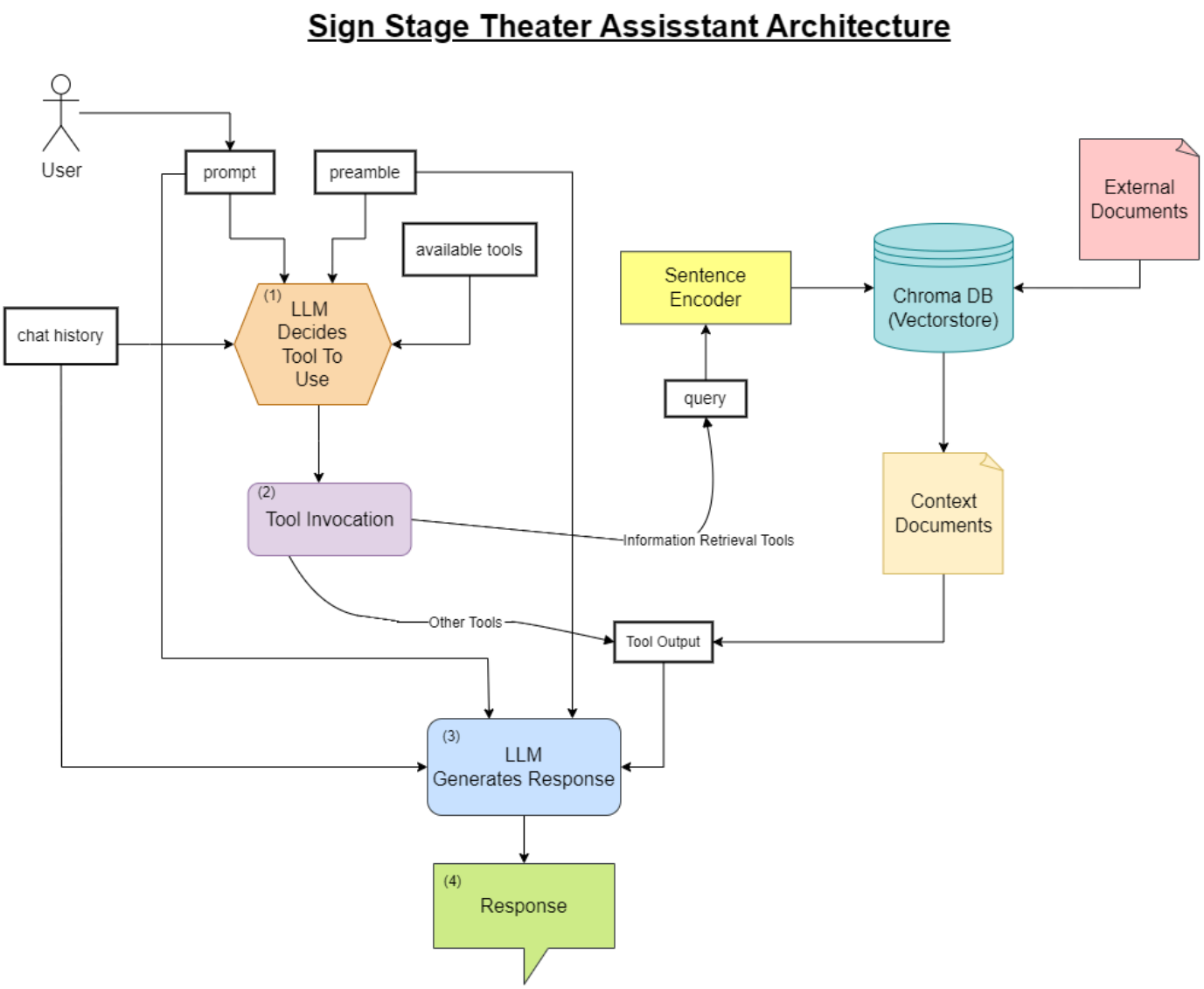
Evaluation Method IV

Usability Inspection

The usability inspection was conducted by another team of three students attending the HCI course at our university. This inspection provided valuable insights on how to improve our final prototypes to enhance the user interface's usability. Key points from the inspection include:

1. Better organization of the settings page (better category names and bigger logout button)
2. The starting position of the chatbot bubble should be the top right instead of the bottom right, which we were considering, because that way it doesn't get in the way of text, forcing the user to move it.
3. On the theater info screen, make the buttons for calling and sending an email more obvious.
4. Have the chat send a pre-defined text, the first time the user opens it by pressing the assistant bubble, containing information about what functions can be performed using it.
5. On the payment screen, make it evident that there can be multiple credit cards available and provide the option to give concise instructions on how to use them.
6. Make a pop-up confirmation screen before ticket purchases and cancellations.

Extra: LLM chatbot explanation



PipeLine Explanation:

(1): At first, the user's prompt is passed through the LLM, along with the preamble, a description of the available tools and how and when to use each one as well as the last 4 exchanges of the conversation, if they exist. In this phase the LLM recognises the user's intent and decides what is the appropriate tool to use to respond to their request.

(2): Then the LLM invokes the appropriate tool.

The tools can be put in two groups as seen in the diagram above.

(a) The first group has two tools that query an Index to retrieve the most relevant documents (using similarity search) to the user's prompt and return them along with the instructions on how to respond to the user.
(b) The second group has the rest of the tools (another 9 tools) that do not need extra context to respond. These tools just return instructions on how to respond to the user.

Each tool also returns a custom "code" that indicates the user's intent.

(3): In this step, the LLM takes as input once again the preamble (slightly modified than the one passed in the first step to denote that its task now is to generate a response based both on the general information and the output of the tool), the output of the invoked tool, the chat history (at most the 4 previous exchanges) and the original prompt of the user to "remind" it of the user's request and generates the final response.

(4): The user is displayed the Assistant's response. If the requested action needs the user to navigate to another screen, we display below the Assistant's response, a button (according to the custom "code" sent) that when pressed does just that.

Sign_Stage_Theater_Assistant_Architecture.png

You can find the source code here:

https://github.com/Ippo03/sign-stage

Conclusion

In this project, we developed a user interface for a text-based chat application to improve theater-goers' experiences by enabling seamless communication with a hypothetical theater's box office. Utilizing the spiral model, we iteratively refined our design based on user feedback, ensuring inclusivity and accessibility for diverse user groups, including individuals with hearing and visual impairments. Key features included ticket booking, cancellation, providing information about current and upcoming plays, and contacting a theater employee. We used tools like Draw.io and Figma for prototyping and integrated Definite Clause Grammars (DCGs) in Prolog and a large language model (LLM) for robust chatbot interactions. This iterative, user-centric approach resulted in a comprehensive, functional, and user-friendly interface.