# 

**CHAPTER 1**

**BACKGROUND & MOTIVATION**

# 1.1 INTRODUCTION: AN OVERVIEW OF THE SYSTEM

Our group created an Android Application that deals with this situation. This Daycare App will help Parents to stay connected to their children. Parents like having access to information about their children so having this app will give them peace of mind knowing that information is there whenever they need it. Giving them a way to pay you with just a few clicks on their device, will reduce the hassle of paying childcare fees. By looking at our daycare app features, parents will be able to know if they are a good fit for the app and daycare. Our app is also convenient for preschool and kindergarten teachers. They are an excellent tool to use in the classroom. It promotes making kids learn new abilities and skills and make them future-ready.

# 1.2 BACKGROUND: EXISTING SYSTEM

In today's era, the world seems to be run on apps. Well-designed apps are eye-catching and make you want to use them, whilst poorly designed ones are just frustrating. While some basic features such as usability, functionality, and layout facilitate the user to have an excellent experience, one of the most complained about issues with apps is the features, be it lack of expected ones or bad use of those.

The particular features for circumstances that lack similar existing apps are:-

1. Lesser Communication

2. Speed

3. Schedule and Data Keeping

4. Flexibility

5. Digital Payment Options

6. User Friendly

To sum up, everything that has been stated so far, Here comes our app which makes it unique.

# 1.3 MOTIVATION

It’s no secret that the COVID-19 pandemic has changed the state of play for marketers irreversibly, with society covering a “decade in days” with regards to digital adoption.

Although we’ve seen some degree of normality in 2021, markets are still unpredictable making it increasingly difficult to plan for the year ahead.

Due to the current scenario of the pandemic, parents are now busy with their jobs and going out leaving their kids alone at home unattended. Handing them off to someone who doesn’t know them, such as a nanny, might be scary.

If a child is starting school or daycare for the first time, the parents might be a bit apprehensive. After all, they have never been away from their children. Therefore for this purpose, we have done the research for such parents to put together our simple overview of the app which is more convenient for preschool and kindergarten teachers.

This motivated us to develop the Day Care app that will facilitate the parents to stay connected with their kids using the app.

# 1.4 PROBLEM STATEMENT

* Daycare software that offers an **easy-to-use communication** system is one of the top priorities you should look for when choosing a daycare software.
* It creates a **safer environment** for kids eliminating various social problems.
* Daycares probably have many children to look after each day so being able to input data quickly and easily is important. By using the app the staff can **spend less time on “paperwork**” and more time looking after kids.
* If you have an app that can help parents pay for daycare services and **schedule payment reminders**, it will make it easier for Daycare and the parents of the kids you look after.
* This app provides many new features which will be **user-friendly** and **easy** to use.
* It enhances the chances of learning **new abilities and skills** in kids.

# 1.5 ADVANTAGES

* Child Care Business Management
* Parent Engagement
* Secure & Contactless
* Tuition Collection and Payment
* Staff Management
* Track Attendance
* Monitor & Report Staff-Child Ratios
* Drop off & Pick up
* Online Registration
* Early Childhood Learning

**CHAPTER 2**

**LITERATURE SURVEY**

**2.1 INTRODUCTION**

A literature survey or a literature review in a project report is that section that shows the various analyses and research made in the field of your interest and the results already published, taking into account the various parameters of the project and the extent of the project. If a child is starting school or daycare for the first time, the parents might be a bit apprehensive. After all, they have never been away from their children. They know their routine and know how to handle their moods. Handing them off to someone who doesn’t know them such as a daycare provide might be a bit scary How will it be for them, would they be happy there? What if there was a quick and easy way to get updates on your child during the day if you start feeling anxious?

For this purpose, our group created an Android Application that deals with this situation. This Daycare App will help Parents to stay connected to their children. Parents like having access to information about their children so having this app will give them peace of mind knowing that information is there whenever they need it. By giving them a way to pay you with just a few clicks on their device, it will reduce the hassle of paying childcare fees. By looking at our daycare app features, parents will be able to know if they are a good fit for the app and daycare. Our app is also convenient for preschool and kindergarten teachers. They are an excellent tool to use in the classroom.

**2.2 RESEARCH PAPERS.**

**Paper 1**

# Title: An Ergonomic Approach on Design of Daycare Centers in the Philippines

# Author’s: [Ma. Janice J. Gumasing](https://ieeexplore.ieee.org/author/37086399526); Kayla Katriz Mae C. Antonio

**Dept. name of the Organization:** School of Industrial Engineering and Engineering Management, Mapua University, Manila, Philippines

A Daycare center is a nursery place for the supervision of children ages 2-6 years old while the parents are at work. This is one of the places where children build and enhance their characters and skills. Having a comfortable and safe environment for both teachers and kids is very important especially because daycare centers or schools are their second home. Currently, daycare centers in the Philippines do not have a standard design for facilities, only the public classrooms under DepEd have a standard design [1]. Additionally, according to a previous study, the Philippines' daycare facilities and their resources are insufficient, even basic resources are lacking, and facilities are low because the facility ratio to the kids does not meet the acceptable minimum limit [2]. As a result, many issues and problems are emerging in current daycare centers in the country. To address this, the researchers aim to propose an ergonomic design for daycare centers. Principles of ergonomics were used in this study such as anthropometry and work environment design to determine the possible recommendation for the daycare facility design. Also, design tools were used to support the analysis of the study such as Quality Function Deployment (QFD), Failure Mode Evaluation and Analysis (FMEA), 3D design modeling, and cost-benefit analysis. By analyzing the data gathered from the study, the researchers were able to develop a standard design for daycare facilities based on the recommended measurements from 150 subjects in the study.

**Paper 2**

# Title: Decision Support System for Choosing Daycare in Surabaya City Using Analytical Hierarchy Process (AHP)

**Author’s:** [Ira Prasetyaningrum](https://ieeexplore.ieee.org/author/37086077096); [Choirunnisa Lutfi Hariyati](https://ieeexplore.ieee.org/author/37086870429)

**Dept. name of the Organization:** Department of Informatics and Computer Engineering Polotiknik Elektronika Negeri, (PENS)

Economic problems led to changes in women's roles where women are choosing to work, so they cannot always be with the children and tend to choose child care in daycare when they are working as a solution. The routines make them lack any information about daycares, especially profiles and locations. So parents need an information system that can provide daycare information and advise the decision. The purpose of this research is to provide a system that gives information on each daycare and daycare recommendation so that it can be used by parents. In this research, a decision-making system was conducted using Analytical Hierarchy Process (AHP), which is a multi-criteria decision-making method. A recommendation is made by user's rank. The conclusion of this research is that the system is able to provide information and recommendations about daycare and able to meet the needs of parents.

**Paper 3**

# Title: Nursery Science: A Mobile Application for Learning Science for Pre-Schoolers

**Author’s:** [Clifford A. Calleja](https://ieeexplore.ieee.org/author/37087089436); [Paul John E. Candelaria](https://ieeexplore.ieee.org/author/37087091643)

**Dept. name of the Organization:** National University- Manila, College of Computing and Information Technologies

Mobile Learning or M-Learning has been gaining attention in research to determine its impact on the learning experiences of the students. The use of technology, particularly of mobile learning, is essential in education due to its capability to provide individual learning experiences. M-Learning can be defined as Learning using mobile devices like Smart Phones and Tablets. This platform can be integrated into Blended Learning where students learn inside and outside the four walls of the classroom. Given the advantages of M-Learning for students, this study sought to design and develop a mobile application that helps preschoolers enhance their learning experience in science in one Daycare Center in the Philippines. The application includes lessons from the Department of Social Welfare and Development (DSWD). Quizzes as a form of assessment can be taken after each lesson. Succeeding lessons are locked and cannot be accessed until the teacher unlocks these lessons. The mobile application was assessed using ISO 9126 to determine the acceptability among the teachers and parents of the pupils of the Daycare Center. The overall result yielded an average mean of 4.0 which is verbally interpreted as Very Satisfactory. A longitudinal-experimental study for the mobile application is recommended to further assess the impact of the application on the learning experience of the pupils.

**Paper 4**

# Title: Project Management Skills Needed for Mobile App Development

# Author’s: Dr. Randall McCoy

**Dept. name of the Organization:** College of Business and Technology Morehead State University Morehead, KY 40380, USA

The problem that this study deals with is the need for research to determine some of the competencies needed by mobile application (app) developers in the year 2020 and beyond. This study uses a Delphi instrument to determine the general needs for the preparation of technology competencies of app developers for the near future of the 21st century. The panel of experts contributing to the data includes participants from the International Association of Internet Professionals (IAIP), and top mobile app developers recognized internationally by their peers.

**Paper 5**

# Title: Child development: comparison between children who attend or do not attend public daycare centers.

# Author’s: [AMARO, Lívia Lúcio de Mattos](http://pepsic.bvsalud.org/cgi-bin/wxis.exe/iah/?IsisScript=iah/iah.xis&base=article%5Edlibrary&format=iso.pft&lang=p&nextAction=lnk&indexSearch=AU&exprSearch=AMARO,+LIVIA+LUCIO+DE+MATTOS)

**Dept. name of the Organization:** Study Design and Scientific Writing Laboratory

**OBJECTIVE**: To evaluate and compare the development of children attending public nursery school from their peers who remain only in the home environment **METHODS**: We performed an observational, cross-sectional study with 167 children aged between 11 and 57 months from Alto Vale do Jequitinhonha, Minas Geris, Brazil. The subjects were divided into two groups (nursery and home areas’), homogeneous in age, gender, socioeconomic status, and maternal education. The development areas: personal-social, language, and fine and gross motor were evaluated through the Denver II test and the quality of environment of the five public nursery school ' through the Infant/Toddler Environment Rating Scale-Revised Edition (ITERS-R). We used the chi-square statistical test for comparison between groups **RESULTS**: There were no significant statistical differences between groups for the overall test result for Denver II, nor for the different areas of the test. The environments presented daycare quality between 'inadequate' and 'minimal'. Thus, there was no difference between attending a low quality daycare or just staying in the home for child development **CONCLUSION**: No difference was observed when comparing the development areas: personal-social, language, and gross and fine motor of child who attend public nursery school and their peers who remained only in the home environment. The daycare environments analyzed showed quality considered 'inadequate', which may have influenced the results as the literature has pointed out the need for a quality childcare environment to positively influence child development

**Paper 6**

# Title: Free daycare policies, family choices and child development

# Author’s: [AnnaBusse](https://www.sciencedirect.com/science/article/abs/pii/S0167268120302857" \l "!) ChristinaGathmann

**Dept. name of the Organization:** University of Heidelberg, Bergheimer Strasse 20, 69115 Heidelberg, Germany

Over recent decades, many governments have expanded state-subsidized daycare for preschool children. Using the staggered introduction of free universal daycare for children between the ages of two and six, we show that free daycare increases attendance only among 2-3-year-old children. There is no effect for older children for which attendance rates have been high even before the reforms. Similarly, labor market attachment increases for mothers with 2-3-year-olds, while mothers of 4-6-year-olds respond mostly at the intensive margin. Short-run development for the average child is largely unaffected by the reforms. Responses and short-run benefits are much more pronounced for poorer and low-skilled families than for the average household. Hence, despite being a universal program, free daycare helps to level the playing field for children from disadvantaged backgrounds – provided the policy is focused on age ranges with low prior attendance.

**Paper 7**

# Title: The impact of daycare programs on child health, nutrition and development in developing countries: a systematic review

**Author’s:** [Jef L. Leroy](https://www.tandfonline.com/author/Leroy%2C+Jef+L) [Paola Gadsden](https://www.tandfonline.com/author/Gadsden%2C+Paola); [Maite Guijarro](https://www.tandfonline.com/author/Guijarro%2C+Maite)

**Dept. name of the Organization:** International Initiative for Impact Evaluation

The authors conducted a systematic review of impact evaluations of daycare interventions on the health, nutrition and development of children under five in low-income and middle-income countries. The review was limited to intervention studies controlling for potential self-selection of beneficiaries. Papers were excluded based on study scope, type and quality. Only six studies (conducted in Latin America) met the inclusion criteria. Large positive effects on child development were found. Because of the limited evidence, no conclusions can be drawn regarding the impact on child health and nutrition. More rigorously conducted studies on the impact of daycare programs in developing countries are needed.The reviewed studies suggest that daycare programs have a positive impact on child development. The evidence on child nutrition and health outcomes is less clear. More rigorously conducted studies on the impact of daycare programs in low-income and middle-income countries are needed. These should be conducted in a variety of settings and provide a clear description of the counterfactual care scenarios. There is a particular need for studies documenting the impact on health outcomes. These studies should use children not exposed to the program as the control to avoid the problem described for the Colombia study. Studying how child age changes the (relative) importance of each pathway (for example, the trade-off between income and care) would provide important insights as well. Additionally, the extent to which the characteristics of the care provided modify its effect should be studied. Finally, future studies should consider alternative evaluation designs that minimize the risk of selection bias.

**Paper 8**

# Title: Daycare Worker Crimes that cause Injuries to Children

**Author’s:** [Russell Button](https://www.buttonlawfirm.com/bio/russell-button.cfm)

**Dept. name of the Organization:** The Button Law firm

**Criminal Charges Are Separate from Civil Claims**

If your child is the victim of physical or sexual abuse, the state will file the appropriate criminal charges. In many cases, the perpetrator will be charged with a felony and face the possibility of several years behind bars. Your child’s name will not be released to the public, and special precautions will be taken if your child is required to provide a statement to the court.

Unfortunately, criminal cases involving the physical or sexual abuse of a child can be difficult to win. Often, there is little direct evidence, or the child’s injuries aren’t definitively an indication of abuse. However, you still have a right to file a civil claim on behalf of your child even if the responsible party is found not guilty of the criminal charges.

**Civil Case Have a Lower Burden of Proof**

In a criminal case, the defendant must be found guilty beyond a reasonable doubt. In a civil case, the plaintiff must only prove that the defendant is guilty by a preponderance of the evidence. This simply means that it is more likely than not that the allegations are true.

A criminal conviction can be used to support your civil case, but you may still have a strong case even if the criminal charges against the daycare worker who abused your child were unsuccessful.

**The Daycare center may also be Liable**

The person who directly harmed your child may not be the only party who can be held legally responsible. Daycare centers, whether they are run as an in-home operation or as a standalone facility, are required to take steps to provide a safe and secure environment for the children in their care. This includes actions to prevent physical or sexual abuse.

* A daycare facility has a responsibility to conduct thorough background checks to ensure that workers do not have a history of mistreating children or other vulnerable individuals.
* The facility is required by law to implement a number of security practices that reduce the risk of abuse, such as policies that reduce the time where adults are alone with children.
* All daycare employees are mandatory reporters who are legally required to report suspicions of child abuse to the Texas Department of Family and Protective Services within 48 hours. This includes abuse by other employees of the facility or by relatives of those associated with an in-home care provider.

When a daycare fails to uphold its duty of care to the children it serves, it can be named as a defendant in a civil suit.

**Paper 9**

# Title: Policy for daycare centers soon: govt.

# Author’s: Alok Deshpande

**Dept. name of the Organization:** The Hindu

Minister of State for Home Ranjit Patil told the Legislative Assembly on Friday that the Ministry of Women and Child Welfare has been asked to come up with a policy for supervision of and training for day care centers and their staff members. “The policy will include inspection of daycare center staff, listing of food being given to the kids and installation of CCTVs. It has also been observed that these centres, though specialised in taking care of children, can operate with just a shop license; no other criterion is applied. The whole procedure will be made stringent through the policy,” he said.

Council chairman Ramraje Naik-Nimbalkar too directed the State government to take measures. “This is clearly a commercial activity, and in that case the entire staff must be specially trained to carry out this job of taking care of children. The State government can also bring special amendments in the IPC to tackle such heinous crimes. Also, the police officer who doesn’t register the crime immediately should be suspended,” he said.

Shiv Sena MLC Ravindra Fatak, who raised the issue, asked if other children at Purva Daycare and Play School were drugged, as the video footage shows them sleeping when the employee was assaulting the infant. Mr. Patil accepted that the police did search for the possible use of drugs, which turned out to be false.

Dr. Neelam Gorhe of the Sena demanded action against Kharghar police station officers who initially refused to register the complaint, instead asking the aggrieved parents to produce CCTV footage. “Yes, the actions of the concerned police officer are suspicious. We have ordered an inquiry against him and strictest of action will be taken following the completion of the inquiry,” Mr. Patil said.

**Paper 10**

# Title: Daycare: Revised Edition

# Author’s: Alison Clarke-Stewart

**Dept. name of the Organization:** Harvard University Press Cambridge, Massachusetts 1993

**The problem:**

Twenty-five years ago in the United States, the ratio of men to women who showed up at the doctor's office with ulcers was twenty to one. Today it is two to one. These statistics reflect significant changes in women's lives over the past two decades, changes that have brought more and more women into the workforce and require them to perform a delicate balancing act as they juggle the demands of home, family, and work.

Today most women with children are stretched thin, working at two full-time jobs. They spend forty hours a week in employment outside the home, another thirty six hours at home looking after the house and children, and in between they commute. The average worker now spends ten hours more a week commuting and working than a generation ago. As one mother reported, "I'm always on edge because I'm so far away from my children. If there were an emergency, it would take me over an hour to get home."2

These women must plan and schedule their activities rigidly. They may do less housework than they would if they were not working; they may complete their house hold chores in less time. But they still spend almost as much time taking care of their children as they would if they were not employed. The tasks of bathing, feeding, dressing, teaching, and transporting children do not dis appear just because the mother is employed.

Although husbands may help with the housework and children, there is no equal sharing of tasks between spouses. The primary responsibility and time commitment in most families is still the mother's, and even if she has hired help to assist with these tasks-a cleaning person, housekeeper, laundry service, daycare provider it is still her responsibility to arrange for and manage such assistance. Three quarters of the mothers employed in full-time jobs claim they don't have enough time to do everything. They cope with the situation by giving up time for themselves, spending fewer hours on hobbies, reading, gardening, socializing, and other personal pleasures; they have no free time at all.

This double duty of homemaker and job taker takes its toll in physical and psychological stress. As a result of their constant efforts to deal with the demands of family and job, most mothers holding full-time jobs feel tired and overworked. Many feel lonely and isolated. All feel harried. There are compensations, of course. They feel good about getting out into the world and having new experiences. They get more satisfaction from outside work than nonworking women do from housework. They feel better about themselves as individuals and as competent achievers. They feel healthier, more independent, and in control of their own lives. But they also feel rushed, tied down, and under pressure, especially when they are first adjusting to their dual role. These problems are especially acute for mothers with limited resources. Mothers with low incomes must deal with the stresses of inadequate living conditions and concerns about how the family will survive, in addition to the stresses of combining work and parenthood. Single mothers must cope with the demands of work and children alone, enduring the physical strain of doing everything by themselves, the psychological strain of isolation, loneliness, and bitterness, and the economic strain of limited family income. Even with adequate resources and a partner's help, however, working mothers experience stress as they cope with physical fatigue, with conflict between their work and their husband's, and with the challenge of balancing their maternal, wifely, and working roles

**Paper 11**

# Title: Kharghar creche fallout: Top cop cracks down on child day care centres

**Author’s:** [George Mendonca](https://timesofindia.indiatimes.com/toireporter/author-George-Mendonca-479210199.cms)

**Dept. name of the Organization:** The Times of India

NAVI MUMBAI: The Navi Mumbai police commissioner, using his discretionary powers, on Friday issued a notification instructing all child care centres to implement several precautionary measure after a 10-month-old girl was severely thrashed by a caretaker at a day care center in Kharghar. Police commissioner Hemant Nagrale said, "It has been made compulsory for all establishments of child day care centres to install CCTV cameras on their premises. They should be Internet Protocol (IP) address-enabled and with a provision of at least 30 days back up of the CCTV footage. The CCTV camera's user ID should be linked to the computer system or mob phones of the parents of the children so that they have direct access to the recordings. Simultaneously, the centre's authorized person should check the CCTV camera recording every 24 hours and if any objectionable content is observed, it should be brought to the notice of the local police station."The circular was issued under the Criminal Procedure Code, 1973 under section 144 (2) which empowers the police chief to take necessary steps in urgent cases of nuisance or apprehended danger. "All authorized persons of crèches, play schools, child day care centres should get the police verification done for the character certificate of their staff within 90 days of issuing of this notice, which should be the in-time and out-time of the child at the centre. Even the in-time and out-time of the child at the daycare center should be maintained and the acknowledgement of parents obtained at the time of handing over their child. Those not abiding the directive will face action as per the provisions of the law," Nagrale added. All private play schools, creches, child day care centres and individuals baby-sitting in residential premises will have to implement the measures.

**Paper 12**

# Title: Mobile e-services using HTML5

**Author’s:** [**Karl Andersson**](https://www.computer.org/csdl/search/default?type=author&givenName=Karl&surname=Andersson)**,** [**Dan Johansson**](https://www.computer.org/csdl/search/default?type=author&givenName=Dan&surname=Johansson)**,**

**Dept. name of the Organization:** Pervasive and Mobile Computing Laboratory, Department of Computer Science, Electrical and Space Engineering, Luleå University of Technology, SE-931 87 Skellefteå, Sweden

New mobile networks and new devices like smartphones and tablets are rapidly changing opportunities for public sector units delivering smart, mobile e-services to their citizens. Moreover, the upcoming HTML5 standard allows for cross-device and cross-browser support making service development and deployment much more easier than before at lower costs. This paper analyzes the most important features of HTML5, CSS3, and WebGL and their applicability for mobile e-services in the public sector. Also, we present a novel architecture for mobile e-services using HTML5, mashups, and RESTful web APIs as important building blocks.

