



ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΕΠΙΚΟΙΝΩΝΙΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πτυχιακή Εργασία

Τίτλος Πτυχιακής Εργασίας	Σχεδιασμός και Υλοποίηση Εφαρμογής Ιστοσελίδας για την Εύρεση Μπρέιμπισίτερ χρησιμοποιώντας C#, .NET, και React.js Design and Implementation of a Babysitter Finding Web Application using C#, .NET, and React.js
Όνοματεπώνυμο Φοιτητή	Ρεντίνα Τζαφεράι(Redina Xhaferaj)
Πατρώνυμο	Μυζαφέρ(Myzafer)
Αριθμός Μητρώου	Π/14178
Επιβλέπων	Ευθύμιος Αλέπης, Αναπληρωτής Καθηγητής

Ημερομηνία Παράδοσης 07/2024

Copyright ©

Απαγορεύεται η αντιγραφή, αποθήκευση και διανομή της παρούσας εργασίας, εξ ολοκλήρου ή τμήματος αυτής, για εμπορικό σκοπό. Επιτρέπεται η ανατύπωση, αποθήκευση και διανομή για σκοπό μη κερδοσκοπικό, εκπαιδευτικής ή ερευνητικής φύσης, υπό την προϋπόθεση να αναφέρεται η πηγή προέλευσης και να διατηρείται το παρόν μήνυμα.

Οι απόψεις και τα συμπεράσματα που περιέχονται σε αυτό το έγγραφο εκφράζουν αποκλειστικά τον συγγραφέα και δεν αντιπροσωπεύουν τις επίσημες θέσεις του Πανεπιστημίου Πειραιώς.

Ως συγγραφέας της παρούσας εργασίας δηλώνω πως η παρούσα εργασία δεν αποτελεί προϊόν λογοκλοπής και δεν περιέχει υλικό από μη αναφερόμενες πηγές.

Contents

1. Introduction	5
1.1 Overview of the Problem Domain.....	5
1.2 Motivation Behind Developing the Software	5
1.3 Objectives and Scope	5
1.4 Outline	6
2. Literature Review	7
2.1 Babysitting Services, Booking Systems, and Related Technologies.....	7
2.2 Challenges Faced by Parents in Finding Babysitters and Limitations of Traditional Methods.....	7
2.3 Analysis of Similar Software Solutions or Platforms Available in the Market.....	8
3. Methodology	9
3.1 Description of the Methodology	9
3.2 Software Development Lifecycle (SDLC) Model.....	9
3.3 Technologies, Programming Languages, and Tools.....	10
4. Requirement Analysis	10
4.1 Functional and Non-functional Requirements.....	10
4.2 Stakeholder Analysis and User Personas	11
4.3 Use Case Diagrams and Scenarios	12
5. System Design.....	23
5.1 Description of the Architectural Design of the Software System	23
5.2 Component Diagrams	23
5.3 Class Diagrams.....	25
5.4 Sequence Diagrams.....	25
5.5 The Entity-Relationship Diagram (ERD)	27
Relationships	28
6. Implementation	29
6.1 Detailing the Implementation Process	29
6.2 Showcasing Specific Features and Functionalities	29

6.3 Discussing Challenges and Solutions	30
7. Testing and Evaluation.....	30
7.1 Explanation of Testing Strategies.....	30
7.2 Presentation of Testing Results	30
7.3 Evaluation Against Requirements	31
8. Results and Analysis.....	31
8.1 Summary of Key Findings	31
8.2 Discussion of Usability, Performance, and Scalability	31
8.3 Comparison with Existing Solutions	32
9. Future Work	32
9.1 Potential Enhancements or Features	32
9.2 Areas for Further Research or Expansion	32
9.3 Potential Impact on the Babysitting Industry and Childcare Services	33
10. Conclusion	33
10.1 Recap of Main Contributions	33
10.2 Reflection on Challenges and Lessons Learned	34
10.3 Recommendations for Practitioners and Researchers.....	34
11. References	34
11.1 Software Documentation	34
11. 2 Websites and Online Resources	35
11.3 Inspiration and Case Studies	35
12. Appendices	35
12.1 Code Snippets	35
12.1.1 Searching for Babysitters by supporting areas	35
12.1.2 Create Appointment	36
12.1.3 Notify Customer and Babysitter	36
12.2 User Manual.....	37
12.2.1 Homepage, Register and Login.....	37
12.2.2 Babysitter Manual	40
12.2.3 Customer Manual.....	46

1. Introduction

1.1 Overview of the Problem Domain

In today's fast-paced world, the demand for babysitters has surged as couples juggle their careers and family life. Yet, the quest for a dependable babysitter remains a daunting challenge for many parents. Typically reliant on word-of-mouth recommendations or social media advertisements, the search process is arduous and time-consuming, often leading to the apprehension of hiring unreliable individuals.

There is a pressing need for a more efficient solution, one that connects parents with certified professionals swiftly and reliably. Streamlining this process would not only alleviate the stress of finding suitable childcare but also ensure peace of mind for parents entrusting their children's care to qualified caregivers.

1.2 Motivation Behind Developing the Software

The inspiration to create this software stemmed from witnessing the challenges faced by new couples in finding trustworthy babysitters. Through diligent research, I discovered a significant gap in the market for a solution addressing this pressing need. This software aims to offer couples access to a broader network of babysitters, including those in their local area. By facilitating online appointments at their convenience, it seeks to streamline the scheduling process.

Moreover, this platform emphasizes transparency by showcasing babysitters' qualifications and ratings from fellow parents. This ensures that parents can make informed decisions when selecting a caregiver for their children. Ultimately, our goal is to provide peace of mind and convenience to busy parents while simultaneously offering opportunities for qualified babysitters to connect with families in need of their services.

1.3 Objectives and Scope

This thesis aims to develop a user-friendly software platform that addresses the challenges parents encounter in securing reliable babysitters. Our research and software development endeavor seek to create a comprehensive solution that simplifies the process of connecting parents with qualified childcare providers.

Our specific objectives include implementing a robust booking system that efficiently matches babysitters with available childcare needs, considering factors such as location, availability, and user preferences. Furthermore, we aim to assess the usability and effectiveness of the software through thorough user testing and feedback. This

evaluation process will help us identify areas for improvement and propose enhancements to elevate the overall user experience and functionality of the platform.

While our primary focus lies in the technical development and usability evaluation of the software, we recognize the broader societal implications of childcare services. However, our research primarily centers on the practical aspects of software development and usability testing, with the intention of offering valuable insights into the potential impact of the software on the babysitting industry and childcare services.

1.4 Outline

This document will delve deeply into the methodology behind the software, encompassing the requirement analysis, system design, implementation, testing, and evaluation phases.

We will thoroughly examine the process of gathering and analyzing requirements, ensuring that the software meets the needs of both parents and childcare providers effectively. Subsequently, we will detail the system design, outlining the architecture and functionality of the platform to ensure its usability and efficiency.

The implementation section will provide insights into the development process, including the technologies utilized and the strategies employed to bring the software to fruition. Following this, we will discuss the testing and evaluation procedures, including rigorous testing protocols and user feedback mechanisms.

The results and analysis section will present findings from testing and evaluation, offering insights into the software's performance and user satisfaction.

Furthermore, we will outline future work, discussing potential enhancements and features to improve the software's functionality and address emerging needs in the childcare industry. Finally, we will draw conclusions based on our research, summarizing key findings and implications for both the software development process and the broader childcare landscape.

2. Literature Review

2.1 Babysitting Services, Booking Systems, and Related Technologies

To delve into babysitting services, booking systems, and relevant technologies, it's crucial to define key terms. Babysitting services involve temporary childcare by qualified individuals, often sought by busy parents. Booking systems facilitate scheduling babysitting appointments between parents and caregivers, using online platforms or mobile applications. Relevant technologies include online platforms and communication tools designed to connect parents with reliable childcare providers.

A systematic review of literature offers insights into childcare services and technology. Academic journals, conference papers, and industry reports discuss trends and challenges in childcare. Research explores technological solutions like AI-driven matching algorithms and real-time communication platforms to enhance childcare management.

Synthesizing findings reveals recurring challenges like finding trustworthy babysitters and improving parent-caregiver matching. The literature also emphasizes the potential of technological innovations to address these challenges and enhance childcare services. Identifying these insights lays the groundwork for further research and solutions meeting the evolving needs of parents and caregivers.

2.2 Challenges Faced by Parents in Finding Babysitters and Limitations of Traditional Methods

Parents face numerous challenges when seeking reliable babysitters, including limited availability, uncertainty about qualifications, and safety concerns. Babysitters often have restricted schedules, making it difficult for parents to find someone who can accommodate their needs. Additionally, there's a lack of transparency regarding a babysitter's qualifications and experience, leaving parents unsure about entrusting their child's care to them. Safety and trust are paramount, with parents often grappling with concerns about leaving their child with a stranger.

Traditional methods of finding babysitters, such as word-of-mouth referrals or classified ads, have significant limitations. While word-of-mouth recommendations can be valuable, they're subjective and may not always align with a parent's specific needs or preferences. Classified ads and community bulletin boards lack the sophistication and reliability needed to ensure a suitable match between parents and babysitters. These

methods often result in a time-consuming and frustrating search process for parents, with no guarantee of finding a suitable caregiver.

Real-world examples vividly illustrate the challenges parents face in navigating the childcare landscape. Stories abound of parents struggling to find available babysitters during crucial times, such as date nights or work emergencies. Moreover, anecdotes reveal instances where parents have had to settle for less-than-ideal caregivers due to a lack of viable options or information. These experiences underscore the urgency for more efficient and reliable solutions in the realm of childcare services, where parents can confidently entrust their child's well-being to qualified professionals.

2.3 Analysis of Similar Software Solutions or Platforms Available in the Market

Numerous software solutions and platforms aim to streamline the booking of babysitters and childcare services. These range from dedicated mobile applications to online booking platforms, each offering various features and functionalities. Some popular examples include platforms like Care.com, UrbanSitter, and Sittercity. These platforms allow parents to search for and book babysitters based on criteria such as availability, location, and specific requirements.

When comparing these platforms, it's essential to assess their strengths, weaknesses, and areas for improvement. Care.com, for instance, boasts a vast network of caregivers and comprehensive background checks, but some users have criticized its subscription fees and variable quality of caregivers. UrbanSitter emphasizes real-time booking and trusted reviews, but its availability may be limited in certain areas. Sittercity offers a range of features like background checks and caregiver profiles but may lack the same level of user interface polish as its competitors.

Factors such as user interface design, ease of use, reliability, security measures, and customer support play crucial roles in evaluating these platforms. A platform with an intuitive interface, robust security measures, and responsive customer support is likely to attract and retain users. Success stories or case studies of similar software solutions can provide valuable insights into best practices and lessons learned. By studying these examples, developers can glean valuable insights into what works well and where improvements can be made in designing software solutions for booking babysitters and childcare services.

3. Methodology

3.1 Description of the Methodology

In developing the babysitter booking software, I embraced a methodical approach to ensure a smooth project journey. Grounded in Agile principles, known for their flexibility and adaptability, I broke down tasks into manageable sprints, keeping myself on track as a solo developer.

This methodology guided my planning phase, enabling me to define clear objectives and prioritize features based on thorough research and stakeholder feedback. Regular self-assessment and adjustment of the development roadmap ensured alignment with project requirements. During execution, Agile facilitated a dynamic and responsive development process, allowing me to incorporate user feedback and refine features as needed.

Ultimately, this methodology played a pivotal role in successfully delivering the babysitter booking software. Embracing Agile principles helped me maintain focus and deliver a high-quality product that meet user needs. As a one-person team, this approach fostered a sense of ownership and accountability, ensuring a product that was not only functional but also responsive to the evolving needs of users and the childcare industry.

3.2 Software Development Lifecycle (SDLC) Model

The Software Development Life Cycle (SDLC) model employed in the development process of the babysitter booking software was Agile. Agile is characterized by its iterative and incremental approach to software development, allowing for flexibility and adaptability throughout the project lifecycle.

The key phases of the Agile SDLC model include requirements gathering, design, implementation, testing, deployment, and maintenance. In the requirements gathering phase, I engaged in thorough research and stakeholder consultations to define clear project objectives and user needs. The design phase involved creating mockups and prototypes to visualize the software's interface and functionality. During implementation, I translated the design into code, focusing on incremental development to deliver features in small, manageable increments.

Testing was conducted continuously throughout the development process, with regular feedback loops to identify and address issues promptly. Deployment involved releasing new features to users in a controlled manner, ensuring a smooth transition from development to production. Maintenance involved ongoing support and updates to

address any issues that arose post-deployment. Throughout each phase, I adapted the Agile methodology to accommodate the unique requirements and constraints of the project, prioritizing flexibility and responsiveness to changes in user needs and industry trends.

3.3 Technologies, Programming Languages, and Tools

In the development of the babysitter booking software, a combination of technologies, including .NET 8, C#, and React.js, formed the backbone of the application. .NET 8, renowned for its versatility and scalability, was chosen as the primary framework for backend development, providing robust support for server-side logic, data processing, and database management. Complementing .NET, C# served as the programming language of choice for its seamless integration with the .NET framework, facilitating efficient development and maintenance of backend functionalities.

React.js, a popular JavaScript library, played a pivotal role in frontend development, enabling the creation of dynamic and interactive user interfaces. Its component-based architecture and virtual DOM rendering ensured a smooth and responsive user experience across different devices and browsers. The selection of React.js was driven by its flexibility, performance, and extensive ecosystem of libraries and tools, which streamlined the development of modern and visually appealing UI components.

Integrated development environments (IDEs) such as Visual Studio 2022 provided a comprehensive toolkit for coding, debugging, and testing, enhancing developer productivity and efficiency. Version control systems like Git were employed to manage code changes and facilitate collaboration among team members, ensuring seamless integration of new features and updates. Testing frameworks and project management tools were also integrated into the development workflow to ensure quality assurance and effective project planning and execution. Together, these technologies and tools played crucial roles in realizing the desired functionalities and objectives of the babysitter booking software, delivering a seamless and intuitive platform for parents and childcare providers alike.

4. Requirement Analysis

4.1 Functional and Non-functional Requirements

To begin with, the requirements of the babysitter booking software are categorized into functional and non-functional aspects. Functional requirements outline what the system

should do, encompassing features and functionalities essential for its operation. Non-functional requirements, on the other hand, specify qualities or constraints of the system, such as performance, reliability, security, usability, and scalability.

