

BABY SITTING IOS APPLICATION

Project report submitted in partial fulfillment of the
requirement for the degree of Bachelor of Technology

In

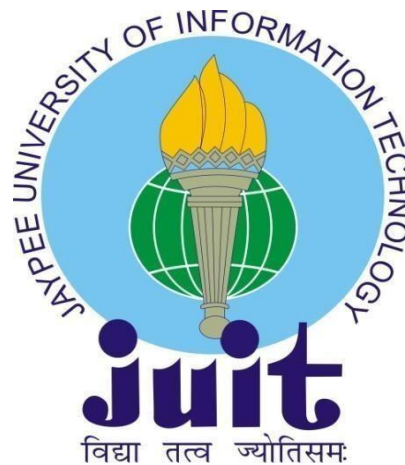
**Computer Science and
Engineering/Information Technology**

By

**Kalpita Bansal
(191537)**

Under the supervision of

**Dr. Pankaj Dhiman
(Assistant Professor, CSE)**



Department of Computer Science & Engineering
Information Technology

Candidate's Declaration

I hereby declare that the work presented in this report entitled “ **Baby sitting iOS application**” in partial fulfilment of the requirements for the award of the degree of **Bachelor of Technology in Computer Science and Engineering/Information Technology** submitted in the department of Computer Science & Engineering and Information Technology, Jaypee University of Information Technology Waknaghat is an authentic record of my own work carried out over a period from Feb 2023 to May 2023 under the supervision of **Dr. Pankaj Dhiman, Assistant Professor (SG)**. I also authenticate that I have carried out the above-mentioned project work under the proficiency stream Cloud Computing.

The matter embodied in the report has not been submitted for the award of any other degree or diploma.

Kalpita Bansal, 191537

This is to certify that the above statement made by the candidate is true to the best of my knowledge.

Dr. Pankaj
Dhiman
Assistant
Professor (SG)
Computer Science and Engineering
Dated: 06/05/2023

Plagiarism Certificate

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, WAKNAGHAT

PLAGIARISM VERIFICATION REPORT

Date:

Type of Document (Tick): PhD Thesis M.Tech Dissertation/ Report B.Tech Project Report Paper

Name: _____ Department: _____ Enrolment No _____

Contact No. _____ E-mail. _____

Name of the Supervisor: _____

Title of the Thesis/Dissertation/Project Report/Paper (In Capital letters): _____

UNDERTAKING

I undertake that I am aware of the plagiarism related norms/ regulations, if I found guilty of any plagiarism and copyright violations in the above thesis/report even after award of degree, the University reserves the rights to withdraw/ revoke my degree/report. Kindly allow me to avail Plagiarism verification report for the document mentioned above.

Complete Thesis/Report Pages Detail:

- Total No. of Pages =
- Total No. of Preliminary pages =
- Total No. of pages accommodate bibliography/references =

(Signature of Student)

FOR DEPARTMENT USE

We have checked the thesis/report as per norms and found **Similarity Index** at.....(%). Therefore, we are forwarding the complete thesis/report for final plagiarism check. The plagiarism verification report may be handed over to the candidate.

(Signature of Guide/Supervisor)

Signature of HOD

FOR LRC USE

The above document was scanned for plagiarism check. The outcome of the same is reported below:

| Copy Received on | Excluded | Similarity Index (%) | Generated Plagiarism Report Details (Title, Abstract & Chapters) | |
|---------------------|--|----------------------|--|--|
| Report Generated on | <ul style="list-style-type: none"> • All Preliminary Pages • Bibliography/Images/Quotes • 14 Words String | Submission ID | Word Counts | |
| | | | Character Counts | |
| | | | Total Pages Scanned | |
| | | | File Size | |

Checked by

Name & Signature

.....

Librarian

Please send your complete thesis/report in (PDF) with Title Page, Abstract and Chapters in (Word File) through the supervisor at plagcheck.juit@gmail.com

ACKNOWLEDGEMENT

Firstly, I express my heartiest thanks and gratefulness to almighty God for His divine blessing makes it possible to complete the project work successfully.

I am grateful and wish my profound indebtedness to Supervisor **Dr. Pankaj Dhiman Assistant Professor (SG)**, Department of CSE Jaypee University of Information Technology, Wakhnaghat. Deep Knowledge & keen interest of my supervisor in the field of “**iOS application development**” to carry out this project. This project was made possible by his never-ending patience, academic leadership, constant encouragement, constant and energetic supervision, constructive criticism, insightful counsel, reading numerous subpar versions and fixing them at all levels.

I would like to express my heartiest gratitude to **Dr. Pankaj Dhiman**, Department of CSE, for his kind help to finish my project.

I would also generously welcome each one of those individuals who have helped me straightforwardly or in a roundabout way in making this project a win. In this unique situation, I might want to thank the various staff individuals, both educating and non- instructing, which have developed their convenient help and facilitated my undertaking.

Finally, I must acknowledge with due respect the constant support and patients of my parents.

(Student Signature)

Project Group No.:

Student Name: Kalpit Bansal

Roll No.: 191537

Table of Content

| Title | Page No. |
|--|-----------------|
| Certificate | (I) |
| Plagiarism Certificate | (II) |
| Acknowledgement | (III) |
| Table of Content | (IV) |
| List of Abbreviations | (V) |
| List of Figures | (VI) |
| List of Tables | (VII) |
| Abstract | (VIII) |
| Chapter-1 (Introduction) | 1-12 |
| Chapter-2 (Literature Survey) | 13-17 |
| Chapter-3 (System Design, Analysis/Design/Development/ Algorithm) | 18 - 45 |
| Chapter-4 (Performance Analysis) | 46 - 60 |
| Chapter-5 (Conclusion) | 61 - 63 |
| References | 64 |

LIST OF ABBREVIATIONS

| | |
|------|------------------------------------|
| iOS | iPhone Operating System |
| URL | Uniform Resource Locator |
| DFD | Data Flow Diagram |
| SWOT | Strength weakness objective threat |
| UI | User Interface |
| API | Application programming interface |
| JSON | JavaScript Object Notation |
| URL | Uniform Resource Locator |
| RAD | Rapid Application Development |

LIST OF FIGURES

| Figure No. | Figure Title | Page No. |
|-------------------|------------------------------------|-----------------|
| 1 | RAD diagram | 9 |
| 2 | ER diagram | 11 |
| 3 | DFD diagram | 12 |
| 4 | Storyboard Architecture | 22 |
| 5 | List of Firebase features | 23 |
| 6 | Payment Gateway | 26 |
| 7 | Use case diagram of the system | 33 |
| 8 | Activity diagram | 34 |
| 9 | Parent panel | 38 |
| 10 | Nanny panel | 40 |
| 11 | Admin panel | 41 |
| 12 | User authentication using firebase | 42 |
| 13 | Location shown in the app | 43 |
| 14 | Authentication database | 50 |
| 15 | Realtime database | 51 |
| 16 | Launch Screen | 52 |
| 17 | Onboarding Screens | 53 |
| 18 | Login Screen | 54 |
| 19 | Register Screen | 55 |
| 20 | Verification Screen | 56 |
| 21 | Choosing Profile Screen | 57 |
| 22 | Nanny Jobcard Screen | 58 |
| 23 | Feedback and Reviews Screen | 59 |

LIST OF TABLES

| Table Number | Table Title | Page No. |
|---------------------|---------------------------------------|-----------------|
| 1 | Technologies used for the application | 12 |
| 2 | List of screens of Parent's Panel | 37 |
| 3 | List of screens of Nanny's Panel | 49 |
| 4 | List of screens of Admin Panel | 41 |

ABSTRACT

Baby sitters are basically the people who take care of the small children temporarily in the absence of their parents that is working parents hire a baby sitter to look after their small ones. They are in charge of making sure the kids under their care are safe, happy and entertained. Parents and families can employ babysitters to care over their kids for a set amount of time, such as nights, weekends, or special occasions. Parents who need someone dependable and responsible to watch over their kids when they are away frequently turn to babysitters. In order to obtain experience in childcare or as a part-time job, many teenagers and young adults work as babysitters. Other babysitters may gain experience through personal or family connections, while some may have professional training in early childhood education or related disciplines.

The purpose of developing a babysitting application is to give parents or guardians an easy way to identify dependable and reputable babysitters for their kids. Using a babysitting app makes it simpler for parents to look for babysitters in their neighborhood who are available and who meet their specific needs. Using their cellphones or other gadgets, they may do this while relaxing in their own homes. Parents may easily locate babysitters who are available during the specified time period, possess the essential abilities or qualifications, and fall within their financial range by using an app. The matching process can be facilitated by the app based on geography, availability, experience, and other pertinent factors. It may offer message or chat tools that make it simple for parents and babysitters to connect and go through specifics like schedules, fees, and directions. This guarantees efficient cooperation and reduces miscommunication. This app can enable safe and practical payment options, doing away with the necessity for cash exchanges. Parents and babysitters can both benefit from its scheduling, booking management, and reminder capabilities, which make it simpler to keep track of appointments.

This babysitting application is divided into two panels – one for parents known as parents panel and second for babysitters known as nanny panel. In both the panels, at first user sign up and user sign in pages are developed along with verification page in which verification would be done via sending OTP to user's respective phone numbers. In the sign in and sign up pages there is a button which redirects to a page in which the user selects the particular flag along with the country code which gets passed on the sign in and sign up page. There is also a button in sign in page for the purpose of like if the user is new so by clicking on that button user would be redirected to the register page and similarly on the sign-up page there is a button for sign in for the purpose like if the user already exists then the button would redirect directly to the sign in page. After verification, a page will be popped up confirming whether the user wants to login in as nanny or parent. If the user is nanny, then user will fill up the form talking about the details like address, fees, timing etc else if the user acts as a parent, then a nanny selection page would be popped up from which according to the ratings and reviews, the user could select the particular nanny of their choice. A notification page is also developed which would provide required notifications to nanny as well as to the parent. At the end ,we also created a page in which user profile come up with details like name, address etc and a page for any kind of editing in profile where we used proper validation in all the sections and along with it we used dropdown button.

A launch screen is also being developed so that when the application starts, the launch screen would be presented on the screen and rest of the screens are being developed in storyboard by using table view delegates and data source, collection view, scroll view, button, text field, dropdown and labels with proper constraints so that the screen (buttons of the screen or text fields) could fit in all devices irrespective of the size of the device screen. We have also applied proper validation like in phone number, the number cannot exceed more than ten digits and must be all digits or like in email address which would contain proper domain name. This application would be best suitable for nanny as well as for the parent as it would increase the employment and more opportunities for nanny and a safe, easy and reliable way to search for one for parents.

CHAPTER-1

INTRODUCTION

1.1 Introduction

Finding proper child care choices might be difficult in the hectic world of today. It goes without saying that parents won't always be available to watch over their children. Parents have responsibilities, a need for some alone time with their partners or friends, and a need to work in order to provide for their family. That is where the concept of babysitting came from. A babysitter, according to the website Health of Children, is a person who takes care of a child when the child's parents are not there. Depending on the terms of the contract between the parent and babysitter, the babysitter may finish their tasks in a defined period and get paid for it. We are aware of the difficulties parents encounter while entrusting others with their children's care. This is why we developed a thorough babysitting program that makes locating, scheduling, and managing babysitters simple and stress-free.

Baby sitters are basically the people who take care of the small children temporarily in the absence of their parents that is working parents hire a baby sitter to look after their small ones. They oversee making sure the kids under their care are safe, happy and entertained. Parents and families can employ babysitters to care over their kids for a set amount of time, such as nights, weekends, or special occasions. Parents who need someone dependable and responsible to watch over their kids when they are away frequently turn to babysitters.

Computer science, bringing new tools, products, and technology in today's lives, brings such technologies to help people develop things for mankind and their ease of life. iOS application development basically develops a mobile application for Apple hardware. The software is written in the Swift programming language and then deployed to the App Store for users to download.

When planning childcare services, flexibility is essential. We understand that parents have various schedules and frequently need babysitting services at various times. By providing flexible scheduling choices, our program enables parents to reserve babysitters in accordance with their unique requirements, whether it's for a few hours in the evening, an entire weekend, or even at short notice. Trust between parents and babysitters is mostly based on communication. With the help of our application, parents may maintain contact with the babysitters of their choice thanks to open communication channels. In order to keep them informed and connected throughout the babysitting session, parents will receive updates, notifications, and real-time information about their child's activities and wellbeing.

The problem here is how the parent and babysitter would be matched, and most crucially, how they would be the proper fit for one another. To help with that, there is MyBabysitter. The two parties will be connected via this Android smartphone application, as was already described. Each party should be content with the other based on the characteristics and functionalities of the other. Parents can browse the various babysitters who have registered with this app and select the one who best suits their preferences. They can do this by utilising a function called "swipe cards," which allows parents to express interest in a babysitter by swiping the babysitter's card to the right.

Depending on the terms of the contract between the parent and babysitter, the babysitter may finish their tasks in a defined period of time and get paid for it. On the PC, Xcode was built up and installed in order to construct this program and the language used is Swift. Apple Inc. created the programming language Swift. Swift seeks to give developers a more comprehensible and effective language by complementing Objective-C, another popular programming language for Apple devices. Swift was created with a heavy emphasis on safety and does away with common programming mistakes that can cause errors and crashes. It has attributes like type inference, optional types, automatic memory management, and tools for handling errors.

Swift Playgrounds offer a dynamic setting for fast testing out code and viewing the outcomes in real-time. It makes it easier for developers to learn and prototype Swift code. Adding external libraries and frameworks to this Xcode projects was made easier with the help of CocoaPods, a dependency manager for iOS and macOS projects. It offers a practical approach to handle versioning, manage dependencies, and update libraries in this project. The setup of the bundle identifier then happens after installation. This babysitting application is divided into two panels – one for parents known as parents panel and second for babysitters known as nanny panel. In both the panels, at first user sign up and user sign in pages are developed along with verification page in which verification would be done via sending OTP to user's respective phone numbers.