**Paper 13**

# Title: A Cross-platform Application Development Environment Supported by Cloud Service

**Author’s:** [Baixing Quan](https://www.computer.org/csdl/search/default?type=author&givenName=Baixing&surname=Quan)   
[Tianzhou Chen](https://www.computer.org/csdl/search/default?type=author&givenName=Tianzhou&surname=Chen)   
[Hongjun Dai](https://www.computer.org/csdl/search/default?type=author&givenName=Hongjun&surname=Dai)   
[Bin Peng](https://www.computer.org/csdl/search/default?type=author&givenName=Bin&surname=Peng)   
[Minghui Wu](https://www.computer.org/csdl/search/default?type=author&givenName=Minghui&surname=Wu)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

Along with the mobile Internet growing up, mobile terminal device are becoming more and more multiplicity and the application development are become more difficult. In this paper, we propose a cross-platform application development environment supported by cloud service. This development environment abstract hardware capability makes developers needn't consider the hardware of target device, and it can generates different platform applications which run on different devices and service application run on the cloud. The cloud has the ability to provide service to four kinds of mobile device operating system and four terminal devices. At last, we implement the prototype of the application development environment and a campus application based on cloud.

**Paper 14**

# Title: A Framework for Cross-Platform Mobile Web Applications Using HTML5

**Author’s:** [Christos Bouras](https://www.computer.org/csdl/search/default?type=author&givenName=Christos&surname=Bouras)   
[Andreas Papazois](https://www.computer.org/csdl/search/default?type=author&givenName=Andreas&surname=Papazois)   
[Nikolaos Stasinos](https://www.computer.org/csdl/search/default?type=author&givenName=Nikolaos&surname=Stasinos)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

In recent years the use of smart mobile devices has become an integral part of everyday life leading to the expansion of applications development for the various mobile platforms. Each of these platforms requires separate software development process, which subsequently increases dramatically the corresponding effort. With the emergence of HTML5 these issues can be addressed efficiently since application development is allowed in a cross-platform manner. An additional important benefit is that users can access the application immediately without any need for installation. In this paper, we investigate the potentials of mobile application development with web technologies and we present a development framework that utilizes the most important state-of-art web technologies for the support of mobile devices. This framework can be used for the implementation of mobile web applications and also for the investigation and experimentation on the main features that HTML5 offers for this specific type of devices.

**Paper 15**

# Title: Design of Rich Client Web Architecture Based on HTML5

**Author’s:** [Li-Li Chen](https://www.computer.org/csdl/search/default?type=author&givenName=Li-Li&surname=Chen)   
[Zheng-Long Liu](https://www.computer.org/csdl/search/default?type=author&givenName=Zheng-Long&surname=Liu)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

HTML5 is a great development of Web standard in nearly more than 10 years. It provides a powerful support for RIA(Applications of Rich Internet).Based on a comparison with other rich client technologies, a rich client Web architecture based on HTML5 is proposed. The paper emphatically analyzed the theory and technology of the client control in the architecture. The experiment proves that the structure is effective in realizing rich client application without any plug-in units. It can achieve personalized settings and provide normal service in unstable network conditions.

**Paper 16**

# Title: A Network-Application Framework Based on Application-Component

**Author’s:** [Gao Huaiyan](https://www.computer.org/csdl/search/default?type=author&givenName=Gao&surname=Huaiyan)   
[Tang Jiahua](https://www.computer.org/csdl/search/default?type=author&givenName=Tang&surname=Jiahua)   
[Liu Junhui](https://www.computer.org/csdl/search/default?type=author&givenName=Liu&surname=Junhui)   
[Liao Hongzhi](https://www.computer.org/csdl/search/default?type=author&givenName=Liao&surname=Hongzhi)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

This paper will describe an application-component framework (ACF), which is made up of application-component designer (ACD), application-component container (ACC), application module configuration file (AMCF), application-component access controller (ACAC), and application-component browser (ACB). Application-component framework provides perfect foundation for the development, deployment, authorization, and consumption of application-component. In the application level, ACF supports distributed network application, rapid development and rapid deployment of application system, and plug-and-play of application-component. By using ACF, enterprise’s complex business system can be conveniently and quickly divided into many small function modules, that is, application-component. Application-component is a visual interface for users, completes a relative independent function, and provides a kind of given service to authorized users through ACB in Internet. In ACD, the developers may make full use of many advantages of C/S development environment to develop and deploy application-component, and the users can achieve B/S effect and convenience through ACB. Since application-component can be developed in ACD which environment is very similar to Delphi, its interactive, interface attractive, and producing efficiency are very outstanding far developing the internet applications.

**Paper 17**

# Title: Beacon-based multi-person activity monitoring system for day care center

**Author’s:** [Kiyoaki Komai](https://www.computer.org/csdl/search/default?type=author&givenName=Kiyoaki&surname=Komai),    
[Manato Fujimoto](https://www.computer.org/csdl/search/default?type=author&givenName=Manato&surname=Fujimoto),    
[Yutaka Arakawa](https://www.computer.org/csdl/search/default?type=author&givenName=Yutaka&surname=Arakawa),    
[Hirohiko Suwa](https://www.computer.org/csdl/search/default?type=author&givenName=Hirohiko&surname=Suwa),    
[Yukitoshi Kashimoto](https://www.computer.org/csdl/search/default?type=author&givenName=Yukitoshi&surname=Kashimoto),    
[Keiichi Yasumoto](https://www.computer.org/csdl/search/default?type=author&givenName=Keiichi&surname=Yasumoto),

**Dept. name of the Organization:** Graduate School of Information Science, Nara Institute of Science and Technology, 8916-5, Takayama, Ikoma, Nara 630-0192, Japan

Recently, as elderly people population grows, the burden on caretakers are getting larger. In day care center, caretakers are taking care records aiming to improve care receiver's Quality of Life (QoL). However, in the present situation, it is difficult for caretakers to record care receiver's activity in detail because each care worker needs to take care of several care receivers at the same time and it is a large burden. To reduce the burden of caretakers, many elderly monitoring systems have been proposed so far, but most of them are not effective in the sense that they force care receivers to use dedicated device such as smart phone and/or particular applications that are obtrusive and cumbersome for care receivers. In this paper, we propose a novel elderly monitoring system which can monitor movements/activity of multiple care receivers at the same time by estimating existence area of each of the care receivers, without burdening them. Our proposed system estimates multiple care receivers existence area only using RSSI (Received Signal Strength Indication) of BLE (Bluetooth Low Energy). The feature of our proposed system is that it takes Movable-Beacon and Fixed Scanner style. We have validated the proposed system and confirmed that we can estimate multi-person's existence area at high accuracy using only BLE devices.

**Paper 18**

# Title: Mobile Application Monitoring

**Author’s:** [Sebastian Vasquez](https://www.computer.org/csdl/search/default?type=author&givenName=Sebastian&surname=Vasquez)   
[Jocelyn Simmonds](https://www.computer.org/csdl/search/default?type=author&givenName=Jocelyn&surname=Simmonds)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

To date, there are more than 800.000 applications available on each of the two largest mobile application markets (Google Play and Apple App Store), each reporting around 40 billion downloads in total. At this scale, it is impossible to analyze each application for malware behavior before making them available for download. Vendors mostly rely on user feedback to remove malware from their marketplaces, so it can take a while for these applications to be removed. In practice, users must be very careful when picking which applications to install. Moreover, it can be hard to detect a malware application once it is installed, unless the application has noticeable side-effects. To improve this situation, we propose a framework for performing runtime monitoring of mobile applications against behavioral correctness properties. These properties specify forbidden and desired interactions between applications and the phone's resources as sequences of events. The user can then determine which applications to uninstall based on which properties they violate. We have implemented a prototype of this framework for the Android operating system. In this first version of the framework, the user must specify the properties to be monitored directly as automata.

**Paper 19**

# Title: BehaviorDroid: Monitoring Android Applications

**Author’s:** [Alexis Silva](https://www.computer.org/csdl/search/default?type=author&givenName=Alexis&surname=Silva)   
[Jocelyn Simmonds](https://www.computer.org/csdl/search/default?type=author&givenName=Jocelyn&surname=Simmonds)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

Mobile app developers declare permissions, but these do not guarantee that apps will behave as expected. Existing work focuses on checking predefined app properties, e.g., clone detection and API analysis. We propose BehaviorDroid, a framework for monitoring general app properties at runtime. Properties are currently specified using automata, describing desired and unwanted interactions between an app and phone resources. BehaviorDroid is a robust, extensible and configurable framework that can simultaneously monitor multiple apps and properties, showing reasonable CPU and memory usage during execution. Initial experiments show that we can improve memory usage by combining automata that have similar alphabets.

**Paper 20**

# Title: ACCUSE: Helping Users to Minimize Android App Privacy Concerns

**Author’s:** [Majda Moussa](https://www.computer.org/csdl/search/default?type=author&givenName=Majda&surname=Moussa)   
[Massimiliano Di Penta](https://www.computer.org/csdl/search/default?type=author&givenName=Massimiliano&surname=Di%20Penta)   
[Giuliano Antoniol](https://www.computer.org/csdl/search/default?type=author&givenName=Giuliano&surname=Antoniol)   
[Giovanni Beltrame](https://www.computer.org/csdl/search/default?type=author&givenName=Giovanni&surname=Beltrame)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

System permissions play a crucial role in the Android security architecture. They are used to restrict app operations only to resources (e.g., file system, network) that the user has agreed to share. In this paper, we present ACCUSE (Android Confidentiality Concern User Support Environment),an approach aimed at helping Android users and developers to compare and contrast the risk level of a set of Android apps. First, ACCUSE assigns to each app risk factors based on the Android classification of permissions. Then, such factors are combined with the app rating and downloading information to produce a new, overall risk factor. ACCUSE has been evaluated on a realworld dataset of 11,576 android apps as well as on a baseline of around 1000 known malware apps. Our results show that ACCUSE always assigns high risk to known malware apps, and outperforms the state-of-the-art.

**Paper 21**

# Title: A Methodology for Learning Across Application Domains for Database Design Systems

**Author’s:** [V.C. Storey](https://www.computer.org/csdl/search/default?type=author&givenName=V.C.&surname=Storey)   
[D. Dey](https://www.computer.org/csdl/search/default?type=author&givenName=D.&surname=Dey)

**Dept. name of the Organization:** IEEE Computer Society Digital Library

Although database design tools have been developed that attempt to automate (or semiautomate) the design process, these tools do not have the capability to capture common sense knowledge about business applications and store it in a context-specific manner. As a result, they rely on the user to provide a great deal of “trivial” details and do not function as well as a human designer who usually has some general knowledge of how an application might work based on his or her common sense knowledge of the real world. Common sense knowledge could be used by a database design system to validate and improve the quality of an existing design or even generate new designs. This requires that context-specific information about different database design applications be stored and generalized into information about specific application domains (e.g., pharmacy, daycare, hospital, university, manufacturing). Such information should be stored at the appropriate level of generality in a hierarchically structured knowledge base so that it can be inherited by the subdomains below. For this to occur, two types of learning must take place. First, knowledge about a particular application domain that is acquired from specific applications within that domain are generalized into a domain node (e.g., entities, relationships, and attributes from various hospital applications are generalized to a hospital node). This is referred to as within domain learning. Second, the information common to two (or more) related application domain nodes is generalized to a higher-level node; for example, knowledge from the car rental and video rental domains may be generalized to a rental node. This is called across domain learning. This paper presents a methodology for learning across different application domains based on a distance measure. The parameters used in this methodology were refined by testing on a set of representative cases; empirical testing provided further validation.

**Paper 22**

# Title: A survey of web log data and their application in use-based design

**Author’s:** [M.C. Burton](https://www.computer.org/csdl/search/default?type=author&givenName=M.C.&surname=Burton),    
[J.B. Walther](https://www.computer.org/csdl/search/default?type=author&givenName=J.B.&surname=Walther)

**Dept. name of the Organization:** IBM Corp., USA

Web-based logs contain potentially useful data with which designers can assess the usability and effectiveness of their choices. Most guides to World Wide Web (Web) design derived from artistic or usability principles feature no empirical validation, while empirical studies of Web use typically rely on observer ratings. Several sources of unobtrusive usage data are available to Web designers, including Web server logs, client-side logs, and other data. The naturally-occurring traces recorded in these logs offer a rich data source, amenable to normative use assessments and to experimental research comparing alternative Web designs. Identification of types of Web server logs, client logs, types and uses of log data, and issues associated with the validity these data, are enumerated. Finally, frameworks that outline how sources of use-based data can be triangulated to assess Web design are illustrated, and an approach to experimentation that overcomes many log data validity issues is presented.

**CHAPTER 3**

**SCOPE OF PROJECT**

**3.1 Problem Statement**

* Daycare software that offers an **easy-to-use communication** system is one of the top priorities you should look for when choosing a daycare software.
* It creates a **safer environment for kids** eliminating various social problems.
* Daycares probably have many children to look after each day so being able to input data quickly is important. By using the app the staff can spend **less time on “paperwork”** and more time looking after kids.
* If you have an app that can help parents pay for daycare services and **schedule payment reminders,** it will make it easier for Daycare and the parents of the kids you look after.
* This app provides many new features which will be **user-friendly and easy to use.**
* It enhances the chances of **learning new abilities** and skills in kids.

**3.2 Objectives and Project Scope**

* Child Care Business Management,
* Parent Engagement,
* Secure & Contactless,
* Tuition Collection and Payment, Staff Management,
* Track Attendance,
* Drop off & Pick up,
* Early Childhood Learning,

These are the basic objectives of this application.

**CHAPTER 4**

**METHODOLOGY**

**4.1 EXPLANATION**

To choose the best methodology for our project we analyzed aspects like requirements, available time, human resources and financial requirements.

**We got the following results in our analysis:**

* The requirements of our project are pretty much clear and there are very little chances of any changes in them
* Our team was lacking experience and it was not capable of working in development of multiple phases in the same time.
* There are different modules in our project which required testing of them in their development phase.
* The progress in developing this project were required to be reported to our guide.
* We had very short time to get this project ready.

**Our analysis showed us that we should go with such software development methodology which has the following benefits:**

* Generates working software quickly and early during the software life cycle.
* More flexible – less costly to change scope and requirements.
* Easier to test and debug during a smaller iteration.
* Easier management of risk.
* Each iteration should be an easily managed milestone.

**Incremental Model** was best suitable for our needs so we decided to go with it.

**The following are the outcome of each release in the development lifecycle:**

**Server-Releases:**

**Release 1:** User management modules of server side were created and tested.

**Release 2:** Modules responsible for securely downloading a file in server were created and tested.

**Release 3:** The Compression module and Communication modules was created and tested.

**Release 4:** Enhancement in Handling request and response in communication modules were made.

**Client-side Releases:**

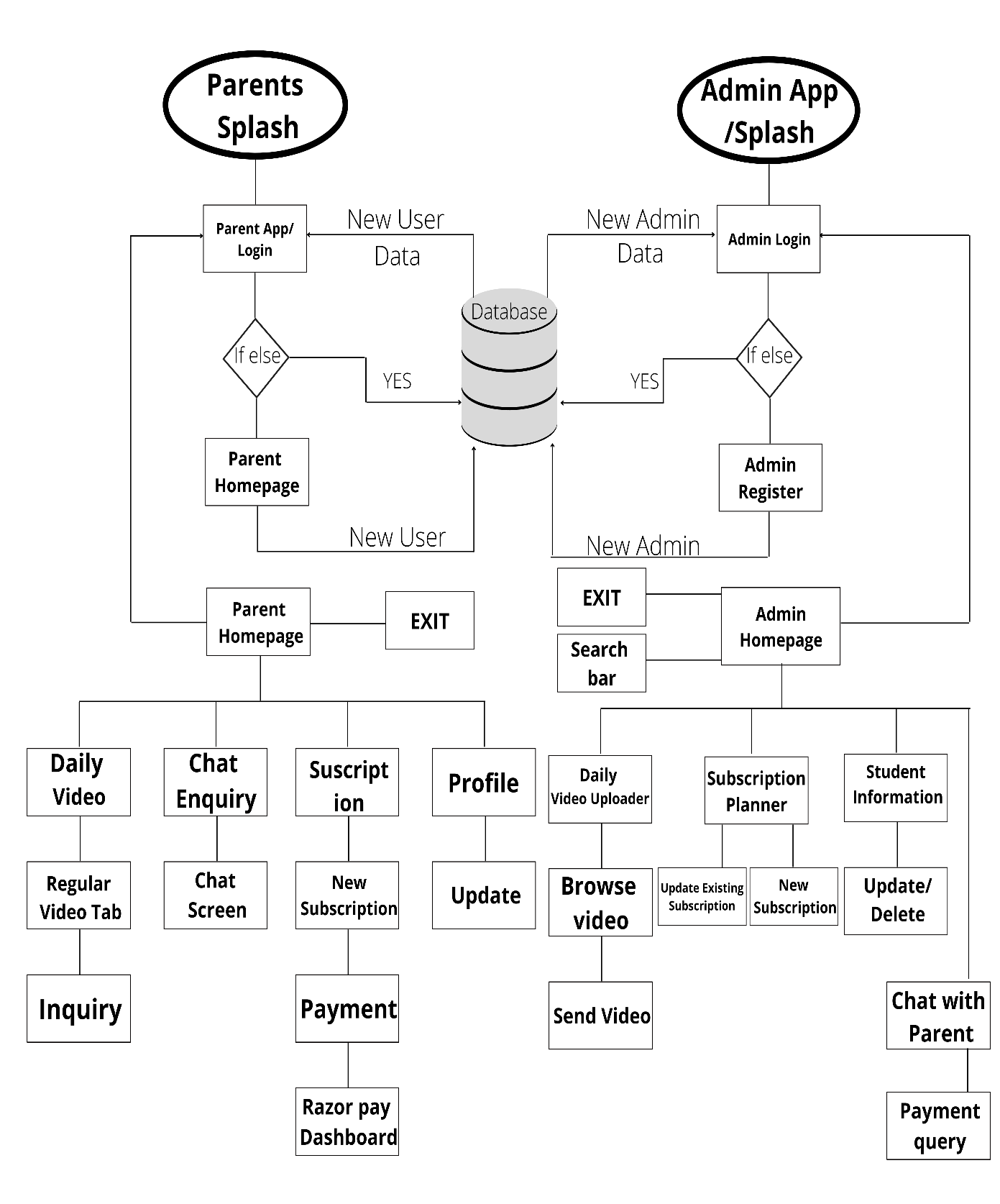
**Release 1:** GUI layouts and components of client-side application were created.

**Release 2:** Handlers were attached with their respective GUI components.

**Release 3:** Core libraries for functionality of client-side application were developed and tested.

**Release 4:** Handlers for different operation were created and tested.

**4.2 SYSTEM FLOW CHART**

****

**4.3 ALGORITHM**

**Parents Application**

**Step 1:** Start  
**Step 2:** Registration And Login, If already logged in,Go to Step 4.

**Step 3:** Daily Videos and Enquiry

**Step 4:** Chat with Daycare

**Step 5:** Subscribe to payment plans and monthly and billing desk

**Step 6:** Child Profile Management and Update desk

**Step 7:** Exit

**Owner Application**

**Step 1:** Start  
**Step** **2:** Registration And Login, If already logged in,Go to Step 4.

**Step 3:** Daily Videos Uploader and Enquiry Checker

**Step 4:** Chat with Parent

**Step 5:** Manage to payment plans and request payment tab

**Step 6:** Daycare Profile Management and Update desk

**Step 7:** Exit

**CHAPTER 5**

**DETAILS OF DESIGN WORKING AND PROCESS**

**5.1 INTRODUCTION**

Software analysis and design include all activities, which help the transformation of requirement specification into implementation. Requirement specifications specify all functional and non-functional expectations from the software. These requirement specifications come in the shape of human-readable and understandable documents, with which a computer has nothing to do.

Software analysis and design is the intermediate stage, which helps human-readable requirements to be transformed into actual code.Requirements, available time, human resources, and financial requirements.

**Data Flow Diagram:**

The data flow diagram is a graphical representation of the flow of data in an information system. It is capable of depicting incoming data flow, outgoing data flow, and stored data. The DFD does not mention anything about how data flows through the system. There is a prominent difference between DFD and Flowchart. The flowchart depicts the flow of control in program modules. DFDs depict the flow of data in the system at various levels. DFD does not contain any control or branch elements.

**Types of DFD:**

Data Flow Diagrams are either Logical or Physical.

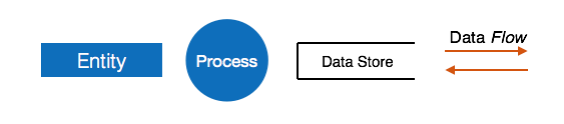
• **Logical DFD:**

This type of DFD concentrates on the system process, and flow of data in the system. For example in a Banking software system, how data is moved between different entities.

**• Physical DFD:**

This type of DFD shows how the data flow is implemented in the system. It is more specific and close to the implementation.

**DFD Components**

DFD can represent the Source, destination, storage, and flow of data using the following set of components

**• Entities:**

Entities are the source and destination of information data. Entities are represented by a rectangle with their respective names.

**• Process:**

Activities and actions taken on the data are represented by circles or Round-edged rectangles.

**• Data Storage:**

There are two variants of data storage - it can either be represented as a rectangle with an absence of both smaller sides or as an open-sided rectangle with only one side missing.

**• Data Flow:**

The movement of data is shown by pointed arrows. Data movement is shown from the base of the arrow as its source towards the head of the arrow as the destination.

**5.2 SYSTEM ANALYSIS**

**5.2.1 INTRODUCTION TO SYSTEM PLANNING**

Before starting a software-hardware project, it is essential to determine the tasks to be performed and properly manage the allocation of tasks among individuals involved in the software development. Hence, planning is important as it results in effective software development. Project planning is an organized and integrated management process, which focuses on activities required for the successful completion of the project. It prevents obstacles that arise in the project such as changes in projects or organization's objectives, non-availability of resources, and so on. Project planning also helps in better utilization of resources and optimal usage of the allotted time for a project. The other objectives of project planning are listed below.

• It defines the roles and responsibilities of the project management team members.

• It ensures that the project management team works according to the healthcare/covid objectives.

• It checks the feasibility of the schedule and user requirements.

• It determines project constraints. Several individuals help in planning the project.