Functional requirements of the software include user registration, allowing parents and babysitters to create accounts with unique login credentials. Profile creation functionality enables users to input and manage personal information, including contact details, availability, qualifications, and preferences. A robust search functionality is essential, allowing parents to find babysitters based on location. The booking system facilitates the scheduling of babysitting appointments, providing a seamless and efficient process for both parents and caregivers. Messaging features enable communication between parents and babysitters, facilitating coordination and addressing any queries or concerns. Finally, payment processing functionality ensures secure and hassle-free transactions for booking babysitting services.

Non-functional requirements related to performance focus on the system's responsiveness and speed, ensuring timely access to information and functionalities. Reliability requirements dictate the system's ability to perform consistently and accurately under varying conditions, minimizing downtime and errors. Security measures must be in place to protect user data and transactions, including encryption protocols and secure authentication mechanisms. Usability requirements emphasize intuitive and user-friendly interfaces, ensuring ease of navigation and interaction for all users. Scalability requirements address the system's ability to handle increasing volumes of users and data without compromising performance or functionality. These non-functional requirements are measurable and testable, ensuring the software meets quality standards and user expectations.

4.2 Stakeholder Analysis and User Personas

During the requirement gathering phase of my thesis project, a thorough stakeholder analysis was conducted to identify individuals or groups with a vested interest in the babysitter booking software. As the primary stakeholder, I played a dual role as both the developer and the end-user. Additionally, academic advisors and committee members were considered key stakeholders, as their feedback and expectations would influence the project's direction. Understanding their perspectives was crucial in ensuring that the project met academic standards and addressed practical user needs.

To represent the different types of users who would interact with the system, I developed detailed user personas. These personas included parents seeking babysitting services and babysitters offering their services. For parents, the persona focused on characteristics such as busy schedules, the need for reliable and trustworthy childcare, and the desire for a user-friendly booking process. For babysitters, the persona

highlighted aspects such as varying availability, the importance of showcasing qualifications and experience, and the need for a seamless way to manage bookings and communicate with parents.

The characteristics, needs, goals, and pain points of each user persona were integral to informing the design and prioritization of system features. For parents, key features prioritized included an intuitive search functionality to find qualified babysitters, secure payment processing, and a reliable booking system. Addressing the pain points of trust and reliability, features like detailed babysitter profiles and reviews from other parents were emphasized. For babysitters, the design focused on easy profile creation, flexible scheduling options, and clear communication tools to connect with parents. These insights ensured that the software was tailored to meet the specific needs and preferences of its primary users, enhancing overall user satisfaction and functionality.

4.3 Use Case Diagrams and Scenarios

Actors:

1. Parent
2. Babysitter

Use Cases:

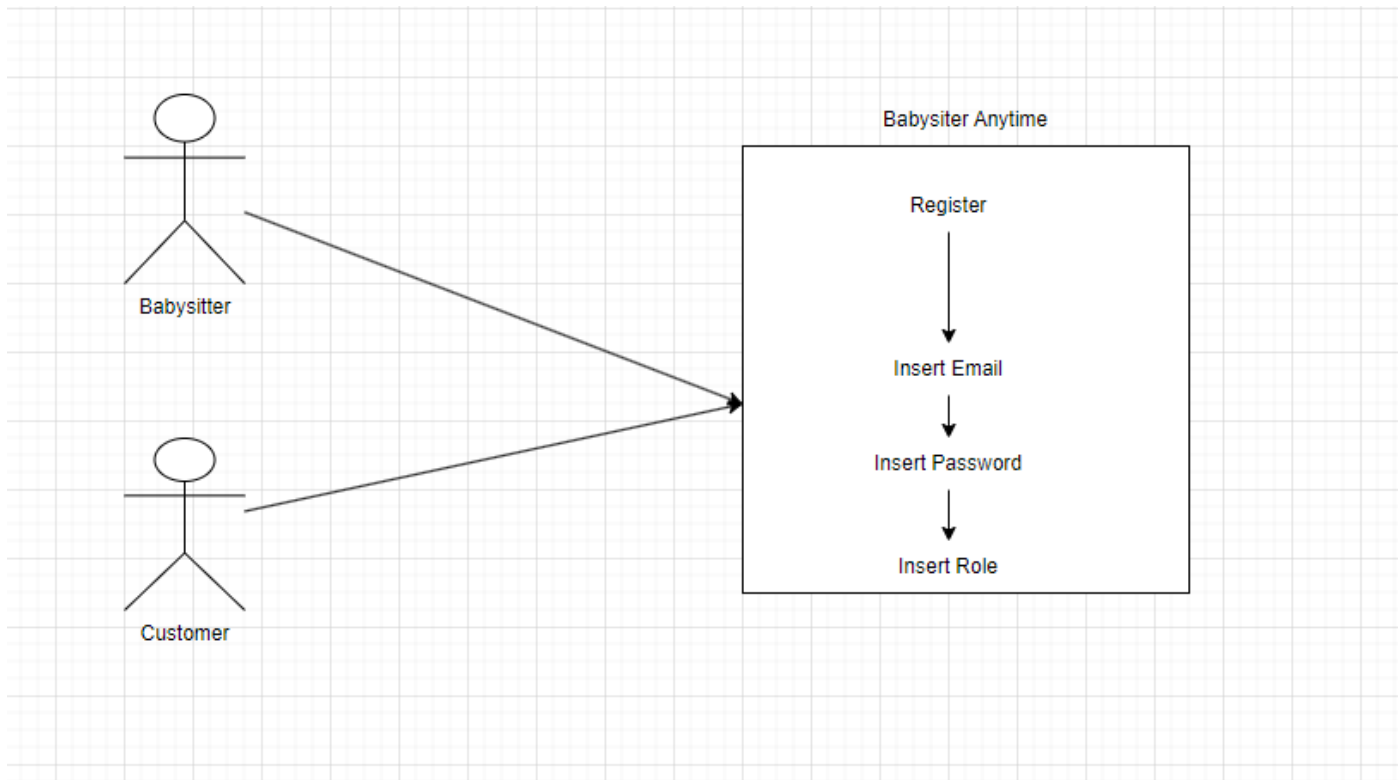
1. User Registration

On this page the user can register. In order to complete registration the user needs to complete the following information:

Email: The user's email address.

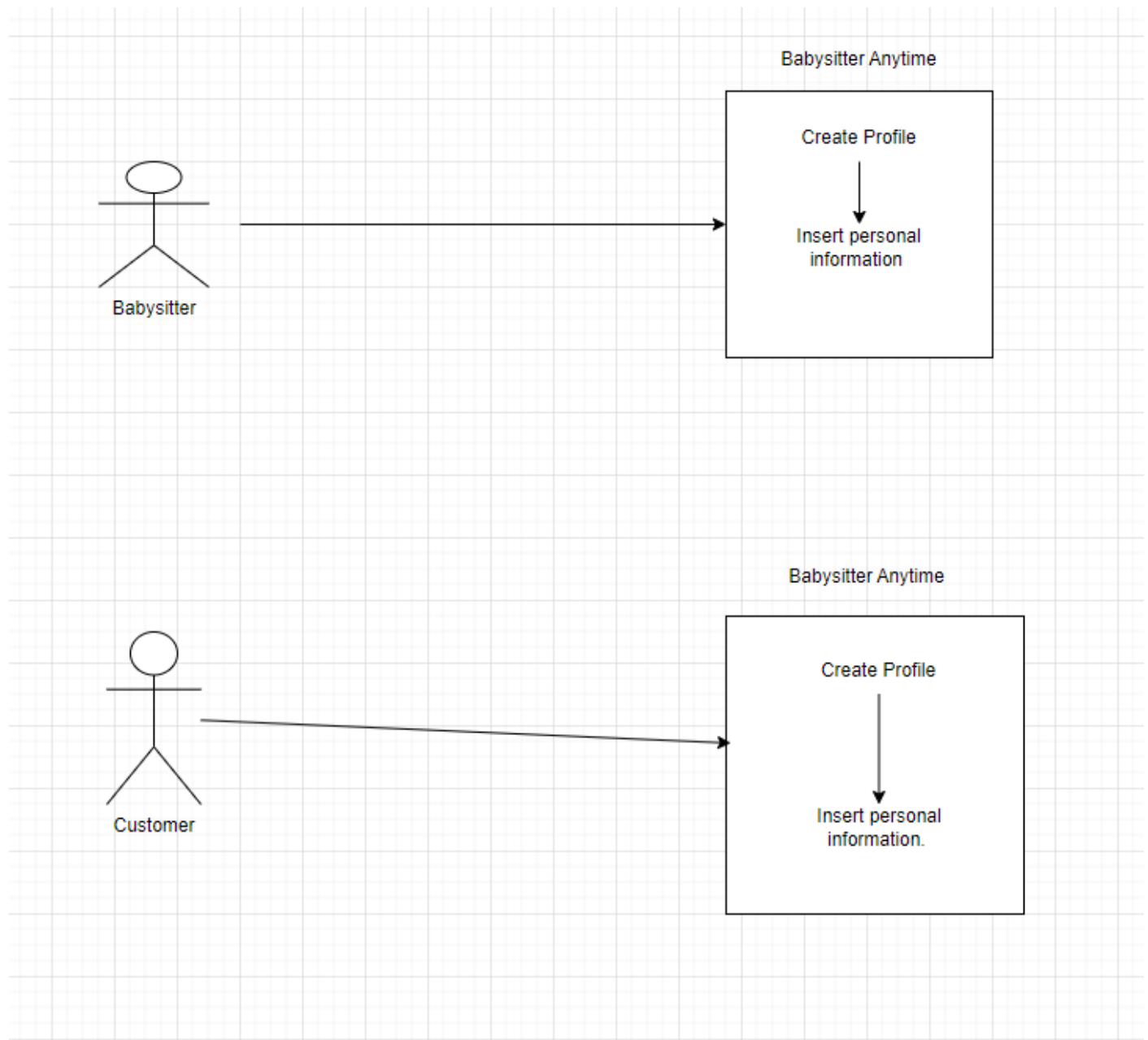
Password: The password of the user for the application. The password needs to be more than 10 character, have a capital letter, a numeric and a special character.

Role: The user needs to choose Babysitter if he is a babysitter and Customer if he is a customer.



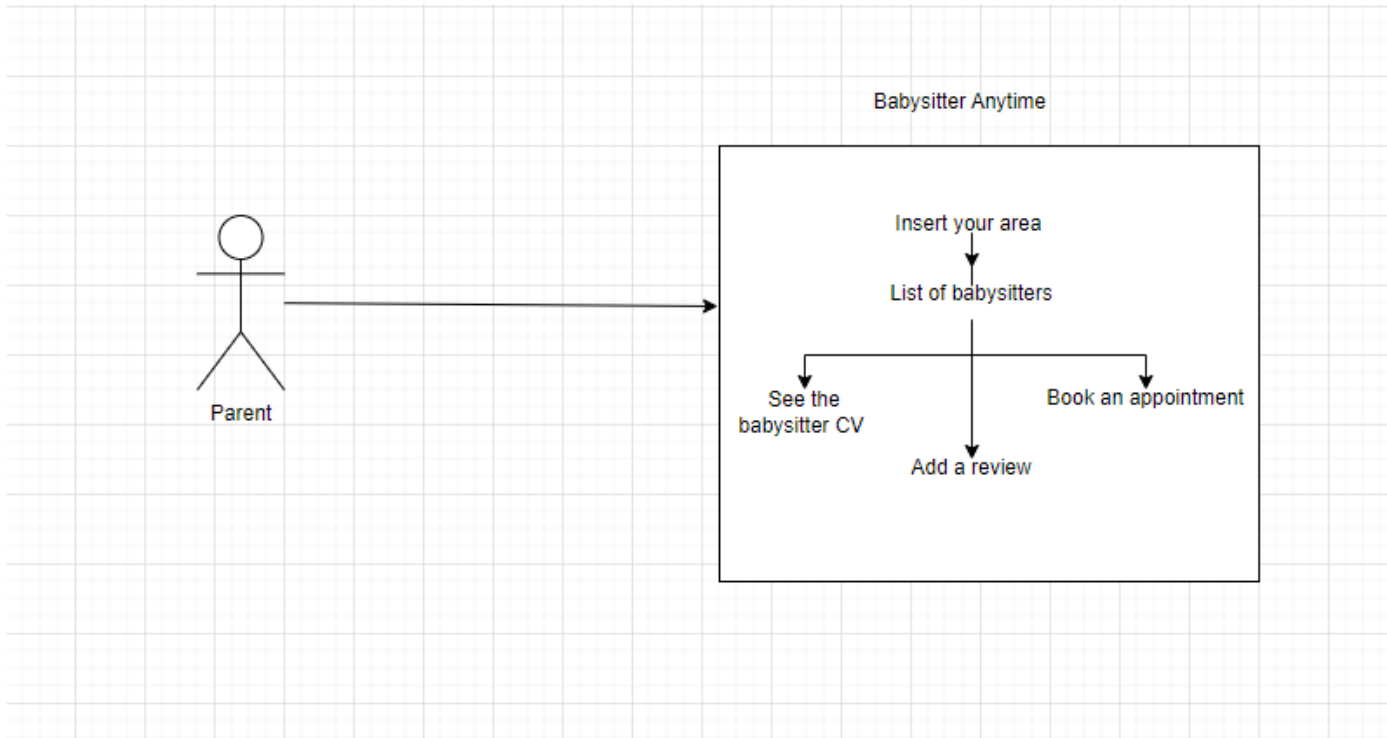
2. Profile Creation

After registration the babysitter and the customer are directed to the create profile page where they can insert their personal information and submit their profile.



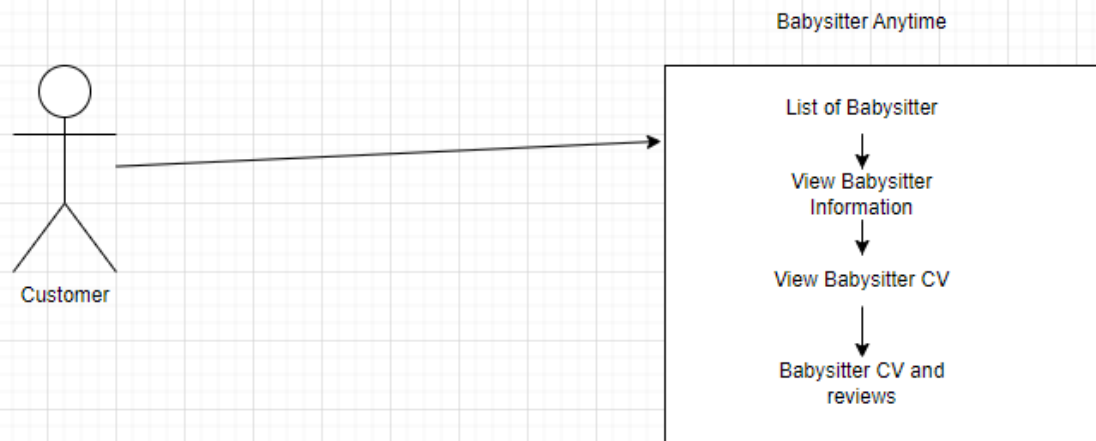
3. Search for Babysitters

The customer's home page consists of a searching bar where the customer can insert the area of their choice. After the area is selected, the babysitter that are available on that area will appear. The list of the babysitters will show their profile information and a few more options. The customer is presented with the option to view the babysitters CV, the option to book an appointment, and the option to add a review for the babysitter.