There is a button on the sign-in and sign-up pages that leads to a page where the user chooses the specific flag and country code that are passed on the sign-in and sign-up pages. Similarly, on the sign-up page, there is a sign in button for the purpose of like if the user already exists then the button would redirect directly to the sign in page. There is also a button in the sign in page for the purpose of like if the user is new so by clicking on that button user would be redirected to the register page. After verification, a page will be popped up confirming whether the user wants to login in as nanny or parent.

If the user is nanny, then user will fill up the form talking about the details like address, fees, timing etc else if the user acts as a parent, then a nanny selection page would be popped up from which according to the ratings and reviews, the user could select the particular nanny of their choice. A notification page is also developed which would provide required notifications to nanny as well as to the parent. At the end, we also created a page in which user profile come up with details like name, address etc and a page for any kind of editing in profile where we used proper validation in all the sections and along with it we used dropdown button.

A launch screen is also being developed so that when the application starts, the launch screen would be presented on the screen and rest of the screens are being developed in storyboard by using table view delegates and data source, collection view, scroll view, button, text field, dropdown and labels with proper constraints so that the screen (buttons of the screen or text fields) could fit in all devices irrespective of the size of the device screen. We have also applied proper validation like in phone number, the number cannot exceed more than ten digits and must be all digits or like in email address which would contain proper domain name.

Using a babysitting app makes it simpler for parents to look for babysitters in their neighborhood who are available and who meet their specific needs. Using their cellphones or other gadgets, they may do this while relaxing in their own homes. Parents may easily locate babysitters who are available during the specified time period, possess the essential abilities or qualifications, and fall within their financial range by using an app. The matching process can be facilitated by the app based on geography, availability, experience, and other pertinent factors. It may offer message or chat tools that make it simple for parents and babysitters to connect and go through specifics like schedules, fees, and directions. This guarantees efficient cooperation and reduces miscommunication. The payment procedure has also been streamlined, making it safe and simple for both parents and babysitters.

Our software offers a smooth payment system that guarantees fair remuneration for babysitters and offers parents clear billing and invoicing. Bid adieu to cash transactions and welcome to a dependable and easy way to pay. Parents and babysitters can both benefit from its scheduling, booking management, and reminder capabilities, which make it simpler to keep track of appointments as well. This application would be best suitable for nanny as well as for the parent as it would increase the employment and more opportunities for nanny and a safe, easy and reliable way to search for one for parents.

1.2 Problem Statement

As we are aware that finding proper child care choices might be difficult in the hectic world of today. It goes without saying that parents won't always be available to watch over their children. Parents have responsibilities, a need for some alone time with their partners or friends, and a need to work in order to provide for their family. That is where the concept of babysitting came from. A babysitter, according to the website Health of Children, is a person who takes care of a child when the child's parents are not there. Depending on the terms of the contract between the parent and babysitter, the babysitter may finish their tasks in a defined period and get paid for it. We are aware of the difficulties parents encounter while entrusting others with their children's care. This is why we developed a thorough babysitting program that makes locating, scheduling, and managing babysitters simple and stress-free. Babysitters oversee making sure the kids under their care are safe, happy and entertained.

Some of the key obstacles as to why we are building this application are:

- 1): Parents frequently struggle to find dependable and experienced babysitters within their local community or network. Limited access to reputable nannies. They require a platform that gives them access to a list of verified babysitters who have had their backgrounds examined.

- 2): Parental anxieties about the safety and security of their children when they are placed in the care of others are understandable. To secure the safety and wellbeing of their kids, they need a solution that includes stringent screening procedures including background checks and references.

- 3): Building trust and ensuring that parents receive updates and notifications about their children's well-being depend on open and honest communication between parents and babysitters. Parents should also be able to comment on and rate babysitters, assisting other parents in making wise choices.

4): Managing money and handling payments between parents and babysitters can be difficult. The app should offer a safe and practical payment system that guarantees fair remuneration for babysitters and offers parents clear billing and invoicing.

5): Babysitting services may be needed by parents at varied times, including nights, weekends, or on short notice because of their frequently variable schedules. Each family should be able to conveniently schedule babysitters thanks to the application's varied scheduling choices that are tailored to their specific requirements.

This babysitting application would provide a trustworthy and safe babysitter to parents with convenient notification page and better employment to babysitters.

1.3 Objectives

India is a developing country with around thousands of children being born daily and in these busy times, working parents need babysitters for babysitting of their small ones that is to look after them in their absence and make them feel safe, happy, and entertained.

Hence our main objective is to create an iOS application of babysitting for nanny and parents – for nanny to find better opportunities and reliable families paying them according to their wish and for parents who want to find safe and trustworthy nanny for their children. iOS application development basically develops a mobile application for Apple hardware. The software is written in the Swift programming language and then deployed to the App Store for users to download. This application would be best suitable for nanny as well as for the parent as it would increase the employment and more opportunities for nanny and a safe, easy and reliable way to search for one for parents. This babysitting app makes it simpler for parents to look for babysitters in their neighborhood who are available and who meet their specific needs.

Using their cellphones or other gadgets, they may do this while relaxing in their own homes. Parents may easily locate babysitters who are available during the specified time period, possess the essential abilities or qualifications, and fall within their financial range by using an app. The matching process can be facilitated by the app based on geography, availability, experience, and other pertinent factors. It may offer message or chat tools that make it simple for parents and babysitters to connect and go through specifics like schedules, fees, and directions.

This guarantees efficient cooperation and reduces miscommunication. This app can enable safe and practical payment options, doing away with the necessity for cash exchanges. Parents and babysitters can both benefit from its scheduling, booking management, and reminder capabilities, which make it simpler to keep track of appointments as well.

1.4 Methodology

It is obvious that parents will not always be able to look after their kids. Parents must work to support their families, have obligations, and may even desire some alone time with their partners or friends. That's how the idea of babysitting originated. According to the website Health of Children, a babysitter is a person who looks after a child when his or her parents are not there. The babysitter may complete their duties in a predetermined length of time and get paid for it, depending on the terms of the parent and babysitter's agreement. This is the implementation of an iOS application of babysitting. To build this application, Xcode was installed and set up on the PC. Adding external libraries and frameworks to this Xcode projects was made easier with the help of CocoaPods, a dependency manager for iOS and macOS projects. It offers a practical approach to handle versioning, manage dependencies, and update libraries in this project. Then after installing comes the setting of bundle identifier. The language used for development is Swift and then deployed to the App Store for users to download.

Our main objective is to create an iOS application of babysitting for nanny and parents – for nanny to find better opportunities and reliable families paying them according to their wish and for parents who want to find safe and trustworthy nanny for their children. This babysitting application is divided into two panels – one for parents known as parents panel and second for babysitters known as nanny panel. In both the panels, at first user sign up and user sign in pages are developed along with verification page in which verification would be done via sending OTP to user's respective phone numbers. After verification, a page will be popped up confirming whether the user wants to login in as nanny or parent.

If the user is nanny, then user will fill up the form talking about the details like address, fees, timing etc else if the user acts as a parent, then a nanny selection page would be popped up from which according to the ratings and reviews, the user could select the particular nanny of their choice. A notification page is also developed which would provide required notifications to nanny as well as to the parent. At the end, we also created a page in which user profile come up with details like name, address etc and a page for any kind of editing in profile where we used proper validation in all the sections and along with it we used dropdown button.

A launch screen is also being developed so that when the application starts, the launch screen would be presented on the screen and rest of the screens are being developed in storyboard by using table view delegates and data source, collection view, scroll view, button, text field, dropdown, and labels with proper constraints so that the screen (buttons of the screen or text fields) could fit in all devices irrespective of the size of the device screen. We have also applied proper validation like in phone number, the number cannot exceed more than ten digits and must be all digits or like in email address which would contain proper domain name. This app can enable safe and practical payment options, doing away with the necessity for cash exchanges.

Table 1.1 Technologies used for the application

| | |
|--------------|-------------------------|
| Swift, Xcode | Front-end framework |
| Node.js | Back-end |
| Express.js | API |
| Stripe | Payment gateway |
| Firebase | Back-end authentication |

Rapid Application Development (RAD) is the approach that was selected for this project's methodology of development. It was selected mostly because it takes a "divide and conquer" strategy with the project. It divides the undertaking into various doable stages or tasks. This not only lessens the project's overpowering nature but also gives room for modifications, should they ever be necessary. Other techniques, such as the waterfall model, do not allow for this because it is very hard to adjust a project after it has begun.

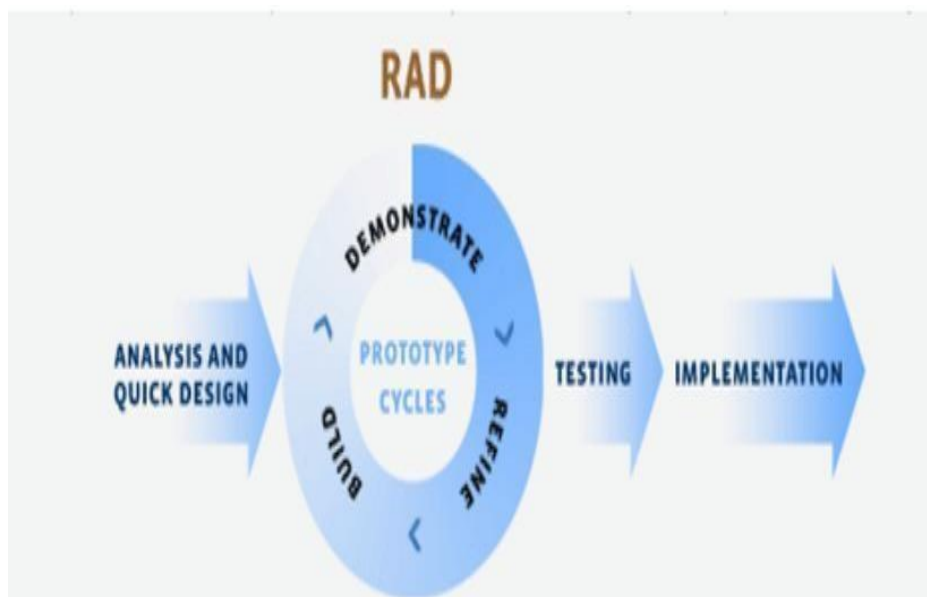


Fig 1: RAD diagram

The Babysitting iOS Application's entities and relationships are depicted more simply in the ER diagram. The following relationships are shown in the ER diagram given below:

1): Booking and Parent have a one-to-many relationship; one parent may make numerous reservations.

2): Babysitter and Booking have a one-to-many relationship; one babysitter may have several appointments.

3): Babysitter and Availability have a one-to-many relationship; one babysitter may have several records of availability.

Entity explanation:

1): Parent: It represents a parent who is utilising the iOS babysitting app. Name, email, phone, and parent_id (primary key) are the attributes.

2): Babysitter: It represents a nanny who has signed up on the platform. Babysitter_id (primary key), name, email, phone, and experience are among the attributes.

3): Booking: It represents a babysitting reservation made by a parent. Booking_id, parent_id (foreign key reference Parent), babysitter_id (foreign key referencing Babysitter), start_time, end_time, and status (such as confirmed, pending, or cancelled) are some of the attributes.

4): Availability: represents a babysitter's availability for certain dates and times. Date, start_time, end_time, babysitter_id (foreign key referencing Babysitter), and other attributes.

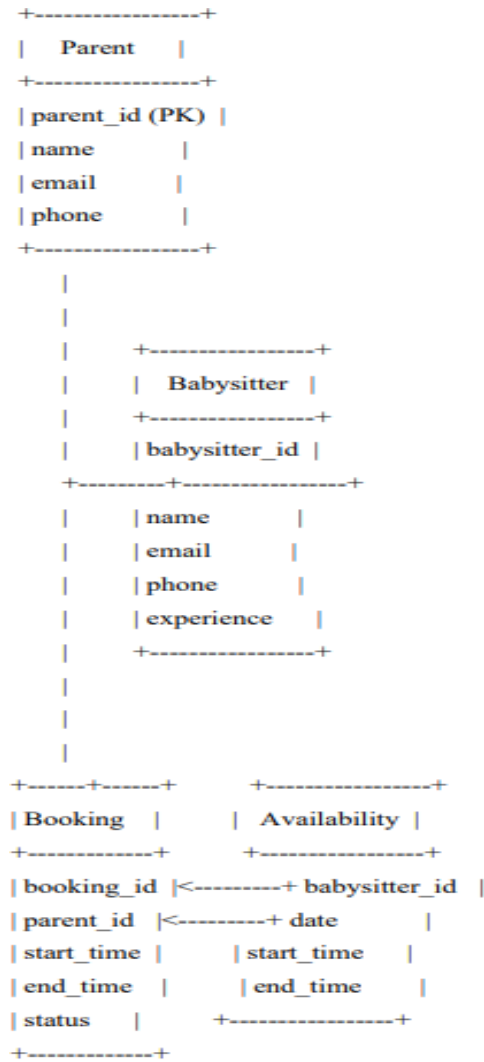


Fig 2: ER diagram

1.5 Data Flow Diagram

The data flow diagram maps the entire information of any process or system by using symbols such as circles, rectangles, and squares. The main purpose is to show and describe the boundaries and scope of any process or system. It describes how the data will enter and leave the system and where that data would be stored.