These include senior management and the project management team. Senior management is responsible for employing team members and 29 providing resources required for the project. The project management team, which generally includes project managers and developers, is responsible for planning, determining, and tracking the activities of the project. Project planning should be effective so that the project begins with well-defined tasks. Effective project planning helps to minimize the additional costs incurred on the project while it is in progress. For effective project planning, some principles are followed. These principles are listed below.

• Planning is necessary: Planning should be done before a project begins. For effective planning, objectives and schedules should be clear and understandable.

• Risk analysis: Before starting the project, senior management and the project management team should consider the risks that may affect the project. For example, the user may desire changes in requirements while the project is in progress. In such a case, the estimation of time and cost should be done according to those requirements (new requirements).

• Tracking of project plan: Once the project plan is prepared, it should be tracked and modified accordingly.

• Meet quality standards and produce quality deliverables: The project plan should identify processes by which the project management team can ensure quality in software. Based on the process selected for ensuring quality, the time and cost for the project are estimated.

• Description of flexibility to accommodate changes: The result of project planning is recorded in the form of a project plan, which should allow new changes to be accommodated when the project is in progress.

Software design approach

Here are two general approaches for software design:

**Top-Down Design**

We know that a system is composed of more than one subsystem and it contains several components. Further, these sub-systems and components may have their onset of sub-system and components and create hierarchical structure in the system. 30 Top-down design takes the whole software system as one entity and then decomposes it to achieve more than one subsystem or component based on some characteristics. Each subsystem or component is then treated as a system and decomposed further. This process keeps on running until the lowest level of the system in the top-down hierarchy is achieved. Top-down design starts with a generalized model of the system and keeps on defining the more specific part of it. When all components are composed the whole system comes into existence. Top-down design is more suitable when the software solution needs to be designed from scratch and specific details are unknown.

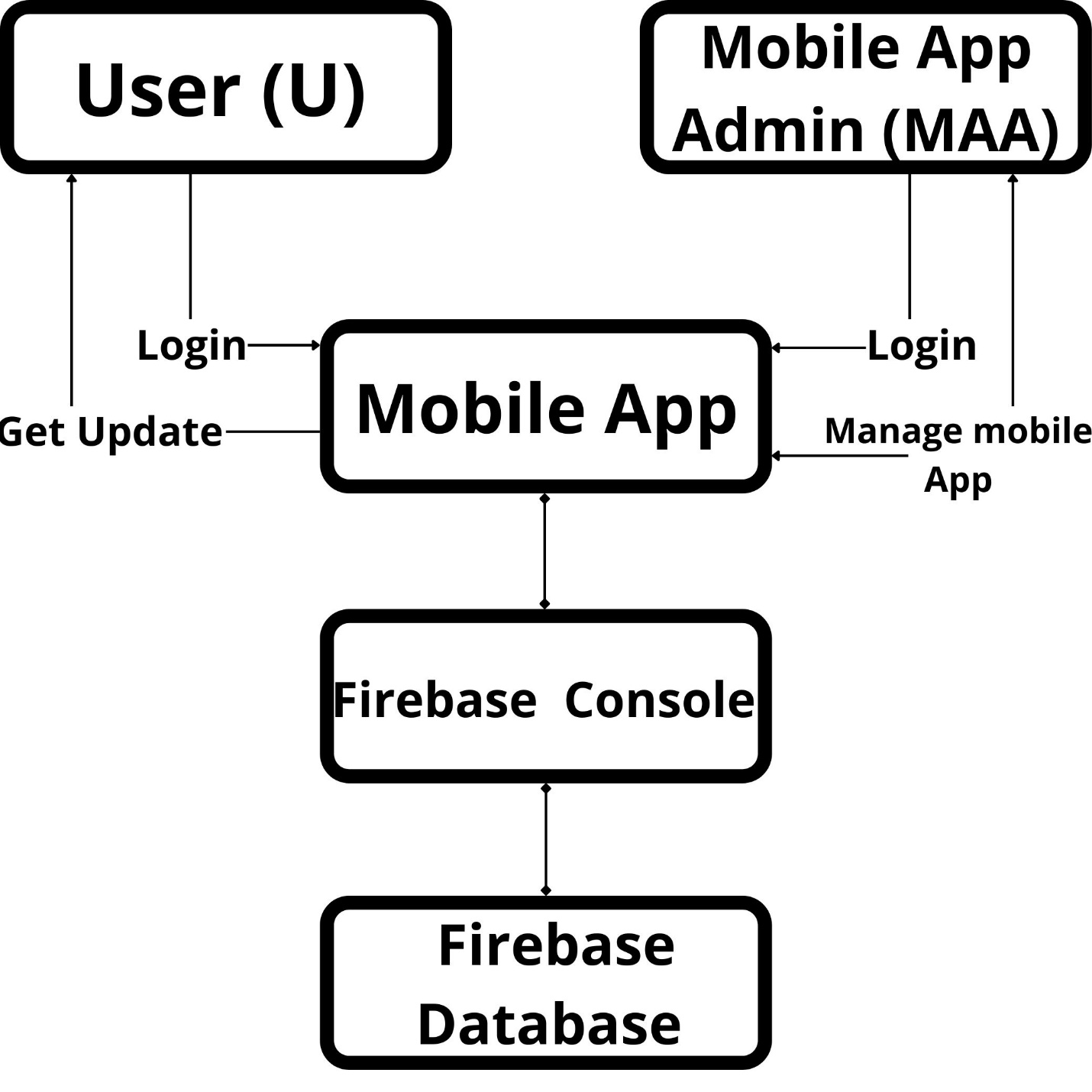
**Bottom-up Design**

The bottom-up design model starts with the most specific and basic components. It proceeds with composing a higher level of components by using basic or lower-level components. It keeps creating higher-level components until the desired system is not evolved as one single component. With each higher level, the amount of abstraction is increased. A bottom-up strategy is more suitable when a system needs to be created from some existing system, where the basic primitives can be used in the newer system.

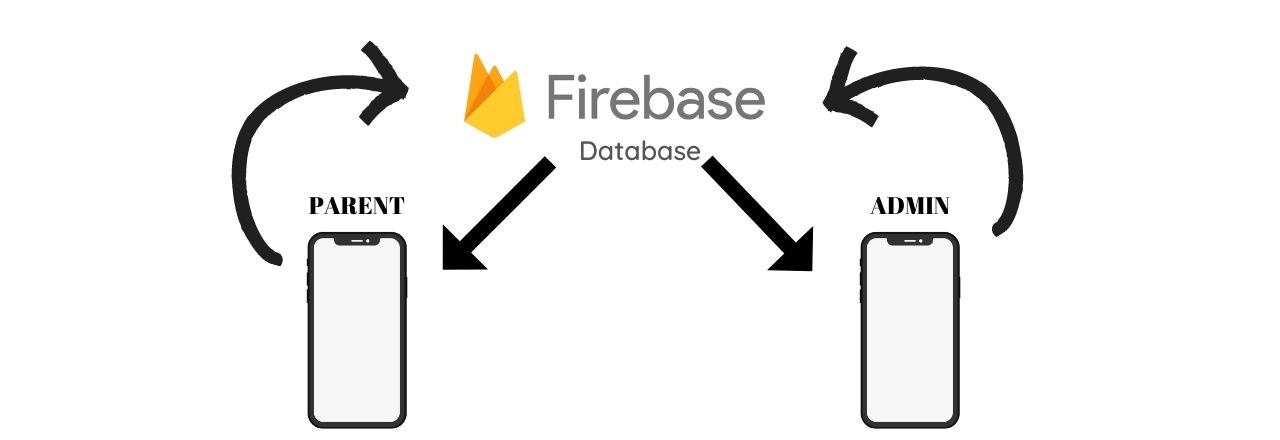
**5.2.2 SOFTWARE DESIGN APPROACH WE CHOSE:**

Bottom-up Design We had decided to take the modular way to develop our project and the bottom-up design approach was best suited for our project development. Firstly, we developed the core components of the Firebase Database and then we kept developing the rest of the components and continuously combined them with their respectively dependent components.

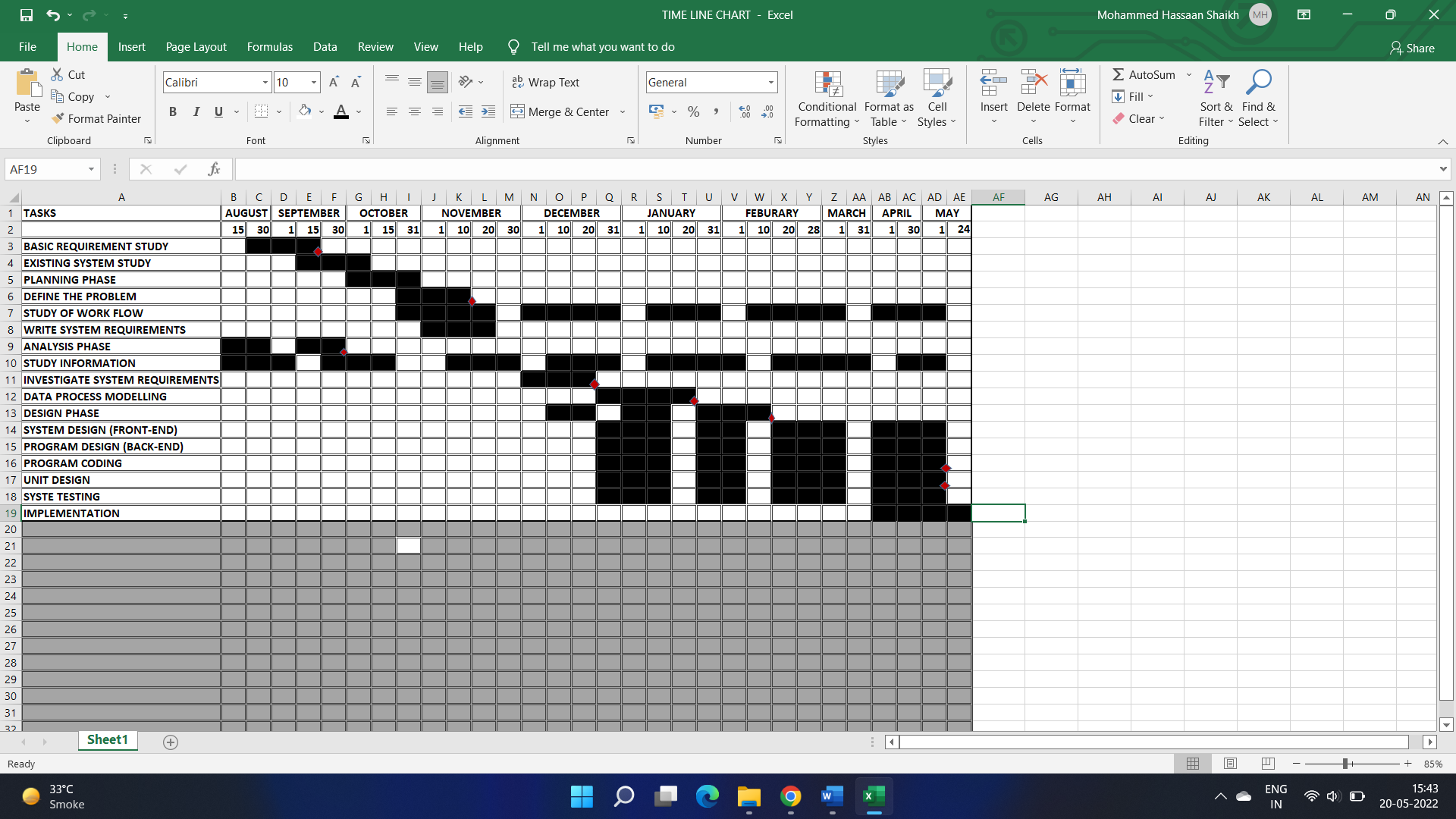
**5.3 BLOCK DIAGRAM**



**5.4 SYSTEM ARCHITECTURE**



**5.5 TIMELINE CHART**



**5.6 COST ESTIMATION**

**Total number of weeks:** 5th Sem + 6th Sem

=13+16 weeks

=29weeks

**Number of hours per week:** 9 hours

**Total number of Hours:** 29\*9

**=** 261 Hours

**Cost of development per Hour:** 80 INR

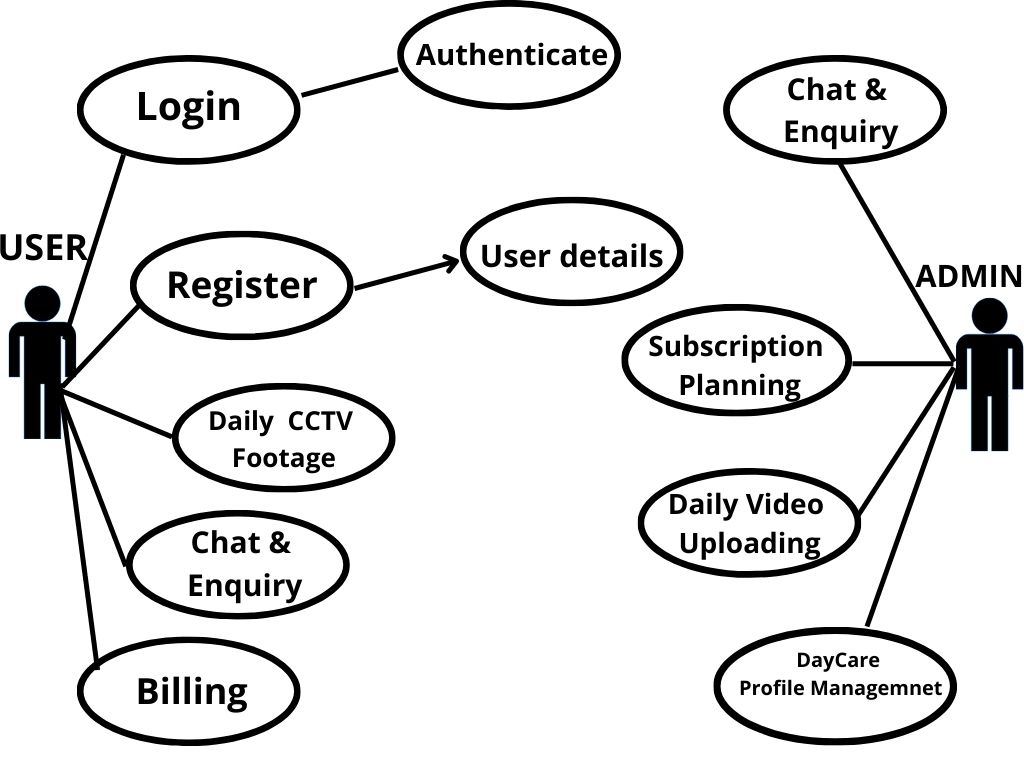
**Total Cost:** Total no. of Hours \* Cost of Development per Hour

=261 \* 80

= **20,880 INR**

**5.7 UML DIAGRAM**

**Use Case Diagram:**

****

**5.8 FEASIBILITY STUDY**

A Feasibility Study is an analysis of how successfully a project can be completed, accounting for factors that affect it such as economic, technological, legal, and scheduling factors. Project managers use feasibility studies to determine potential positive and negative outcomes of a project before investing a considerable amount of time and money into it. There are three types of feasibility, they are defined as follows:

1. **Technical Feasibility**

It has been determined that the technology needed for the proposed system is available and that this technology can be integrated into the application. Technical evaluation has also evaluated the existing system to find that it cannot be upgraded in keeping with the user’s needs. Hence, we need to create an entire system that caters to the specific needs of the user. The end-user can be equipped with the pre-mentioned hardware and software requirements.

**Hardware Requirements**

* Ram: 256 MB or more
* Processor: 2nd generation Intel Core or newer, or AMD CPU
* HDD: 10 GB or more
* Android (Basic)

1. **Economical Feasibility**

The economic feasibility of the system is mainly concerned with its financial aspects. It determines whether the project is economically feasible. As the hardware and software are already available easily in the market, no further investment is to be made in that direction, the only cost involved is that of implementing the system. The entire system is made in Android studio hence no costs are required in that direction however hosting services will begin charging after a certain period.

**It was decided that the project was technically feasible because of the following:**

* Necessary technology exists to do what is suggested.
* The system would be expendable if so decided.
* The system can be integrated with other networking tools and the output of the system can be provided to other tools for high-end analysis.
* This system is technically more secure.

1. **Operational Feasibility**

Operational feasibility is a measure of how well a proposed system solves the identified problems and takes advantage of the opportunities identified in the scope of research. The system should also satisfy the requirements identified in the requirement analysis phase of system development. Since the software is intended in making the encryption process customized, The user must have the fundamental knowledge of the computer(he must know how to do character selection and it is assumed that the user will have the basic skills at the system has its application at a business level). The proposed system is intended at providing a high level of abstraction(AES algorithm is applied just at one click and no coding is shown) to the user so that even any person with average knowledge in the working of a computer is able to use it efficiently. Since all these requirements are easy and affordable, the customized system is operationally feasible.

**4. Scheduling Feasibility**

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization

estimates how much time the project will take to complete

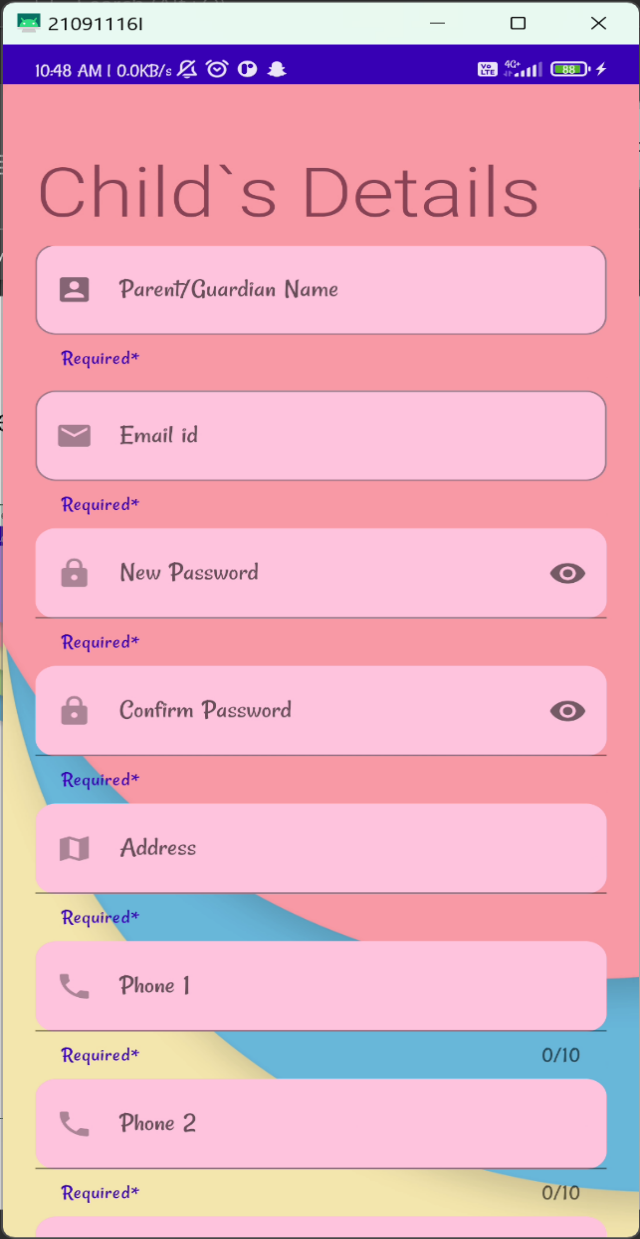
**CHAPTER 6**

**RESULT & APPLICATION**

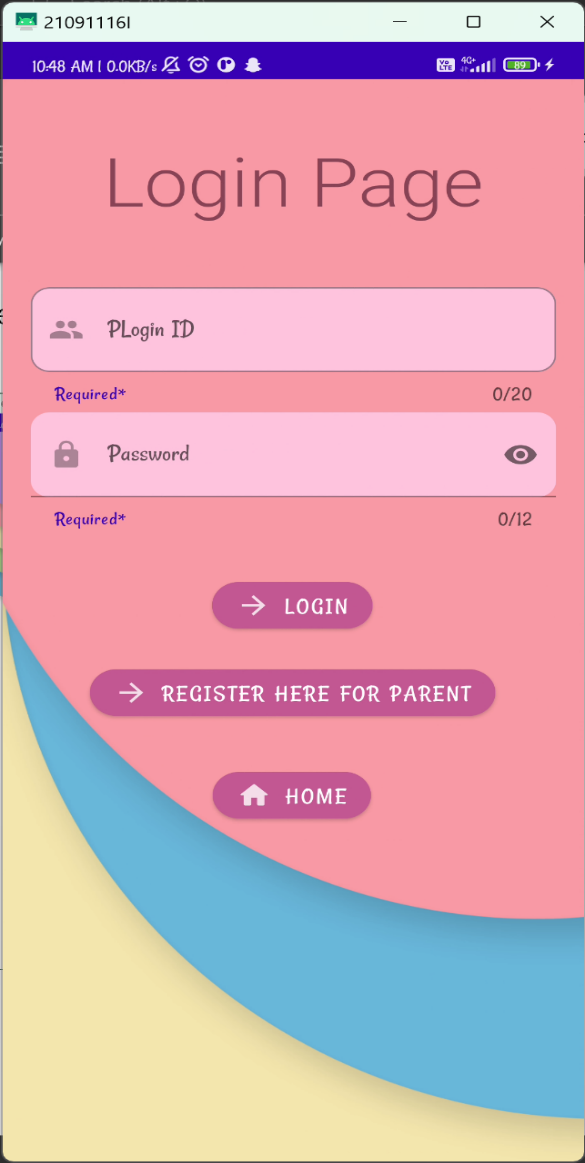
**6.1 SCREENSHOTS**

**Parents App:**

Splash Screen: Registration Screen:

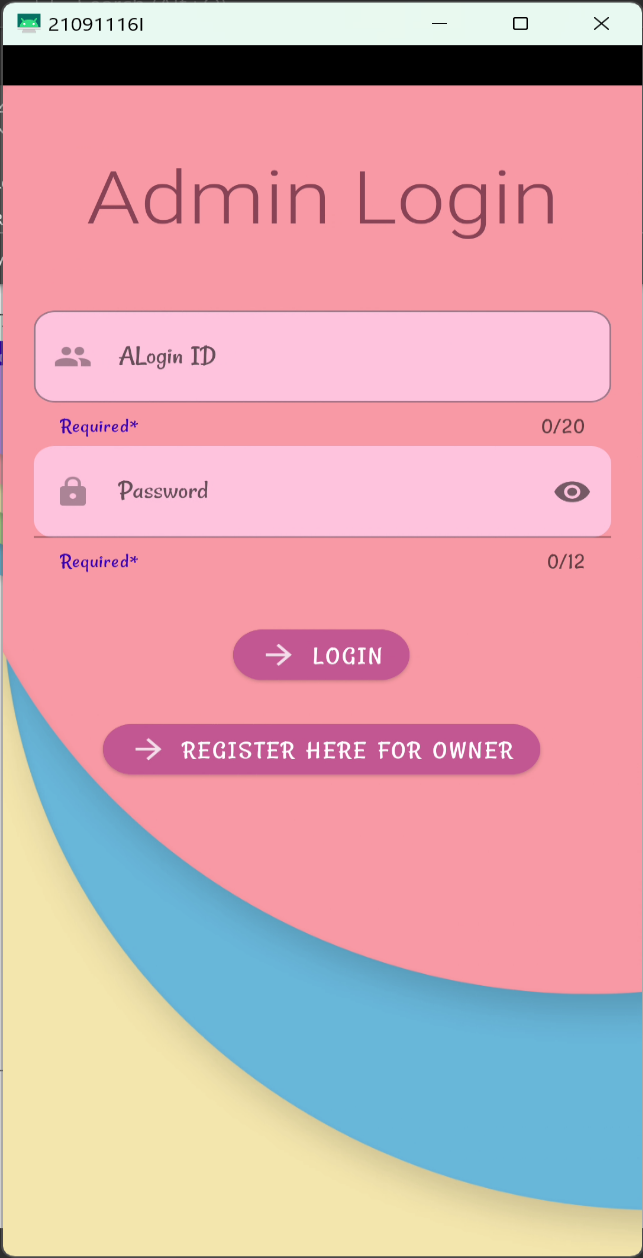
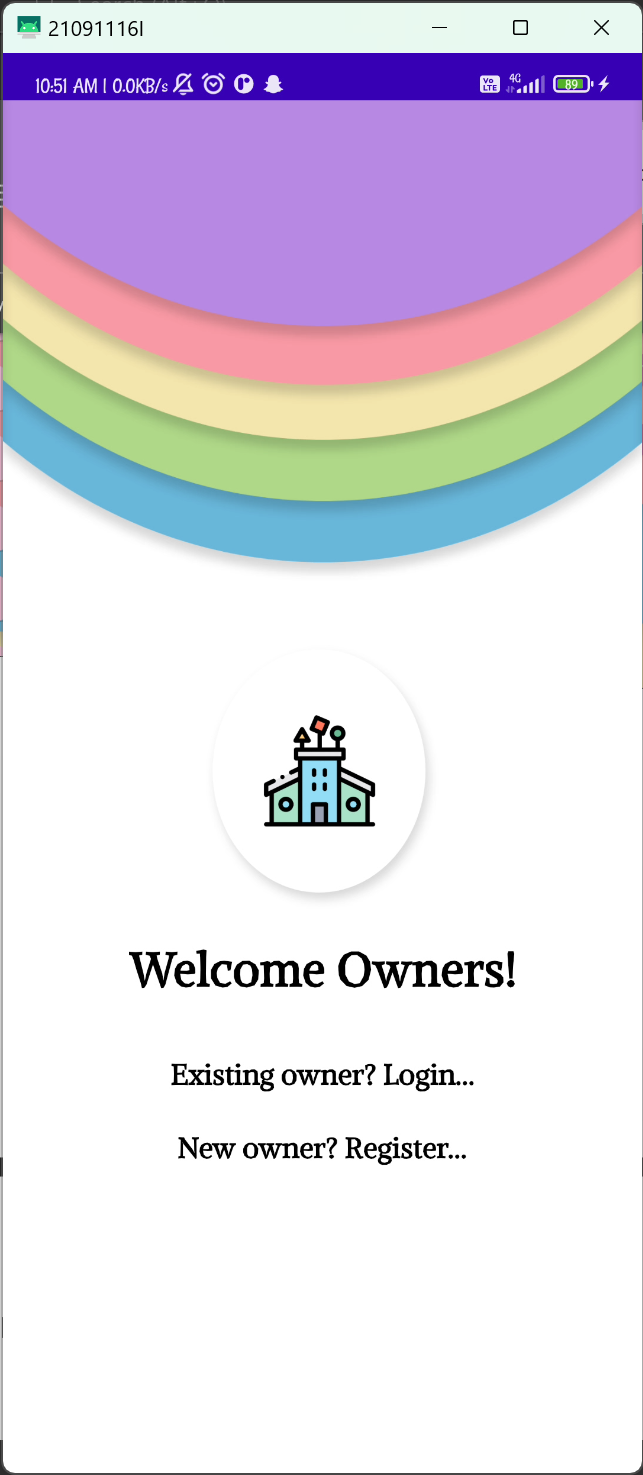


Login Screen: Profile:

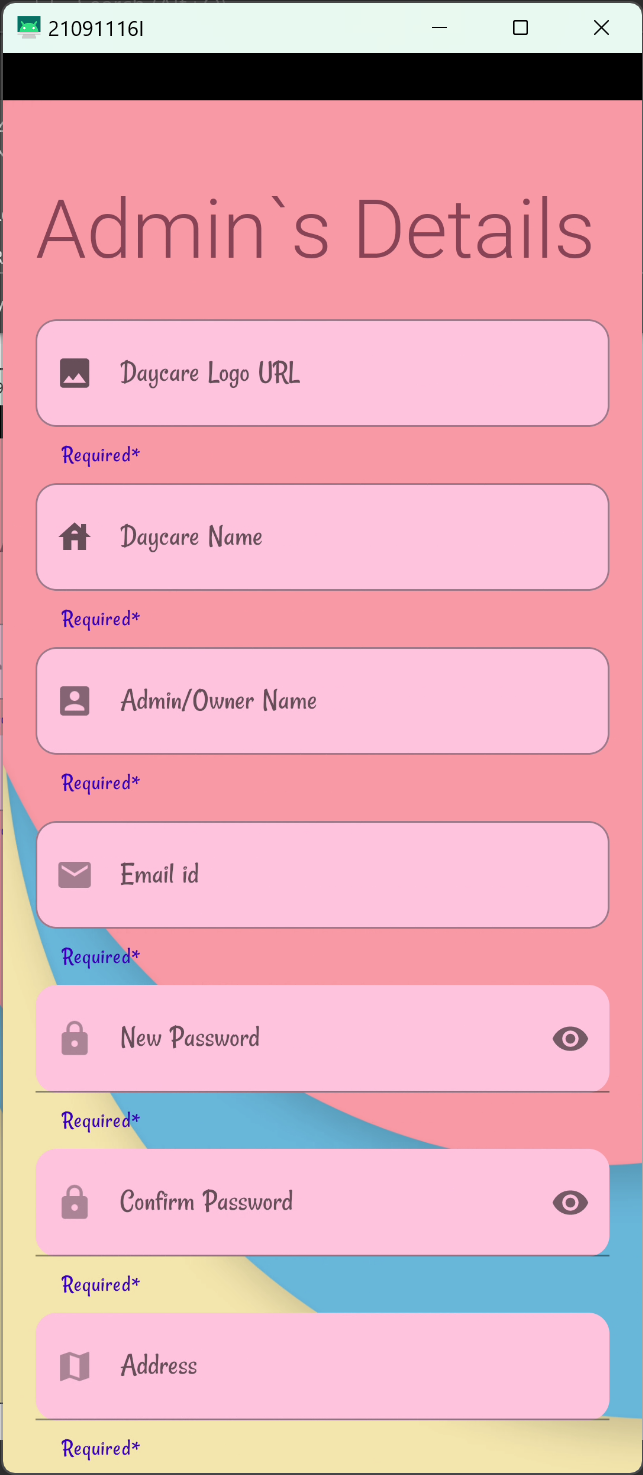


**Owners App (Service Providers):**

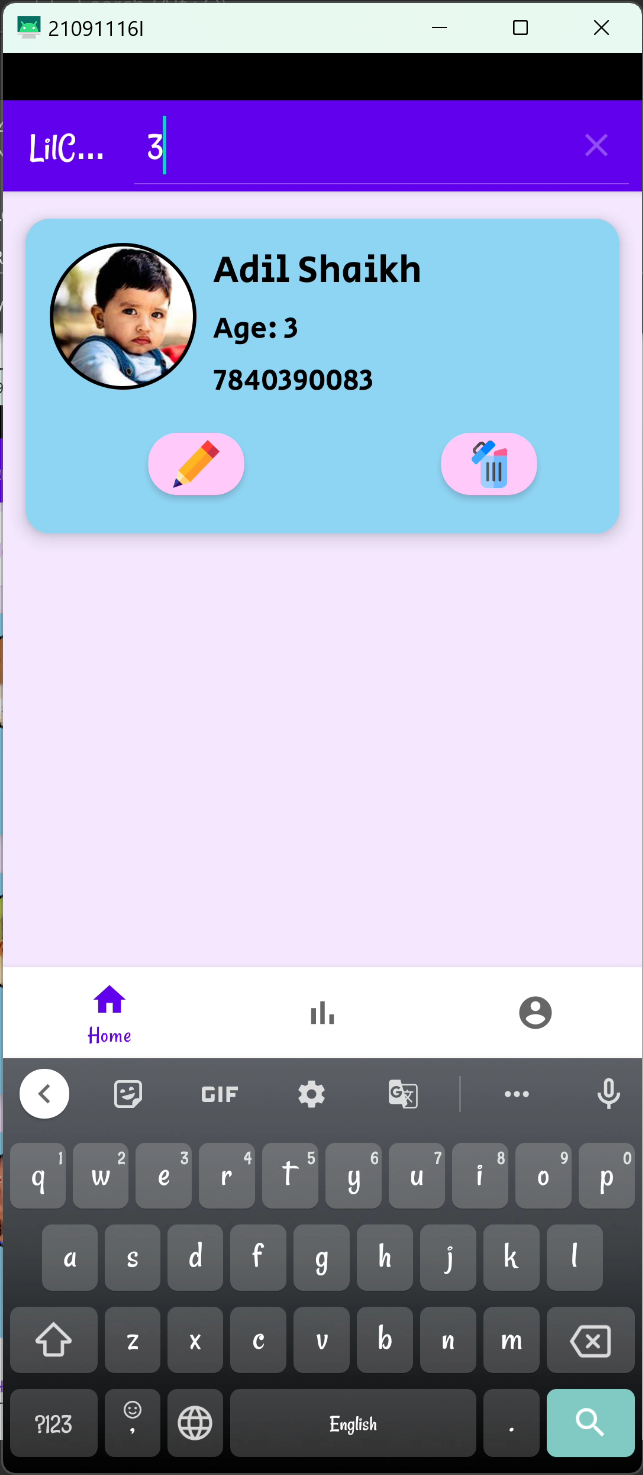
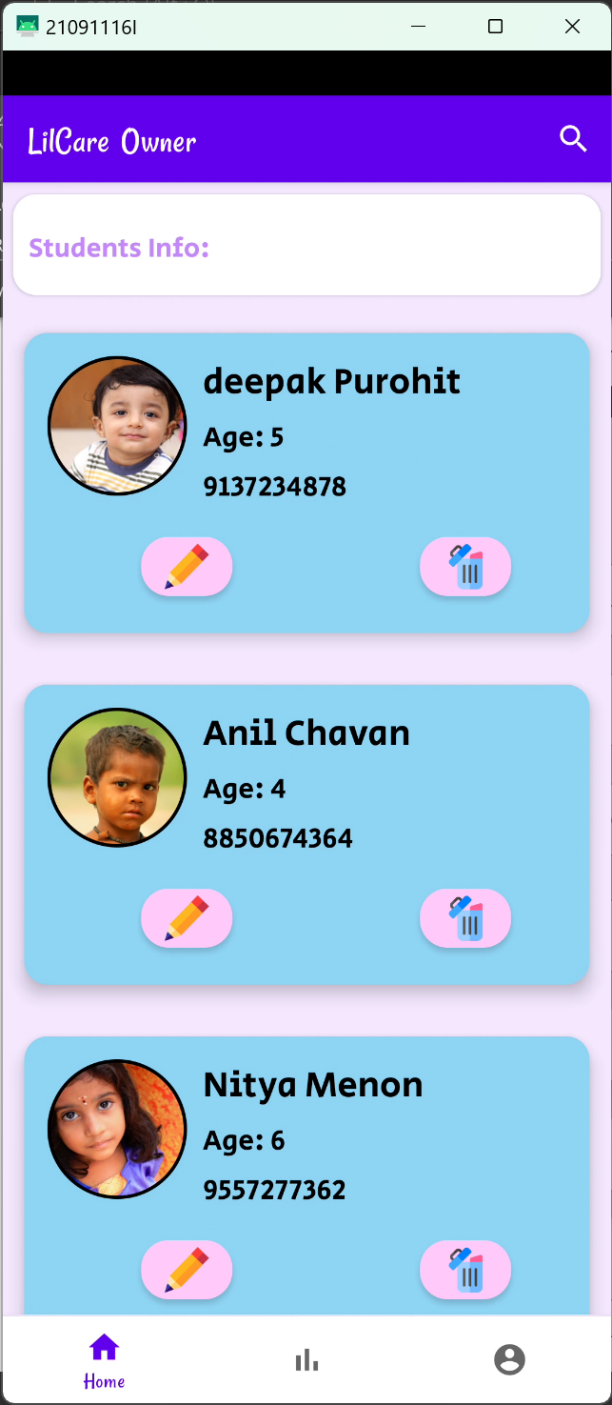
Owner Splash: Owner Login:



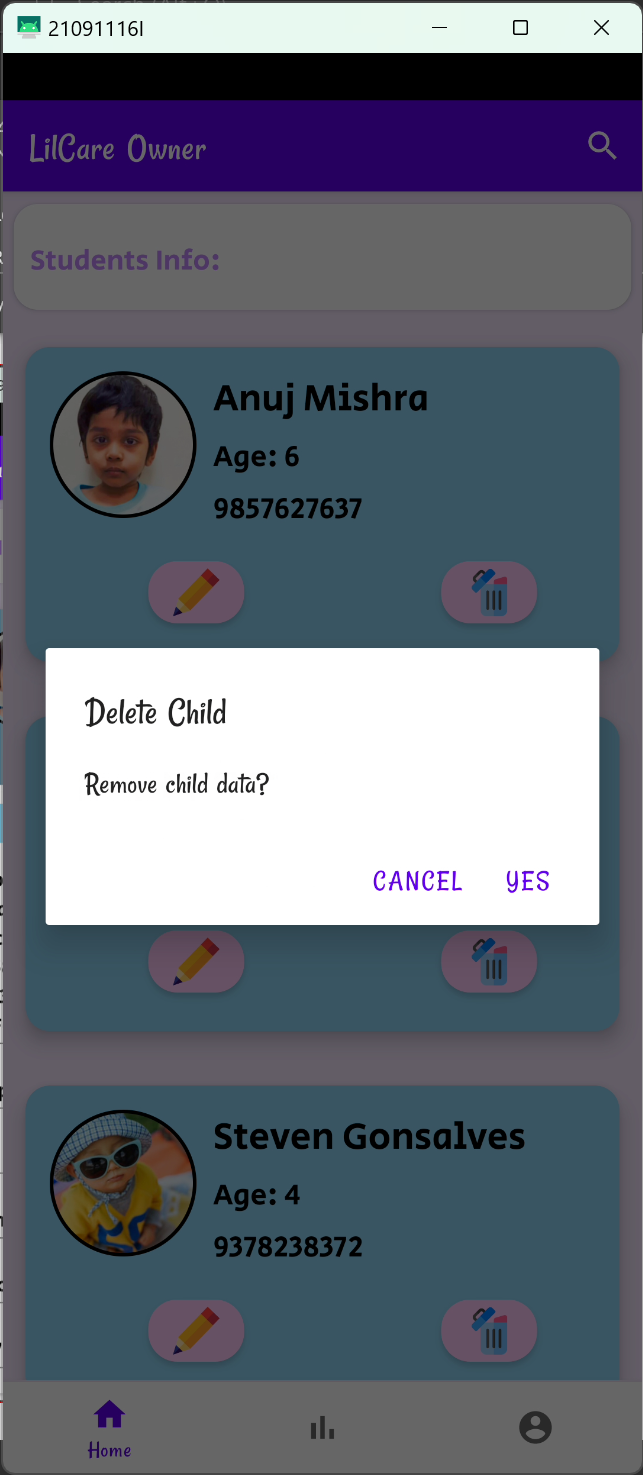
Owner Details:



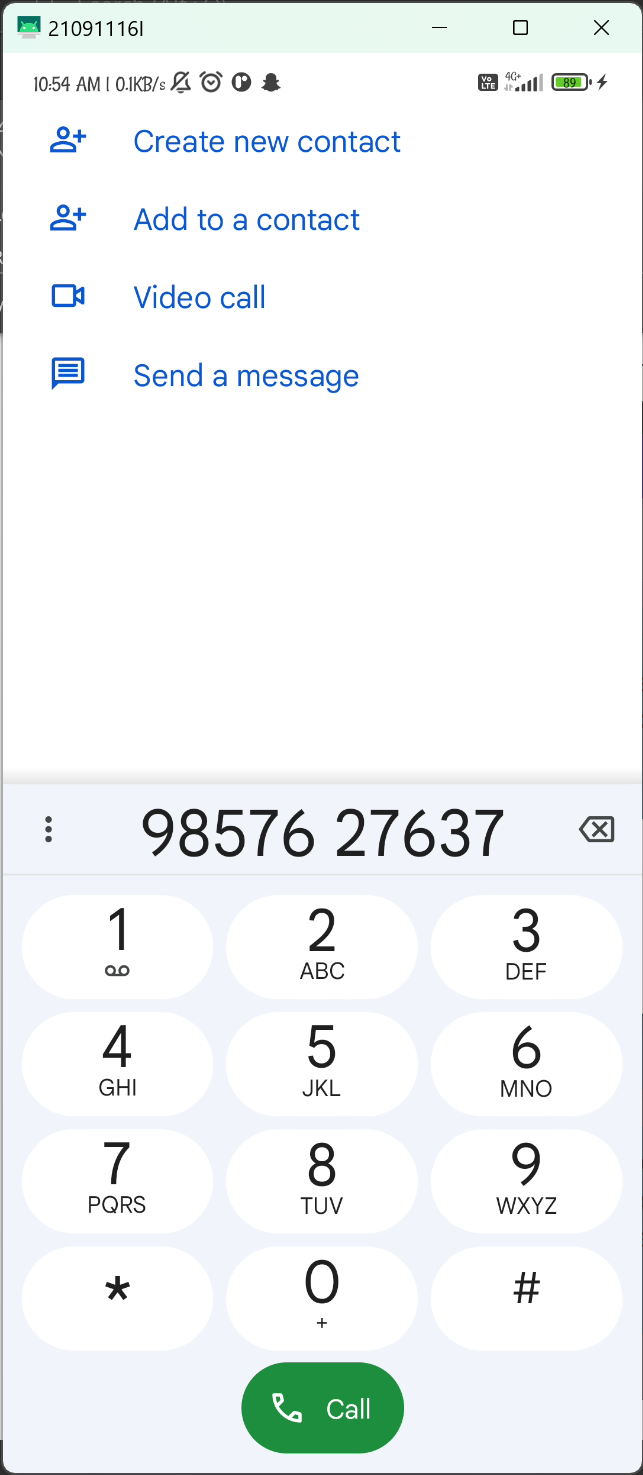
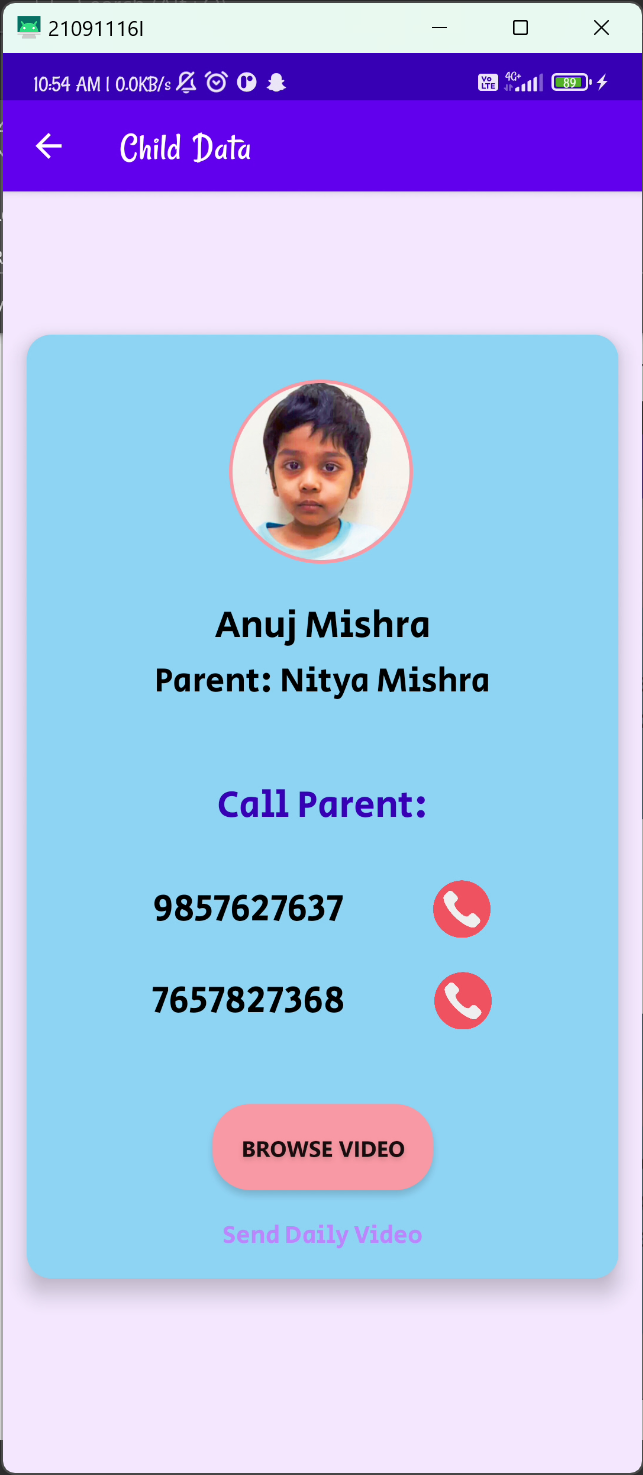
Child Data Home Screen: Home Screen Search Bar:



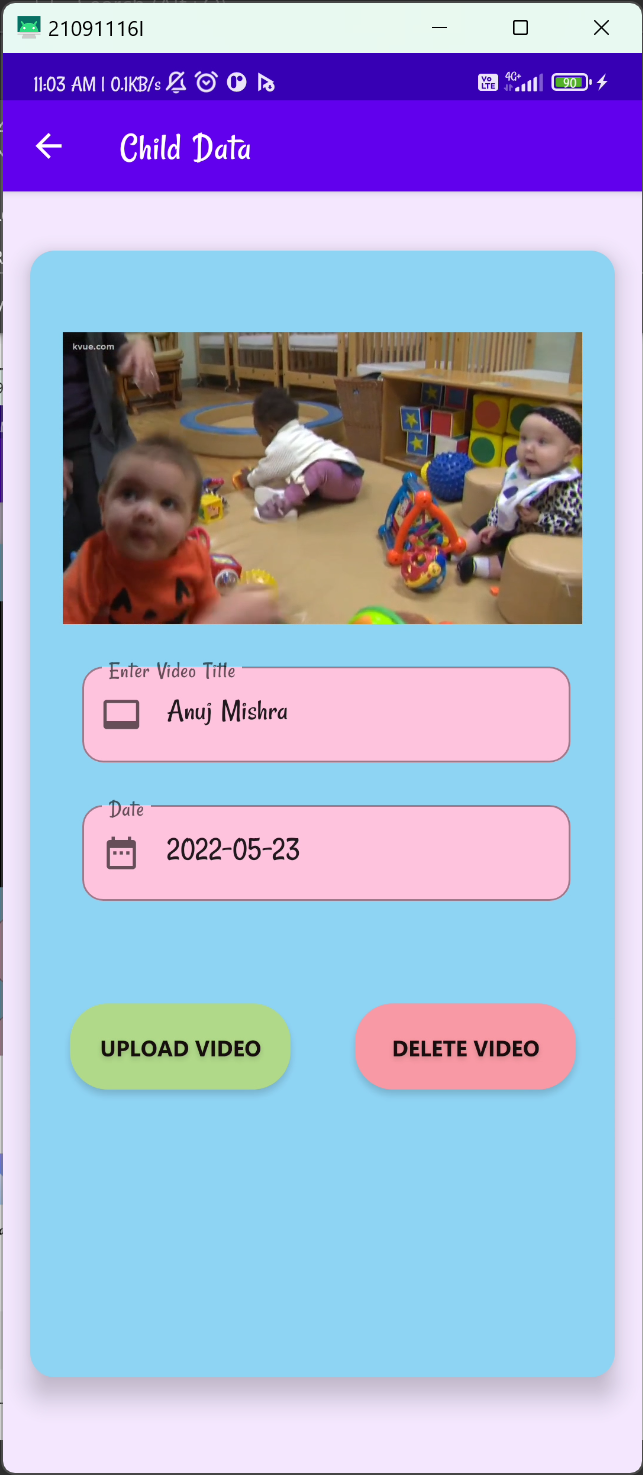
Updating and Deleting the Information:

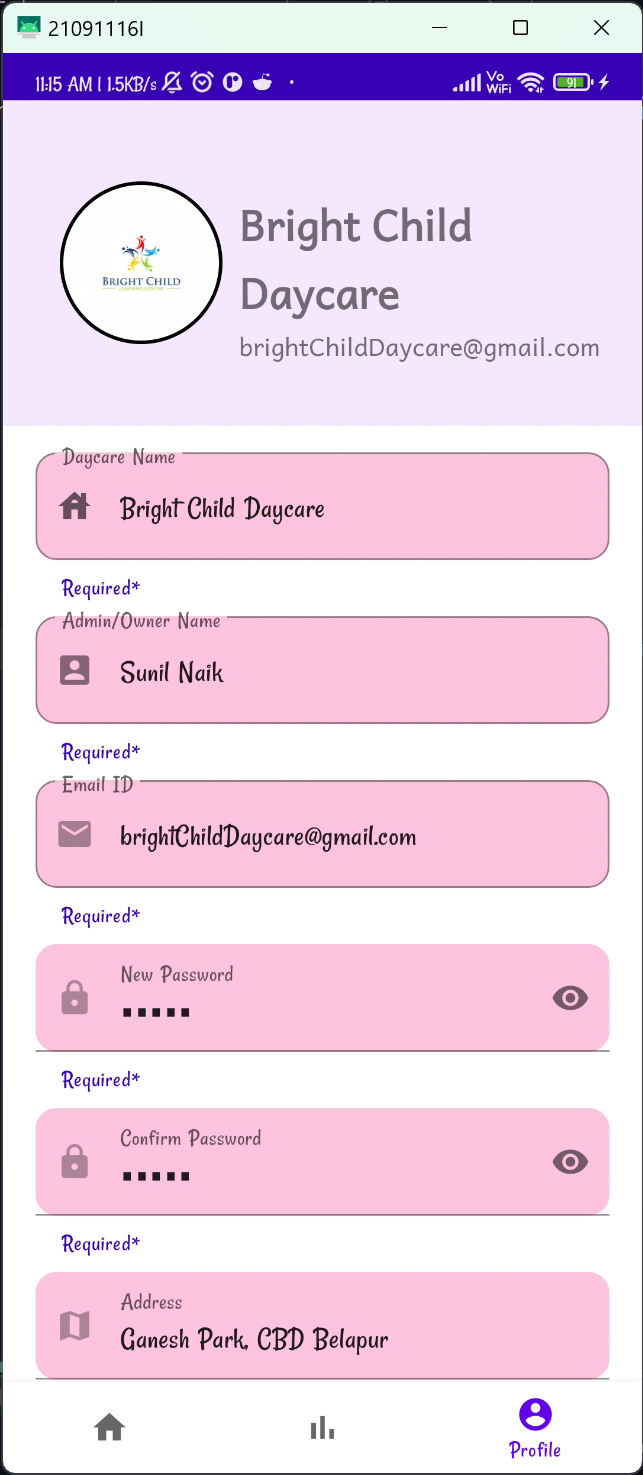
Daily Video uploading screen: Calling Screen:



Video Selection Screen: Video Details:

Profile of Owner:



**6.2 CODING**

**Parent App Coding:**

**Database Files:**

**DatabaseClass.java:**

package com.example.lilcare;

import com.google.android.gms.tasks.Task;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

public class DatabaseClass extends parentRegPage {

private DatabaseReference databaseReference;

public DatabaseClass(){

FirebaseDatabase db = FirebaseDatabase.getInstance();

databaseReference = db.getReference(Users.class.getSimpleName());

}

public Task<Void> add(Users user)

{

return databaseReference.push().setValue(user);

}

}

**User.java:**

package com.example.lilcare;

public class Users {

private String childPhoto;

private String childName;

private String childAge;

private String parentName;

private String parentEmail;

private String newPassword;

private String confPassword;

private String Phone1;

private String Phone2;

private String pAddress;

private String medicState;

public Users(){}

public Users(String childPhoto,

String childName, String childAge,

String parentName, String parentEmail,

String newPassword, String confPassword,

String Phone1, String Phone2,

String pAddress, String medicState){

this.childPhoto = childPhoto;

this.childName = childName;

this.childAge = childAge;

this.parentName = parentName;

this.parentEmail = parentEmail;

this.newPassword = newPassword;

this.confPassword = confPassword;

this.Phone1 = Phone1;

this.Phone2 = Phone2;

this.pAddress = pAddress;

this.medicState = medicState;

}

public String getChildPhoto() {

return childPhoto;

}

public void setChildPhoto(String childPhoto) {

this.childPhoto = childPhoto;

}

public String getParentName() {

return parentName;

}

public void setParentName(String parentName) {

this.parentName = parentName;

}

public String getChildName() {

return childName;

}

public void setChildName(String childName) {

this.childName = childName;

}

public String getChildAge() {

return childAge;

}

public void setChildAge(String childAge) {

this.childAge = childAge;

}

public String getParentEmail() {

return parentEmail;

}

public void setParentEmail(String parentEmail) {

this.parentEmail = parentEmail;

}

public String getNewPassword() {

return newPassword;

}

public void setNewPassword(String newPassword) {

this.newPassword = newPassword;

}

public String getConfPassword() {

return confPassword;

}

public void setConfPassword(String confPassword) {

this.confPassword = confPassword;

}

public String getPhone1() {

return Phone1;

}

public void setPhone1(String phone1) {

Phone1 = phone1;

}

public String getPhone2() {

return Phone2;

}

public void setPhone2(String phone2) {

Phone2 = phone2;

}

public String getpAddress() {

return pAddress;

}

public void setpAddress(String pAddress) {

this.pAddress = pAddress;

}

public String getMedicState() {

return medicState;

}

public void setMedicState(String medicState) {

this.medicState = medicState;

}

}

**App Files:**

**ParentSplash.java:**

package com.example.lilcare;  
import androidx.appcompat.app.AppCompatActivity;  
import android.content.Intent;  
import android.os.Bundle;  
import android.view.View;  
import android.view.animation.Animation;  
import android.view.animation.AnimationUtils;  
import android.widget.ImageView;  
import android.widget.LinearLayout;  
import android.widget.TextView;  
public class MainActivity extends AppCompatActivity {  
 ImageView bgslide, Parent, daycare;  
 LinearLayout textsplash, menu;  
 Animation bganim, frombottom;  
 TextView loginTXT, registerTXT;  
 @Override  
 protected void onCreate(Bundle savedInstanceState) {  
 super.onCreate(savedInstanceState);  
 setContentView(R.layout.activity\_main);  
 bgslide = (ImageView) findViewById(R.id.bgslide);  
 loginTXT = findViewById(R.id.loginTXT);  
 registerTXT = findViewById(R.id.registerTXT);  
 textsplash = (LinearLayout) findViewById(R.id.textsplash);  
 menu = (LinearLayout) findViewById(R.id.menu);  
 bganim = AnimationUtils.loadAnimation(this, R.anim.bganim);  
 frombottom = AnimationUtils.loadAnimation(this, R.anim.frombottom);  
 bgslide.animate().translationY(-1500).setDuration(1000).setStartDelay(800);  
 textsplash.animate().translationY(-1700).alpha(0).setDuration(1000).setStartDelay(800);  
 menu.animate().translationY(-100).setDuration(600).setStartDelay(1200).alpha(1.0f);  
 menu.setAnimation(frombottom);  
 loginTXT.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startActivity(new Intent(MainActivity.this, parentLogPage.class));  
 }  
 });  
 registerTXT.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {  
 startActivity(new Intent(MainActivity.this, parentRegPage.class));  
 }  
 });  
 }  
}

**ParentSpalsh.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".MainActivity">

<LinearLayout

android:id="@+id/menu"

android:gravity="center"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="200dp"

android:elevation="6dp"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="MissingConstraints">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="vertical">

<ImageView

android:id="@+id/Parent"

android:layout\_width="150dp"

android:layout\_height="200dp"

android:src="@drawable/family"

tools:ignore="InvalidId">

</ImageView>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/alike\_angular"

android:text="Welcome Parents!"

android:textColor="@color/black"

android:textStyle="bold"

android:textSize="25sp">

</TextView>

</LinearLayout>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="vertical"/>

</LinearLayout>

<TextView

android:id="@+id/loginTXT"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30sp"

android:fontFamily="@font/alike\_angular"

android:text="Existing user? Login..."

android:textColor="@color/black"

android:textStyle="bold"

android:textSize="15sp" />

<TextView

android:id="@+id/registerTXT"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20sp"

android:fontFamily="@font/alike\_angular"

android:text="New Here? Register..."

android:textColor="@color/black"

android:textStyle="bold"

android:textSize="15sp" />

</LinearLayout>

<ImageView

android:id="@+id/bgslide"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:adjustViewBounds="true"

android:scaleType="fitXY"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:srcCompat="@drawable/splashbg"

tools:ignore="MissingConstraints" />

<LinearLayout

android:id="@+id/textsplash"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="300dp"

android:elevation="6dp"

android:gravity="center"

android:orientation="vertical"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/smallsplash"

/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/secular\_one"

android:text="Lil Care"

android:textColor="#DA000000"

android:textSize="40sp" />

</LinearLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

**ParentLogpage.java:**

package com.example.lilcare;  
import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.Toast;  
import com.example.lilcare.databinding.ActivityParentLogBinding;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;

public class parentLogPage extends AppCompatActivity {

ActivityParentLogBinding parentLogBinding;

private FirebaseAuth auth;

DatabaseReference databaseReference = FirebaseDatabase.getInstance().getReferenceFromUrl("https://userauthenticationapp-4f6d3-default-rtdb.firebaseio.com/");

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

parentLogBinding = ActivityParentLogBinding.inflate(getLayoutInflater());

setContentView(parentLogBinding.getRoot());

auth = FirebaseAuth.getInstance();  
 parentLogBinding.parentLogBTN.setOnClickListener(new View.OnClickListener() {  
 @Override  
 public void onClick(View view) {

String txt\_email = parentLogBinding.parentLogID.getText().toString();

String txt\_pass = parentLogBinding.parentLogPass.getText().toString();

if (txt\_email.isEmpty() || txt\_pass.isEmpty()){

Toast.makeText(parentLogPage.this, "Please enter credentials!", Toast.LENGTH\_SHORT).show();

}

else {

databaseReference.child("Users").addListenerForSingleValueEvent(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

if (snapshot.hasChild(txt\_email)){

final String getPass = snapshot.child(txt\_pass).child("newPassword").getValue(String.class);

if (getPass.equals(txt\_pass)){

Toast.makeText(parentLogPage.this, "abc", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(parentLogPage.this, HomePage.class));

finish();

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

}

});

}

}

});

parentLogBinding.regforparent.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(parentLogPage.this, parentRegPage.class);

startActivity(i);

}

});

parentLogBinding.home.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(parentLogPage.this, HomePage.class);

startActivity(i);

}

});

}

}

**ParentLogpage.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical"

android:background="@drawable/login"

tools:context=".parentLogPage">

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="50dp"

android:fontFamily="@font/roboto\_light"

android:includeFontPadding="false"

android:text="Login Page"

android:textAlignment="center"

android:textAllCaps="false"

android:textColor="#874356"

android:textSize="45sp"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent" />

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout1"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="355dp"

android:layout\_height="wrap\_content"

android:layout\_marginStart="32dp"

android:layout\_marginTop="44dp"

android:layout\_marginEnd="32dp"

android:hint="PLogin ID"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:counterEnabled="true"

app:counterMaxLength="20"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textView"

app:startIconDrawable="@drawable/ic\_owner">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/parentLogID"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout2"

android:layout\_width="355dp"

android:layout\_height="wrap\_content"

android:layout\_marginTop="3dp"

android:hint="Password"

android:layout\_marginStart="32dp"

android:layout\_marginEnd="32dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:counterEnabled="true"

app:counterMaxLength="12"

app:endIconMode="password\_toggle"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout1"

app:startIconDrawable="@drawable/ic\_lock">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/parentLogPass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.button.MaterialButton

android:id="@+id/parentLogBTN"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="32dp"

android:backgroundTint="#C25792"

android:text="Login"

android:textColor="@color/white"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:cornerRadius="20dp"

app:icon="@drawable/ic\_next"

app:iconTint="@color/white"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.498"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout2">

</com.google.android.material.button.MaterialButton>

<com.google.android.material.button.MaterialButton

android:id="@+id/regforparent"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="100dp"

android:backgroundTint="#C25792"

android:text="Register Here For Parent"

android:textColor="@color/white"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:cornerRadius="20dp"

app:icon="@drawable/ic\_next"

app:iconTint="@color/white"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.496"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout2">

</com.google.android.material.button.MaterialButton>

<Button

android:id="@+id/home"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="180dp"

android:backgroundTint="#C25792"

android:text="Home"

android:textColor="@color/white"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:cornerRadius="20dp"

app:icon="@drawable/ic\_home"

app:iconTint="@color/white"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.496"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toBottomOf="@+id/textInputLayout2"/>

</androidx.constraintlayout.widget.ConstraintLayout>

**ParentRegisterpage.java:**

package com.example.lilcare;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Toast;

import com.example.lilcare.databinding.ActivityParentRegBinding;

public class parentRegPage extends AppCompatActivity {

ActivityParentRegBinding parentRegBinding;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

parentRegBinding = ActivityParentRegBinding.inflate(getLayoutInflater());

setContentView(parentRegBinding.getRoot());

DatabaseClass dao = new DatabaseClass();

parentRegBinding.nextbtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Users user = new Users(parentRegBinding.childProfile.getText().toString(),

parentRegBinding.childName.getText().toString(),

parentRegBinding.childAge.getText().toString(),

parentRegBinding.parentName.getText().toString(),

parentRegBinding.parentEmail.getText().toString(),

parentRegBinding.newPass.getText().toString(),

parentRegBinding.confirmPass.getText().toString(),

parentRegBinding.Phone1.getText().toString(),

parentRegBinding.Phone2.getText().toString(),

parentRegBinding.childAddress.getText().toString(),

parentRegBinding.medicState.getText().toString());

if (parentRegBinding.newPass.getText().toString().equals(parentRegBinding.confirmPass.getText().toString())) {

if (!(parentRegBinding.childName.getText().toString() == null)){

dao.add(user).addOnSuccessListener(suc -> {

Toast.makeText(parentRegPage.this, "Record Inserted", Toast.LENGTH\_SHORT).show();

}).addOnFailureListener(er -> {

Toast.makeText(parentRegPage.this, "Record Not Inserted" + er.getMessage(), Toast.LENGTH\_SHORT).show();

});

Intent i = new Intent(parentRegPage.this, HomePage.class);

startActivity(i);

}

}

else{

Toast.makeText(parentRegPage.this, "Confirm password doesn't match!", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

**ParentRegisterpage.xml:**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@drawable/login"

tools:context=".parentRegPage">

<TextView

android:id="@+id/textView"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginTop="50dp"

android:layout\_marginBottom="9dp"

android:fontFamily="@font/roboto\_light"

android:includeFontPadding="false"

android:paddingLeft="17sp"

android:text="Child`s Details"

android:textAllCaps="false"

android:textColor="#874356"

android:textSize="43sp" />

<ScrollView

android:id="@+id/scrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@id/textView"

>

<LinearLayout

android:layout\_marginTop="10dp"

android:id="@+id/linear\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout1"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Child's Photo URL"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditTex

android:id="@+id/childProfile"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL2"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Child's Name"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/childName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL6"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Child's Age"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/childAge"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayou

android:id="@+id/TL3"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Parent/Guardian Name"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/parentName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout2"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Email id"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_email">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/parentEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>  
 <com.google.android.material.textfield.TextInputLayout  
 android:id="@+id/textInputLayout4"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp”

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="New Password"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="password\_toggle"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_lock">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/newPass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL5"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Confirm Password"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="password\_toggle"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_lock">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/confirmPass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Address"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_address">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/childAddress"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout5"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Phone 1"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:counterEnabled="true"

app:counterMaxLength="10"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:prefixText="+91"

app:startIconDrawable="@drawable/ic\_phone">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Phone1"

android:layout\_width="match\_parent"

android:maxLength="10"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout6"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Phone 2"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:counterEnabled="true"

app:counterMaxLength="10"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:prefixText="+91"

app:startIconDrawable="@drawable/ic\_phone">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Phone2"

android:maxLength="10"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout8"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Any serious medical condition, if any?"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text”

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_phone">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/medicState"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.button.MaterialButton

android:id="@+id/nextbtn"

android:layout\_marginTop="14dp"

android:layout\_gravity="right"

android:backgroundTint="#C25792"

app:iconTint="@color/white"

android:text="Next"

android:hapticFeedbackEnabled="true"

android:textColor="@color/white"

app:cornerRadius="20dp"

app:icon="@drawable/ic\_next"

android:layout\_marginBottom="25dp"

android:layout\_marginRight="22dp"

>

</com.google.android.material.button.MaterialButton>

</LinearLayout>

</ScrollView>

</RelativeLayout>

**HomePage.java:**

package com.example.lilcare;  
import android.os.Bundle;  
import android.os.PersistableBundle;  
import android.view.MenuItem;  
import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import androidx.fragment.app.Fragment;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class HomePage extends AppCompatActivity {

@Override

public void onCreate( Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_home);

BottomNavigationView bottomNav= findViewById(R.id.bottom\_navigation);

bottomNav.setOnNavigationItemSelectedListener(navListener);

getSupportFragmentManager().beginTransaction().replace(R.id.fragment\_container,new HomeFragment()).commit();

}

private BottomNavigationView.OnNavigationItemSelectedListener navListener =

new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

Fragment selectedFragment = null;

switch(item.getItemId()) {

case R.id.nav\_home:

selectedFragment = new HomeFragment();

break;

case R.id.nav\_profile:

selectedFragment = new ProfileFragment();

break;

case R.id.nav\_data:

selectedFragment = new StatisticsFragment();

break;

// case R.id.nav\_data:

// selectedFragment = new AboutUs();

// break;