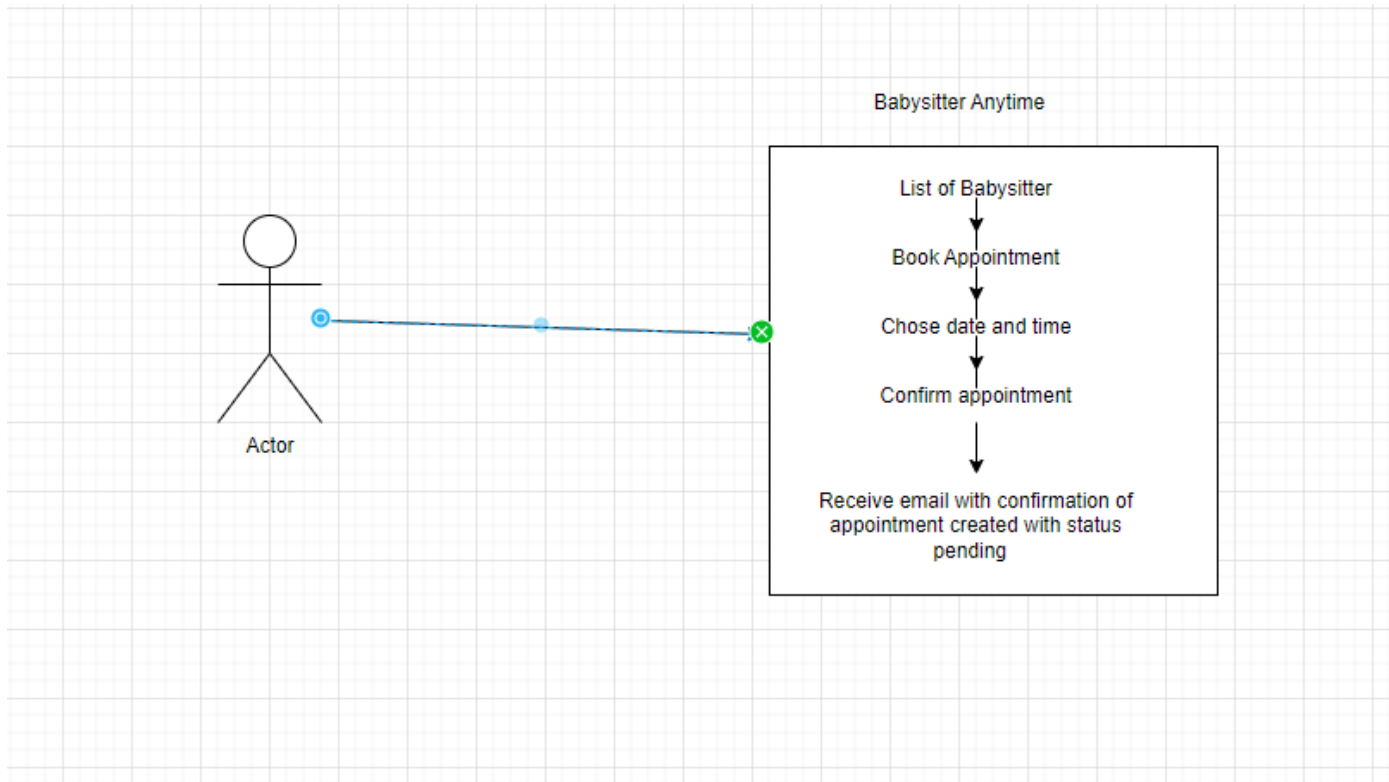
4. View Babysitter Profiles

The babysitter profiles contain their personal information and their contact information.



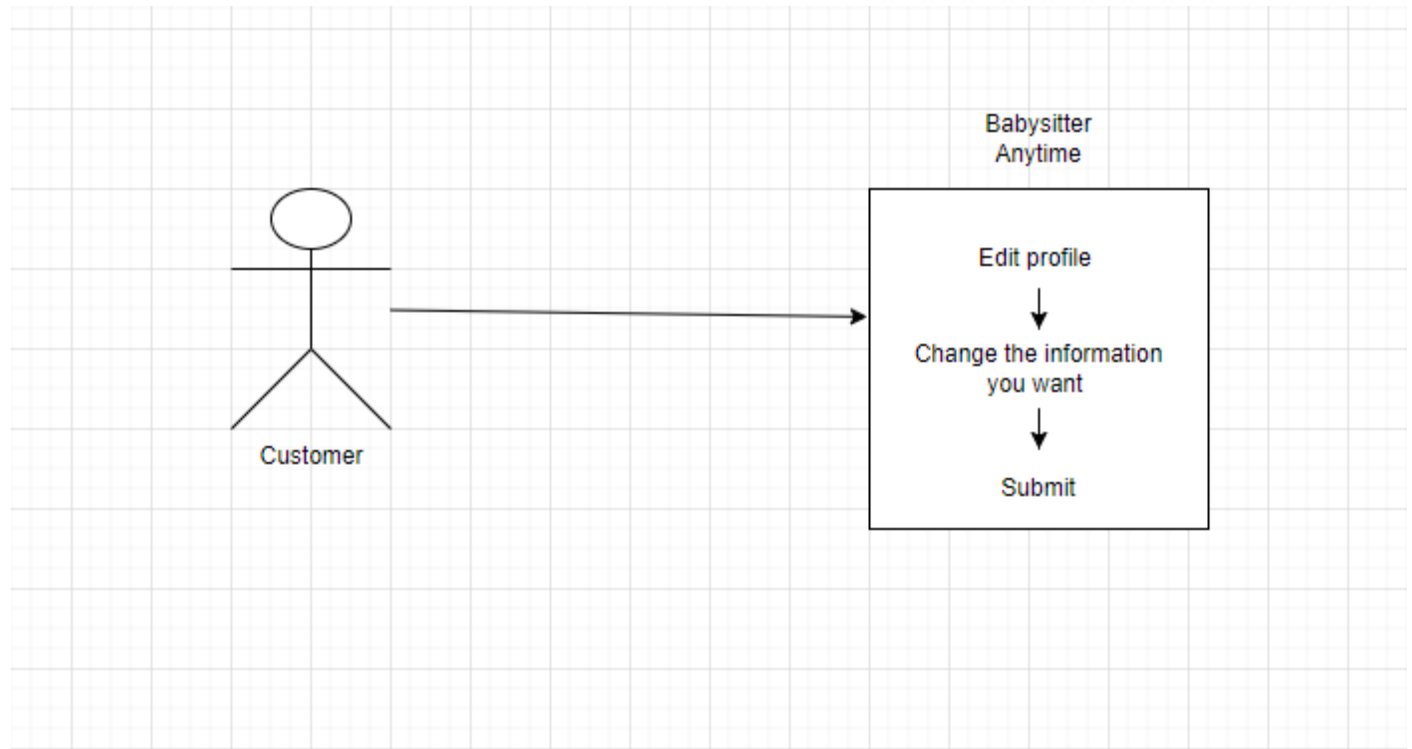
5. Book Babysitter

The customer can book an appointment when clicking on the booking appointment option. The customer can insert the starting date and time of the appointment and the ending date and time of the appointment. On submit, an appointment will be created with a pending status. The customer and the babysitter will both be informed via email with the details of the appointment. The appointment can be accepted or declined by the babysitter.



6. Edit Profile of Customer

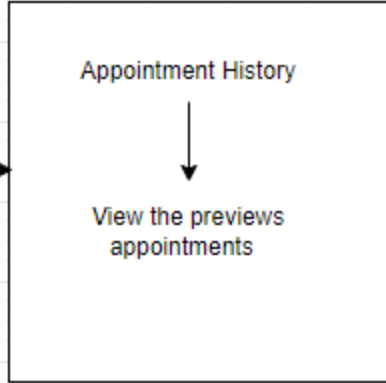
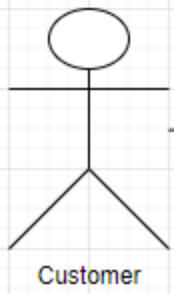
The customer can use this option to change his personal or contact information when needed.



7. View History Appointment of Customer

The customer can use this option to have an overview of his past appointments.

Babysitter Anytime



8. Babysitter Calendar and Manage Appointments

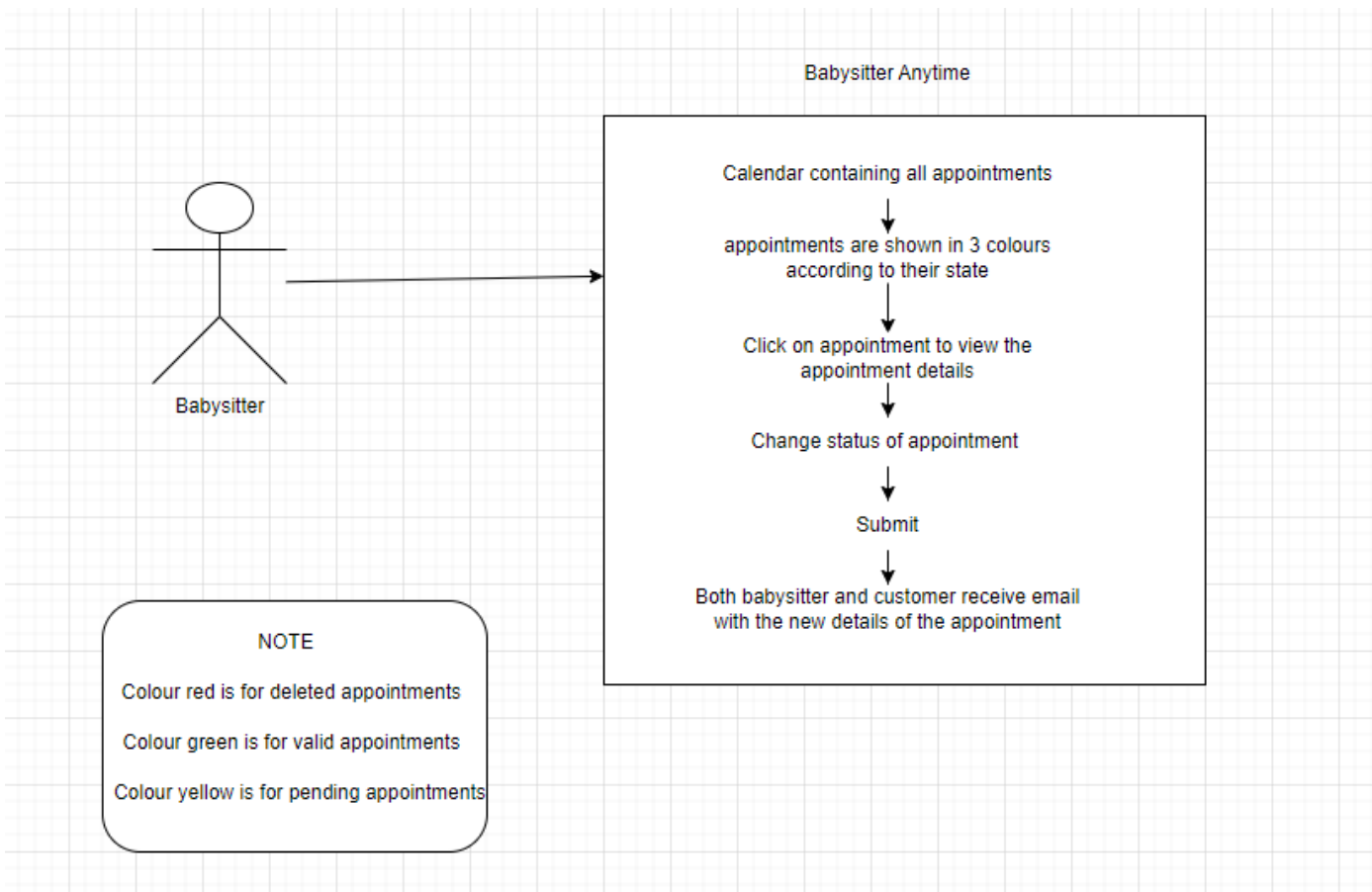
The babysitter's home page consists of a calendar where all his appointments are displayed. The appointments are shown in different colors according to their status.