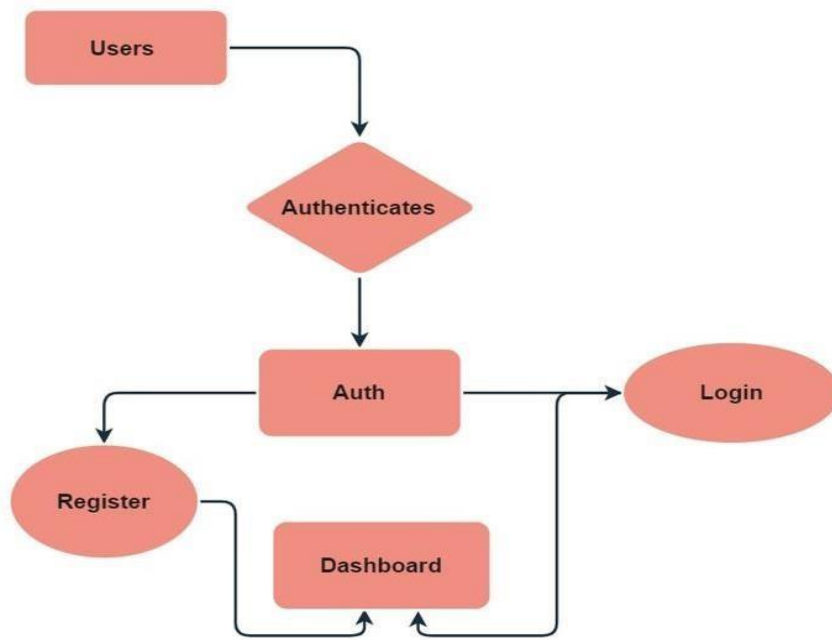


Fig 3: DFD diagram

1.6 Organization

The rest of the paper is organized as follows: In Chapter 2, we present a literature survey that depicts the various approaches used by the authors to create an iOS mobile application for babysitting. Section 3 highlights the methodology and system development of the project. It represents the various computational, experimental, and mathematical concepts of the project.

In addition, we focused on the software and hardware platforms required to implement the model. In Chapter 4, we present a performance analysis of the project. Section 5 presents the conclusions of the project and observations of the results. Application and future scope of the project is also being provided.

CHAPTER-2

LITERATURE SURVEY

2.1. Literature Survey

This chapter presents a list of potential approaches to develop iOS mobile application for babysitting. This poll assists in highlighting the shortcomings of several widely used techniques. The majority of approaches have the disadvantage of not being used in real-time, making it difficult for people to use them. According to a review of the literature on babysitter apps for iOS users, demand for these services is growing as parents look for solutions to manage their busy schedules and keep their kids safe.

The study examined how technology affects parenting, focusing on the advantages and disadvantages of utilizing babysitting applications. Parents who used babysitting apps claimed they provided greater flexibility, convenience, and usability, but they also voiced worries about security and dependability, according to one study. Parents place a higher priority on safety, dependability, and ease of communication, according to another study that examined the qualities and skills that customers value most. The importance of reputation and trust in the babysitting app ecosystem is shown by the fact that many users rely on user ratings and reviews to assess capable babysitters.

Numerous studies have stressed the significance of putting robust protections in place to protect kids and their families, which makes privacy and security other key aspects to consider. For instance, experts have advocated for improved identity verification procedures, identity verification, and secure messaging systems to encourage private communication between parents and babysitters. Overall, the assessment of the research shows that babysitter apps can greatly enhance the childcare experience by offering parents more freedom and accessibility and by providing sitters with career prospects and flexibility.

Since, it is important to consider safety, privacy, and trustworthiness to ensure that these apps are safe and a better and reliable options for families. Numerous studies have emphasized the importance of having strong safeguards in place to protect children and their families, making privacy and security other crucial factors to consider. For instance, to promote safe and secure communication between parents and babysitters, researchers have recommended for improved identification verification procedures, background checks, and secure messaging systems. The literature analysis shows that babysitter apps have a lot of potential to improve the childcare experience by giving parents more convenience and flexibility and by giving sitters job possibilities and flexibility. However, it is essential to thoroughly assess safety, privacy, and trustworthiness in order to make sure that these applications are a safe and reliable option for families.

The remainder of this paper is organized as follows:

[1] M. Suganya, Dayana R, Revathi.R, “Application for babysitter using React Native,” IJCET [2020]

This is an online service that links parents with nannies and babysitters in their area. According to their unique demands, parents can use the marketplace to locate and reserve childcare providers. In order to make it simpler for parents to obtain dependable day-care, this particularly serves metropolitan regions. Parents can look for babysitters on it by area, availability, and other factors. Each sitter has a complete profile on the platform that includes details about their background checks, credentials, experience, and reviews from other parents. Through the platform, parents can also communicate with potential babysitters, arrange interviews, and make reservations. It provides a variety of services, such as last-minute appointments, tutoring, occasional babysitting, part-time or full-time nanny placements, and even pet care. This platform gives parents a quick and easy way to identify and hire dependable babysitters for their kid

[2] Mrugank Gandhi, Shubham Kothavade, “Development of a iOS Mobile Application for online Babysitters,” IRJMETS [2022]

This app links parents with childcare providers like nannies, babysitters, and pet sitters. Parents can use this app to find and hire carers based on their specific needs and preferences. It enables parents to set up an account and post a job posting stating their requirements for child care. They can specify the caregiver's required schedule, working conditions, and any additional needs they may have. The software then connects an employment listing via local caretakers who fit the requirements. Each caretaker on this app has a personal account with a profile that includes information about their education, credentials, background checks, and company endorsements. Parents can look through these websites before selecting a nanny, contact possible employees, and set up appointments or interviews to ascertain compatibility. It also provides other services like background checks, which may reassure parents. Using this website, parents may check the backgrounds of potential carers to make sure they satisfy their safety criteria. Additionally, it offers tools and knowledge that parents and carers can use to efficiently manage their arrangements. This consists of functions including an integrated chat system, interview scheduling, and a speedy and secure payment method.

[3] Yogita Masare, Sneha Mahale, Manjusha Kele, Ashvin Upadhyay, Bhushan R. Nanwalkar, “Subscription-based babysitting service,” IJERT [2021]

Families may utilise this mobile app to find babysitters in their area. It is intended to provide parents with a simple and dependable way to locate and reserve dependable babysitters for their kids. It is televised in several American cities. People can register accounts and search for potential nannies in their region with the aid of smartphone applications. The application provides details about each nanny's history, experience, experience level, and family ratings checks. If the nanny has received any references from friends or other trustworthy sources, parents can inquire.

Parents can request applications from possible carers using the software. Depending on their availability, the babysitter can then decide whether to accept or reject the request. The app also has tools that allow parents to make payments, schedule interviews, and contact with babysitters. The social connection function of this app is one of its distinctive features. Parents get the opportunity to meet babysitters who have been highly recommended by their friends, which helps foster confidence and trust in the final carer.

[4] Vrushali C. Waikar, Sheetal Y. Thorat, A. A. Ghute, and Priya P. Rajput, Mahesh S. Shinde, "Connecting Families with Caregivers Application for Android Mobile Application," IRJET [2020]

This technological platform connects parents with qualified babysitters who have undergone extensive screening. It provides on-demand childcare facilities, which makes it simple for parents to locate dependable nannies when they require them. Parents pay a monthly or yearly subscription fee to access the application's portal and make babysitter reservations. By completing extensive background checks, reference checks, and interviews with potential babysitters, the platform seeks to provide parents peace of mind. This often hires nannies, teachers, or someone with a lot of babysitting experience who have previous childcare experience. Parents can explore local babysitters who are available and check their profiles, which include details about their experience, credentials, and reviews from other parents, using this app or website. For parents to contact with possible babysitters and go through specific childcare requirements, the portal also offers a chat system. With it is flexible booking options, parents may reserve babysitters in advance or call in a last-minute need for childcare. This software makes it simple for parents to handle transactions directly within the platform by facilitating safe payment processing.

[5] Ramesh D, Vishnu B. Vardhan, "Development of a Android Mobile Application for Babysitters," IJARCEE [2013]

This platform links families with carers for a variety of care services, such as child care, geriatric care, pet care, and more. It is one of the biggest and best-known sites for seeking carers, and it is widely used throughout many nations. Families can register on this to build a profile and submit job postings that detail their care need. Based on requirements like availability, experience, certifications, and more, they can search for local carers.. This platform has a messaging feature that enables families and carers to interact and go over specifics of their duties. For increased security, families can also conduct interviews, contact references, and ask for background checks. This platform also provides possibilities for both temporary or infrequent care as well as full- and part-time care. It offers tools and information to help families and carers manage their relationships in addition to carer matching services. This includes tools like an integrated calendar for scheduling, a payment platform for conducting transactions, and access to publications and advice on a range of caregiving-related subjects.

[6] Noof A. Al-Safi, Rawan A. Al-Asiri, Malak A. Al-Malki, Sameera Abar, “Fostering Childcare E-Service: Design and Development of a Software Application,” IJITN [2022]

This research paper was published by the IJITN. They developed a vibrant mobile application of babysitting where the main idea was to provide an online childcare service in which parents could timely track and monitor and take care of their children from a distant place. The proposed strategy's convenience and utility in developing good parenting for young children are confirmed by performance evaluation of its integrated features and functionality. Additionally, this research provides students who are developing their own software applications with a useful reference in a variety of fields.

CHAPTER-3

SYSTEM DEVELOPMENT

3.1 Non-Functional Requirements

3.1.1 User interface

The user interface must be simple and easy to use and understand, and based on the given inputs, the model must be able to efficiently help nanny get better employment and reliable parents and help parents to find suitable and trustworthy nanny to babysit their children. Proper validation and constraints are being used while creating the user interface so that user would not face any difficulty while using the application. The payment gateway being integrated is very safe and secure for hassle free payments.

Here are some things to think about when creating a user interface:

1): Maintain simplicity and ease of use by keeping the UI design simple and uncluttered. Be careful not to overwhelm users with too many graphic components or difficult navigation.

2): Consistency: Keep the app's color scheme, typography, and other design components the same throughout. This guarantees a seamless and simple user experience.

3): Design a navigation structure that is simple to use and allows users to easily switch between screens and access different features. Use the side menus, navigation bars, and other standard iOS navigational elements as needed.

4): Make that the user interface (UI) responds to various device sizes and orientations. This entails utilising auto-layout constraints and adaptive layouts to support different screen sizes, including iPhones and iPads.

5): Feedback and Validation: When users engage with the app, give them unambiguous feedback, for as by displaying success messages for activities that were successful or error messages for inputs that were erroneous. Users can better grasp the results of their activities and misunderstanding is avoided as a result.

3.1.2 Hardware

No special hardware interface is required for the successful implementation of the system other than a PC.

3.1.3 Software

1). Xcode

2). Firebase

3). Node.js

4). Cocoa Pods

5). Stripe

6). Swift programming language

- **Xcode**

Apple Inc. developed Xcode as an integrated development environment (IDE) for creating software programmes for the macOS, iOS, iPadOS, watchOS, and tvOS platforms. It comes with a set of software development tools that give developers the ability to make, test, and upload programmes to the App Store. The intuitive user interface of Xcode, which makes it simple for developers to navigate and personalise their workplace, is one of its primary characteristics. In addition to storyboard, which helps developers to plan an application's flow and individual screens, Xcode offers a variety of tools for developing graphical user interfaces, such as interface builder, which allows developers to graphically design application layouts.

A source code editor is also available in Xcode, and it offers several effective editing capabilities like syntax highlighting, autocomplete, and refactoring tools. Swift, Objective-C, C++, and Python are just a few of the programming languages that are supported by the editor. Another important tool in Xcode that aids developers in finding and fixing coding problems is the debugger. Developers can watch variables and memory use, set breakpoints, and step through code using the debugger, which provides real-time feedback on how the application is behaving. Developers can optimise the performance of their applications with the use of the performance analysis tools included in Xcode.

With the use of these tools, developers may pinpoint performance bottlenecks, such as CPU and memory usage, and receive thorough analysis reports to help them improve their code. Additionally, Xcode provides testing tools that let programmers examine the performance of their apps in a range of conditions, including as user interaction, network connectivity, and load-bearing conditions. All things considered, Xcode is a strong and adaptable tool for developers using Apple's different platforms. It is a crucial step in the software development process for iOS, iPadOS, watchOS, tvOS, and macOS applications because to its extensive array of tools for designing, creating, debugging, and testing programmes.

- **Swift**

Apple Inc. created Swift, a general-purpose, open-source programming language. It is intended to be quick, safe, and contemporary. Applications for iOS, macOS, watchOS, and tvOS are created using Swift. The syntax of Swift is designed to be simple to understand and write, making it friendly for programmers who are just starting out. A variety of programming paradigms like imperative, functional, and object-oriented approaches are also being supported by it. Along with it, it also helps reduce and prevent common programming errors plus enhances its app performance with safety features like optionals, type safety and intelligent memory management. Swift is great for creating a variety of apps reason being it is strong and adaptable language.

- **CocoaPods**

A dependency manager for Swift and Objective-C Cocoa projects is called CocoaPods. As developers can quickly search for and install third-party libraries to their projects through a straightforward command-line interface, it makes the process of adding external libraries and frameworks to iOS projects simpler. For these libraries, CocoaPods automatically manages versioning and updates, making it simpler for developers to maintain and upgrade their code. Additionally, CocoaPods makes it simple to exchange and reuse code, which improves developer collaboration. For iOS developers looking to accelerate their development process and raise the calibre of their code, CocoaPods has grown to be a well-liked solution thanks to its huge collection of community-contributed pods.

- **Storyboard**

A storyboard is a graphic representation of an iOS application's user interface, displaying content screens and the links that connect them. Each scene in a storyboard represents a view controller and its associated views. Scenes are connected by segue objects, which represent a change in control between two view controllers.