}

getSupportFragmentManager().beginTransaction().replace(R.id.fragment\_container,

selectedFragment).commit();

return true;

}

};

**HomePage.xml:**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".HomePage">

<FrameLayout

android:id="@+id/fragment\_container"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_above="@id/bottom\_navigation"/>

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottom\_navigation"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:menu="@menu/nav\_items"

app:labelVisibilityMode="selected"

android:layout\_alignParentBottom="true"

app:itemRippleColor="@color/purple\_100"

app:layout\_behavior="@string/hide\_bottom\_view\_on\_scroll\_behavior"

>

</com.google.android.material.bottomnavigation.BottomNavigationView>

</RelativeLayout>

**HomePageFragment.java:**

package com.example.lilcare;

import android.animation.LayoutTransition;

import android.content.DialogInterface;

import android.os.Build;

import android.os.Bundle;

import android.widget.Button;

import android.transition.AutoTransition;

import android.transition.TransitionManager;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toolbar;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.annotation.RequiresApi;

import androidx.appcompat.app.AppCompatActivity;

import androidx.cardview.widget.CardView;

import androidx.core.widget.NestedScrollView;

import androidx.fragment.app.Fragment;

public class HomeFragment extends Fragment {

@Nullable

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

TextView wel\_text, wel\_text2, wel\_text3;

LinearLayout cardlayout, cardlayout2, cardlayout3;

View view = inflater.inflate(R.layout.fragment\_home,container,false);

//Expandable-Card

wel\_text = view.findViewById(R.id.wel\_text);

wel\_text2 = view.findViewById(R.id.wel\_text2);

wel\_text3 = view.findViewById(R.id.wel\_text3);

cardlayout = view.findViewById(R.id.cardlayout);

cardlayout.getLayoutTransition().enableTransitionType(LayoutTransition.CHANGING);

cardlayout2 = view.findViewById(R.id.cardlayout2);

cardlayout2.getLayoutTransition().enableTransitionType(LayoutTransition.CHANGING);

cardlayout3 = view.findViewById(R.id.cardlayout3);

cardlayout3.getLayoutTransition().enableTransitionType(LayoutTransition.CHANGING);

//Cardview

CardView btn1= view.findViewById(R.id.expand);

CardView btn2= view.findViewById(R.id.expand2);

CardView btn3= view.findViewById(R.id.expand3);

btn1.setOnClickListener(new View.OnClickListener(){  
 @Override

public void onClick(View v) {

int u = (wel\_text.getVisibility() == View.GONE) ? View.VISIBLE : View.GONE;

TransitionManager.beginDelayedTransition(cardlayout, new AutoTransition());

wel\_text.setVisibility(u);

}

});

btn2.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

int w = (wel\_text2.getVisibility() == View.GONE) ? View.VISIBLE : View.GONE;

TransitionManager.beginDelayedTransition(cardlayout2, new AutoTransition());

wel\_text2.setVisibility(w);

}

});

btn3.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

int x = (wel\_text3.getVisibility() == View.GONE) ? View.VISIBLE : View.GONE;

TransitionManager.beginDelayedTransition(cardlayout3, new AutoTransition());

wel\_text3.setVisibility(x);

}

});

return view;

}

}

**HomeFragment.xml:**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/purple\_300"

tools:context="HomeFragment">

<androidx.core.widget.NestedScrollView

android:id="@+id/nestedlayout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<com.google.android.material.card.MaterialCardView

android:id="@+id/mat\_card"

android:layout\_width="match\_parent"

android:layout\_height="300dp"

android:layout\_margin="16dp"

android:animateLayoutChanges="true"

app:cardBackgroundColor="@color/white"

app:cardCornerRadius="15dp"

app:cardElevation="10dp"

app:contentPadding="8dp">

<androidx.core.widget.NestedScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:fontFamily="@font/noto\_sans"

android:text="@string/text"

android:textSize="20sp" />

</androidx.core.widget.NestedScrollView>

</com.google.android.material.card.MaterialCardView>

<com.google.android.material.card.MaterialCardView

android:id="@+id/mat\_card2"

android:layout\_width="match\_parent"

android:layout\_height="300dp"

android:layout\_margin="16dp"

android:animateLayoutChanges="true"

app:cardBackgroundColor="@color/white"

app:cardCornerRadius="15dp"

app:cardElevation="10dp"

app:contentPadding="8dp">

<androidx.core.widget.NestedScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:fontFamily="@font/noto\_sans"

android:text="@string/text"

android:textSize="20sp" />

</androidx.core.widget.NestedScrollView>

</com.google.android.material.card.MaterialCardView>

<androidx.cardview.widget.CardView

android:id="@+id/expand"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:backgroundTint="@color/white"

android:onClick="expand3"

app:cardCornerRadius="18dp"

app:cardElevation="10dp"

app:contentPadding="20dp">

<androidx.core.widget.NestedScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:id="@+id/cardlayout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:animateLayoutChanges="true"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/app\_name"

android:textColor="@color/purple\_700"

android:textSize="30sp" />

<TextView

android:id="@+id/wel\_text3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/text"

android:textSize="20sp"

android:visibility="gone" />

</LinearLayout>

</androidx.core.widget.NestedScrollView>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/expand2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:backgroundTint="@color/white"

android:onClick="expand"

app:cardCornerRadius="18dp"

app:cardElevation="10dp"

app:contentPadding="20dp">

<androidx.core.widget.NestedScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:id="@+id/cardlayout2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:animateLayoutChanges="true"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/app\_name"

android:textColor="@color/purple\_700"

android:textSize="30sp" />

<TextView

android:id="@+id/wel\_text"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/text"

android:textSize="20sp"

android:visibility="gone" />

</LinearLayout>

</androidx.core.widget.NestedScrollView>

</androidx.cardview.widget.CardView>

<androidx.cardview.widget.CardView

android:id="@+id/expand3"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_margin="20dp"

android:backgroundTint="@color/white"

android:onClick="expand2"

app:cardCornerRadius="18dp"

app:cardElevation="10dp"

app:contentPadding="20dp">

<androidx.core.widget.NestedScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

<LinearLayout

android:id="@+id/cardlayout3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:animateLayoutChanges="true"

android:orientation="vertical">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/app\_name"

android:textColor="@color/purple\_700"

android:textSize="30sp" />

<TextView

android:id="@+id/wel\_text2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="@string/text"

android:textSize="20sp"

android:visibility="gone" />

</LinearLayout>

</androidx.core.widget.NestedScrollView>

</androidx.cardview.widget.CardView>

</LinearLayout>

</androidx.core.widget.NestedScrollView>

</RelativeLayout>

**StatisticsFragment.java:**

package com.example.lilcare;

import android.animation.LayoutTransition;

import android.os.Bundle;

import android.transition.AutoTransition;

import android.transition.TransitionManager;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.LinearLayout;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.cardview.widget.CardView;

import androidx.fragment.app.Fragment;

public class StatisticsFragment extends Fragment {

@Override

public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {

TextView wel\_text;

LinearLayout cardlayout;  
View view = inflater.inflate(R.layout.fragment\_statistics,container,false);

wel\_text = view.findViewById(R.id.wel\_text);

cardlayout = view.findViewById(R.id.cardlayout);

cardlayout.getLayoutTransition().enableTransitionType(LayoutTransition.CHANGING);

CardView btn1= view.findViewById(R.id.expand);

CardView btn2= view.findViewById(R.id.expand2);

btn1.setOnClickListener(new View.OnClickListener(){

@Override

public void onClick(View v) {

int u = (wel\_text.getVisibility() == View.GONE) ? View.VISIBLE : View.GONE;

TransitionManager.beginDelayedTransition(cardlayout, new AutoTransition());

wel\_text.setVisibility(u);

}

});

return view;

}

}

**StatisticsFragment.xml:**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/green"

tools:context=".StatisticsFragment">

<androidx.core.widget.NestedScrollView

android:id="@+id/nestedlayout"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

>

<androidx.cardview.widget.CardView

android:id="@+id/expand"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

app:cardCornerRadius="18dp"

app:cardElevation="10dp"

app:contentPadding="20dp"

android:layout\_margin="20dp"

android:backgroundTint="@color/white"

>

<androidx.core.widget.NestedScrollView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:clipToPadding="false"

app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior"

>

<LinearLayout

android:id="@+id/cardlayout"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="vertical"

android:animateLayoutChanges="true">

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Data &amp; Analysis"

android:layout\_gravity="center"

android:textSize="30sp"

android:textColor="@color/purple\_700"/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="This Page Will Ungergo A Massive OverHaul.\nChanges will be implemented later.\nStay Tuned!!!!"

android:textSize="22sp"

android:id="@+id/wel\_text"

/>

</LinearLayout></androidx.core.widget.NestedScrollView>

</androidx.cardview.widget.CardView>

</androidx.core.widget.NestedScrollView>

</RelativeLayout>

ProfileFragment.java:

package com.example.lilcare;  
import android.content.Intent;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.LinearLayout;

import android.widget.TextView;

import android.widget.Toast;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.cardview.widget.CardView;

import androidx.fragment.app.Fragment;

import com.bumptech.glide.Glide;

import com.example.lilcare.databinding.FragmentProfilePageBinding;

import com.google.firebase.database.DataSnapshot;

import com.google.firebase.database.DatabaseError;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.database.ValueEventListener;  
public class ProfileFragment extends Fragment {  
 public static final String TAG = "1";

private DatabaseReference databaseReference;

private FirebaseDatabase firebaseDatabase;

public static final String USERS = "Users";

private FragmentProfilePageBinding binding;

@Nullable

@Override

public View onCreateView( LayoutInflater inflater, ViewGroup container, Bundle savedInstanceState) {

binding = FragmentProfilePageBinding.inflate(inflater,container,false);

// Button btn = view.findViewById(R.id.aboutUs);

// btn.setOnClickListener(new View.OnClickListener() {

// @Override

// public void onClick(View view) {

// Intent i = new Intent(getActivity(), AboutUs.class);

// startActivity(i);

// }

// });

firebaseDatabase = FirebaseDatabase.getInstance();

databaseReference = FirebaseDatabase.getInstance().getReference(USERS);

databaseReference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

for (DataSnapshot ds : snapshot.getChildren()){

if (ds.child("parentEmail").getValue().equals("gonsalves@gmail.com")){

Glide.with(binding.profileImg.getContext()).load(ds.child("childPhoto").getValue(String.class)).into(binding.profileImg);

binding.fullName.setText(ds.child("childName").getValue(String.class));

binding.email.setText(ds.child("parentEmail").getValue(String.class));

binding.childName.setText(ds.child("childName").getValue(String.class));

binding.childAge.setText(ds.child("childAge").getValue(String.class));

binding.parentName.setText(ds.child("parentName").getValue(String.class));

binding.parentEmail.setText(ds.child("parentEmail").getValue(String.class));

binding.newPass.setText(ds.child("newPassword").getValue(String.class));

binding.confirmPass.setText(ds.child("confPassword").getValue(String.class));

binding.childAddress.setText(ds.child("pAddress").getValue(String.class));

binding.Phone1.setText(ds.child("phone1").getValue(String.class));

binding.Phone2.setText(ds.child("phone2").getValue(String.class));

binding.medicState.setText(ds.child("medicState").getValue(String.class));

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

Toast.makeText(getContext(), "Unable to retrieve data", Toast.LENGTH\_SHORT).show();

}

});

return binding.getRoot();

}

}

**ProfileFragment.xml**:

<?xml version="1.0" encoding="utf-8"?>

<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="fill\_parent"

android:orientation="vertical"

tools:context=".ProfileFragment">

<!-- TODO: Update blank fragment layout -->

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="265dp"

android:background="@color/light"

android:padding="20dp">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/profile\_img"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_centerVertical="true"

android:layout\_marginStart="15dp"

android:layout\_marginBottom="10dp"

app:civ\_border\_width="2dp"

app:civ\_border\_color="@color/black"

android:src="@drawable/ic\_account" />

<TextView

android:id="@+id/full\_name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentTop="true"

android:layout\_marginLeft="10dp"

android:layout\_marginTop="70dp"

android:layout\_toRightOf="@+id/profile\_img"

android:fontFamily="@font/andika\_new\_basic"

android:includeFontPadding="false"

android:text="Name"

android:textSize="22sp"

android:textStyle="bold" />

<TextView

android:id="@+id/email"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/full\_name"

android:layout\_marginLeft="10dp"

android:layout\_toRightOf="@+id/profile\_img"

android:fontFamily="@font/andika\_new\_basic"

android:includeFontPadding="false"

android:text="email"

android:textSize="14sp" />

</RelativeLayout>

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="20dp"

android:layout\_marginTop="-100dp">

<com.google.android.material.card.MaterialCardView

android:layout\_width="0dp"

android:layout\_height="125dp"

android:layout\_weight="1"

android:layout\_margin="10dp"

app:cardBackgroundColor="@color/light"

app:cardElevation="10dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="5dp">

<ImageView

android:layout\_width="30dp"

android:layout\_height="30dp"

android:src="@drawable/payment"

android:layout\_centerHorizontal="true"/>

<TextView

android:id="@+id/payment\_label"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="$123"

android:textSize="20sp"

android:fontFamily="@font/roboto\_light"

android:includeFontPadding="false"

android:textStyle="bold"

android:layout\_centerInParent="true"/>

<TextView

android:id="@+id/payment\_desc"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="39"

android:textSize="14sp"

android:fontFamily="@font/roboto\_light"

android:includeFontPadding="false"

android:layout\_below="@+id/payment\_label"

android:layout\_centerHorizontal="true"/>

</RelativeLayout>

</com.google.android.material.card.MaterialCardView>

<com.google.android.material.card.MaterialCardView

android:layout\_width="0dp"

android:layout\_height="125dp"

android:layout\_weight="1"

android:layout\_margin="10dp"

app:cardBackgroundColor="@color/light"

app:cardElevation="10dp">

<RelativeLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:padding="5dp">

<ImageView

android:layout\_width="30dp"

android:layout\_height="30dp"

android:src="@drawable/payment"

android:layout\_centerHorizontal="true"/>

<TextView

android:id="@+id/booking\_label"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="$123"

android:textSize="20sp"

android:fontFamily="@font/roboto\_light"

android:includeFontPadding="false"

android:textStyle="bold"

android:layout\_centerInParent="true"/>

<TextView

android:id="@+id/booking\_desc"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/booking\_label"

android:layout\_centerHorizontal="true"

android:fontFamily="@font/roboto\_light"

android:includeFontPadding="false"

android:text="39"

android:textSize="14sp" />

</RelativeLayout>

</com.google.android.material.card.MaterialCardView>

</LinearLayout>

<ScrollView

android:id="@+id/scrollView"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_below="@id/textView"

>  
 <LinearLayout

android:layout\_marginTop="10dp"

android:id="@+id/linear\_layout"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:orientation="vertical"

>  
 <com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL2"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Child's Name"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/childName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL6"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Child's Age"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/childAge"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL3"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:hint="Parent/Guardian Name"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_account">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/parentName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout2"

style="@style/Widget.MaterialComponents.TextInputLayout.OutlinedBox"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Email id"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_email">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/parentEmail"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:singleLine="true" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout4"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="New Password"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="password\_toggle"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_lock">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/newPass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/TL5"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Confirm Password"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="password\_toggle"

app:helperText="Required\*

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_lock">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/confirmPass"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:inputType="textPassword" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout3"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Address"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_address">  
 <com.google.android.material.textfield.TextInputEditText

android:id="@+id/childAddress"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout5"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Phone 1"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:counterEnabled="true"

app:counterMaxLength="10"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:prefixText="+91"

app:startIconDrawable="@drawable/ic\_phone">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Phone1"

android:layout\_width="match\_parent"

android:maxLength="10"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout6"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Phone 2"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:counterEnabled="true"

app:counterMaxLength="10"

app:endIconMode="clear\_text"

app:helperText="Required\*"

app:helperTextTextColor="@color/purple\_700"

app:prefixText="+91"

app:startIconDrawable="@drawable/ic\_phone">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/Phone2"

android:maxLength="10"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<com.google.android.material.textfield.TextInputLayout

android:id="@+id/textInputLayout8"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginTop="6dp"

android:layout\_marginRight="20dp"

android:layout\_marginLeft="20dp"

android:hint="Any serious medical condition, if any?"

app:boxBackgroundColor="#FEC3DE"

app:boxCornerRadiusBottomEnd="13dp"

app:boxCornerRadiusBottomStart="13dp"

app:boxCornerRadiusTopEnd="13dp"

app:boxCornerRadiusTopStart="13dp"

app:endIconMode="clear\_text"

app:helperTextTextColor="@color/purple\_700"

app:startIconDrawable="@drawable/ic\_phone">

<com.google.android.material.textfield.TextInputEditText

android:id="@+id/medicState"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content" />

</com.google.android.material.textfield.TextInputLayout>

<Button  
 android:id="@+id/button"  
 android:layout\_gravity="end"  
 android:layout\_width="wrap\_content"

android:layout\_marginBottom="10dp"

android:layout\_marginRight="24dp"

android:layout\_marginTop="10dp"

android:layout\_height="wrap\_content"

android:paddingTop="15dp"

android:paddingBottom="10dp"

android:text="Update" />

</LinearLayout>

</ScrollView>

</LinearLayout>

**Owner App:**

**Coding:**

**Database Files:**

**adminModel.java:**

package com.example.recyclerviewdemoproj;

public class adminModel {

private String dcID;

private String dcName;

private String dcAdminName;

private String dcEmail;

private String dcNewPass;

private String dcConfPass;

private String dcAddress;

private String dcP1;

private String dcP2;

private String dcOcc;

public adminModel(){}

public adminModel(String dcID, String dcName,

String dcAdminName, String dcEmail,

String dcNewPass, String dcConfPass,

String dcAddress, String dcP1,

String dcP2, String dcOcc)

{

this.dcID = dcID

this.dcName = dcName;

this.dcAdminName = dcAdminName;

this.dcEmail = dcEmail;

this.dcNewPass = dcNewPass;

this.dcConfPass = dcConfPass;

this.dcAddress = dcAddress;

this.dcP1 = dcP1;

this.dcP2 = dcP2;

this.dcOcc = dcOcc;

}

public String getDcID() {

return dcID;

}

public void setDcID(String dcID) {

this.dcID = dcID;

}

public String getDcName() {

return dcName;

}

public void setDcName(String dcName) {

this.dcName = dcName;

}

public String getDcAdminName() {

return dcAdminName;

}  
 public void setDcAdminName(String dcAdminName) {

this.dcAdminName = dcAdminName;

}

public String getDcEmail() {

return dcEmail;

}

public void setDcEmail(String dcEmail) {

this.dcEmail = dcEmail;

}

public String getDcNewPass() {

return dcNewPass;

}  
 public void setDcNewPass(String dcNewPass) {

this.dcNewPass = dcNewPass;

}

public String getDcConfPass() {

return dcConfPass;

}

public void setDcConfPass(String dcConfPass) {

this.dcConfPass = dcConfPass;

}

public String getDcAddress() {

return dcAddress;

}

public void setDcAddress(String dcAddress) {

this.dcAddress = dcAddress;

}

public String getDcP1() {

return dcP1;

}

public void setDcP1(String dcP1) {

this.dcP1 = dcP1;

}

public String getDcP2() {

return dcP2;

}

public void setDcP2(String dcP2) {

this.dcP2 = dcP2;

}

public String getDcOcc() {

return dcOcc;

}

public void setDcOcc(String dcOcc) {

this.dcOcc = dcOcc;

}

}

**model.java:**

package com.example.recyclerviewdemoproj;

public class model {

private String childPhoto;

private String childName;

private String childAge;

private String parentName;

private String parentEmail;

private String newPassword;

private String confPassword;

private String Phone1;

private String Phone2;

private String pAddress;

private String medicState;

public model(){}

public model(String childPhoto,

String childName, String childAge,

String parentName, String parentEmail,

String newPassword, String confPassword,

String Phone1, String Phone2,

String pAddress, String medicState){

this.childPhoto = childPhoto;

this.childName = childName;

this.childAge = childAge;

this.parentName = parentName;

this.parentEmail = parentEmail;

this.newPassword = newPassword;

this.confPassword = confPassword;

this.Phone1 = Phone1;

this.Phone2 = Phone2;

this.pAddress = pAddress;

this.medicState = medicState;

}

public String getChildPhoto() {

return childPhoto;

}

public void setChildPhoto(String childPhoto) {

this.childPhoto = childPhoto;

}

public String getParentName() {

return parentName;

}

public void setParentName(String parentName) {

this.parentName = parentName;

}

public String getChildName() {

return childName;

}

public void setChildName(String childName) {

this.childName = childName;

}

public String getChildAge() {

return childAge;

}

public void setChildAge(String childAge) {

this.childAge = childAge;

}

public String getParentEmail() {

return parentEmail;

}

public void setParentEmail(String parentEmail) {

this.parentEmail = parentEmail;

}

public String getNewPassword() {

return newPassword;

}

public void setNewPassword(String newPassword) {

this.newPassword = newPassword;

}

public String getConfPassword() {

return confPassword;

}

public void setConfPassword(String confPassword) {

this.confPassword = confPassword;

}

public String getPhone1() {

return Phone1;

}

public void setPhone1(String phone1) {

Phone1 = phone1;

}

public String getPhone2() {

return Phone2;

}

public void setPhone2(String phone2) {

Phone2 = phone2;

}

public String getpAddress() {

return pAddress;

}

public void setpAddress(String pAddress) {

this.pAddress = pAddress;

}

public String getMedicState() {

return medicState;

}

public void setMedicState(String medicState) {

this.medicState = medicState;

}

}

**fileModel.java:**

package com.example.recyclerviewdemoproj;

public class fileModel {

String title,date,vUrl;

public fileModel() {

}

public fileModel(String title, String date, String vUrl){

this.title = title;

this.vUrl = vUrl;

this.date = date;

}

public String getTitle() {

return title;

}

public void setTitle(String title) {

this.title = title;

}

public String getvUrl() {

return vUrl;