Green: Approved

Yellow: Pending

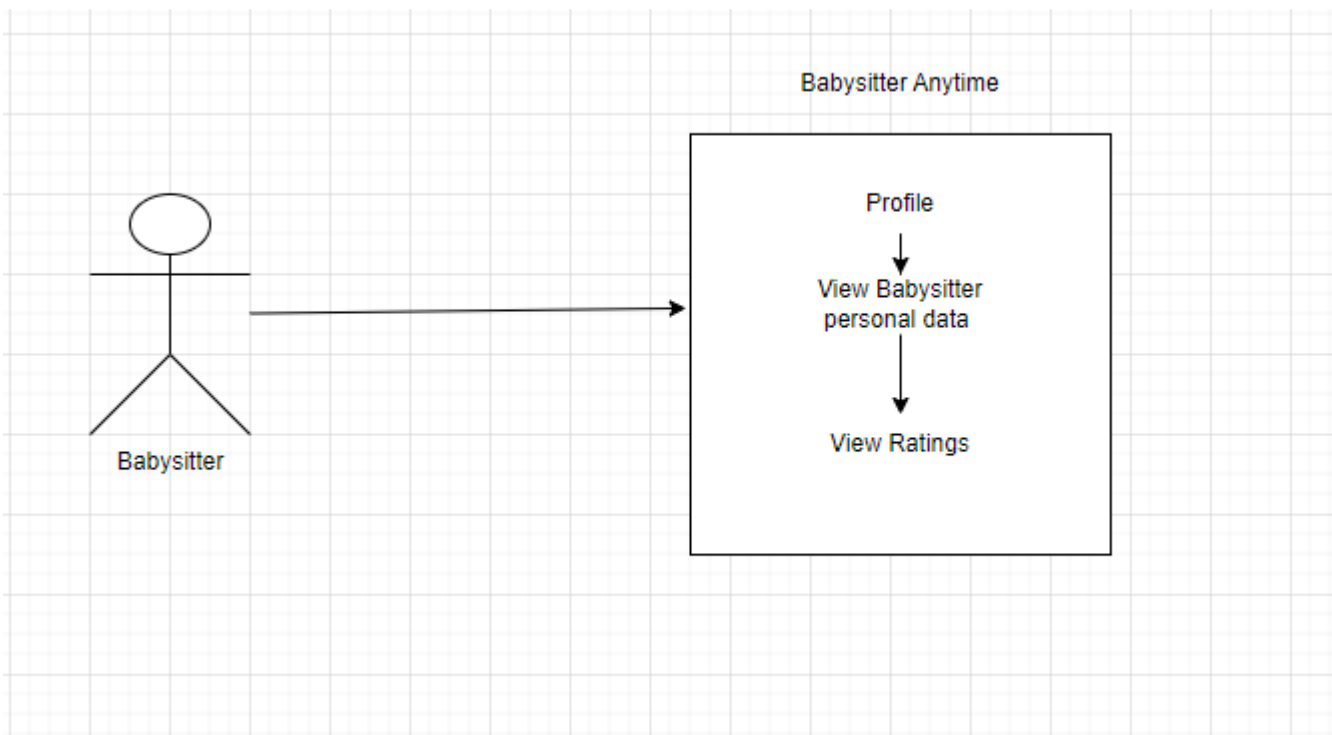
Red: Deleted

Once the babysitter clicks on an appointment he can see the details of the appointment and the status. The babysitter can approve or delete the appointment and submit the changes. Once the changes are submitted, the babysitter and the customer will be notified via email.



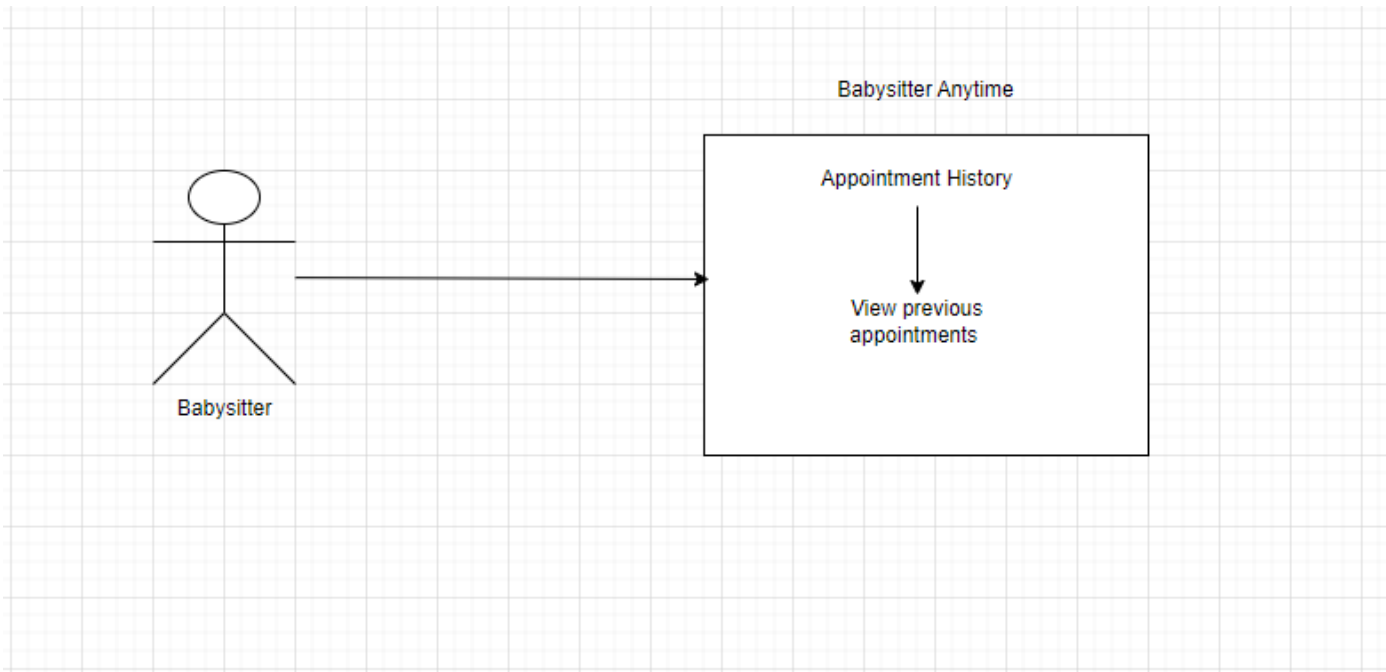
9. View Profile of Babysitter

In this option, the babysitter can take a view of his personal information, his contact information, his reviews as well as the the submitted CV. He will also be presented with the option to add/edit the CV.



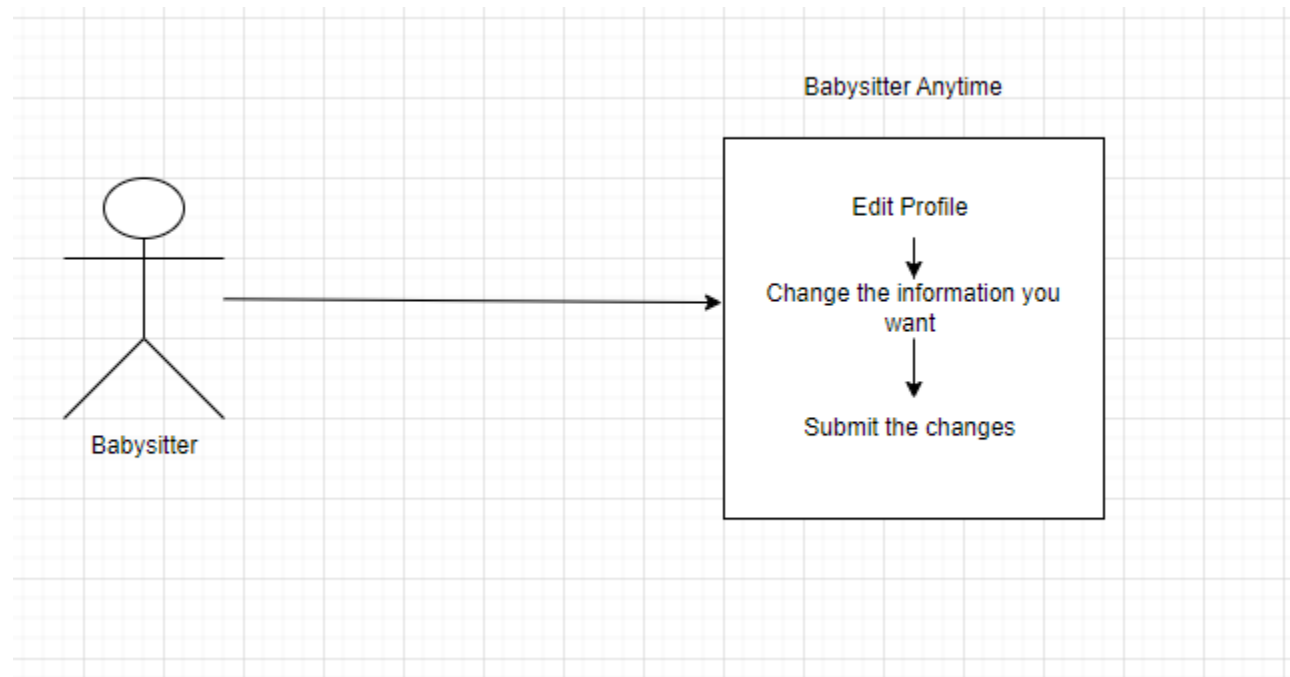
10. View History Appointment of Babysitter

In this option, the babysitter will be presented with his previous appointments.



11. Edit Profile of Babysitter

In this page, the babysitter will be presented with the option to make changes to his personal information or contact information and submit the changes to update the profile.



5. System Design

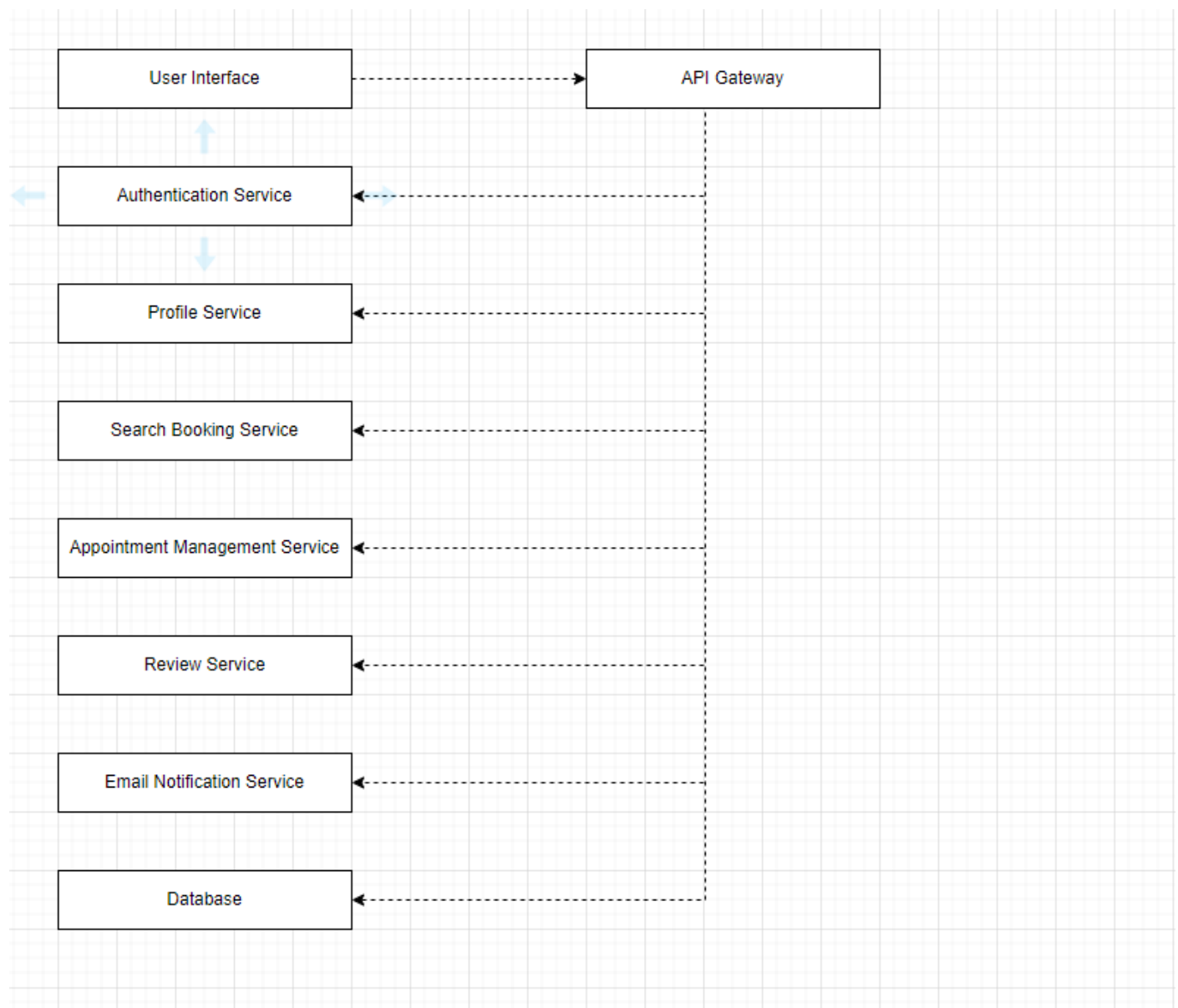
5.1 Description of the Architectural Design of the Software System

The architectural design of the babysitter booking software adheres to a robust client-server model, ensuring a scalable and maintainable system that offers a seamless user experience. The frontend is crafted using React.js, which provides a dynamic and responsive interface for users. React.js enables the creation of reusable components, enhancing the efficiency of the development process and ensuring that the user interface is both intuitive and interactive. The use of HTML and CSS within the React framework ensures that the design is aesthetically pleasing and consistent across different devices and screen sizes. This approach not only improves user engagement but also facilitates easier updates and maintenance of the user interface.