By placing views like buttons, tables, and text views onto scenes, you may layout and design your application's user interface using the visual editor for storyboards provided by Xcode. A storyboard also gives you the ability to govern the data transmission between view controllers and to link a view to its controller object. It is advised to utilise storyboards while creating your application's user interface since they let you see how it will look and function all on one canvas.

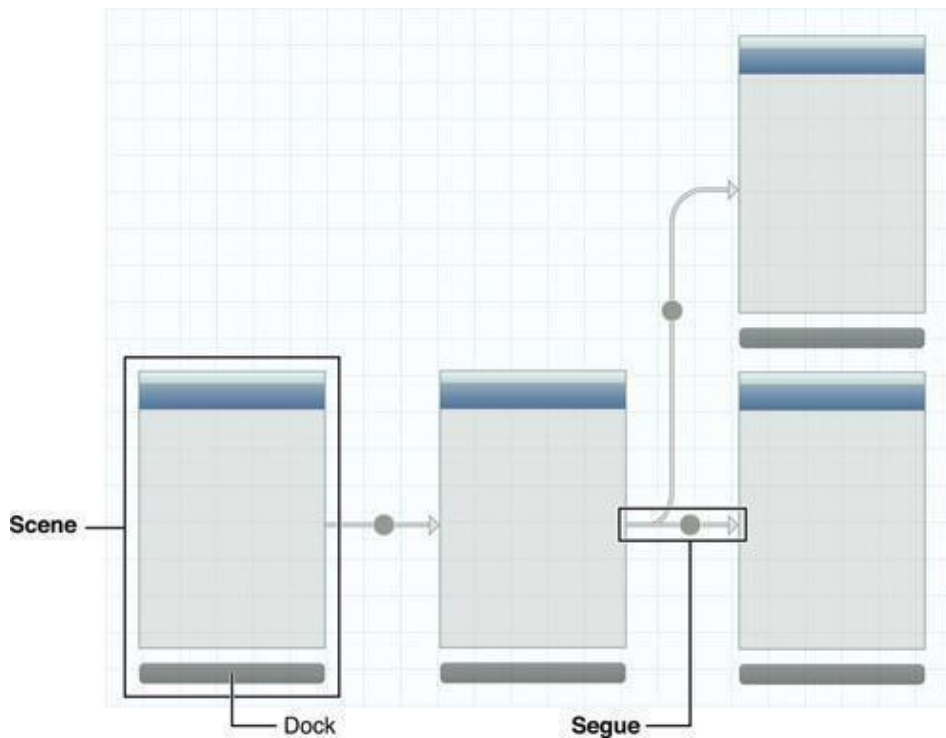


Fig 4: StoryBoard architecture

On the iPhone, each scene corresponds to the entirety of the screen; on the iPad, however, numerous scenes can display simultaneously, for instance when utilizing popover view controllers. Every scene has a dock that shows symbols for the scene's top-level elements. The dock is generally used to connect the view controller's views to its outlets and actions. As with other objects imported from a storyboard, you must override `awakeFromNib` to complete initializing a view controller loaded from a storyboard. via Nib.

- **Firebase by Google**

A platform called Firebase, developed by Google, provides a variety of tools for building and growing mobile and web applications. It is a popular choice for iOS developers because it offers a variety of services and features that can facilitate development and reduce time to market. The features offered by the firebase are listed in the built portion of figure given below.

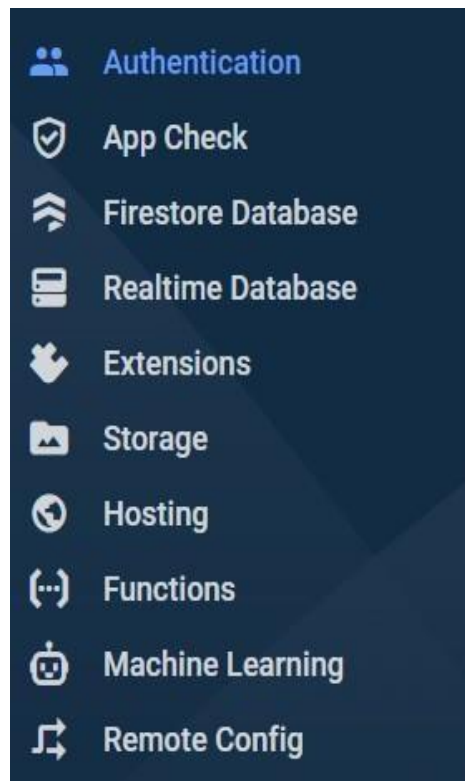


Fig 5: List of Firebase features

The following are key details about Firebase for iOS applications:

1). One choice is to use Firebase Realtime Database, a cloud-hosted database, to rapidly store and sync data. It offers a NoSQL database, making data management and upkeep simple in a flexible, scalable approach. Applications that respond immediately to data changes can be built using the Firebase Live Database, increasing their interactivity and user appeal.

2). Using Google Authentication is a quick and simple way to add authentication to your iOS application. Numerous methods of authentication, such as social media, phone, email, passwords, and others, are supported. With Firebase Identification, you can develop secure apps that request user authentication and permission. In this project, the sole type of authentication we're focusing on is phone number authentication. OTP is generated and sent to the provided mobile number. For added protection, this OTP has a configurable life span.

3). Firebase cloud-based storage makes it simple to store and serve user-generated content with big file sizes, such photos and movies. It enables both local and remote storage, enabling material access from any place on the planet. Firebase Cloud Storage integrates with other Firebase services like Authorization and Dynamic Database to provide a comprehensive storage solution for your iOS application.

4). FCM (Firebase cloud messaging), another functionality of Google Firebase, is available. The stable and scalable Firebase cloud-based messaging service enables message delivery to clients on Android, iOS, and the web. It makes it simple to deliver timely and pertinent communications to people by providing a number of capabilities, such as targeting, planning, and statistical analysis. Because of its ease of use, scalability, and compatibility with other Google services, it is a well-liked choice for iOS.

• **Stripe for Payment**

Companies of any kind may accept and manage payments online thanks to a program called Stripe that handles payments. Online payment processing is made simple and safe by Stripe. Numerous payment options, such as debit and credit card transactions, and payments via mobile devices are supported. All aspects of payment processing, such as identifying fraudulent transactions, reimbursements, and currency conversions, are handled by Stripe. Additionally, it supports memberships and periodic payments, making it simple for businesses to handle customers' payments over time. It holds a PCI Level 1 Service Provider authorization, which is the highest kind of validation available to payment processors. Additionally, Stripe's platform offers fraud identification and avoidance tools like multiple-factor authentication, real-time risk assessment, and machine learning algorithms. Companies of various sizes, including startups, SMBs, and multinationals, use Stripe on a global scale.

It supports all payment types, including debit and credit card payments as well as card payments of all kinds, including MasterCard, visa, and UPI payments. Below are a few of Stripe's important characteristics.

1). Payment options: Stripe makes it easy and secure to process payments online. Numerous payment options, such as debit and credit card transactions, and payments via mobile devices are supported. All aspects of payment processing, such as identifying fraudulent transactions, reimbursements, and currency conversions, are handled by Stripe. Additionally, it supports memberships and periodic payments, making it simple for businesses to handle customers' payments over time.

2). Security: Stripe places a high priority on security and offers a range of solutions to help businesses keep their transactions secure. The greatest level of validation accessible to payment processors is the PCI Level 1 Service Provider accreditation that it possesses. The Stripe platform also provides capabilities for detecting and preventing fraud, including multi-factor authentication, real-time risk analysis, and machine learning algorithms.

3). Global Reach: With around 135 recognized denominations and over 40 operating nations, Stripe makes it simple for business establishments to take payments through clients all around the globe. Stripe handles all aspects of international payments, including currency conversions and compliance with local laws.

4). Billing: A group of software tools referred to as invoicing make it easier for businesses to manage membership and recurring payments. It provides tools for setting up pricing schemes, managing users and payments, and handling declined payments. Stripe Billing collaborates with other Stripe services like Payment Processing and Hawkeye to provide comprehensive billing solutions.

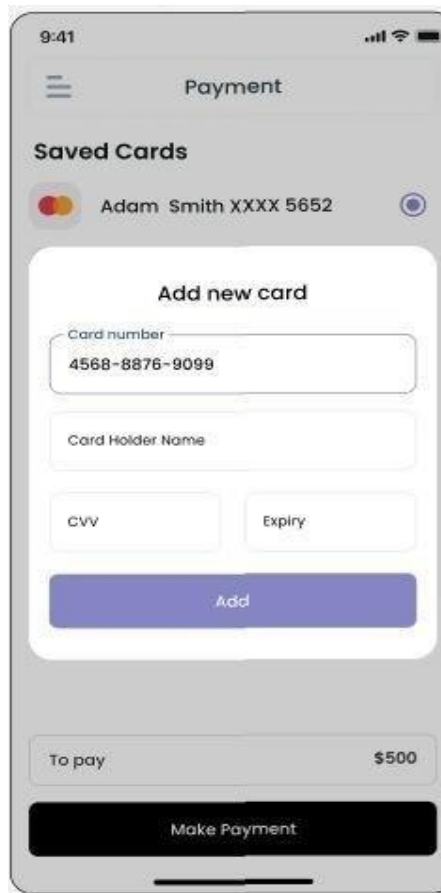


Fig 6: Payment gateway

3.1.4 User Requirements

The user requirements include that the user could comfortably and smoothly navigate through the mobile application and could be able to use all the services easily without runtime lag. It must be able to use without any problem or error or any unexpected crashes of the system. It also includes the constraints on the user that the user must not provide or leak any false information to other users using the application and shall not be able to access any other user's profile or make any false changes in that.

3.1.5 Performance

The babysitting application being built must be able to coordinate properly with nanny as well as with the parent fulfilling each one's needs and satisfying them that is for parents- providing them with trustworthy and reliable nanny and for nanny- providing them with better employment and safe families.

The user must be able to use all the app's features without experiencing any noticeable runtime lag and the system must be able to finish its tasks without any noticeable runtime lag. The mobile application that would be created must provide a simple, intuitive interface which would facilitate the user's navigation on the application. The system must be able to finish its tasks without any noticeable runtime lag. An extremely low failure rate is required for the system. The system must be implemented utilizing Firebase database, XML, and Swift. Nothing dishonest for the users shall be displayed by the system.

The system must not show any other users' personal information about its users. All user-related data will be encrypted by the system. The information must be kept private and only available to the owners. The main objective of this capstone is to develop a mobile application that connects parents with the ideal nanny for their child or children. The primary objective of the feasibility study is to ascertain whether this project is technically and financially feasible. The market, technical, economic, and timetable aspects will be the focus of this study.

It must be possible to utilize it without encountering any issues, errors, or unforeseen system crashes. Additionally, it places restrictions on the user, prohibiting them from accessing another user's profile or making any false changes therein, as well as from giving or leaking any false information to other users of service. This project was built using a variety of technologies and tools. To begin with, Xcode is the IDE used to create the applications. The iOS app being developed, as implied by the name of the IDE, calls for the use of the programming language Swift.

Firebase was selected as the database since it integrates well with Xcode and would house all the user data. It also provides a real-time database. The layout of the app's activities and the interface were both implemented using the markup language XML. This project was theoretically viable thanks to all these technologies.

3.2 Functional Requirements

3.2.1 User Requirements

With the aid of their personal details, the actor should be able to create an account. The actor's registered email and password must be used to access the system. The actor will have access to the cards and online profiles of other performers who have different statuses. The actor should swipe right on a card if they are interested in it. The actor should swipe left on a card to convey disinterest. The performer will have the option to post a photo of their choosing in the settings. It may be modified at any time. The performer must be able to describe himself or herself in relation to the settings. The performer must be able to describe himself or herself in relation to the settings. The actor's age, home address, and personality type are some examples of this. The actor can also describe what he or she is seeking in the opposing position and what they are prepared to offer in the description area.

3.2.2 System Requirements

The actor will have the ability to register with the system by giving the necessary data. Name, status (whether parent or babysitter), email, and password of the performer are all included. Once the necessary data has been provided, the server will verify that the input is correct. The performer will be informed if a mistake exists so that they can fix it. The system must be able to process the actor's email and password, and it must match the information they entered when they originally enrolled in order to provide them access. The technology will be able to record in the database which cards or profiles the actor swiped left or right to.

The system must be capable of storing the image that the actor uploads to his or her card in the database. The system must have the ability to save the actor's description in the database.

3.2.3 Computational metrics

There are several factors to consider while determining the computational metrics for an iOS babysitter app. The following list of key computational metrics includes:

1): Performance Measurements: Calculate how quickly the app responds to user actions like booking appointments, looking for babysitters, or loading data. Watch how long it takes the app to load its initial screens, data, or graphics. Monitor the memory footprint of the app to make sure it uses minimal system resources and runs well.

2): Network Measurements: The time it takes the app to receive responses from external APIs used for features like location services or payment processing is known as the API response time.

3): Batteries Used: Power Consumption: Examine the app's effect on the device's battery life to make sure it will not use up the battery too quickly whether in use or operating in the background.

4): Accident Reporting: Track the frequency with which your app crashes and examine crash logs to find and repair bugs or stability problems.

5): Metrics for User Engagement: Track the number of people who are actively using the app over time to determine how well-liked and persistent it is. Measure the typical length of user sessions to gain insight into how involved users are with the app. Track particular user actions performed within the app, such as reservations booked, messages exchanged, or reviews left.