}

public void setvUrl(String vUrl) {

this.vUrl = vUrl;

}

public String getDate() {

return date;

}

public void setDate(String date) {

this.date = date;

}

}

**App Files:**

**ownerSplash.java:**

package com.example.recyclerviewdemoproj;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

import android.widget.LinearLayout;

import android.widget.TextView;

import java.util.Objects;

public class OwnerSplash extends AppCompatActivity {

ImageView bgslide, Parent, daycare;

LinearLayout textsplash, menu;

Animation bganim, frombottom;

TextView loginTXT, registerTXT;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_owner\_splash);

Objects.requireNonNull(getSupportActionBar()).hide();

bgslide = (ImageView) findViewById(R.id.bgslide);

loginTXT = findViewById(R.id.loginTXT);

registerTXT = findViewById(R.id.registerTXT);

textsplash = (LinearLayout) findViewById(R.id.textsplash);

menu = (LinearLayout) findViewById(R.id.menu);

bganim = AnimationUtils.loadAnimation(this, R.anim.bganim);

frombottom = AnimationUtils.loadAnimation(this, R.anim.frombottom);

bgslide.animate().translationY(-1500).setDuration(1000).setStartDelay(800);

textsplash.animate().translationY(-1700).alpha(0).setDuration(1000).setStartDelay(800);

menu.animate().translationY(-100).setDuration(600).setStartDelay(1200).alpha(1.0f);

menu.setAnimation(frombottom);

loginTXT.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

startActivity(new Intent(OwnerSplash.this, OwnerLogin.class));

finish();

}

});

registerTXT.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

startActivity(new Intent(OwnerSplash.this, HomeRecycle.class));

finish();

}

});

}

}

**ownersplash.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context=".OwnerSplash">

<LinearLayout

android:id="@+id/menu"

android:gravity="center"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="200dp"

android:elevation="6dp"

android:orientation="vertical"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

tools:ignore="MissingConstraints">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:orientation="horizontal">

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="vertical">

<ImageView

android:id="@+id/Parent"

android:layout\_width="150dp"

android:layout\_height="200dp"

android:src="@drawable/daycare"

tools:ignore="InvalidId">

</ImageView

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/alike\_angular"

android:text="Welcome Owners!"

android:textColor="@color/black"

android:textSize="25sp"

android:textStyle="bold">

</TextView>

</LinearLayout>

<LinearLayout

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:gravity="center"

android:orientation="vertical" />

</LinearLayout>

<TextView

android:id="@+id/loginTXT"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="30sp"

android:fontFamily="@font/alike\_angular"

android:text="Existing owner? Login..."

android:textColor="@color/black"

android:textStyle="bold"

android:textSize="15sp" />

<TextView

android:id="@+id/registerTXT"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginTop="20sp"

android:fontFamily="@font/alike\_angular"

android:text="New owner? Register..."

android:textColor="@color/black"

android:textStyle="bold"

android:textSize="15sp" />

</LinearLayout>

<ImageView

android:id="@+id/bgslide"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:adjustViewBounds="true"

android:scaleType="fitXY"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintHorizontal\_bias="0.0"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintTop\_toTopOf="parent"

app:srcCompat="@drawable/splashbg"

tools:ignore="MissingConstraints" />

<LinearLayout

android:id="@+id/textsplash"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_marginEnd="8dp"

android:layout\_marginStart="8dp"

android:layout\_marginTop="300dp"

android:elevation="6dp"

android:gravity="center"

android:orientation="vertical"

app:layout\_constraintTop\_toTopOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:layout\_constraintEnd\_toEndOf="parent">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/smallsplash"

/>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:fontFamily="@font/secular\_one"

android:text="Lil Care"  
 android:textColor="#DA000000"  
 android:textSize="40sp" />  
 </LinearLayout>  
</androidx.constraintlayout.widget.ConstraintLayout>

**ownerLogin.java:**

package com.example.recyclerviewdemoproj;  
import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.util.Log;

import android.util.Patterns;

import android.view.View;

import android.view.WindowManager;

import android.widget.Toast;

import com.example.recyclerviewdemoproj.databinding.ActivityOwnerLoginBinding;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseAuthException;

import java.util.Objects;

import java.util.regex.Pattern;

public class OwnerLogin extends AppCompatActivity {  
 ActivityOwnerLoginBinding loginBinding;  
 private FirebaseAuth mAuth;  
 private static final Pattern PASSWORD\_PATTERN =  
 Pattern.compile("^" +  
 "(?=.\*[@#$%^&+=])" + // at least 1 special character

"(?=\\S+$)" + // no white spaces

".{4,}" + // at least 4 characters

"$");

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

loginBinding = ActivityOwnerLoginBinding.inflate(getLayoutInflater());

setContentView(loginBinding.getRoot());

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,WindowManager.LayoutParams.FLAG\_FULLSCREEN);

Objects.requireNonNull(getSupportActionBar()).hide();

mAuth = FirebaseAuth.getInstance();

loginBinding.adminLogBTN.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

if (validatePassword() | validateEmail()){

loginBinding.bar.setVisibility(View.VISIBLE);  
 String email = loginBinding.adminLogInTB.getEditText().getText().toString();

String password = loginBinding.adminLogPassTB.getEditText().getText().toString();

mAuth.signInWithEmailAndPassword(email,password)

.addOnCompleteListener(OwnerLogin.this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()){

loginBinding.bar.setVisibility(View.INVISIBLE);

Intent intent = new Intent(OwnerLogin.this, HomeRecycle.class);

intent.putExtra("email", mAuth.getCurrentUser().getEmail());

intent.putExtra("password", mAuth.getCurrentUser().getUid());

Toast.makeText(OwnerLogin.this, "Logged in Successfully!", Toast.LENGTH\_SHORT).show();

startActivity(intent);

finish();

}

else {

loginBinding.bar.setVisibility(View.INVISIBLE);

loginBinding.adminLogID.setText("");

loginBinding.adminLogPass.setText("");

Toast.makeText(OwnerLogin.this, "Invalid User details!", Toast.LENGTH\_SHORT).show();

}

}

});

}

}

});

loginBinding.regforowner.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(OwnerLogin.this, OwnerRegister.class);

startActivity(i);

}

});

}

private boolean validateEmail() {

String emailInput = loginBinding.adminLogInTB.getEditText().getText().toString().trim();

if (emailInput.isEmpty()) {

loginBinding.adminLogInTB.setError("Field can not be empty");

return false;

}

else if (!Patterns.EMAIL\_ADDRESS.matcher(emailInput).matches()) {

loginBinding.adminLogInTB.setError("Please enter a valid email address");

return false;

} else {

loginBinding.adminLogInTB.setError(null);

return true;

}

}

private boolean validatePassword() {

String passwordInput = loginBinding.adminLogPassTB.getEditText().getText().toString().trim();

if (passwordInput.isEmpty()) {

loginBinding.adminLogPassTB.setError("Field can not be empty");

return false;

}

else if (!PASSWORD\_PATTERN.matcher(passwordInput).matches()) {

loginBinding.adminLogPassTB.setError("Password is too weak");

return false;

} else {

loginBinding.adminLogPassTB.setError(null);

return true;

}

}

}

**ownerRegister.java:**

package com.example.recyclerviewdemoproj;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity

import android.content.Intent;

import android.os.Bundle;

import android.util.Patterns;

import android.view.View;

import android.view.WindowManager;

import android.widget.Toast;

import com.example.recyclerviewdemoproj.databinding.ActivityOwnerRegisterBinding;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

import java.util.Objects;

import java.util.regex.Pattern;

public class OwnerRegister extends AppCompatActivity {

ActivityOwnerRegisterBinding registerBinding;

private FirebaseAuth mAuth;  
 private static final Pattern PASSWORD\_PATTERN =

Pattern.compile("^" +

"(?=.\*[@#$%^&+=])" + // at least 1 special character

"(?=\\S+$)" + // no white spaces

".{4,}" + // at least 4 characters

"$");

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

registerBinding = ActivityOwnerRegisterBinding.inflate(getLayoutInflater());

setContentView(registerBinding.getRoot());

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,WindowManager.LayoutParams.FLAG\_FULLSCREEN);

Objects.requireNonNull(getSupportActionBar()).hide();

DatabaseClass dbclass = new DatabaseClass();

registerBinding.nextbtn.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Boolean isInputEmpty = validateTextFields();

adminModel adminModel = new adminModel(registerBinding.dcLogo.getText().toString(),

registerBinding.dcName.getText().toString(),

registerBinding.dcAdminName.getText().toString(),

registerBinding.dcAdminEmail.getText().toString(),

registerBinding.dcNewPass.getText().toString(),

registerBinding.dcConfPass.getText().toString(),

registerBinding.dcAddress.getText().toString(),

registerBinding.dcPhone1.getText().toString(),

registerBinding.dcPhone2.getText().toString(),

registerBinding.dcOccupancy.getText().toString());

if (!isInputEmpty){

if (validateEmail() | validatePassword()){

if (registerBinding.dcNewPass.getText().toString().equals(registerBinding.dcConfPass.getText().toString())){

String email = registerBinding.dcAdminEmail.getText().toString();

String password = registerBinding.dcNewPass.getText().toString();

mAuth = FirebaseAuth.getInstance();

mAuth.createUserWithEmailAndPassword(email,password)

.addOnCompleteListener(OwnerRegister.this, new OnCompleteListener<AuthResult>() {

@Override

public void onComplete(@NonNull Task<AuthResult> task) {

if (task.isSuccessful()) {

registerBinding.dcAdminEmail.setText("");

registerBinding.dcNewPass.setText("");

Toast.makeText(OwnerRegister.this, "Registration Successful", Toast.LENGTH\_SHORT).show();

} else {

registerBinding.dcAdminEmail.setText("");

registerBinding.dcNewPass.setText("");

Toast.makeText(OwnerRegister.this, "Registration Failed", Toast.LENGTH\_SHORT).show();

}

}

});

dbclass.add(adminModel).addOnSuccessListener(suc -> {

Toast.makeText(OwnerRegister.this, "Record Inserted", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(OwnerRegister.this, OwnerLogin.class));

}).addOnFailureListener(er -> {

Toast.makeText(OwnerRegister.this, "Record Not Inserted" + er.getMessage(), Toast.LENGTH\_SHORT).show();

});

}else {

registerBinding.dcConfPassTB.setError("Confirm Password not matching!");

}

}

}

}

});

}

private boolean validateEmail() {

String emailInput = registerBinding.dcAdminEmailTB.getEditText().getText().toString().trim();

if (emailInput.isEmpty()) {

registerBinding.dcAdminEmailTB.setError("Field can not be empty");

registerBinding.dcAdminEmailTB.requestFocus();

return false;

}

else if (!Patterns.EMAIL\_ADDRESS.matcher(emailInput).matches()) {

registerBinding.dcAdminEmailTB.setError("Please enter a valid email address");

registerBinding.dcAdminEmailTB.requestFocus();

return false;

} else {

registerBinding.dcAdminEmailTB.setError(null);

return true;

}

}

private boolean validatePassword() {

String passwordInput = registerBinding.dcNewPassTB.getEditText().getText().toString().trim();

if (passwordInput.isEmpty()) {

registerBinding.dcNewPassTB.setError("Field can not be empty");

registerBinding.dcNewPassTB.requestFocus();

return false;

}

else if (!PASSWORD\_PATTERN.matcher(passwordInput).matches()) {

registerBinding.dcNewPassTB.setError("Password is too weak");

registerBinding.dcNewPassTB.requestFocus();

return false;

} else {

registerBinding.dcNewPassTB.setError(null);

return true;

}

}

private Boolean validateTextFields(){

Boolean isEmptyFields = false;

String dcID = registerBinding.dcLogoTB.getEditText().getText().toString().trim();

String dcName = registerBinding.dcNameTB.getEditText().getText().toString().trim();

String dcADMIN = registerBinding.dcAdminNameTB.getEditText().getText().toString().trim();

String dcAddress = registerBinding.dcAddressTB.getEditText().getText().toString().trim();

String dcP1 = registerBinding.dcPhone1TB.getEditText().getText().toString().trim();

String dcP2 = registerBinding.dcPhone2TB.getEditText().getText().toString().trim();

String dcOcc = registerBinding.dcOccupancyTB.getEditText().getText().toString().trim();

if (dcID == null || dcID.equalsIgnoreCase("")){

registerBinding.dcLogoTB.setError("Fields can't be empty");

registerBinding.dcLogoTB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcLogoTB.setError(null);

isEmptyFields = false;

}

if (dcName == null || dcName.equalsIgnoreCase("")){

registerBinding.dcNameTB.setError("Fields can't be empty");

registerBinding.dcNameTB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcNameTB.setError(null);

isEmptyFields = false;

}

if (dcADMIN == null || dcADMIN.equalsIgnoreCase("")){

registerBinding.dcAdminNameTB.setError("Fields can't be empty");

registerBinding.dcAdminNameTB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcAdminNameTB.setError(null);

isEmptyFields = false;

}

if (dcAddress == null || dcAddress.equalsIgnoreCase("")){

registerBinding.dcAddressTB.setError("Fields can't be empty");

registerBinding.dcAddressTB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcAddressTB.setError(null);

isEmptyFields = false;  
 }

if (dcP1 == null || dcP1.equalsIgnoreCase("")){

registerBinding.dcPhone1TB.setError("Fields can't be empty");

registerBinding.dcPhone1TB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcPhone1TB.setError(null);

isEmptyFields = false;

}

if (dcP2 == null || dcP2.equalsIgnoreCase("")){

registerBinding.dcPhone2TB.setError("Fields can't be empty");

registerBinding.dcPhone2TB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcPhone2TB.setError(null);

isEmptyFields = false;

}

if (dcOcc == null || dcOcc.equalsIgnoreCase("")){

registerBinding.dcOccupancyTB.setError("Fields can't be empty");

registerBinding.dcOccupancyTB.requestFocus();

isEmptyFields = true;

}

else {

registerBinding.dcOccupancyTB.setError(null);

isEmptyFields = false;

}

return isEmptyFields;

}

}

**HomeRecycle.java:**

package com.example.recyclerviewdemoproj;

import androidx.annotation.NonNull;

import androidx.appcompat.app.ActionBar;

import androidx.appcompat.app.AppCompatActivity;

import androidx.cardview.widget.CardView;

import androidx.fragment.app.Fragment;

import androidx.recyclerview.widget.LinearLayoutManager;

import androidx.recyclerview.widget.RecyclerView;

import android.animation.LayoutTransition;

import android.content.Intent;

import android.os.Bundle;

import android.text.Layout;

import android.transition.AutoTransition;

import android.transition.TransitionManager;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.view.ViewTreeObserver;

import android.view.Window;

import android.view.WindowManager;

import android.view.animation.Animation;

import android.widget.LinearLayout;

import android.widget.RelativeLayout;

import android.widget.SearchView;

import com.firebase.ui.database.FirebaseRecyclerOptions;

import com.google.android.material.bottomnavigation.BottomNavigationView;

import com.google.firebase.database.FirebaseDatabase;

public class HomeRecycle extends AppCompatActivity {  
 RecyclerView recView;

myAdapter adapter;

BottomNavigationView bottomNavigationView;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

getWindow().setFlags(WindowManager.LayoutParams.FLAG\_FULLSCREEN,WindowManager.LayoutParams.FLAG\_FULLSCREEN);

setContentView(R.layout.activity\_main);

recView = findViewById(R.id.recView);

recView.setLayoutManager(new LinearLayoutManager(this));

// recView.addItemDecoration(new DividerItemDecoration(recView.getContext(), DividerItemDecoration.VERTICAL));  
 FirebaseRecyclerOptions<model> options =

new FirebaseRecyclerOptions.Builder<model>()

.setQuery(FirebaseDatabase.getInstance().getReference().child("Users"), model.class)

.build();

adapter = new myAdapter(options);

recView.setAdapter(adapter);

bottomNavigationView = findViewById(R.id.bottom\_navigation);

bottomNavigationView.setSelectedItemId(R.id.nav\_home);

bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

switch(item.getItemId())

{

case R.id.nav\_home:

return true;

case R.id.nav\_profile:

startActivity(new Intent(getApplicationContext(),Profile.class));

overridePendingTransition(0,0);

return true;

case R.id.nav\_subscribe:

startActivity(new Intent(getApplicationContext(),Subscription.class));

overridePendingTransition(0,0);

return true;

}

return false;

}

});

}

@Override

protected void onStart(){

super.onStart();

adapter.startListening();

}

// @Override

// protected void onStop(){

// super.onStop();

// adapter.stopListening();

// }

@Override

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.searchmenu, menu);

MenuItem item = menu.findItem(R.id.search);

SearchView searchView = (SearchView)item.getActionView();

searchView.setOnQueryTextListener(new SearchView.OnQueryTextListener() {

@Override

public boolean onQueryTextSubmit(String s) {

processSearch(s);

return false;

}

@Override

public boolean onQueryTextChange(String s) {

processSearch(s);

return false;

}

});

return super.onCreateOptionsMenu(menu);

}

private void processSearch(String s) {

FirebaseRecyclerOptions<model> options =

new FirebaseRecyclerOptions.Builder<model>() setQuery(FirebaseDatabase.getInstance().getReference().child("Users").orderByChild("childAge").startAt(s).endAt(s+"\uf8ff"), model.class)

.build();

adapter = new myAdapter(options);

adapter.startListening();

recView.setAdapter(adapter);

}

}

**HomeRecycle.xml:**

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="@color/daycareBG"

android:gravity="center"

tools:context=".HomeRecycle">

<androidx.cardview.widget.CardView

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:layout\_above="@id/recView"

android:elevation="5dp"

android:isScrollContainer="true"

app:cardCornerRadius="15dp"

app:cardUseCompatPadding="true">

<TextView

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:fontFamily="@font/secular\_one"

android:paddingLeft="10dp"

android:paddingTop="5dp"

android:layout\_gravity="center"

android:text="Students Info:"

android:textAlignment="gravity"

android:textColor="#C388FA"

android:textSize="15sp" />

</androidx.cardview.widget.CardView>

<androidx.recyclerview.widget.RecyclerView

android:id="@+id/recView"

android:layout\_width="match\_parent"

android:layout\_height="650dp"

android:layout\_above="@id/fragment\_container"

android:layout\_marginStart="1dp"

android:layout\_marginEnd="1dp"

android:layout\_marginBottom="1dp" />

<FrameLayout

android:id="@+id/fragment\_container"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_above="@id/bottom\_navigation"

/>

<com.google.android.material.bottomnavigation.BottomNavigationView

android:id="@+id/bottom\_navigation"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"  
 android:layout\_alignParentBottom="true"

app:itemRippleColor="#7767CD"

app:labelVisibilityMode="selected"

android:outlineSpotShadowColor="@color/black"

app:layout\_behavior="@string/hide\_bottom\_view\_on\_scroll\_behavior"

app:layout\_constraintBottom\_toBottomOf="parent"

app:layout\_constraintEnd\_toEndOf="parent"

app:layout\_constraintStart\_toStartOf="parent"

app:menu="@menu/nav\_items">

</com.google.android.material.bottomnavigation.BottomNavigationView>

</RelativeLayout>

**RecyclerView Adpater files used on this screen:**

**myAdapter.java:**

package com.example.recyclerviewdemoproj;

import android.annotation.SuppressLint;

import android.app.ActionBar;

import android.app.AlertDialog;

import android.content.DialogInterface;

import android.content.Intent;

import android.os.Bundle;

import android.view.LayoutInflater;  
import android.view.View;

import android.view.ViewGroup;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import androidx.annotation.NonNull;

import androidx.recyclerview.widget.RecyclerView;

import com.bumptech.glide.Glide;  
import com.firebase.ui.database.FirebaseRecyclerAdapter;  
import com.firebase.ui.database.FirebaseRecyclerOptions;  
import com.google.android.gms.tasks.OnFailureListener;  
import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.database.FirebaseDatabase;

import com.orhanobut.dialogplus.DialogPlus;

import com.orhanobut.dialogplus.ViewHolder;

import java.util.HashMap;

import java.util.Map;

import de.hdodenhof.circleimageview.CircleImageView;

public class myAdapter extends FirebaseRecyclerAdapter<model,myAdapter.myViewHolder>{

public myAdapter(@NonNull FirebaseRecyclerOptions<model> options) {

super(options);

}

@SuppressLint("SetTextI18n")

@Override

protected void onBindViewHolder(@NonNull myViewHolder holder, @SuppressLint("RecyclerView") final int position, @NonNull model model) {

holder.cName.setText(model.getChildName());

holder.cAge.setText("Age: "+model.getChildAge());

holder.cPhone.setText(model.getPhone1());

Glide.with(holder.pImg.getContext()).load(model.getChildPhoto()).into(holder.pImg);

holder.editICON.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

final DialogPlus dialogPlus = DialogPlus.newDialog(holder.pImg.getContext())

.setContentHolder(new ViewHolder(R.layout.dialogcontent))

.setExpanded(true, 1480 )

.create();

View myView = dialogPlus.getHolderView();

final EditText url = myView.findViewById(R.id.profileURL);

final EditText name = myView.findViewById(R.id.cName);

final EditText age = myView.findViewById(R.id.cAge);

final EditText parent = myView.findViewById(R.id.pName);

final EditText address = myView.findViewById(R.id.pAddress);

final EditText p1 = myView.findViewById(R.id.phone1);

final EditText p2 = myView.findViewById(R.id.phone2);

Button update = myView.findViewById(R.id.update);

url.setText(model.getChildPhoto());

name.setText(model.getChildName());

age.setText(model.getChildAge());

parent.setText(model.getParentName());

address.setText(model.getpAddress());

p1.setText(model.getPhone1());

p2.setText(model.getPhone2());

dialogPlus.show();

update.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Map<String,Object> map = new HashMap<>();

map.put("childPhoto",url.getText().toString());

map.put("childName",name.getText().toString());

map.put("childAge",age.getText().toString());

map.put("parentName",parent.getText().toString());

map.put("pAddress",address.getText().toString());

map.put("phone1",p1.getText().toString());

map.put("phone2",p2.getText().toString());