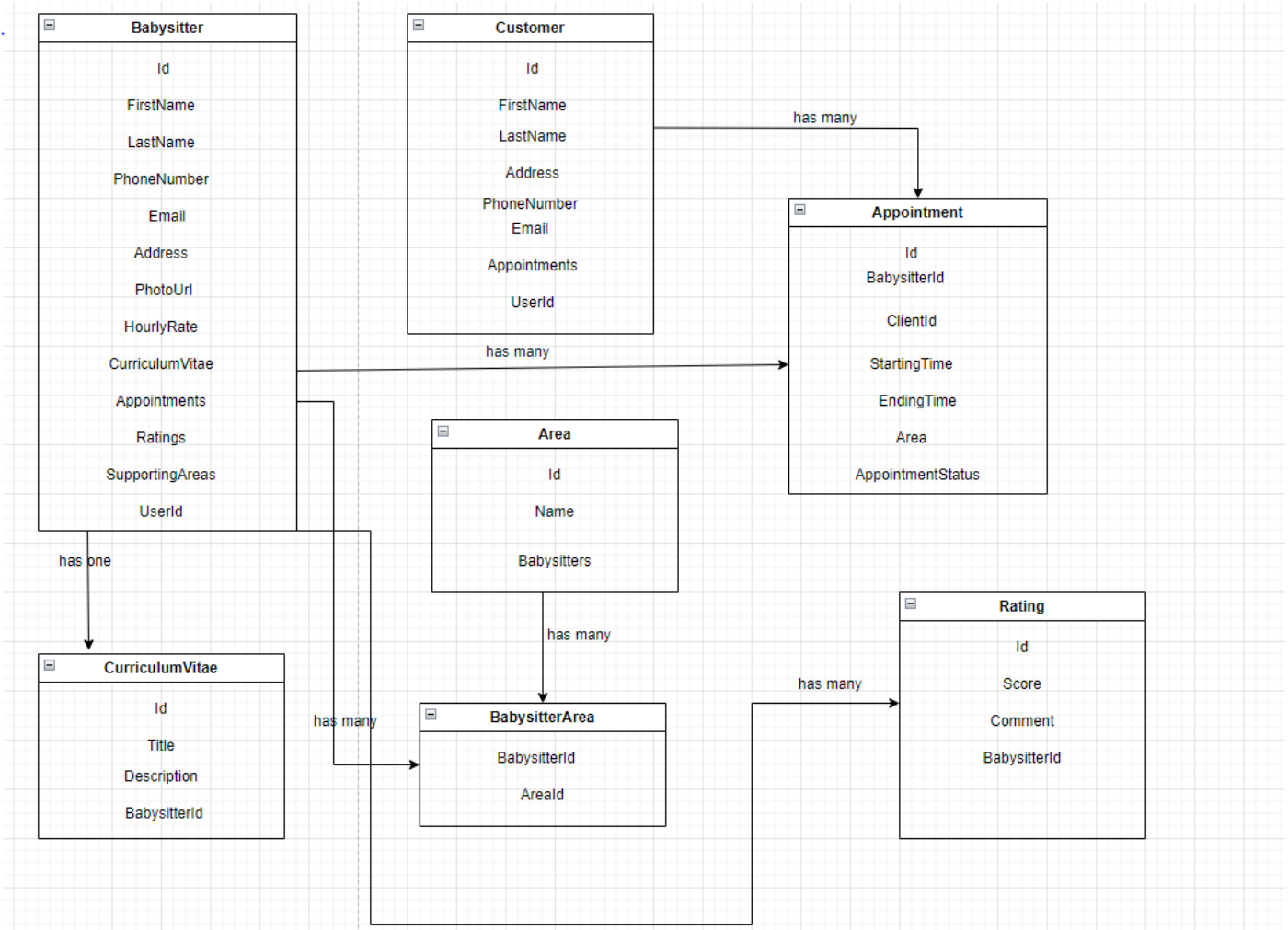
On the server side, the backend is implemented using .NET 8 and C#. This choice leverages the robust capabilities of the .NET framework, known for its performance, reliability, and extensive library support. The backend is responsible for handling the core business logic, processing data, and managing communications between the frontend and the database. By using .NET 8 and C#, the system benefits from strong type safety, a powerful development environment, and seamless integration with various services. This setup ensures that the backend can efficiently manage user requests, process bookings, and handle other critical operations with high performance and reliability.

The database layer is managed using SQL Server Management Studio, which stores all essential data, including user information, booking details, and reviews. SQL Server is chosen for its robustness, scalability, and comprehensive tools for database management. It ensures data integrity, security, and efficient retrieval and storage operations. This three-tier architecture, consisting of a React.js frontend, a .NET 8 backend with C#, and an SQL Server database, promotes a clear separation of concerns. Each layer of the architecture is designed to be independently scalable and maintainable, allowing for efficient updates and modifications as the system evolves. This design not only meets current requirements but also ensures that the software can adapt to future needs and expansions.

5.2 Component Diagrams

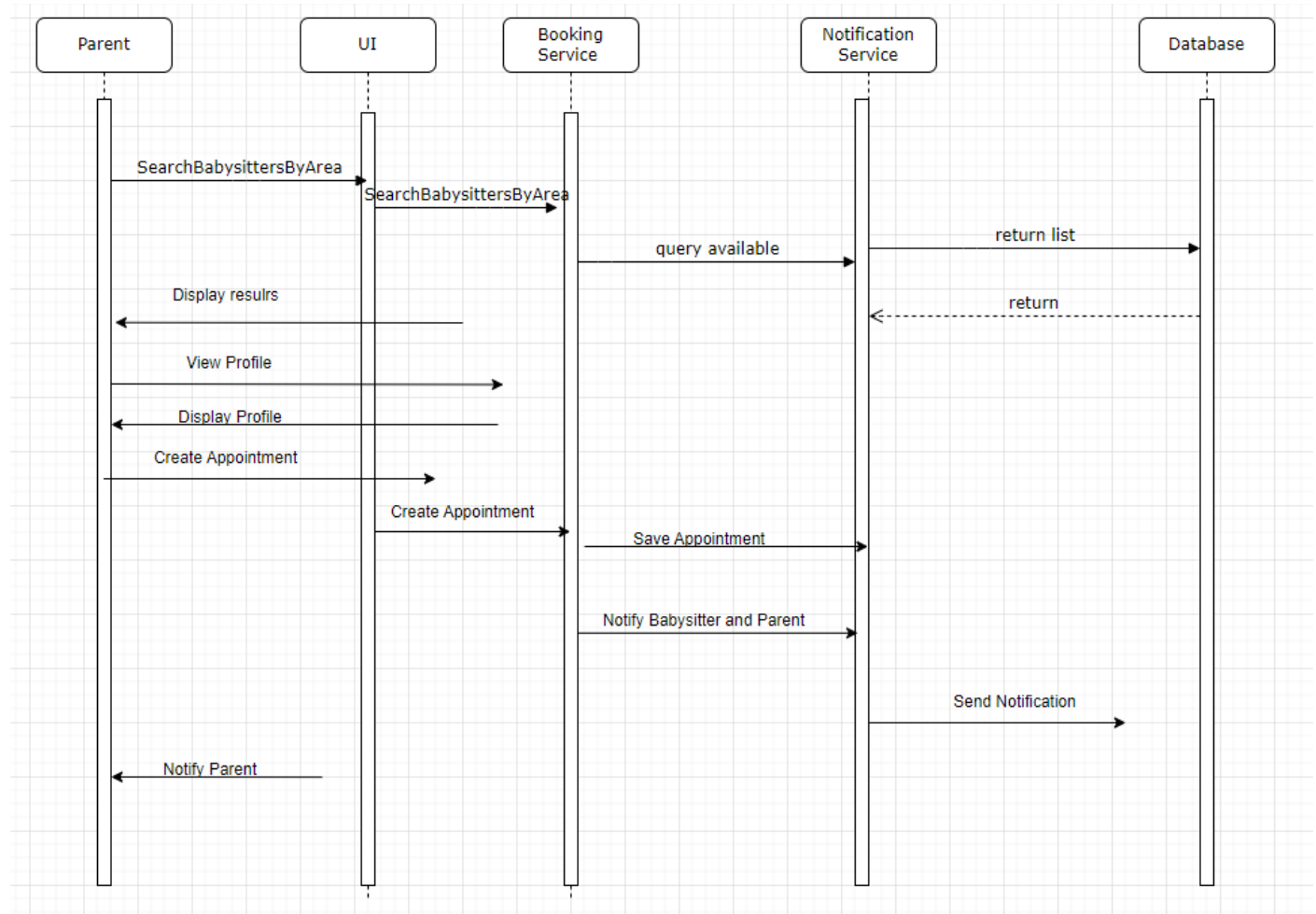


5.3 Class Diagrams

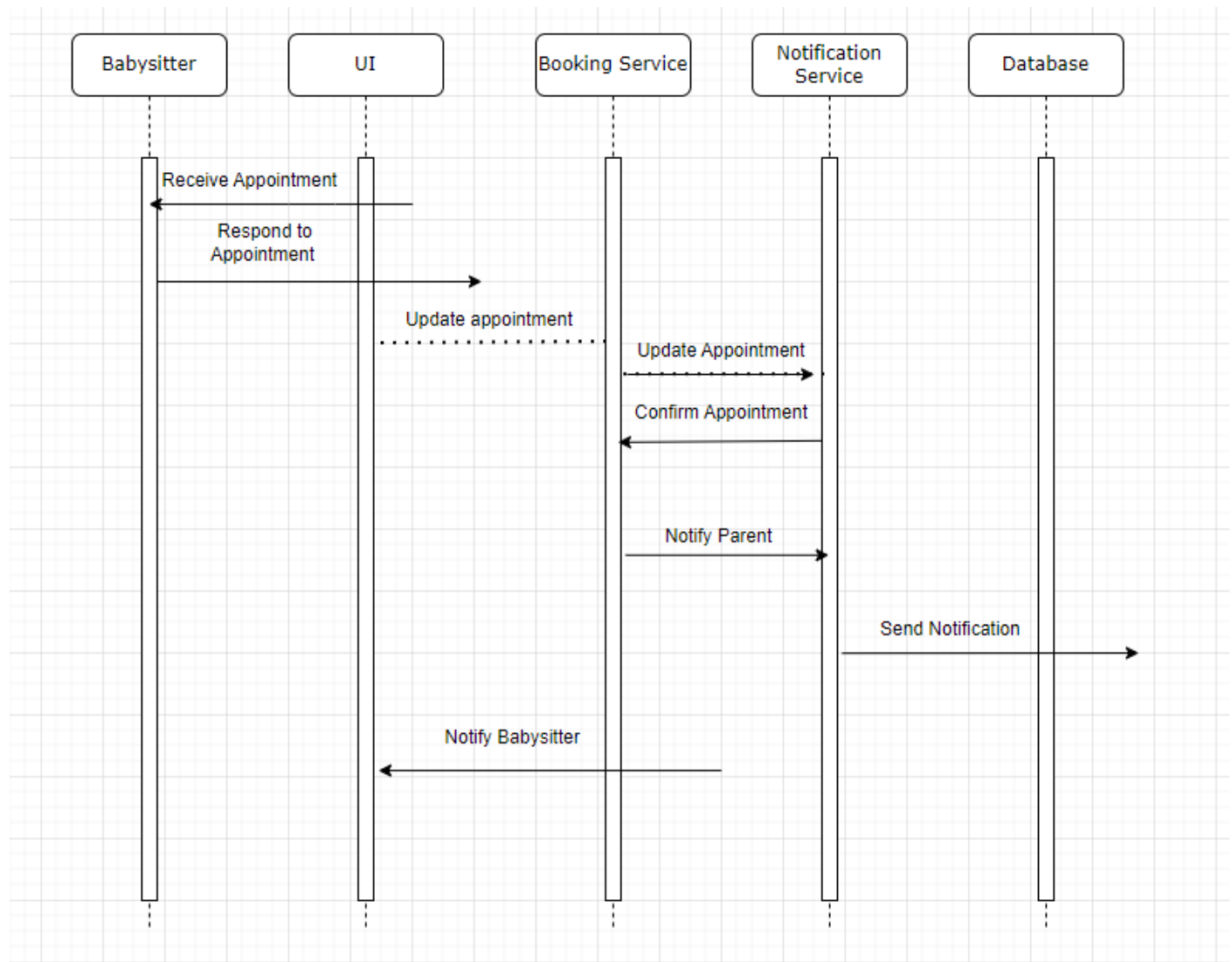


5.4 Sequence Diagrams

Parent Sequence Diagram:



Babysitter Sequence Diagram:



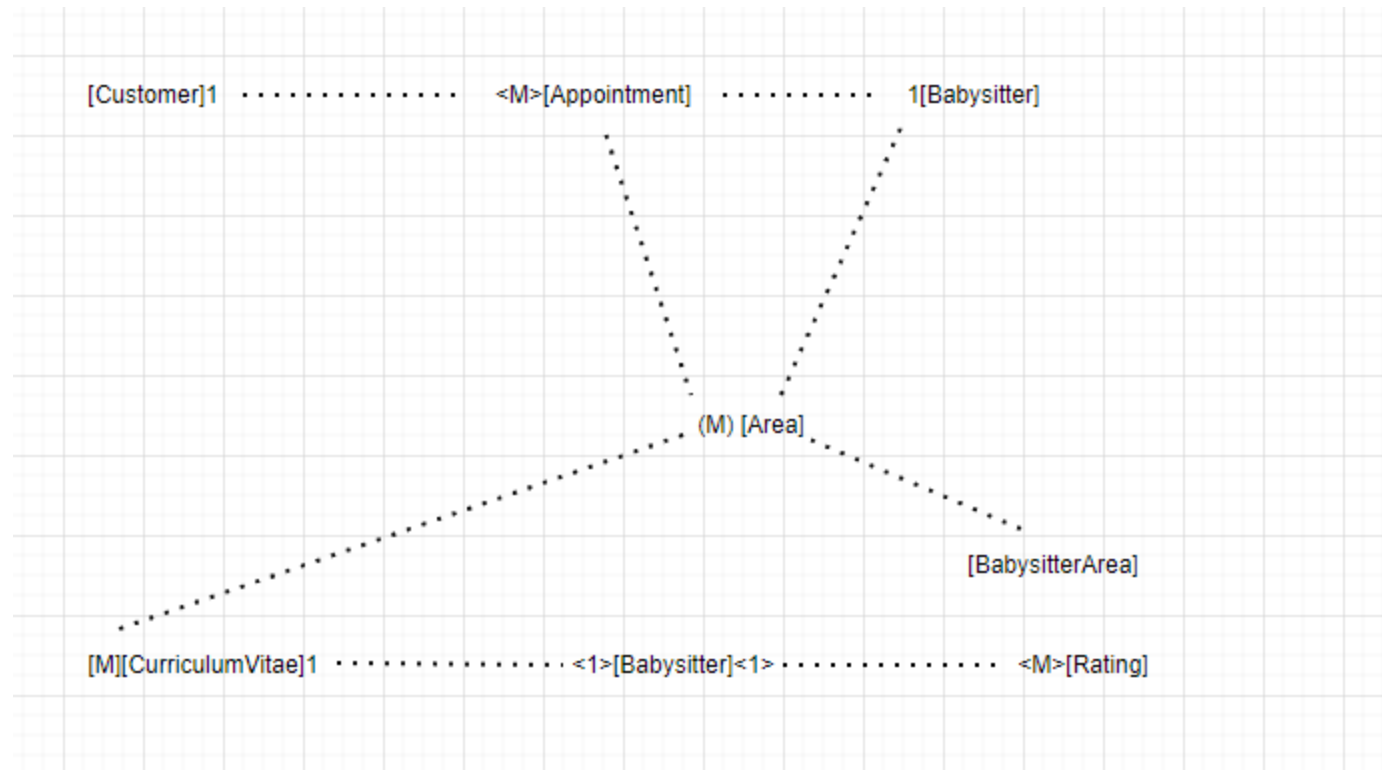
5.5 The Entity-Relationship Diagram (ERD)

1. **Babysitter**
 - Id (Primary Key)
 - FirstName
 - LastName
 - PhoneNumber
 - Email
 - Address
 - PhotoUrl

- HourlyRate
- CurriculumVitae (Foreign Key)
- UserId (Foreign Key)
- 2. **Customer**
 - Id (Primary Key)
 - FirstName
 - LastName
 - Address
 - PhoneNumber
 - Email
 - UserId (Foreign Key)
- 3. **Appointment**
 - Id (Primary Key)
 - BabysitterId (Foreign Key)
 - ClientId (Foreign Key)
 - StartingTime
 - EndingTime
 - Area
 - AppointmentStatus
- 4. **Area**
 - Id (Primary Key)
 - Name
- 5. **BabysitterArea**
 - BabysitterId (Foreign Key)
 - AreaId (Foreign Key)
- 6. **CurriculumVitae**
 - Id (Primary Key)
 - Title
 - Description
 - BabysitterId (Foreign Key)
- 7. **Rating**
 - Id (Primary Key)
 - Score
 - Comment
 - BabysitterId (Foreign Key)

Relationships

- **Babysitter** has many **Appointments**
- **Customer** has many **Appointments**
- **Area** has many **Babysitters** through **BabysitterArea**
- **Babysitter** has one **CurriculumVitae**
- **Babysitter** has many **Ratings**



6. Implementation

6.1 Detailing the Implementation Process

The implementation process for the babysitter booking software followed a structured approach to ensure code quality and maintainability. Utilizing best coding practices, I used Git for version control and hosted the repository on GitHub, adhering to the Git Flow workflow. This methodology allowed for efficient management of changes through feature branches and pull requests, facilitating a smooth development process. Although formal testing was not conducted, code reviews and manual testing were performed to identify and fix any immediate issues.