3.3 Constraints And Assumptions

3.3.1 Constraints

There are various limits to consider while building an iOS application for babysitting. Here are some typical restrictions that could be present:

1): Privacy and Security: Babysitting apps deal with sensitive information such as identifying information and location data. It is essential to guarantee the security and privacy of user data. The app needs to follow privacy regulations and use safe authentication and data encryption procedures.

2): Legal Compliance: Local laws and regulations, particularly those relating to child safety and labour legislation, may require babysitting apps to adhere to. To stay out of trouble with the law, it's critical to comprehend and comply with these rules.

3): Availability and scheduling: Babysitters frequently have different schedules; therefore, the app must efficiently handle availability and scheduling information. It needs to enable parents to plan appointments, check the availability of babysitters, and manage their schedules.

4): Reviews and Ratings: Including a system that allows parents to rate and review babysitters can be helpful, but it also needs to be moderated properly to protect the system's integrity and stop abuse or inappropriate content.

5): Payment processing: If the application has a payment mechanism, it must be secure and adhere to all financial rules. The management of transactions safely and integrating with a reputable payment gateway are essential.

6): Communications and notifications: The software should enable communication between parents and babysitters, whether it is through text messages or phone conversations. It should also notify both parties of pertinent alerts, including changes or reminders for appointments.

7): Geolocation and Maps: Geolocation and maps are two elements that many babysitting applications offer to help parents find local nannies. Although using geolocation and maps can improve user experience, proper implementation is necessary to guarantee precise position tracking and effective integration with mapping services.

3.3.2 Assumptions

Some common assumptions are:

1): Parental Need: It is assumed that parents need easy access to dependable babysitters for a variety of reasons, including work obligations, social obligations, crises, or personal time. This requirement is intended to be met by the app, which offers a platform for locating and hiring dependable babysitters.

2): Babysitter Availability: It is presumed that there are people willing to watch children who are available. The app assumes that there is a pool of babysitters who are interested in interacting with parents who are looking for their services and who have varying availability schedules and preferences.

3): Verification and screening: It is assumed that parents desire to protect their children's welfare and safety. The program assumes that parents anticipate that nannies would go through a verification and screening procedure, which may involve reference checks, background checks, and even credentials like CPR training.

4): Maps and geolocation: It is assumed that the software will make use of maps and geolocation to assist parents in finding babysitters nearby. Users might anticipate seeing a map-based interface that shows the locations of available babysitters in relation to the parents' home.

5): Booking and Scheduling: It is anticipated that parents wish to use the app to easily book babysitters. The services that users demand include the option to view babysitter availability, pick suitable timeslots, and make appointments. It is anticipated that the software will handle conflicts, cancellations, and rescheduling well.

6): Payment processing: It is assumed that the app would provide parents with a safe and practical manner to pay babysitters for their services. Users anticipate connectivity with well-known payment gateways, enabling smooth and simple transactions.

7): User profiles: It is anticipated that both parents and babysitters will make thorough profiles on the app. Personal information, pertinent experience, qualifications, availability, and perhaps ratings and reviews from prior engagements are all frequently included in these profiles. Users anticipate having access to this data so they may make educated judgements.

3.4 Use case diagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. They are like the blueprints of the system. The use cases are represented by either circles or ellipses. It depicts what a user would do in proposed undertaken work. The cases of use are that parents can create accounts on the babysitting app. The Parent can look online to find babysitters who are available. The Babysitter has the option to accept or reject a parent's request for babysitting. The babysitter may alter their schedule of available times. The Administrator could add or remove babysitters from the app. The System notifies the appropriate actors of updates, requests, or other critical information by sending notifications.

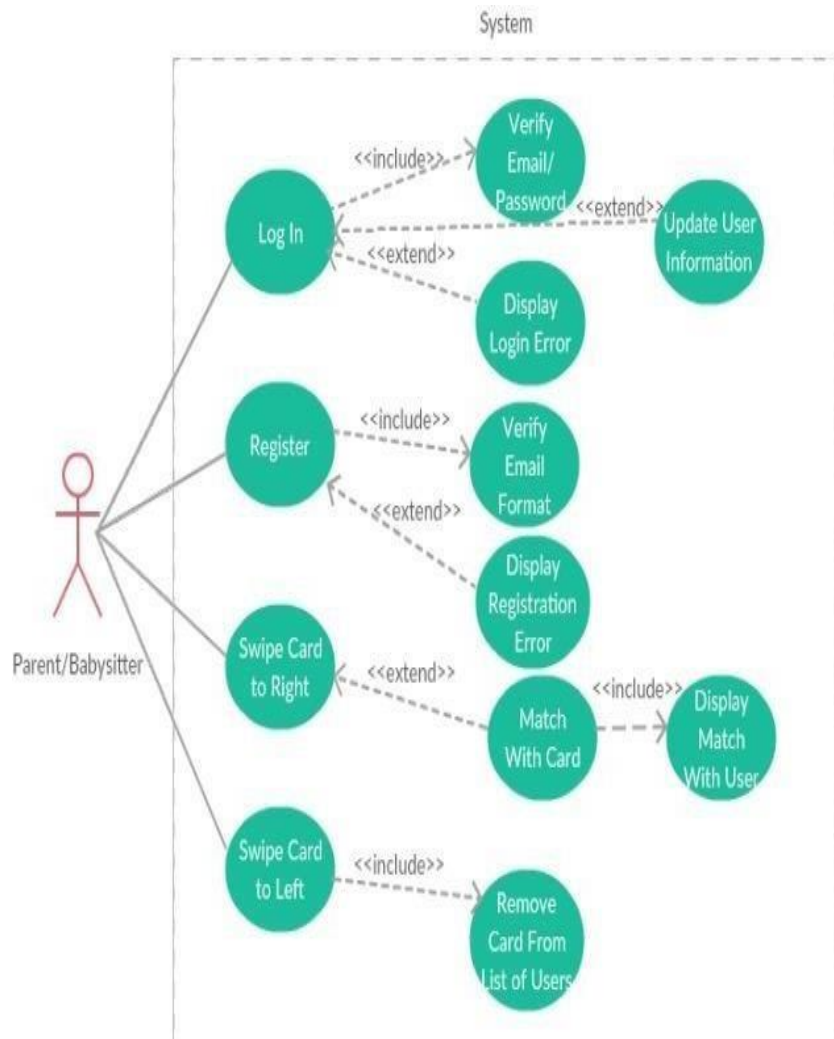


Fig 7 : Use case diagram of the system

3.5 Activity Diagram

A UML (Unified Modelling Language) diagram called a "activity diagram" shows the progression of actions or activities within a system or process. It is used to depict the activities, choices, and interactions of various system components or entities. In essence, it is a flowchart that shows how one activity leads to another. The action might be referred to as a system operation. One operation leads to the next in the control flow. This flow may be parallel, contemporaneous, or branched. Activity diagrams are used to document the implementation of system processes as well as to define business processes and use cases.

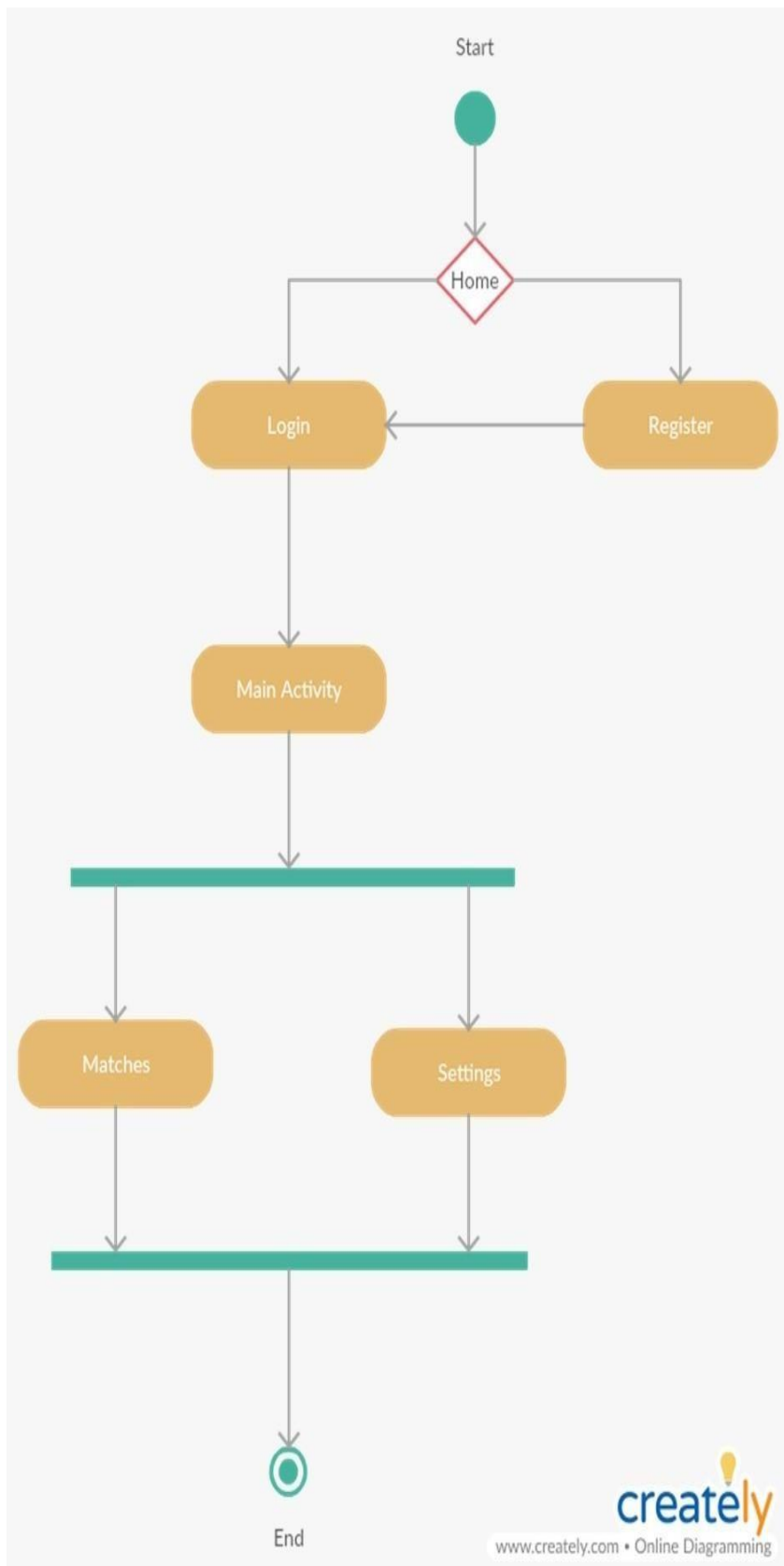


Fig 8: Activity diagram

3.6 Analysis

An examination of the project's technological viability should be done before designing a babysitter app. The needs for the app, including its features and functionalities, should be evaluated as part of this analysis. The analysis should also consider the technological limitations, such as the mobile platforms on which the app will run and the app's device compatibility. The cost of development, including the resources and time needed to create the app, as well as any potential security or data privacy concerns that need to be handled, are additional aspects to consider in the technical analysis.

Developers can detect potential issues or challenges that may arise during development and take action to address them before starting the project by doing a thorough technical analysis. This can ensure that the app is created successfully and efficiently, and that it satisfies the requirements of its consumers. For handling the backend of a babysitting app, Firebase might be a fantastic choice. The real-time database function would be especially helpful for keeping track of babysitting sessions because it will let parents and sitters see updates right away. A further degree of protection might be added by using the authentication feature to make sure that only allowed users can access the app.

Additionally, the hosting and cloud capabilities of Firebase can aid the app's performance and scalability, enabling it to manage a high amount of requests and users. This is crucial for a babysitting app since it must always be accessible and responsive. The server-side runtime environment Node.js, on the other hand, enables programmers to create scalable and quick web applications. With its extensive library of modules and packages, it is simple to design and construct a backend system that is tailored to the unique requirements of a babysitter app.

3.7 Design

This process entails gathering, analysing, figuring out, assembling, and assessing the application's functionality needs. Finding the application's key components, such as new user registration, signing in as a registered user, identifying the pick-up and delivery locations for current users, calculating the delivery cost, processing payments, and delivery tracking, is vital. Based on these specs, we created a system design that addresses the architecture, user interface, and database structure. Create a design paper outlining the technical requirements for the system. Xcode and Firebase (for backend services) are two crucial platforms and technologies that we must use to build the system. Create the user interfaces, integrate the back-end functions, and implement the system logic.

We must test our application after implementation. The main focus of the implementation phase is on building the platform using Swift, Firebase, and other relevant resources and structures. The backend services are built using Firebase, which provides functions including cloud computing, database storage, and authentication. Popular backend-as-a-service (BaaS) platform Firebase For mobile and web apps, Firebase provides a wide range of capabilities, including cloud storage, real-time databases, verification, and more. The following are some advantages of using Firebase in iOS projects is that using Firebase in OS projects has many benefits, including easy integration, scalability, real-time updates, and robust authentication. As part of the design process, this phase entails developing a technological plan for the product's delivery application.

This includes the architecture, user interface, and database schema. The structure of the system design includes descriptions of the backend services, APIs (Application Programming Interfaces), and user interface components of the system as well as how they interact. The user interface layout should follow design best practises in order to guarantee accessibility and simplicity. The database design should specify the data model, entity relationships, and data storage methods. The model was divided into three portions as part of the project's architecture and flow.

3.8 Parents Panel

For parents seeking for a babysitter, there is a consumer panel. They may use it to look up babysitters, filter the results, view their profiles, and reserve a sitter for a particular day and time. Parents can browse their previous reservations, rate and review babysitters, and pay for their bookings using the customer panel. Starting with the Launch Screen, this project has a number of modules and pages. The user interface of the application is made up of many, many screens, or "views". With different features and functionalities, each screen functions as a separate part of the mobile application.