FirebaseDatabase.getInstance().getReference().child("Users")

.child(getRef(position).getKey()).updateChildren(map)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void unused) {

dialogPlus.dismiss();

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

dialogPlus.dismiss();

}

});

}

});

}

});

holder.deleteICON.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

AlertDialog.Builder builder = new AlertDialog.Builder(holder.pImg.getContext());

builder.setTitle("Delete Child");

builder.setMessage("Remove child data?");

builder.setPositiveButton("Yes", new DialogInterface.OnClickListener() {

@Override

public void onClick(DialogInterface dialogInterface, int i) {

FirebaseDatabase.getInstance().getReference().child("Users")

.child(getRef(position).getKey()).removeValue();

}

});

builder.setNegativeButton("Cancel", new DialogInterface.OnClickListener() {  
 @Override  
 public void onClick(DialogInterface dialogInterface, int i) {

}

});

builder.show();

}

});

holder.card.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Intent i = new Intent(holder.pImg.getContext(), Data.class);

Bundle bundle = new Bundle();

bundle.putString("cName", model.getChildName());  
 bundle.putString("cURL", model.getChildPhoto());  
 bundle.putString("pname", model.getParentName());  
 bundle.putString("phone1", model.getPhone1());  
 bundle.putString("phone2", model.getPhone2());  
 i.putExtras(bundle);  
 view.getContext().startActivity(i);  
 }

});

}

@NonNull

@Override

public myViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {

View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.singlerow,parent,false);

return new myViewHolder(view);

}

class myViewHolder extends RecyclerView.ViewHolder

{

RelativeLayout card;

CircleImageView pImg;

TextView cName, cAge, cPhone;

ImageView editICON, deleteICON;

public myViewHolder(@NonNull View itemView) {

super(itemView );

pImg = itemView.findViewById(R.id.pURL);

cName = itemView.findViewById(R.id.childName);

cAge = itemView.findViewById(R.id.childAge);

cPhone = itemView.findViewById(R.id.childPhone);

editICON = itemView.findViewById(R.id.editICON);

deleteICON = itemView.findViewById(R.id.deleteICON);

card = itemView.findViewById(R.id.card);

}

}

}

**singlerow.xml:**

<?xml version="1.0" encoding="utf-8"?>

<androidx.cardview.widget.CardView xmlns:android="http://schemas.android.com/apk/res/android"

xmlns:app="http://schemas.android.com/apk/res-auto"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

app:cardBackgroundColor="#8FD4F4"

android:foreground="?android:attr/selectableItemBackground"

android:clickable="true"

app:cardElevation="8dp"

app:cardCornerRadius="15dp"

app:cardUseCompatPadding="true"

>

<RelativeLayout

android:id="@+id/card"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:padding="15dp">

<de.hdodenhof.circleimageview.CircleImageView

android:id="@+id/pURL"

android:layout\_width="90dp"

android:layout\_height="90dp"

app:civ\_border\_color="#000000"

app:civ\_border\_width="2dp"

app:srcCompat="@drawable/ic\_launcher\_foreground" />

<TextView

android:id="@+id/childName"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="5dp"

android:layout\_toRightOf="@+id/pURL"

android:fontFamily="@font/secular\_one"

android:text="TextView"

android:textColor="#000"

android:textSize="20sp"

/>

<TextView

android:id="@+id/childAge"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/childName"

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="5dp"

android:layout\_toRightOf="@+id/pURL"

android:fontFamily="@font/secular\_one"

android:text="TextView"

android:textColor="#000"

android:textSize="18dp" />

<TextView

android:id="@+id/childPhone"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@+id/childAge"

android:layout\_marginLeft="10dp"

android:layout\_marginBottom="5dp"

android:layout\_toRightOf="@+id/pURL"

android:fontFamily="@font/secular\_one"

android:text="TextView"

android:textColor="#000"

android:textSize="18dp" />

<LinearLayout

android:layout\_width="378dp"

android:layout\_height="50dp"

android:layout\_below="@id/childPhone"

android:layout\_alignParentEnd="true"

android:layout\_gravity="center"

android:layout\_marginTop="10dp"

android:layout\_marginEnd="0dp"

android:orientation="horizontal">

<ImageView

android:id="@+id/editICON"

android:layout\_width="180dp"

android:layout\_height="match\_parent"

android:layout\_gravity="center"

app:srcCompat="@drawable/update" />

<ImageView

android:id="@+id/deleteICON"

android:layout\_width="180dp"

android:layout\_height="match\_parent"

android:layout\_gravity="center"

app:srcCompat="@drawable/delete" />

</LinearLayout>

</RelativeLayout>

</androidx.cardview.widget.CardView>

**Data.java:**

package com.example.recyclerviewdemoproj;

import androidx.annotation.NonNull;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import android.Manifest;

import android.app.ProgressDialog;

import android.content.Intent;

import android.net.Uri;

import android.os.Bundle;

import android.view.View;

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.webkit.MimeTypeMap;

import android.widget.MediaController;

import android.widget.Toast;

import android.widget.VideoView;

import com.bumptech.glide.Glide;

import com.example.recyclerviewdemoproj.databinding.ActivityDataBinding;

import com.google.android.gms.tasks.OnFailureListener;

import com.google.android.gms.tasks.OnSuccessListener;

import com.google.firebase.database.DatabaseReference;

import com.google.firebase.database.FirebaseDatabase;

import com.google.firebase.storage.FirebaseStorage;

import com.google.firebase.storage.OnProgressListener;

import com.google.firebase.storage.StorageReference;

import com.google.firebase.storage.UploadTask;

import com.karumi.dexter.Dexter;

import com.karumi.dexter.PermissionToken;

import com.karumi.dexter.listener.PermissionDeniedResponse;

import com.karumi.dexter.listener.PermissionGrantedResponse;

import com.karumi.dexter.listener.PermissionRequest;

import com.karumi.dexter.listener.single.PermissionListener;

import java.text.SimpleDateFormat;

import java.util.Date;

import java.util.Locale;

import de.hdodenhof.circleimageview.CircleImageView;

public class Data extends AppCompatActivity {

private ActivityDataBinding activityDataBinding;

Animation cardsanim;

Uri videoURI;

MediaController mediaController;

VideoView videoView;

StorageReference storageReference;

DatabaseReference databaseReference;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

activityDataBinding = ActivityDataBinding.inflate(getLayoutInflater());

setContentView(activityDataBinding.getRoot());

getSupportActionBar().setTitle("Child Data");

getSupportActionBar().setDisplayHomeAsUpEnabled(true);

cardsanim = AnimationUtils.loadAnimation(this, R.anim.cardsanim);

activityDataBinding.cards.animate().translationY(50).setDuration(500).alpha(1.0f);

activityDataBinding.cards.setAnimation(cardsanim);

videoView = findViewById(R.id.videoView);

mediaController = new MediaController(this);

activityDataBinding.videoView.setMediaController(mediaController);

activityDataBinding.videoView.start();

String date = new SimpleDateFormat("yyyy-MM-dd", Locale.getDefault()).format(new Date());

storageReference = FirebaseStorage.getInstance().getReference();

databaseReference = FirebaseDatabase.getInstance().getReference("myVideos");

Bundle bundle = getIntent().getExtras();

activityDataBinding.cname.setText(bundle.getString("cName"));

activityDataBinding.pname.setText("Parent: "+bundle.getString("pname"));

CircleImageView cm = findViewById(R.id.pURL);

Glide.with(getApplicationContext()).load(bundle.getString("cURL")).into(activityDataBinding.pURL);

activityDataBinding.phone1.setText(bundle.getString("phone1"));

activityDataBinding.phone2.setText(bundle.getString("phone2"));

activityDataBinding.call1.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Uri u = Uri.parse("tel:"+ bundle.getString("phone1"));

Intent i = new Intent(Intent.ACTION\_DIAL, u);

try {

startActivity(i);

}

catch (SecurityException s){

Toast.makeText(Data.this, "An error occurred", Toast.LENGTH\_SHORT).show();

}

}

})

activityDataBinding.call2.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Uri u = Uri.parse("tel:"+ bundle.getString("phone2"));

Intent i = new Intent(Intent.ACTION\_DIAL, u);

try {

startActivity(i);

}

catch (SecurityException s){

Toast.makeText(Data.this, "An error occurred", Toast.LENGTH\_SHORT).show();

}

}

});

activityDataBinding.browseICON.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view)

Dexter.withContext(getApplicationContext())

.withPermission(Manifest.permission.READ\_EXTERNAL\_STORAGE)

.withListener(new PermissionListener() {

@Override

public void onPermissionGranted(PermissionGrantedResponse permissionGrantedResponse) {

Intent intent = new Intent();

intent.setType("video/\*");

intent.setAction(Intent.ACTION\_PICK);

activityDataBinding.cards.setVisibility(View.INVISIBLE);

activityDataBinding.videoCard.animate().translationY(-100).setDuration(500).alpha(1.0f);

activityDataBinding.videoCard.setAnimation(cardsanim);

activityDataBinding.videoCard.setVisibility(View.VISIBLE);

activityDataBinding.titleTXT.setText(bundle.getString("cName"));

activityDataBinding.dateTXT.setText(date);

startActivityForResult(Intent.createChooser(intent,"Select Video!"),101);

}

@Override

public void onPermissionDenied(PermissionDeniedResponse permissionDeniedResponse) {

}

@Override

public void onPermissionRationaleShouldBeShown(PermissionRequest permissionRequest, PermissionToken permissionToken) {

permissionToken.continuePermissionRequest();

}

}).check();

}

});

activityDataBinding.uploadBTN.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

processVideoUploading();

}

});

activityDataBinding.backBTN.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

activityDataBinding.videoCard.setVisibility(View.INVISIBLE);

activityDataBinding.cards.setVisibility(View.VISIBLE);

}

});

}

public String getExtension(){

MimeTypeMap mimeTypeMap = MimeTypeMap.getSingleton();

return mimeTypeMap.getExtensionFromMimeType(getContentResolver().getType(videoURI));

}

private void processVideoUploading() {

final ProgressDialog pd = new ProgressDialog(this);

pd.setTitle("Media Uplaoder");

pd.show();

final StorageReference uploder = storageReference.child("myVideos/"+System.currentTimeMillis()+","+getExtension());

uploder.putFile(videoURI)

.addOnSuccessListener(new OnSuccessListener<UploadTask.TaskSnapshot>() {

@Override

public void onSuccess(UploadTask.TaskSnapshot taskSnapshot) {

uploder.getDownloadUrl().addOnSuccessListener(new OnSuccessListener<Uri>() {

@Override

public void onSuccess(Uri uri) {

fileModel obj = new fileModel(activityDataBinding.titleTXT.getText().toString(),activityDataBinding.dateTXT.getText().toString(), videoURI.toString());

databaseReference.child(databaseReference.push().getKey()).setValue(obj)

.addOnSuccessListener(new OnSuccessListener<Void>() {

@Override

public void onSuccess(Void unused) {

pd.dismiss();

Toast.makeText(Data.this, "Succesfully Uploaded!", Toast.LENGTH\_SHORT).show();

startActivity(new Intent(getApplicationContext(), HomeRecycle.class));

finish();

}

}).addOnFailureListener(new OnFailureListener() {

@Override

public void onFailure(@NonNull Exception e) {

pd.dismiss();

Toast.makeText(Data.this, "Uploading Failed", Toast.LENGTH\_SHORT).show();

}

});

}

});

}

})

.addOnProgressListener(new OnProgressListener<UploadTask.TaskSnapshot>() {

@Override

public void onProgress(@NonNull UploadTask.TaskSnapshot snapshot) {

float percent = (100\*snapshot.getBytesTransferred()/snapshot.getTotalByteCount());

pd.setMessage("Uplaoded: "+(int)percent+"%");

}

});

}

@Override

public boolean onSupportNavigateUp() {

finish();

return super.onSupportNavigateUp();

}  
 @Override

protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {

super.onActivityResult(requestCode, resultCode, data);

if(requestCode==101 && resultCode==RESULT\_OK)

{

videoURI=data.getData();

videoView.setVideoURI(videoURI);

}

}

}

**Profile.java:**

package com.example.recyclerviewdemoproj;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.MenuItem;

import android.widget.Toast;

import com.bumptech.glide.Glide;  
import com.example.recyclerviewdemoproj.databinding.ActivityProfileBinding;  
import com.google.android.material.bottomnavigation.BottomNavigationView;  
import com.google.firebase.database.DataSnapshot;  
import com.google.firebase.database.DatabaseError;  
import com.google.firebase.database.DatabaseReference;  
import com.google.firebase.database.FirebaseDatabase;  
import com.google.firebase.database.ValueEventListener;

public class Profile extends AppCompatActivity {

public static final String TAG = "1";

DatabaseReference databaseReference;

FirebaseDatabase firebaseDatabase;

public static final String ADMINMODEL = "adminModel";

ActivityProfileBinding profileBinding;

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

profileBinding = ActivityProfileBinding.inflate(getLayoutInflater());

setContentView(profileBinding.getRoot());

getSupportActionBar().hide();

Bundle bundle = getIntent().getExtras();

firebaseDatabase = FirebaseDatabase.getInstance();

databaseReference = FirebaseDatabase.getInstance().getReference(ADMINMODEL);

databaseReference.addValueEventListener(new ValueEventListener() {

@Override

public void onDataChange(@NonNull DataSnapshot snapshot) {

for (DataSnapshot ds : snapshot.getChildren()){

if (ds.child("dcEmail").getValue().equals("brightChildDaycare@gmail.com")){ //When users login it will capture the email and retrieve its data

Glide.with(profileBinding.dcName.getContext()).load(ds.child("dcID").getValue(String.class)).into(profileBinding.dcProfile);

profileBinding.dcName.setText(ds.child("dcName").getValue(String.class));

profileBinding.dcAdminName.setText(ds.child("dcAdminName").getValue(String.class));

profileBinding.dcAdminEmail.setText(ds.child("dcEmail").getValue(String.class)); profileBinding.dcNewPass.setText(ds.child("dcNewPass").getValue(String.class));  
profileBinding.dcConfPass.setText(ds.child("dcConfPass").getValue(String.class));  
profileBinding.dcAddress.setText(ds.child("dcAddress").getValue(String.class));  
profileBinding.dcPhone1.setText(ds.child("dcP1").getValue(String.class));  
profileBinding.dcPhone2.setText(ds.child("dcP2").getValue(String.class));  
profileBinding.dcOccupancy.setText(ds.child("dcOcc").getValue(String.class));  
profileBinding.daycareName.setText(ds.child("dcName").getValue(String.class));  
profileBinding.daycareEmail.setText(ds.child("dcEmail").getValue(String.class));

}

}

}

@Override

public void onCancelled(@NonNull DatabaseError error) {

Toast.makeText(Profile.this, "Unable to retrieve data", Toast.LENGTH\_SHORT).show();

}

});

BottomNavigationView bottomNavigationView = findViewById(R.id.bottom\_navigation);

bottomNavigationView.setSelectedItemId(R.id.nav\_profile);

bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

switch(item.getItemId())

{

case R.id.nav\_home:

startActivity(new Intent(getApplicationContext(),HomeRecycle.class));

overridePendingTransition(0,0);

return true;

case R.id.nav\_profile:

return true;

case R.id.nav\_subscribe:

startActivity(new Intent(getApplicationContext(),Subscription.class));

overridePendingTransition(0,0);

return true;

}

return false;

}

});

}

}

**Subscription.java:**

package com.example.recyclerviewdemoproj;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.MenuItem;

import com.google.android.material.bottomnavigation.BottomNavigationView;

public class Subscription extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_subscription);

BottomNavigationView bottomNavigationView = findViewById(R.id.bottom\_navigation);

bottomNavigationView.setSelectedItemId(R.id.nav\_subscribe);

bottomNavigationView.setOnNavigationItemSelectedListener(new BottomNavigationView.OnNavigationItemSelectedListener() {

@Override

public boolean onNavigationItemSelected(@NonNull MenuItem item) {

switch(item.getItemId())

{

case R.id.nav\_home:

startActivity(new Intent(getApplicationContext(),HomeRecycle.class));

overridePendingTransition(0,0);

return true;

case R.id.nav\_profile:

startActivity(new Intent(getApplicationContext(),Profile.class));

overridePendingTransition(0,0);

return true;

case R.id.nav\_subscribe:

return true;

}

return false;

}

});

}

}

**6.3 SYSTEM TESTING**

**System Testing** is a type of software testing that is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. In system testing, integration testing passed components are taken as input. The goal of integration testing is to detect any irregularity between the units that are integrated together. System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested. System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both. System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS). System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing. We uses the black-box testing technique to test our project’s vulnerability, in which we tested different units of our project. Some testing processes are listed below:

**Test Case 1**

**Description**

In this Test Case. We will check whether the validation of the sign-up page is working properly or not.

**Input**

The user will give their username, first name, last name, date of birth, email, and password as an input to register in our app.

**Expected Output**

If the user enters the correct input and clicks on the Register button, then the app will send a verification email to users for authentication purposes.

**Actual Output**

After entering the correct input, users successfully logins into the App

**Result**

Testing is conducted successfully, users successfully logins into the App and the user is granted permission to access the app and the main page of our app appears to the user.

**Test Case 2**

**Description**

In this Test Case, we will check whether the app is working properly by Sending the payment due date notification to the parent.

**Input**

The user will click on the notification and the user will redirected to the payment page.

**Expected Output**

The referred payment app should open properly when the user clicks on the due date notification which is notified by the Daycare app

**Actual Output**

While testing, the payment build desk was not opening with the specified payment value when the user clicks on the notification.

**Result**

When the error/ bud was recognized by the developers of our team they start to work on the error and find out the solution for that error. That error was timeout() which is known for the timeout of the screen. So for this error, they made changes in the code by disabling the timeout(). Through these changes, we get the expected output and the payment build desk was opening with the specified payment value when the user clicks on the notification.

So we resolved this error with the help of the black-box testing technique

**6.4 APPLICATION**

Our project is developed for users only.Application of our project are:

* With the use of our app, Parents will be able to connect to the Owner of Child Care.
* Our app will provide Video Uploading Feature, Payment Reminders, and easy access to chat.
* Owners will be able to focus on child rather than paper work.
* Attendance tracking will be done in the app.

**CHAPTER 7**

**CONCLUSION & FUTURE SCOPE**

**7.1 CONCLUSION & FUTURE SCOPE**

**Conclusion**

The project was made possible because of our team. Our App helps to ensure the promise of all-around development of the child and their safety. Our admins will help parents discover the potential of their small ones. This app will make them enable them to discover their habits and make necessary changes. We ensure that the safety of their will is our responsibility.

**Future Scope**

* We will implement the QR and Face recognition system for attendance in the future which will automate the manual attendance data work.
* We will implement a Daycare selection form for parents to help them find suitable Daycare for their kids sorting with the reviews of Daycare, staff review, and fee estimation.
* We will implement an E-learning model for kids, containing the material like Rhyme videos, Vocabulary learning, etc provided by the respective Daycare.
* We will enhance the UI design of the apps to make them more interactive and user-friendly.

**7.2 REFERENCE AND BIBLIOGRAPHY**

**Links for Research Papers:**

* <https://www.computer.org/csdl/proceedings-article/mobilesoft/2017/07972728/12OmNBt3qlO>
* <https://www.computer.org/csdl/proceedings-article/hicss/2001/00926541/12OmNx6g6eu>
* <https://www.computer.org/csdl/journal/tk/2002/01/k0013/13rRUygBw7o>
* <https://www.computer.org/csdl/proceedings-article/mobilesoft/2016/07832957/12OmNvTBB3d>
* <https://www.computer.org/csdl/proceedings-article/sccc/2013/0426a030/12OmNBBzol7>
* <https://www.computer.org/csdl/proceedings-article/percomw/2016/07457140/12OmNAle6ut>
* <https://www.computer.org/csdl/proceedings-article/snpd/2009/3642a181/12OmNxdVgTC>
* <https://www.computer.org/csdl/proceedings-article/iccis/2012/4789b009/12OmNzxyizZ>
* <https://www.computer.org/csdl/proceedings-article/ficloud/2014/4357a420/12OmNvFHfJF>
* <https://www.computer.org/csdl/proceedings-article/hpcc-icess/2012/4749b421/12OmNvHoQop>
* <https://www.computer.org/csdl/proceedings-article/lcnw/2012/06424068/12OmNzw8iZn>
* <https://timesofindia.indiatimes.com/city/mumbai/kharghar-creche-fallout-top-cop-cracks-down-on-child-day-care-centres/articleshow/55642635.cms>
* <https://books.google.co.in/books?hl=en&lr=&id=G0BzKsp-b6wC&oi=fnd&pg=PA1&dq=daycare+on+child+development&ots=_aKnk1yz9E&sig=tVprtc2INauo7Wd3tQE8KLonTrU&redir_esc=y#v=onepage&q=daycare%20on%20child%20development&f=false>
* <https://www.thehindu.com/news/cities/mumbai/Policy-for-daycare-centres-soon-govt./article16786712.ece>
* <https://www.buttonlawfirm.com/library/what-parents-need-to-know-about-daycare-worker-crimes.cfm>
* <https://www.tandfonline.com/doi/full/10.1080/19439342.2011.639457>
* <https://www.sciencedirect.com/science/article/abs/pii/S0167268120302857>
* <http://pepsic.bvsalud.org/scielo.php?pid=S0104-12822015000200006&script=sci_abstract&tlng=en>
* <https://www.engpaper.com/mobile-app-2019-2.htm>
* <https://ieeexplore.ieee.org/document/8896249>
* <https://ieeexplore.ieee.org/document/8751262>
* <https://ieeexplore.ieee.org/document/9436696>

**Links for Tutorials:**

* <https://youtu.be/SV9pJqR41KI>
* <https://youtu.be/PJ3RdfJ4Np8>
* <https://youtu.be/mXjZQX3UzOs>

**Books Referred:**

* The Definitive Guide to Firebase by Laurence Moroney
* Mobile Application Development by Nirali Prakashan