6.2 Showcasing Specific Features and Functionalities

A key feature of the software is the user registration and authentication system. Users can register by completing a form that includes client-side validation for all input fields. Upon registration, an email verification link is sent to the user to confirm their email address, enhancing security. Passwords are securely encrypted using SHA-256 before being stored in the database. The authentication process uses JWTs (JSON Web Tokens) to manage user sessions securely and efficiently.

The search functionality enables parents to find babysitters based on their location and availability. Parents can book a babysitter by selecting their preferred date and time, and both parties receive a confirmation notification. Additionally, an integrated messaging system allows parents and babysitters to communicate directly within the platform, ensuring that all interactions are securely documented and easily accessible.

6.3 Discussing Challenges and Solutions

One significant challenge encountered during implementation was ensuring efficient search functionality without external APIs. This was addressed by implementing a robust database query system that optimizes search results based on location and availability. Ensuring quick and relevant search results involved fine-tuning database indexes and optimizing queries.

Usability testing indicated that some users found the profile creation process to be cumbersome. To address this, I simplified the form fields and added tooltips to provide users with guidance throughout the process.

By detailing the implementation process, showcasing specific features, and discussing challenges and their solutions, this section offers a comprehensive overview of how the babysitter booking software was developed and refined to meet user needs and ensure a high-quality, secure, and accessible platform.

7. Testing and Evaluation

7.1 Explanation of Testing Strategies

To ensure the quality and reliability of the babysitter booking software, a combination of manual testing and user feedback was employed. Manual testing involved systematically testing each feature, such as user registration, profile creation, and the booking system. Issues were documented in a shared spreadsheet and tracked until resolved. While formal automated testing was not conducted, I relied on rigorous manual testing to identify and fix issues. Additionally, informal user feedback was gathered from a small group of parents and babysitters to evaluate the software's usability and functionality in real-world scenarios.

7.2 Presentation of Testing Results

During manual testing, several critical bugs were identified, including issues with the booking confirmation process and profile updates. These bugs were promptly fixed, and retesting confirmed their resolution. User feedback highlighted the need for a more streamlined profile creation process and improved search filtering options. Based on this

feedback, enhancements were made to simplify the profile setup and refine the search functionality, ensuring a more user-friendly experience. The mail notification system also faced initial challenges due to the integration of Gmail's two-factor authentication. However, these issues were resolved by configuring the application to securely send emails using the appropriate authentication methods.

7.3 Evaluation Against Requirements

An evaluation of the software against the defined functional requirements showed that most features, such as user registration, search functionality, and the booking system, were fully implemented and functional. The mail notification system, which sends confirmation and notification emails to users, faced initial difficulties but was successfully integrated and tested. Non-functional requirements, such as performance and security, were generally met. Usability testing revealed the need for additional improvements in the user interface design to enhance accessibility and ease of use, aligning the software more closely with user expectations and requirements.

By detailing the testing strategies, presenting the results, and evaluating the software against the requirements, this section provides a thorough overview of the testing and evaluation process, highlighting both successes and areas for future improvement.

8. Results and Analysis

8.1 Summary of Key Findings

Through extensive testing and evaluation, we uncovered several key insights into the usability, performance, and scalability of our babysitter booking software. Users highlighted the need for improvements in the search functionality, while performance testing revealed latency issues during peak usage times. Scalability testing also identified potential bottlenecks in database queries, signaling areas for future optimization and enhancement.

8.2 Discussion of Usability, Performance, and Scalability

Overall, our software demonstrated good usability, with users appreciating its easy navigation. However, we noted that the search process could be more user-friendly, prompting us to explore ways to simplify and streamline this aspect of the platform. Performance testing showed acceptable response times under normal conditions, though there were delays observed during peak hours, indicating the importance of further optimization efforts to ensure consistent performance across varying usage scenarios. Scalability testing suggested potential challenges as our user base grows, particularly in managing concurrent bookings and efficiently storing and retrieving data.

These findings underscore the importance of implementing scalable infrastructure and robust database management strategies to support future growth and maintain optimal system performance.

8.3 Comparison with Existing Solutions

Compared to other options in the market, our software offers some distinct advantages, such as a simplified booking process and integrated messaging features, which enhance communication between parents and babysitters. Yet, we acknowledge limitations, including performance issues during peak times and scalability concerns with database management. Despite these challenges, our software presents a valuable solution to the ongoing challenge of finding reliable childcare services. We are committed to addressing the identified issues and enhancing the overall user experience, particularly in improving the search process to make it more intuitive and efficient for users.

9. Future Work

9.1 Potential Enhancements or Features

In upcoming iterations, our software could undergo exciting enhancements to further refine the user experience. One major improvement could involve implementing sophisticated matching algorithms. These algorithms would analyze various factors like babysitter availability, skill set, and user preferences to create more tailored matches between parents and babysitters, enhancing satisfaction on both ends. Additionally, integrating advanced communication tools such as video calling features could facilitate smoother interactions between parents and babysitters. Real-time video calls would enable a deeper connection, helping parents gauge the suitability of a babysitter before booking. Furthermore, incorporating a secure payment system directly within the platform would streamline transactions, ensuring seamless and hassle-free payments for all parties involved. Lastly, developing a dedicated mobile application would enhance accessibility, allowing users to access the platform on-the-go, thereby increasing convenience and engagement.

9.2 Areas for Further Research or Expansion

Areas for further exploration include localization efforts to adapt the platform to different cultural and linguistic contexts, expanding its reach to a more diverse user base. Strengthening safety and security measures remains crucial, potentially through identity verification protocols and enhanced background checks. Moreover, delving into user behavior and preferences through data analytics could offer valuable insights for optimizing features and enhancing user satisfaction. Additionally, fostering community-

building features and partnerships with childcare services could enrich the platform's value proposition, promoting user engagement and providing a comprehensive solution for childcare needs."

These potential enhancements and areas for further research not only promise to improve the platform's functionality but also underscore its potential to make a significant impact on the babysitting industry and childcare services as a whole.

9.3 Potential Impact on the Babysitting Industry and Childcare Services

The potential impact of these enhancements extends beyond mere convenience, promising to reshape the landscape of the babysitting industry and childcare services as a whole. By improving access to childcare services for parents, the platform empowers them to find reliable and trustworthy babysitters with greater ease and efficiency. This, in turn, enables parents to better balance their personal and professional lives, fostering greater well-being and productivity.

Furthermore, by increasing the visibility and earning potential of babysitters, the platform elevates their status within the childcare industry, recognizing their invaluable contributions and expertise. Through the platform, babysitters gain access to a broader pool of potential clients, allowing them to expand their clientele and build their reputation as trusted child care providers. This not only benefits individual babysitters but also contributes to raising standards across the industry, promoting professionalism and excellence in childcare services.

In summary, the software's potential impact on the babysitting industry and childcare services is significant, promising to foster a culture of trust, transparency, and quality. By leveraging technology to enhance accessibility, communication, and safety, the platform has the potential to revolutionize the way parents and babysitters connect and interact, ultimately improving the overall childcare experience for families.

10. Conclusion

10.1 Recap of Main Contributions

This thesis has significantly contributed to the childcare services domain through the development of a robust babysitter booking software. The software serves as a user-friendly platform facilitating seamless connections between parents and babysitters. Key contributions include the implementation of features such as profile creation, search functionality, and booking management, all while prioritizing security and transparency for users.

10.2 Reflection on Challenges and Lessons Learned

Throughout the project, various challenges were encountered, ranging from technical complexities to nuanced considerations of user needs. Technical hurdles, such as integrating real-time communication and implementing secure payment systems, required innovative solutions. Additionally, understanding and accommodating the diverse requirements of parents and babysitters necessitated careful attention to detail and iterative design processes.

Reflecting on these challenges, several invaluable lessons have been learned. Firstly, the centrality of user feedback in shaping effective solutions cannot be overstated. Iterative design based on continuous user input proved instrumental in refining the software and ensuring its usability. Secondly, effective project management practices, including clear communication and agile methodologies, were essential in overcoming obstacles and meeting project milestones.

10.3 Recommendations for Practitioners and Researchers

For practitioners in the childcare industry and researchers, several recommendations emerge from this work. Firstly, investing in user-centered design principles and rigorous testing methodologies is crucial for the successful development and deployment of childcare software solutions. Additionally, fostering collaboration between stakeholders, including parents, babysitters, and software developers, can lead to more tailored and effective solutions that meet the diverse needs of end-users. Finally, continued research into emerging technologies, such as AI-driven matching algorithms and IoT integration for childcare monitoring, holds promise for further enhancing the efficacy and accessibility of childcare services.

In conclusion, this thesis represents a significant step forward in leveraging technology to address the challenges faced by parents and babysitters in the childcare industry. By recapitulating the main contributions, reflecting on challenges and lessons learned, and offering recommendations for practitioners and researchers, this work aims to inspire continued innovation and improvement in the field of childcare services.

11. References

11.1 Software Documentation

- React. (n.d.). React – A JavaScript library for building user interfaces. Retrieved from <https://reactjs.org/docs/getting-started.html>
- Microsoft. (n.d.). C# documentation. Retrieved from <https://learn.microsoft.com/en-us/dotnet/csharp/>

- Microsoft. (n.d.). ASP.NET Core documentation. Retrieved from <https://learn.microsoft.com/en-us/aspnet/core/>
- Microsoft. (n.d.). .NET 8 documentation. Retrieved from <https://learn.microsoft.com/en-us/dotnet/core/dotnet-eight>

11. 2 Websites and Online Resources

- Stack Overflow. (n.d.). Questions tagged [reactjs]. Retrieved from <https://stackoverflow.com/questions/tagged/reactjs>
- GitHub. (n.d.). React repository. Retrieved from <https://github.com/facebook/react>
- FreeCodeCamp. (n.d.). Learn C# – Free Interactive C# Tutorial. Retrieved from <https://www.freecodecamp.org/news/learn-c-sharp-free-interactive-c-tutorial/>

11.3 Inspiration and Case Studies

- Doctoranytime. (n.d.). Doctoranytime - Find a Doctor Near You. Retrieved from <https://www.doctoranytime.gr/en>

12. Appendices

12.1 Code Snippets

12.1.1 Searching for Babysitters by supporting areas

```
2 references
public async Task<List<Babysitter>> GetBabysittersBySupportingArea(Guid areald)
{
    var babysitters = await _appDbContext.BabysitterArea
        .Where(ba => ba.Areald == areald)
        .Select(ba => ba.Babysitter)
        .ToListAsync();

    return babysitters;
}
```

12.1.2 Create Appointment

```
2 references
public async Task CreateAppointment(Appointment appointment)
{
    _appDbContext.Appointments.Add(appointment);
    await _appDbContext.SaveChangesAsync();

    // Re-fetch the appointment with related data
    appointment = _appDbContext.Appointments
        .Include(a => a.Babysitter)
        .Include(a => a.Client)
        .FirstOrDefault(a => a.Id == appointment.Id);

    await NotifyAppointmentDetails(appointment, true);
}
```

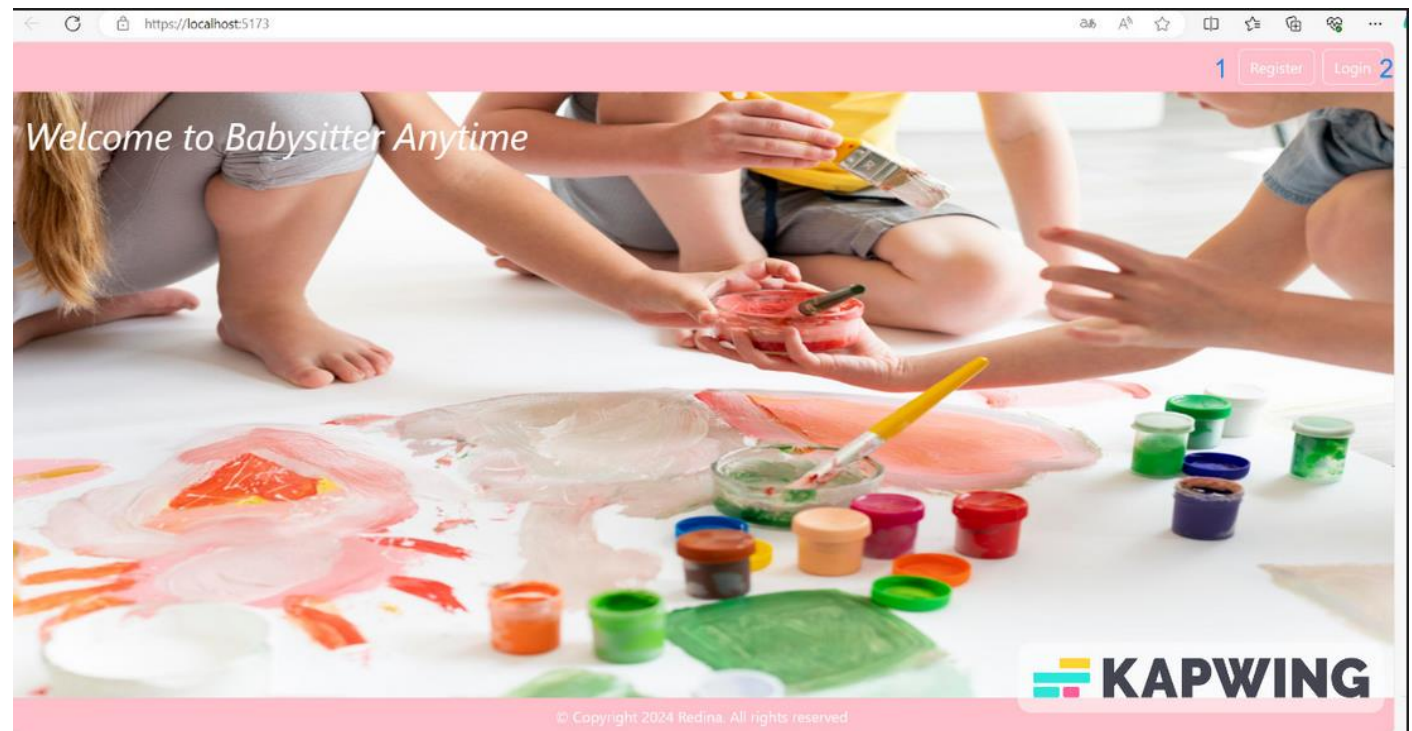
12.1.3 Notify Customer and Babysitter

```
< references
69 private async Task NotifyAppointmentDetails(Appointment appointment, bool isNewAppointment)
70 {
71     var babysitterEmail = appointment.Babysitter.Email; // Assuming Email property exists
72     var clientEmail = appointment.Client.Email; // Assuming Email property exists
73     var subject = isNewAppointment ? "New Appointment Created" : "Appointment Updated";
74     var body = $"{@
75     Hello,
76
77     An appointment has been {(isNewAppointment ? "created" : "updated")} with the following details:
78
79     - Starting Time: {appointment.StartingTime}
80     - Ending Time: {appointment.EndingTime}
81     - Area: {appointment.Area}
82     - Status: {appointment.AppointmentStatus}
83
84     Thank you.
85     ";
86
87     await _emailService.SendEmailAsync(babysitterEmail, subject, body);
88     await _emailService.SendEmailAsync(clientEmail, subject, body);
89 }
90 }
```