The following screens could be found on the parents' panel.

Table 3.1 List of screens of Parent's panel

| S. No. | Screen Title | Functionality |
|---------------|---------------------|---|
| 1. | Launch Screen | This is the launch screen of the app i.e. the initial screen that is displayed when the app is opened by the user. |
| 2. | Welcome Screen | This screen welcomes the user to the app and gives a brief introduction to the product. |
| 2. | Login Screen | In this screen the user needs to enter the registered number and press submit and he will move to the next screen. |
| 3. | Register Screen | If the user is not registered (new user), he has to enter a mobile number to get himself registered. |
| 4. | Verification Screen | In this screen there is a user input of 6 digits in which the user needs to enter the OTP which he got via SMS on the entered mobile number. |
| 5. | Home Screen | This screen lets the user select whether he/she wants to look for a nanny, or babysit. This screen navigates which panel the user will be navigated to. |
| 6. | Location Screen | In this screen the user needs to enter the location where they need babysitting services |

| | | |
|-----|---------------------------|--|
| | | at. |
| 7. | User Profile Screen | This screen gives a personalized information of the account holder and helps user find their account information, notifications, reviews etc |
| 8. | Job Card Screen | This screen allows the user to select their requirements and specifications like timings, price range etc. |
| 9. | Select Nanny Screen | This displays all the nannies near the user that have been filtered out by the previous requirements filled by the customer. |
| 10. | Hiring Nanny Screen | Gives insight of the nanny selected, and lets the user hire or invite accordingly. |
| 11. | Payment Screen | After the nanny is hired, the user is redirected to this screen to complete payment. |
| 12. | Rating and Reviews Screen | This is a feature screen of this application where the user can rate and review the nanny for his/her service. |
| 13. | Notification Screen | This screen shows the notifications for the user. |
| 14. | Edit Profile Screen | The user can edit his/ her personal profile and update his/her details using this screen. |

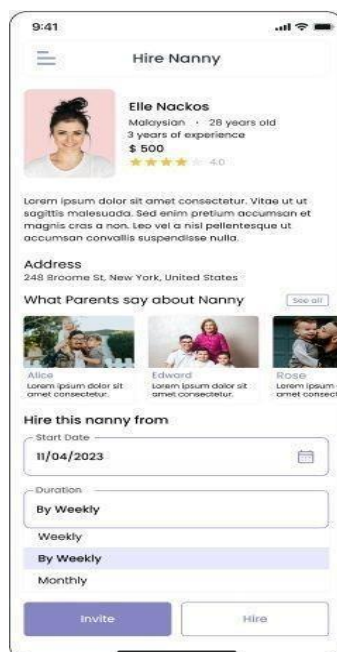


Fig 9: Parent panel

3.9 Nanny Panel

Babysitters looking for employment can use the nanny panel. They can make a profile, indicate their available times, and approve or decline appointments. Babysitters can contact with parents, examine their previous bookings, and get paid through the nanny panel. after being transferred from the home screen to the nanny panel.

Table 3.2: Lists of screens of the Nanny Panel.

| S. No. | Screen Title | Functionality |
|--------|----------------------------|--|
| 1. | Location Screen | In this screen the user needs to enter the location where they need babysitting services at. |
| 2. | User Profile Screen | This screen gives a personalised information of the account holder and helps user find their account information, notifications, reviews etc |
| 2. | Job Card Screen | This screen allows the nanny to fill in their details like hourly rate, availability etc. |
| 3. | Search Family Screen | Allows the nanny to search families which have registered accounts to provide long term services. |
| 4. | Application Screen | Allows the nanny to apply for a family |
| 5. | Requests Home Screen | Allows the nanny to get a list of the customers along with their details that have invited him/her. |
| 6. | Notification Screen | In this screen the nanny can see all the notifications/ communications |
| 7. | Ratings and Reviews Screen | This screen is for keeping track of the ratings and reviews the nanny has been awarded by previous clients. |
| 8. | User Profile Screen | This screen is the personalised account holder screen |
| 9. | Add account Screen | Lets the nanny create an account and add her bank details to receive payments from clients. |
| 10. | Edit Profile Screen | This screen enables the nanny to edit her public profile. |

| | | |
|-----|------------------|--|
| 11. | Set Price Screen | Allows the nanny to fix a price which will be publicly displayed for his/her services. |
|-----|------------------|--|

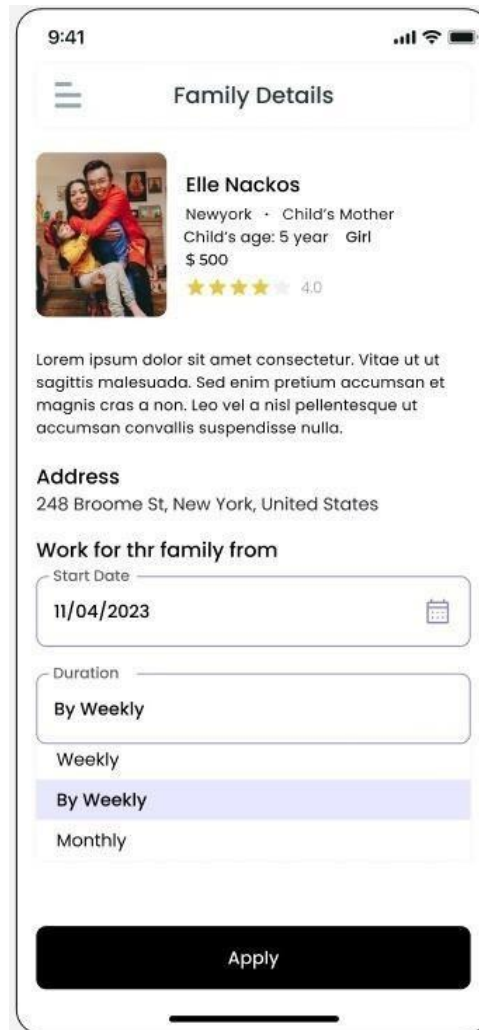


Fig 10: Nanny panel

3.10 Admin Panel

The administrators of the app, who oversee its operations, are the target audience for the admin panel. They may use it to examine and manage every appointment made via the app, keep track of payments, maintain the database of babysitters and clients, and settle any disagreements or problems that could come up. Administrators may examine analytics and create reports on app usage, user behaviour, and revenue using the admin panel.

The Admin Panel included the following screens:

Table 3.3 List of screens of admin panel

| S. No. | Screen Title | Functionality |
|--------|-------------------|---|
| 1. | Manage Users | Lets the admin manage both customers and nanny profiles. |
| 2. | Manage Complaints | Keeps a track of the complaints issued by the customers. |
| 2. | Payments | Keeps a record of all the payments made by the customers. |
| 3. | Customer Support | Keeps track of customer support. |
| 4. | Invoices | Keep a track of invoices. |

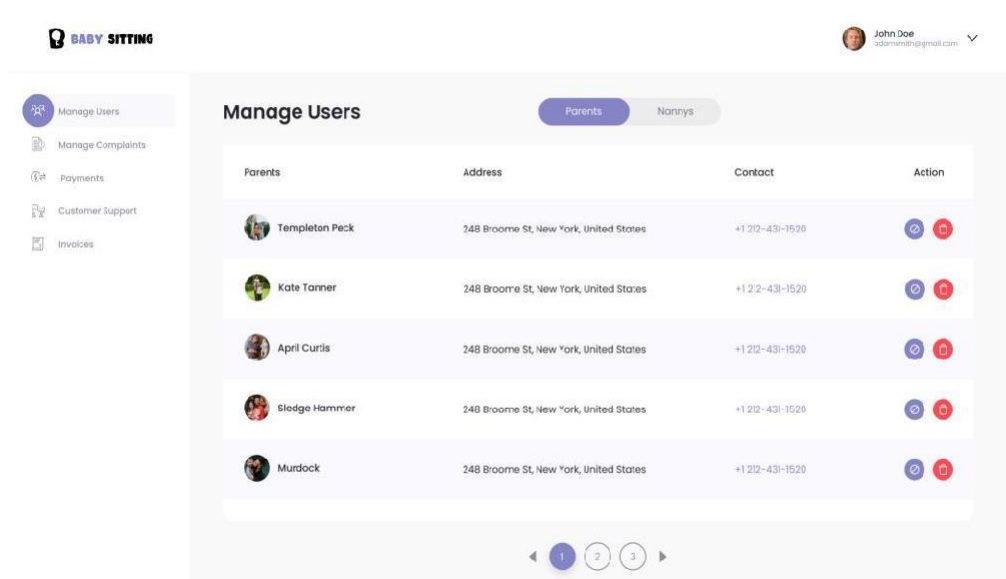


Fig 11: Admin panel

3.11 User authentication

Verifying the identification of a person or other entity trying to access a system, application, or resource is known as user authentication. It makes sure that only authorised users are admitted, keeping out unauthorised ones. It is a feature that all three panels share. If a person has never used this program before, they must sign up. If they have already registered, they only need to sign in. In all situations, the user can only verify their identity using their mobile number. On their registered mobile number, the user receives an OTP. OTPs are frequently time-based or event-based to increase security. Event-based OTPs change each time a new OTP is requested, whereas time-based OTPs change after a defined amount of time (for example, every 30 seconds). A mix of cryptographic methods and secret keys is typically used to generate OTPs, assuring their uniqueness and unpredictable nature.

```
implementation 'com.google.firebase:firebase-core:16.0.8'
```

```
implementation 'com.google.firebase:firebase-auth:16.2.0'
```

```
implementation 'com.google.firebase:firebase-database:16.1.0'
```

Fig 12: User authentication using firebase

3.12 Location, Pricing Selection

Moving further in this application after signing-in the user will get an option of in this module of our application the user have to fill following details:

- Address
- City
- Timings
- Price Range

The user is then routed to the make payment page after choosing the desired nanny. The user will receive a complete delivery invoice as soon as the source and destination points are verified, along with a payment page. The capability to add a new card for payment is available to the user. Fig 3.5 displays that functionality. The user will select the card from which he wants to make a payment each time. If the user wishes to add a new card, a dialogue box will open in the centre of the screen, requiring him to enter the card's information. The front-end properly verifies whether the card information is accurate or not. On the backend, Stripe is used to check whether the card details match those of an existent card or not. We will talk more about this technology in the next section of this chapter. The user of this app will also receive a variety of functions, such as the ability to view and amend their profile, rate, and review the nanny, examine order history, and receive notifications.



Fig 13: Location shown in the app

3.13 Implementation

Step by Step approach for the implementation model

Step 1: At first, we will download the XCode and then setting up of storyboard.

Step 2: Then we would be downloading the suitable libraries and Cocoapods wherever required.

Step 3: We would be creating the on-boarding screens.

Step 4: We would then be creating the login/register page for new users to register and registered user to login using their login credentials.

Step 5: Then we created a verification page where the OTP would be generated by using firebase authentication.

Step 6: Next we would be creating a page where the user would be choosing the profile – nanny or parent.

Step 7: Confirming about the user's location using google API is also a mandatory step.

Step 8: Then if the user chooses the parent profile -a page pops up where user could select a nanny of its choice, else if the user chooses to be nanny – a page will pop up to choose a suitable family to work with.

Step 9: As a parent, user could select a particular nanny by seeing its past reviews and same would happen in case of nanny that is it can view old ratings of the family (if they being rated by any).

Step 10: After selecting a nanny, the user acting as a parent would request nanny to babysit their small one and if nanny accepts the request, then they will be directly directed to payment gateway page where they have to pay accordingly.

Step 11: A feedback page also created where parents could rate about nanny they hired.

Step 12: A notification page is also being created for user acting as nanny as well as parent where they would get notifications from each other regarding like invitations, ratings.

Step 13: A payment page is created for nanny as well, where nanny would fill in its card details for receiving the payment.

Step 14: There will be an invoice page for nanny and along with it a profile page and profile edit page where user can view and edit their profiles.

CHAPTER-4

PERFORMANCE ANALYSIS

4.1. Introduction

There are several factors you may consider while doing a performance analysis of an iOS babysitter app. These are some crucial areas to evaluate like we have assessed the responsiveness of the program by touching buttons, scrolling through lists, or loading data to see how quickly it reacts to user activities. The user experience of the app should be fluid and free from any glaring latency or delays. We have also measured the amount of time it takes for the application to load its first displays and subsequent features. In order to avoid annoyance, users want applications to launch swiftly and display content rapidly. Then comes keeping a track of how much memory the application uses in various conditions.

Performance problems, such as sluggish reaction times, crashes, or other unexpected behaviour, can be brought on by excessive memory utilization and along with it evaluating how well the application performs in different network environments, such as Wi-Fi, 4G, or weak signal. Examine the app's handling of network-related tasks including sending and receiving messages and loading external data and making sure that while the network is functioning, the app is fast and gives the user the right feedback. Examining how the app affects battery life. Applications with low battery consumption are frequently desired since users want their devices to last all day. Along with it we must reduce pointless background operations, improve network usage, and make sure device resources are used effectively to cut down on battery consumption. Then evaluating the application's performance under load, or when many users or processes are running at once and testing the app's performance under demanding circumstances, such as a heavy user load or multiple simultaneous tasks.

Checking out the application's handling of faults and exceptions. The application should avoid crashes and data loss and should gracefully recover from faults. The user experience and overall performance of the software are improved by proper error handling. User testing sessions to get input from actual users on how well the programme performs is also important which also includes observing how people use the software, note any usability problems or bottlenecks, and gather comments for enhancements.