12.2 User Manual

12.2.1 Homepage, Register and Login

12.2.1.1 Homepage



In the Homepage the user is presented with the options:

1. Register
2. Login

12.2.1.2 Register

Welcome to Babysitter Anytime

Register

- 1 Username
- 2 Password
10 characters with a numeric, character and capital.
- 3 Role
Customer
- 4 Register

© Copyright 2024 Redina. All rights reserved

In order to achieve a successful registration the user must provide:

1. Username
2. A password that must be more than 10 character, and should have at least one capital letter, one special character and one numeric.
3. Their role (Customer/Babysitter)
4. After this information is provided, the user can click on the 'Register' button and complete the registration.

12.2.1.3 Create Profile

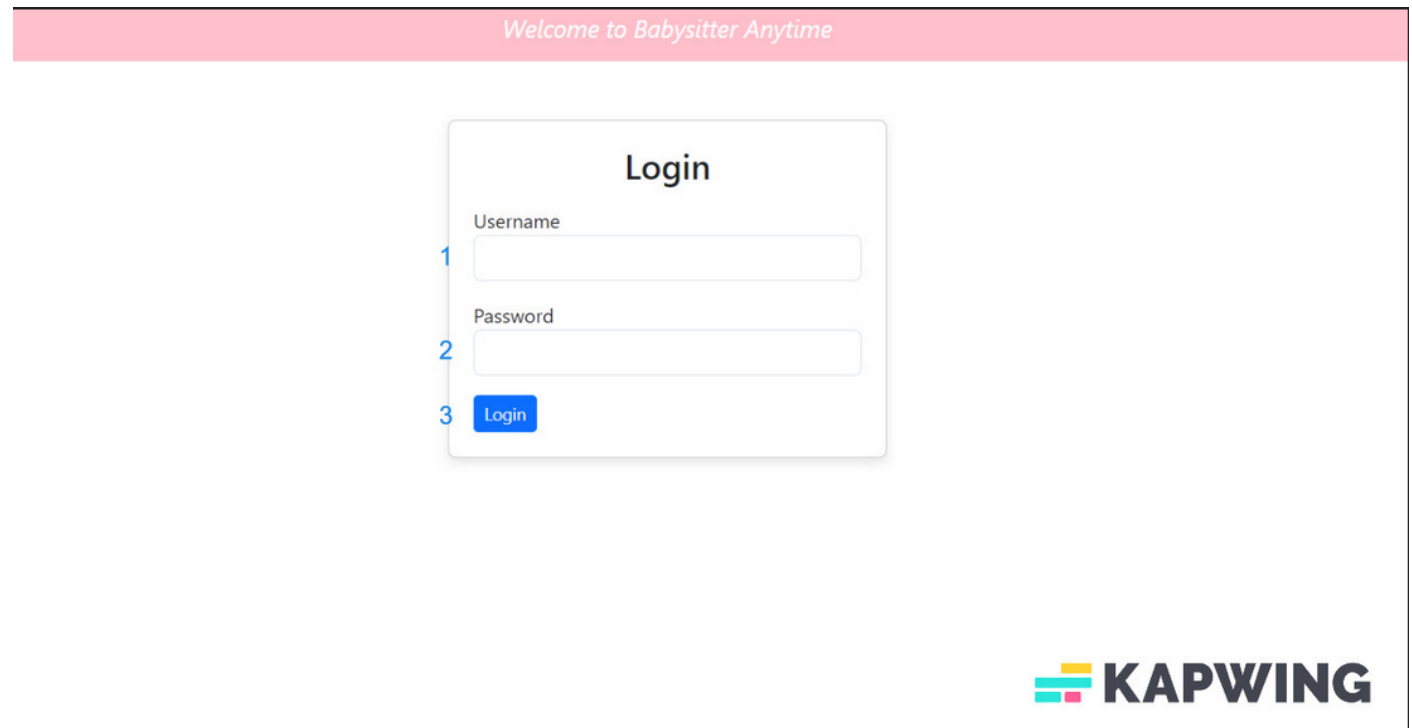
Create Your Profile

- 1 First Name
- 2 Last Name
- 3 Address
- 4 Phone Number
- 5 Email
- 6 Create Profile

After a successful registration the user is prompted to create his profile. In order to complete this action the user must insert:

1. First name
2. Last name
3. Address
4. Phone number
5. Email
6. After this information is inserted, the user can click on the 'Create Profile' button and complete the action

12.2.1.4 Login



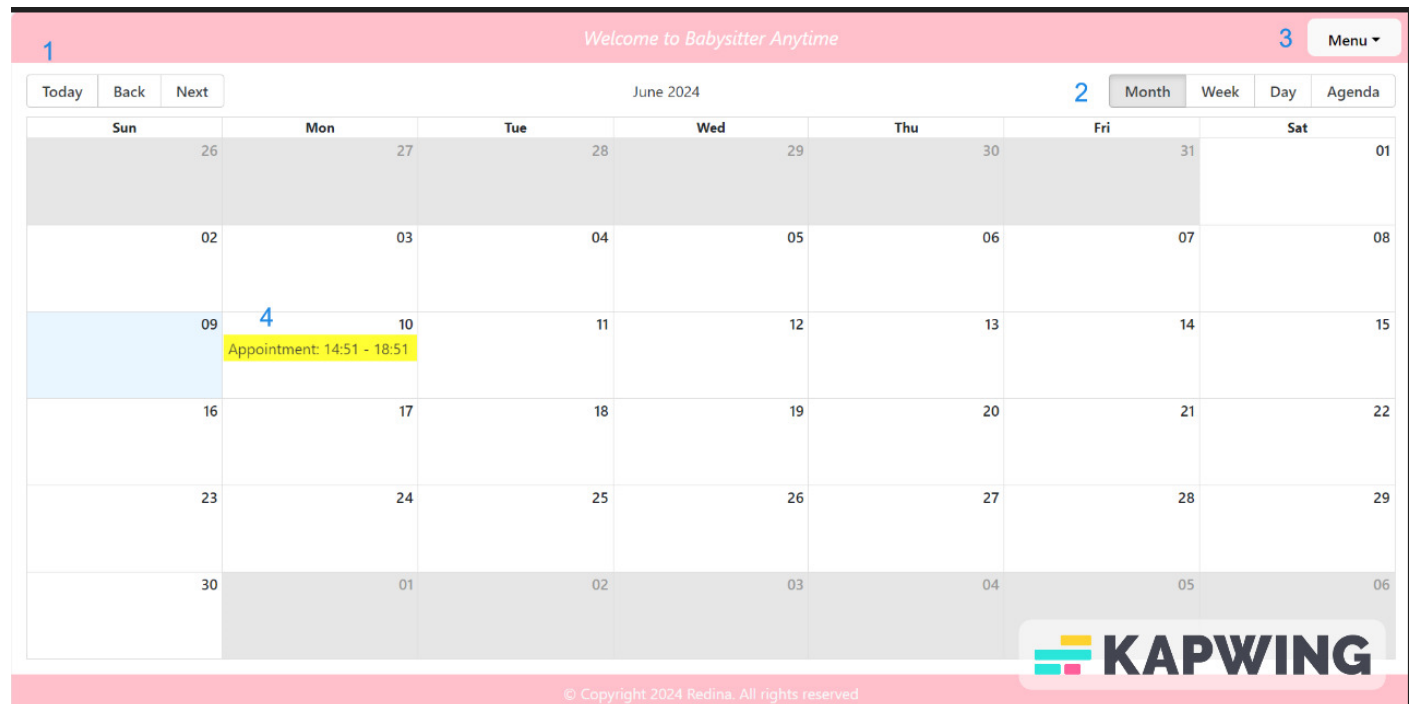
The screenshot shows a web page with a pink header bar containing the text "Welcome to Babysitter Anytime". Below the header is a white login form titled "Login". The form contains three elements: a "Username" input field with a blue "1" to its left, a "Password" input field with a blue "2" to its left, and a blue "Login" button with a blue "3" to its left. The KAPWING logo is visible in the bottom right corner of the page.

If a user already has an account, he can log in by using the Login option. In order to complete a successful login the user must provide:

1. Username
2. Password
3. Click on the 'Login' button and complete this action

12.2.2 Babysitter Manual

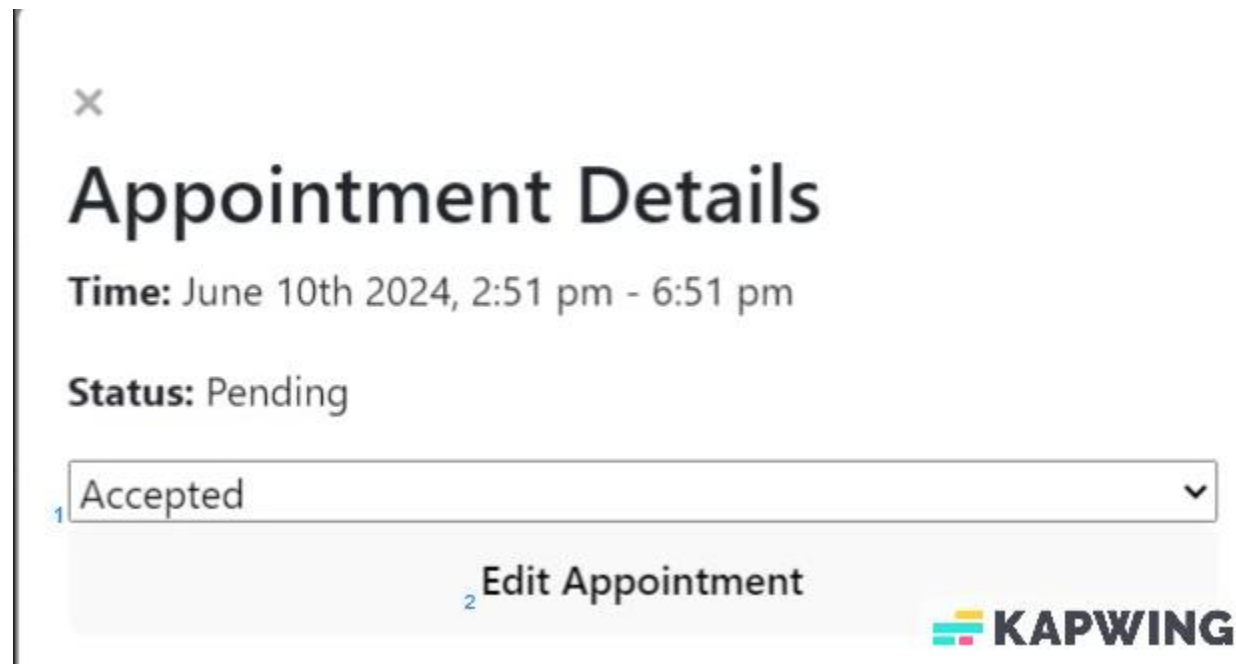
12.2.2.1 Babysitter Home page



The babysitter's home page features a comprehensive calendar with several key navigation options:

- 1. Calendar Navigation:**
 - **Today:** Displays the current day's calendar.
 - **Back:** Navigates to the previous month.
 - **Next:** Navigates to the next month.
- 2. Viewing Options:** Allows the user to switch between different calendar views:
 - **Month View:** Displays the entire month.
 - **Week View:** Displays the week's schedule.
 - **Day View:** Displays the day's schedule.
 - **Agenda View:** Displays a list of scheduled appointments.
- 3. Menu Access:** Expands to reveal various application options and settings.
- 4. Appointment Requests:** Shows examples of how appointment requests are displayed to the user.

12.2.2.2 Edit Appointment Status



×


Appointment Details

Time: June 10th 2024, 2:51 pm - 6:51 pm

Status: Pending

1 Accepted

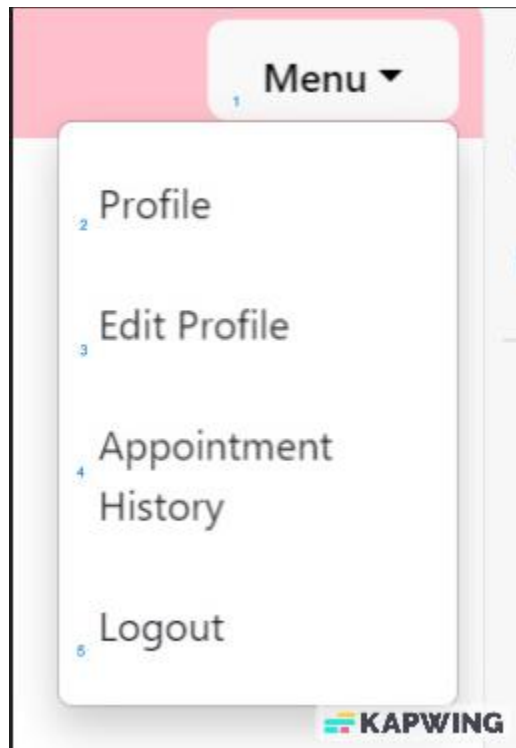
2 Edit Appointment



When the user clicks on an appointment, the details of the appointment are displayed.

- 1. Status Dropdown:**
 - There is a dropdown menu containing the available statuses for the appointment.
 - The user can change the status of the appointment by selecting a new option from the dropdown.
- 2. Edit Appointment Button:**
 - To finalize any changes, the user must click the "Edit Appointment" button.

12.2.2.3 Babysitter Menu



1. **Menu**
2. **Profile:** Allows the user to see their profile.
3. **Edit Profile:** Enables the user to edit their profile details.
4. **Appointment History:** Displays the user's appointment history.
5. **Logout:** Logs the user out of the system.

12.2.2.4 Babysitter Profile

Welcome to Babysitter Anytime

Menu

Babysitter Profile

Curriculum Vitae

2 Title: Test CV 7

Description: Testy testy CV 7

3 Edit CV

Score: 3.5

4 Show Comments

1

First Name: Redina

Last Name: Redi

Phone Number: 6262626262

Email: redtzaferai@gmail.com

Address: TEstt

Hourly Rate: 30

© Copyright 2024 Redina, All rights reserved

KAPWING

In the babysitter profile page the user is able to see:

1. Personal and contact information
2. CV details
3. Option to modify CV
4. Option to view comments and ratings

X

Edit Curriculum Vitae

1 Test CV 7

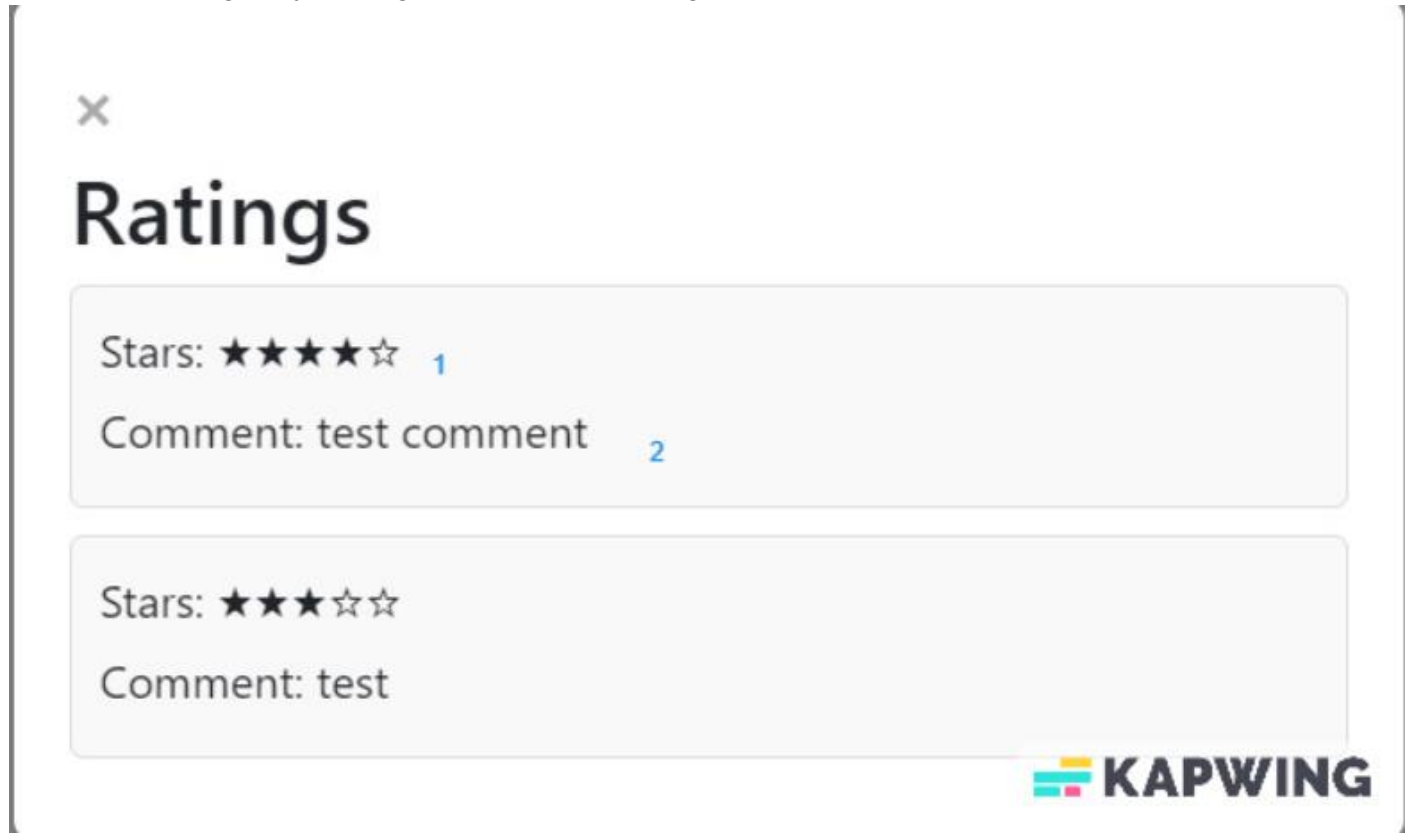
2 Testy testy CV 7

2 Submit Changes

KAPWING

In the Edit Curriculum Vitae option the user can:

1. Change the title of the CV and the details.
2. Submit the changes by clicking on the 'Submit Changes' button.



In the Ratings option, the user can see:

1. The score the customer has left them in stars
2. The comment the customer has left them

12.2.2.5 Babysitter Edit Profile

Welcome to Babysitter Anytime Menu ▾

Edit Profile


1

First Name: Redina Last Name: Redi

Phone Number: 6262626262 Email: redtzaferai@gmail.com

Address: TEstt Hourly Rate: .30

Photo: Επιλογή αρχείου Δεν έχει επιλεγεί αρχείο



2

Update Profile

© Copyright 2024 Redina. All rights reserved. **KAPWING**

On the Edit Profile for Babysitter option the user is able to:

1. Look at their current information and change them
2. Submit the changes by clicking the 'Update Profile' button

12.2.2.6 Babysitter Appointment History

Welcome to Babysitter Anytime Menu ▾

Appointments history

1

	Start Time	End Time	Status
1	9/5/2024, 1:35:00 π.μ.	9/5/2024, 3:35:00 π.μ.	Completed
2	8/5/2024, 5:43:00 μ.μ.	8/5/2024, 7:43:00 μ.μ.	Pending
3	8/5/2024, 5:57:00 μ.μ.	8/5/2024, 7:57:00 μ.μ.	Pending
4	19/5/2024, 8:44:00 μ.μ.	18/5/2024, 10:45:00 π.μ.	Completed
5	19/5/2024, 8:44:00 μ.μ.	18/5/2024, 10:45:00 π.μ.	Completed
6	10/6/2024, 2:51:00 μ.μ.	10/6/2024, 6:51:00 μ.μ.	Completed

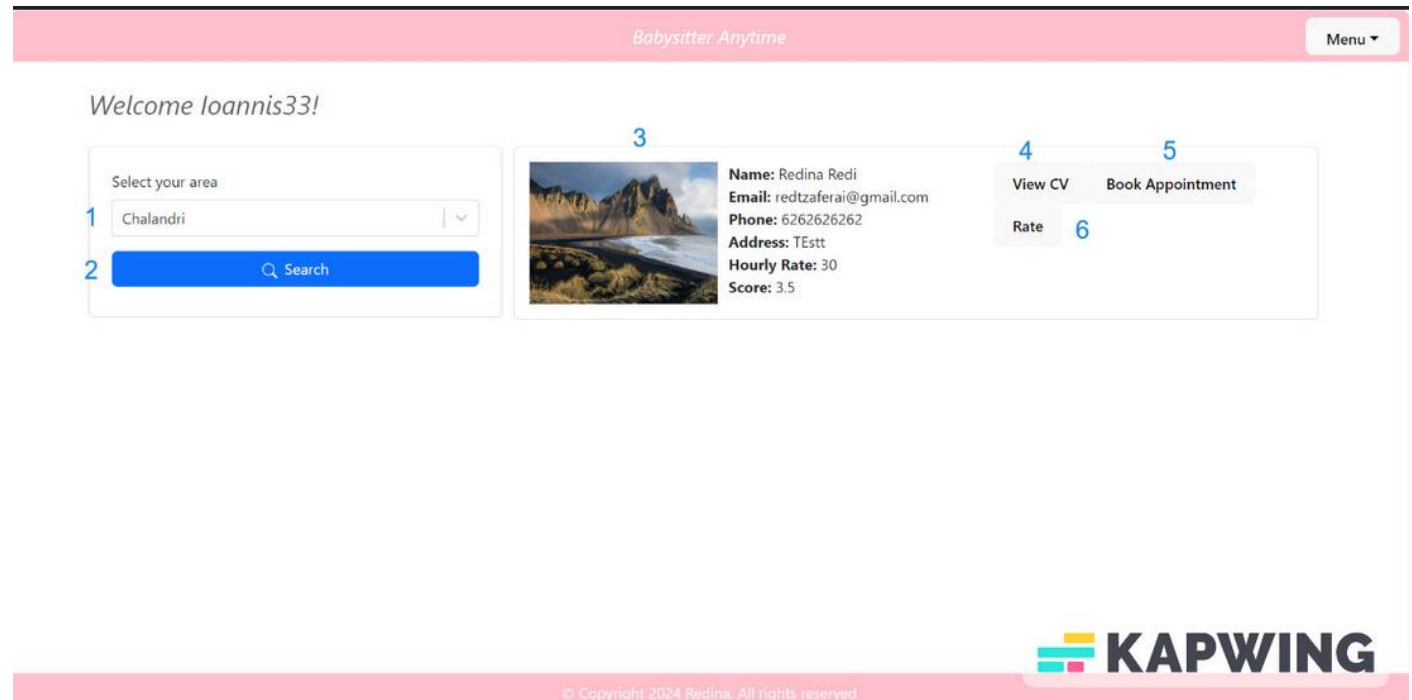
KAPWING

In this option the user is able to see:

1. The previous appointments and their details.

12.2.3 Customer Manual

12.2.3.1. Customer Home Page



In the Customer Home Page the user is able to:

1. Chose their area in order to find the babysitters that are available
2. Submit their choice
3. View the babysitters available and their information
4. Option to view the babysitters CV
5. The option to book an appointment
6. The option to rate the babysitter.

The image shows a 'Book Appointment' form with the following elements:

- Start Date:** A text input field with the placeholder 'ηη/χιλ./εεεε' and a calendar icon. A blue number '1' is positioned above the label.
- End Date:** A text input field with the placeholder 'ηη/χιλ./εεεε' and a calendar icon. A blue number '2' is positioned above the label.
- Start Time:** A text input field with the placeholder '--:-- --' and a clock icon. A blue number '3' is positioned above the label.
- End Time:** A text input field with the placeholder '--:-- --' and a clock icon. A blue number '4' is positioned above the label.
- Confirm Appointment:** A blue button with white text. A blue number '5' is positioned to its right.
- KAPWING:** The logo is located in the bottom right corner of the form.

In the Book Appointment option the user is able to:

1. Pick the Starting Date
2. Pick the Ending Date
3. Pick the Starting Time
4. Pick the Ending Time
5. Submit the appointment by clicking the 'Confirm Appointment' button


×

Rate Babysitter

★ ★ ★ ★ ☆ 1

Leave a comment (optional)
2

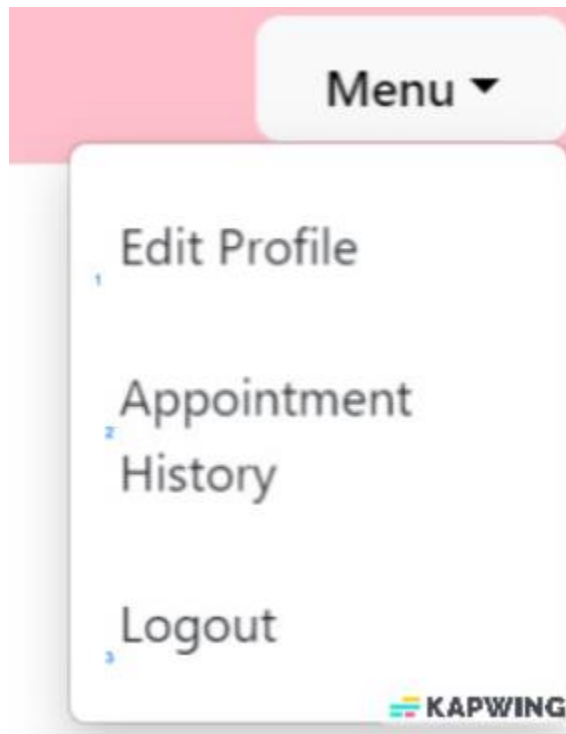
3 Submit Rating

 KAPWING

In the Rate Babysitter option the user is able to:

1. Place the stars for the babysitter
2. Leave a comment on the babysitter
3. Submit the rating by clicking on the 'Submit Rating' button

12.2.3.2 Customer Menu



Through the menu the user is able to navigate to their options

1. Edit Profile
2. Appointment History
3. Logout

12.2.3.3 Customer Edit Profile

Babysitter Anytime

Menu ▾

Edit Profile

1

First Name
Ioannis

Last Name
Ion

Phone Number
69696969

Email
m@m.com

Address
test1

Update Profile 2

© Copyright 2024 Redina. All rights reserved

In the Edit Profile option the user is able to:

1. See and change their information
2. Submit the changes by clicking the 'Update Profile' button

12.2.3.4 Customer Appointment History

Welcome to Babysitter Anytime

Menu ▾

Appointments history

1

	Start Time	End Time	Status
1	25/3/2024, 8:14:00 μ.μ.	25/3/2024, 10:14:00 π.μ.	Completed
2	8/5/2024, 5:57:00 μ.μ.	8/5/2024, 7:57:00 μ.μ.	Pending
3	19/5/2024, 8:44:00 μ.μ.	18/5/2024, 10:45:00 π.μ.	Completed
4	19/5/2024, 8:44:00 μ.μ.	18/5/2024, 10:45:00 π.μ.	Completed
5	10/6/2024, 2:51:00 μ.μ.	10/6/2024, 6:51:00 μ.μ.	Completed

In the Appointment history option the user is able to:

1. See the previous appointments and their details.