Along with-it Integrating analytics tools within the programme to collect information on user behavior, performance indicators, and any performance bottlenecks and utilizing these insights to pinpoint areas that require improvement and monitor the effects of performance enhancements. Comparing the app's performance to other childcare apps offered in the App Store is known as a competitor analysis. In terms of performance and user experience, we must evaluate their advantages and disadvantages and then examine for any areas where your app lags and take inspiration from their achievements.

4.2 SWOT Analysis

The SWOT analysis, on the other hand, is a tool for examining the overall strengths, weakness, opportunity, and threat of the project made. It helps identify the person what all advantages and disadvantages the project holds and then implement on the project further. It is a strategic planning technique which provides assessment tools to identify the core strengths, weakness, opportunity, and threats of the computer.

Here is a SWOT analysis of a babysitting iOS application:

1). Strengths: The app offers parents a simple way to look for and reserve babysitters, saving them time and energy in the process. The app connects parents with a variety of babysitters who have a range of availability, abilities, and certifications. The addition of a ratings and reviews system enables parents to make knowledgeable choices based on the experiences of other users, building responsibility and trust.

The app may put in place extensive screening and verification procedures for babysitters, protecting children's security and safety. By integrating geolocation, parents may find local babysitters, making it simpler for them to find qualified carers in their neighborhood.

2). Weakness: The app's operation depends on having connection to the internet, it cannot be used in locations with adequate network coverage or when there is a network outage. Depending on the area and user base of the app, babysitter availability may change locally. There may be fewer options available to parents in less populous locations. Despite safety precautions, there is still a chance of running into unreliable or dishonest babysitters, which could damage the app's trustworthiness.

3). Opportunities: In order to meet a wider range of customer demands, the app may decide to include services besides babysitting, such as tutoring, pet sitting, or housekeeping. Working with neighborhood companies, such as children's shops or places that cater to families, may lead to mutually beneficial alliances and focused marketing initiatives. Users can be drawn in and kept on the app by regularly adding new features, enhancing the user experience, and offering customization possibilities.

4). Threats: Existence of rival babysitter applications or well-known childcare providers in the market. Competitors might provide comparable or superior features, which could entice consumers away from the app. The app must abide by local laws and ordinances pertaining to kid safety, data protection, and babysitting services. Legal repercussions may result from breaking these rules. Poor user experiences, unfavorable reviews, or incidents involving babysitters may harm the app's reputation and discourage new users.

4.3 PESTLE Analysis

4.3.1 Political

This mobile app has no impact at all on the politics of a place. Furthermore, BabySitting Application has no influence whatsoever on any political parties.

4.3.2 Economical

Due to the use of the free products Xcode and Firebase, this project's implementation was made possible. The general public can get the smartphone app for no cost. Through the app, no monetary transactions are possible. The payment arrangement between the parent and babysitter should be worked out in a separate setting from the mobile app.

4.3.3 Social

The social repercussions include enabling parents and families to continue with their day at work, an event, or even a night out while knowing that someone is watching their child(ren) at home. A excellent way for babysitters to promote themselves and improve their chances of being hired by parents is through this smartphone app. Additionally, using the app to find babysitters is more handy than using the conventional methods of posting job openings for babysitters on newspapers or fliers because everyone is continuously using their cellphones in the digital age.

4.3.4 Technological

Because technology is constantly developing, we can nearly finish all of our daily tasks with a single click. This updated app, BabySitting, makes use of existing technology like IDEs, programming languages, and database systems to help parents connect with the best babysitters for their kids from the comfort of their homes using smartphones.

4.3.5 Legal

Since Google itself gave all the tools needed to develop the mobile app for free, it does not violate any regulations. Furthermore, because it is stored in a well-designed database, user information is kept private and safe.

4.3.6 Environmental

BabySitting environmental impact is minimal, except from the fact that it doesn't consume any natural resources or raw materials, which prevents any valuable resources from being depleted. Additionally, it doesn't release any harmful wastes or pollute the environment in any way.

4.4 Implementation details of the solution

4.4.1 Authentication Database

With the aid of Firebase Authentication, a feature provided by Google's Firebase platform, we can easily add user authentication and identity management to your applications. Even though Firebase Authentication primarily focuses on authentication capabilities, Firebase also offers a real-time database service called Firebase Realtime Database that may relate to Firebase Authentication to securely store user-related data.

Firebase Realtime Database, a NoSQL cloud-hosted database, enables real-time synchronisation and data storage for web and mobile apps. This JSON-based database's data is organised into a hierarchical structure of key-value pairs.

PhoneOTP

Authentication

Users | Sign-in method | Templates | Usage | Settings | Extensions NEW

Search by email address, phone number or user UID Add user Refresh More

| Identifier | Providers | Created ↓ | Signed in | User UID |
|---------------|-----------|------------|------------|---|
| +918558867664 | 📞 | 5 May 2023 | 5 May 2023 | tcXWVxkaFcZWNEGwq7I3KA6Knt... |
| +918559060991 | 📞 | 5 May 2023 | 5 May 2023 | 5X80JMSp1tbZ0r9ZLselaI3vRB2 |
| +918766260943 | 📞 | 5 May 2023 | 5 May 2023 | BUV23udgOLcUaOKoJlJWsk9glz1 Copy More |
| +917807444104 | 📞 | 5 May 2023 | 5 May 2023 | FZVQ06fIZ7YSeKnooHky9bwKMJ... |
| +918368332763 | 📞 | 5 May 2023 | 5 May 2023 | b5QyDDvD9APzm6xyFxF0bQqyL... |
| +916261932942 | 📞 | 5 May 2023 | 9 May 2023 | q2RpV9jkiGWgdKTsbiPowmLLSKR2 |

Rows per page: 50 | 1 - 6 of 6 | Previous Next

Fig 14: Authentication database

4.4.2 Realtime database

This real-time database synchronizes the data of numerous users. The database's structure can be seen in the tree below. The mobile app's registered users can be seen under Users. Whether a user is a parent or a babysitter is determined by the status attribute. Of course, it also keeps the user's name, phone number, profile picture, and description. The matches a user has with other users are likewise stored.

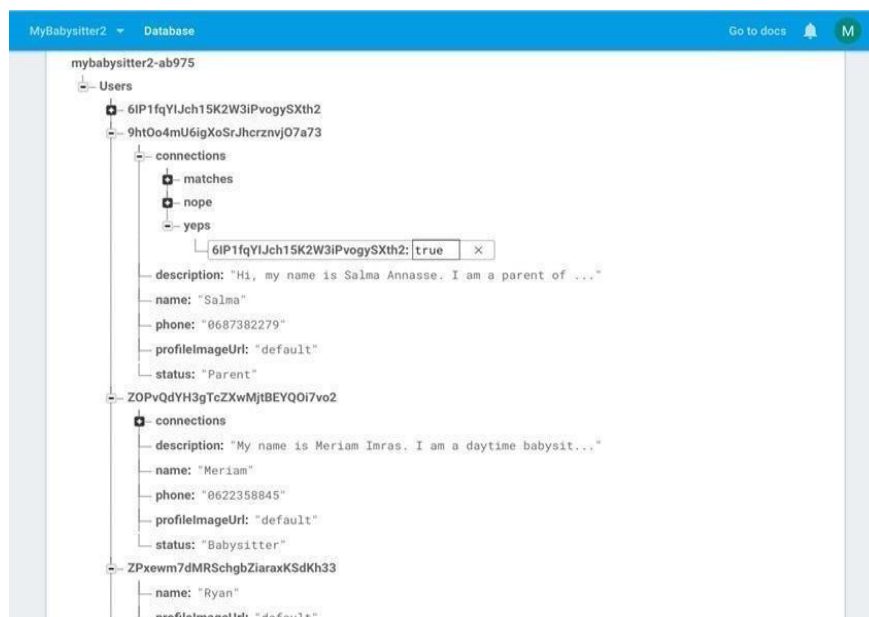


Fig 15: Realtime database

4.4.3 Launch Screen

A launch screen, commonly referred to as a splash screen, is a visual component that appears when a software programme or application is opened. It acts as the user's first visual cue that the application is loading or initialising. A launch screen's major functions are to provide users with quick feedback that the application is starting up and to make the transition from the app launch to the main user interface as seamless as possible. By decreasing the perception of waiting time, it aids in enhancing the application's perceived performance. Before the user sees the application's primary interface, launch screens are frequently shown for a brief duration, usually just a few seconds.

The launch screen shown below will appear when a user initially launches the mobile application.



Fig 16: Launch screen

4.4.4 Onboarding screen

Users are generally shown an onboarding screen when they first open an application, commonly referred to as a welcome screen or introduction screen. It assists users in getting started and provides a succinct overview of the features, functionality, and advantages of the program.

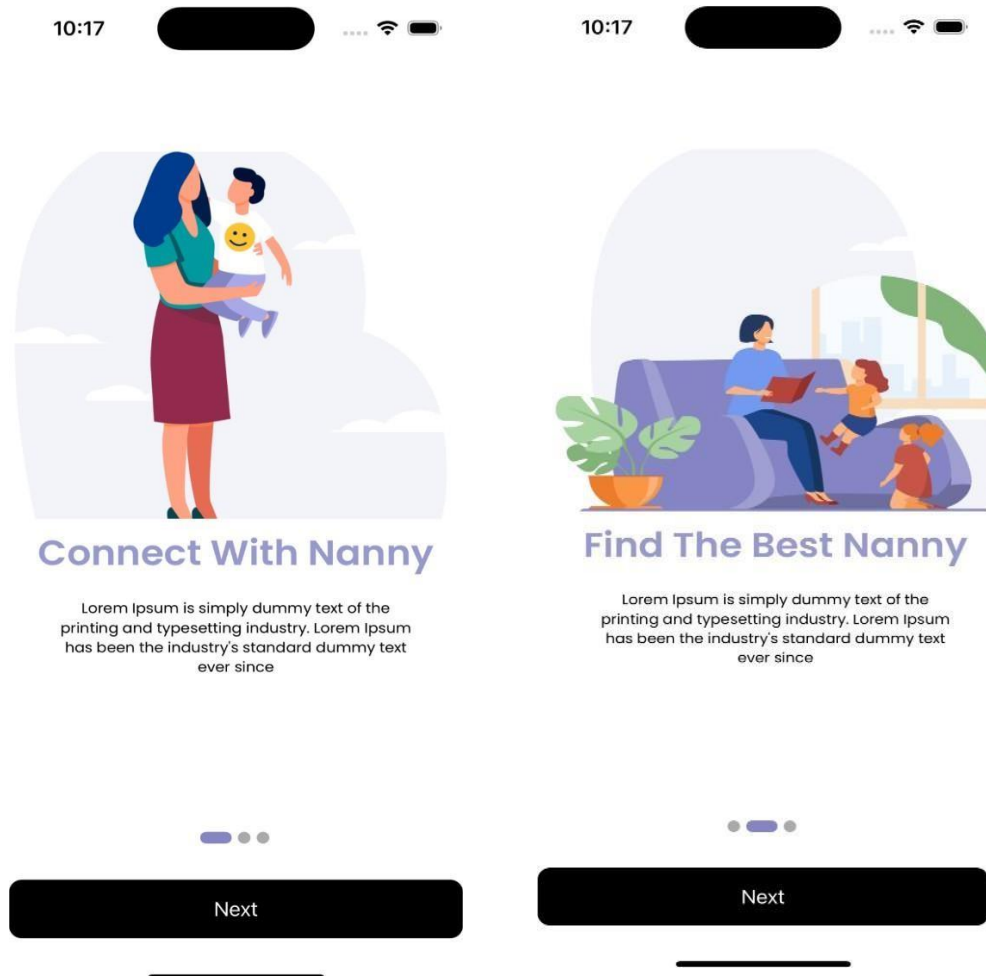


Fig 17: Onboarding Screens

4.4.5 Login page

The user is prompted to enter his email address and password on this page, which is the Login Activity page. If the user already exists, then the user can login by entering their respective login credentials or else if the user is new to the application, then the user must sign up. Below is the sign in page we have created for our application.

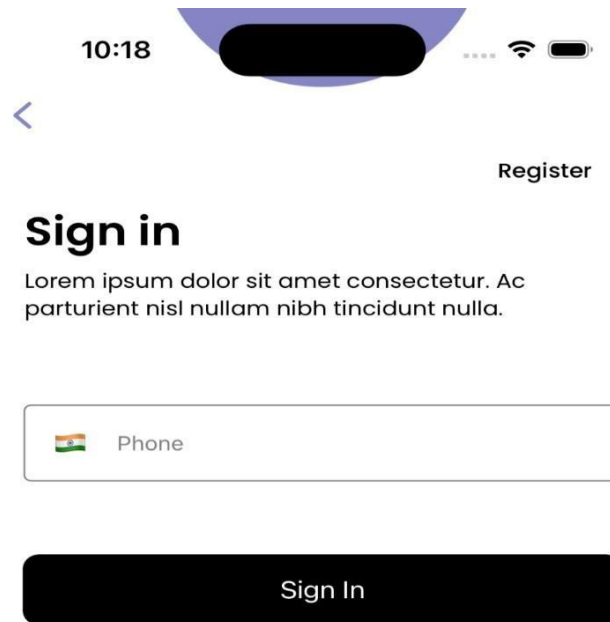


Fig 18: Login Screen

4.4.6 Register page

The User is then prompted to enter information such as Email and Password on the registration page. If the user is new to the application, then the user will register by entering its details or else if the user already exists, then on tapping the sign in button, the user can login.

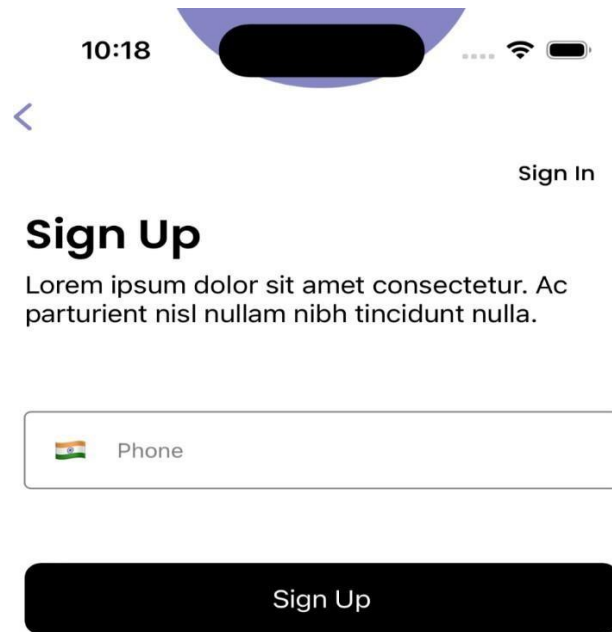


Fig 19: Register Screen

4.4.7. User Verification

A user's status with the company is periodically checked through the user verification process. It was put up to maintain the system's accuracy and security. User identification, authorization, and subsequent determination of a user's ability to do a certain action are all supported by the process of user verification. After entering the mobile number, the OTP would be generated using firebase authentication and below is the user interface design of verification page of the application.

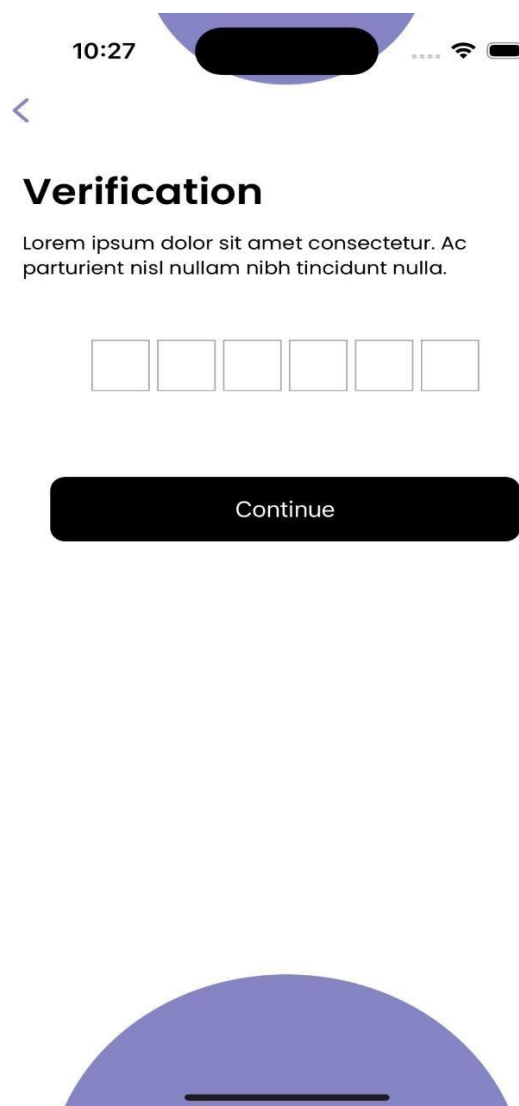


Fig 20: Verification Screen

4.4.8. Choosing profile – nanny or parent (done)

This page only allows the user to choose the profile that is whether the user wants to babysit as nanny or wants a babysitter for their children as a parent.

Here is the screen we have created:



Fig 21: Choosing Profile Screen

4.4.9. Nanny's job card description

A work description is a general breakdown of the obligations placed on a nanny. Depending on the demands of the individual family, there may be additional duties and particular criteria. If the nanny is new to the application, then nanny must mention all the details like address, pricing, city, time slots etc. Below is the job card being created for the application.

The screenshot displays a mobile application interface for creating a job card. At the top, the status bar shows the time 10:28, a blacked-out name, and icons for signal strength, Wi-Fi, and battery. The main header is a white bar with a back arrow on the left and the title "Job Card" in the center. Below the header are several input fields: a large "Address" field with a location pin icon on the right; two smaller fields for "City" and "Pin"; and a "Country" field. Under the heading "Timings", there are two fields: "From" and "To", each with a clock icon. Below the timings is a "Pricing" section featuring a horizontal slider. The slider is currently set to 110\$, with "0\$" at the left end and "500\$" at the right end. At the bottom of the screen is a prominent black button with the text "Continue" in white.

Fig 22: Nanny Jobcard Screen

4.4.10. Feedback page

Parents and babysitters can discuss their experiences, evaluate the service, and leave feedback on a feedback page in an iOS babysitting app. Along with it, we have added a rating system that enables users to express their opinions about the babysitting service, such as a star system or a number scale. This evaluation may consider elements like dependability, professionalism, communication, and general satisfaction. A "Submit" button has been included to enable users to provide comments. Additionally, we have given a feedback form so that people can provide specifics about their experiences. Encourage them to discuss any specifics, ideas, or problems they had with the babysitting service.

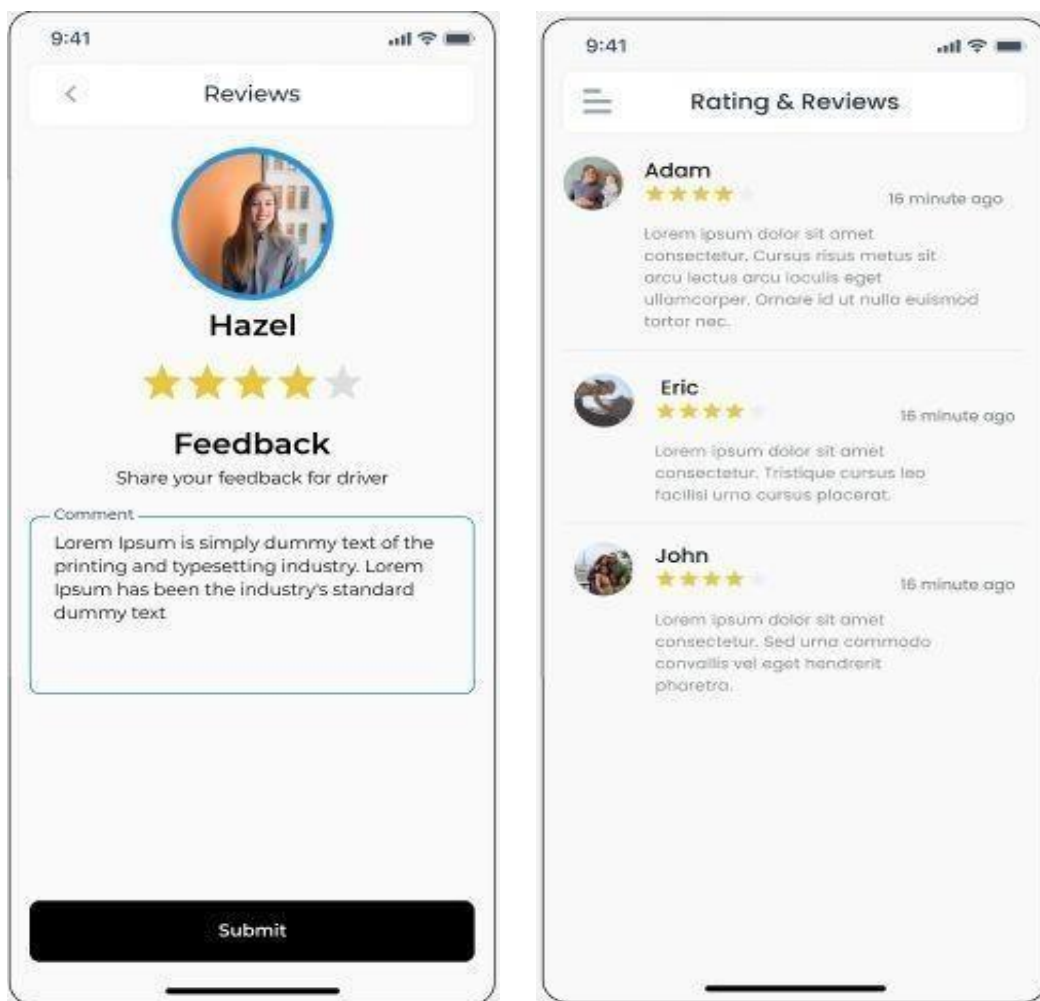


Fig 23: Feedback and Reviews Screens

4.4.11. Notification page

A notification page would normally act as the main hub of an iOS babysitting app to display significant notifications and updates pertaining to the babysitting services. The main portion of the page is the notification list, where notifications are shown in reverse chronological order. A title, message, timestamp, and optional icon or thumbnail pertaining to the notification type are normally included in each notification item. Depending on their type or relevance, notifications can be grouped. Booking inquiries, schedule adjustments, payment reminders, parent communications, and emergency notifications are typical categories in a babysitting app. Based on these categories, users can choose to filter or order their alerts.



Fig 24: Notification page

CHAPTER-5

CONCLUSIONS

5.1 Conclusion

The word "working mum" has become widely used today. In many families, both parents must work to support their families and build better futures for themselves. But this raises the age-old issue of who will take care of the kids. As a result, there is a rising need for nannies and babysitters.

With more people using mobile apps for everything from making appointments to placing food orders, there is a need for such an app. Homemakers and working women who require childcare support can also utilize the app. These nanny and babysitter search applications have a lot of advantages. It takes emotional elements out of the equation and lets parents select a carer they feel at ease with rather than putting all their faith in a single organization.

Currently, a mobile app for finders and sitters is essential. It saves the parents and the caregivers—who frequently struggle with the difficulties of finding a babysitter or nanny—time and money. Baby sitters are essentially people who look after young children temporarily while their parents are at work; working parents employ a baby sitter to take care of their young children. They oversee seeing to it that the children in their charge are secure, content, and amused. Babysitters are available for hire by parents and families to watch over their children for a predetermined period, such as nights, weekends, or special occasions. Finding and arranging babysitters can be made easier with a well-designed iOS babysitting app. It offers parents an easy way to look for reliable babysitters, check out their profiles, and schedule their services.

Comparing this convenience to more conventional means of hiring babysitters, parents can save a lot of time and work. When it comes to babysitting, safety comes first. In order to build confidence between parents and babysitters, a trustworthy iOS babysitting app can provide features like background checks, reference verification, and user reviews. This makes it possible for parents to feel secure and at ease when leaving their kids with a babysitter they located using an app and we have developed one such application. This mobile app's capabilities guarantee that both parties will discover the ideal user—whether that user is a parent or a babysitter—to negotiate a "deal" that meets their needs.

This would give parents a place to post their need for a babysitter and a platform for babysitters to promote themselves. This mobile app's capabilities guarantee that both parties will discover the ideal user—whether that user is a parent or a babysitter—to negotiate a "deal" that meets their needs. This report demonstrates how feasible it is to carry out this project. It was accomplished by acquiring the required equipment. creating accurate diagrams. maintaining a well-planned routine. putting the app into use in accordance with the diagrams. getting support when required. To ensure everything is in working order, we have tested it several times. Finally, we have to remember to take pleasure in the high points and draw lessons from the low points of this project's roller coaster ride.

Such an application can assist in connecting parents with certified and experienced babysitters given the rising need for dependable and trustworthy childcare. For parents to assure their child's safety, the app may have features like a secure payment system, background checks for babysitters, and real-time GPS tracking. In order to improve communication and foster trust between parents and sitters, the app may also provide features like scheduling, texting, ratings, and reviews. Overall, a babysitting app that is well-designed and user-friendly may give parents peace of mind and create a practical and effective platform for babysitters to find work.

5.2 Future Scope

The need for at-home services is growing as societies become more developed. This programme can be upgraded and improved in a variety of ways.

These potential future app scope points are listed below:

1). In-app video conferencing: As online learning and distant employment grow more common; parents may want to be able to check in on their kids and talk to babysitters through the app.

2). AI-powered matching: Using a babysitter's location, preferences, and availability, artificial intelligence algorithms may be used to connect parents with nannies. This can make it simpler for parents to locate the ideal nanny promptly and effectively.

3). Integration with smart home appliances: Babysitting apps may be able to work with smart home appliances like thermostats, webcams, and door locks, giving parents the ability to monitor and manage their children's environment from a distance.

4). On-demand services: A babysitting app might provide on-demand babysitting services for parents who require immediate or last-minute childcare, like other on-demand services like food delivery or transportation.

5). Educational resources: Parents and babysitters may be able to get information on child development, safety, and other pertinent subjects through babysitting applications.

6). Virtual activities: As more activities shift online, babysitting applications may provide children with interactive games, educational exercises, or storytime as virtual activities to perform with their babysitters.

REFERENCES

[A] Journals

[1] M. Suganya, Dayana R, Revathi.R, “Application for babysitter using React Native,” IJCET [2020]

[2] Mrugank Gandhi, Shubham Kothavade, “Development of a iOS Mobile Application for online Babysitters,” IRJMETS [2022]

[3] Yogita Masare, Sneha Mahale, Manjusha Kele, Ashvin Upadhyay, Bhushan R. Nanwalkar, “Subscription based service for Babysitter,” IJERT [2021]

[4] Vrushali C. Waikar, Sheetal Y. Thorat, A. A. Ghute, and Priya P. Rajput, Mahesh S. Shinde, “Connecting Families with Caregivers Application for Android Mobile Application,” IRJET [2020]

[5] Ramesh D, Vishnu B. Vardhan, “Development of a Android Mobile Application for Babysitters,” IJARCEE [2013]

[6] Noof A. Al-Safi, Rawan A. Al-Asiri, Malak A. Al-Malki, Sameera Abar, “Fostering Childcare E-Service: Design and Development of a Software Application,” IJITN [2022]