

Oluwaseye Omowa

**DEVELOPMENT OF AN E-SERVICE APP ON THE ANDROID
PLATFORM (FRONT-END)**

**DEVELOPMENT OF AN E-SERVICE APP ON THE ANDROID
PLATFORM (FRONT-END)**

Oluwaseye Omowa
Bachelor's Thesis
Autumn 2016
Information Technology
Oulu University of Applied Sciences

ABSTRACT

Oulu University of Applied Sciences
Degree Programme in Information Technology

Author: Oluwaseye Omowa

Title of the bachelor's thesis: Development of an E-service App on the Android Platform (Front-end)

Supervisor: Väisänen Veijo

Term and year of completion: Autumn 2016 Number of pages: 45

The aim of this Bachelor's thesis was to make it as easy as possible for parents and guardians to get the services of caregivers or a nanny for their children. My aim was to reduce the problems of time and availability in seeking the service of a nanny.

The idea of this thesis was formed by me and my thesis project partner due to our personal experiences in Nigeria. As graduating students in the field of information technology, we saw the need to solve this problem with our acquired skills.

The BabyShift application was developed on the Android Studio version 2.2.0.12 with Android API 21 using Java programming language and SQLite for the database. The application was created with two different user interfaces, one for parents and the other for nannies. Nannies will be able to register, sign-in and make business profiles regarding to their services and also parents will be able to register, sign-in and search for a suitable nanny who meets their requirements using a location and a username as search criteria.

The marketing strategy was based on the Nigerian market and it was carried out in our previous study at Oulu University of Applied Sciences.

Finally, the development of the BabyShift application came to realisation, the application ran successfully on all Android mobile phones. However, the payment module for the application will be incorporated as the BabyShift application is open for a further development.

Keywords: Android, Java Programming Language, SQLite

PREFACE

This thesis project was carried out at Oulu University of Applied Sciences and the idea was formed by me and my project partner.

During my study at Oulu University of Applied Sciences, I was able to develop a mobile Android application that archives information on how to repair and fix various bicycle parts. For this reason I developed interest in Android development and the knowledge I acquired actually helped in carrying out my thesis project.

I would like to thank my supervisor Veijo Väisänen and my thesis project partner Alli Ibrahim Adesanya for his support and mentorship. They were helpful throughout my thesis project.

Oulu, 20.8.2016

Oluwaseye Omowa

CONTENTS

ABSTRACT	3
PREFACE	4
TABLE OF CONTENTS	5
1 INTRODUCTION	7
2 PRODUCT FUNCTIONS	9
2.1 High level architecture	9
2.1.1 Software architecture	11
2.1.2 Logical view	12
3 FUNCTIONAL AREA DECOMPOSITION OF BABYSHIFT APP	13
3.1 System features and functional requirements	14
Expanded use case: Register with BabyShift	14
Expanded Use Case: Login and reset a password	16
Expanded Use Case: Search nanny services	18
Expanded Use Case: Call or send message to a nanny	20
Expanded use case: Rate a nanny	23
Expanded Use Case: Make a profile on the system	25
Expanded Use Case: Edit a profile and upload a photo	27
4 USER INTERFACES	30
Installation permission	30
Login	31
Resetting a password	32
Parent registration	34
Making a nanny profile	34
Searching nanny services	35
Call or message nanny	36
Rating a nanny	37
Uploading a photo	39
Logging out	40
4.1 Incorporation with the knowledge of HCI principles	40
4.2 Themes	40

4.3 Non-functional requirements	41
5 TESTING	43
6 CONCLUSION	44
7 REFERENCES	45

1 INTRODUCTION

In the world we live in today, there is hardly no one without a smart phone. Users might acquire a phone based on the existing platforms or operating systems, and the most popular ones are Android, IOS and Windows. In the course of planning the marketing strategy of the BabyShift app, it was considered that not everyone, who can use smart phones, are computer literate. Also, users get things done quickly by using apps on their smart phones. For this reason, the Android mobile platform was targeted as it is widely acquired by smart phone users.

As finding a nanny on a short notice could be very challenging, the BabyShift app has been developed to help parents find capable nannies for their children. It is a simple application that connects both parents and nannies.

How the BabyShift app works

The app would work in the following way for a normal user:

1. The normal user downloads the application on the Google Play store. This application is available free of charge.
2. After the app has been installed, the user will be asked to grant the access rights to a phone call application and a message application. The application also states that a phone call and a message may charge the user some money.
3. When the user grants the rights to the app, it is installed on the Android phone.
4. Afterwards, the application requires a registration. It provides two options for the registration within the app, i.e. either as a nanny or as a parent.
5. After the successful registration as a particular type of a user, a user can enjoy the full functionality of the application.

6. A nanny can make a profile and update his/her business page by editing a profile. Moreover, nannies are rated on the basis of their services.
7. Parents can register to find nannies through a search and they can place a call to the nanny with the help of the application. Furthermore parents can rate the nannies after hiring them according to the quality of the services they provided.

2 PRODUCT FUNCTIONS

BabyShift is a user friendly system that provides a time efficient search for services of nannies and a platform for nannies to offer services. Moreover, it provides a platform for parents as well to search for the services of nannies.

Major functions for an easy market are registration for parents and nannies, profiling of nannies, searching nanny services, calling and sending a message to a nanny and rating a nanny.

2.1 High level architecture

The high level architecture design will help to understand the following elements of this system:

- Understanding the requirements
- Architectural patterns
- Components and their interfaces
- Interactions between components
- Data model of the components and interfaces

This system uses two kinds of interfaces:

- Parent interface
- Nanny interface

Figure 1 describes the architecture of this system.

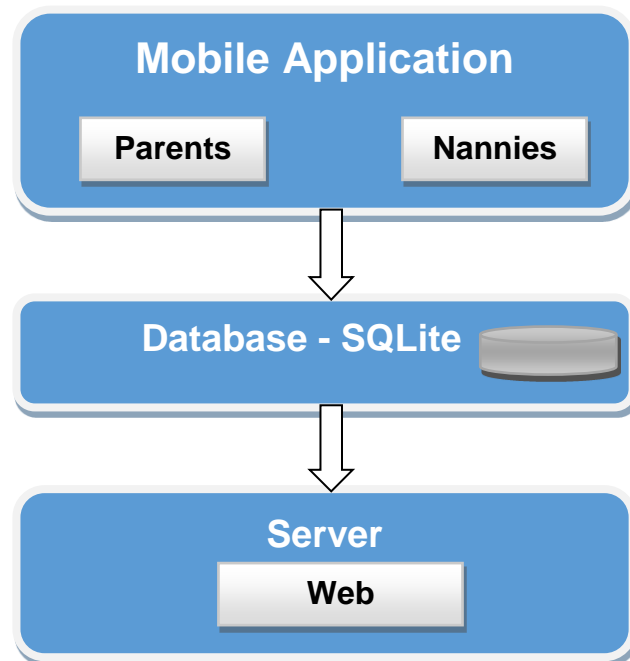


FIGURE 1. High Level Architecture of BabyShift App

2.1.1 Software architecture

The basic software architecture of this proposed system provides a better understanding. It follows a simple architecture, which uses a local database known as SQLite and it uses the internet to install the application from a webserver. (See figure 2).

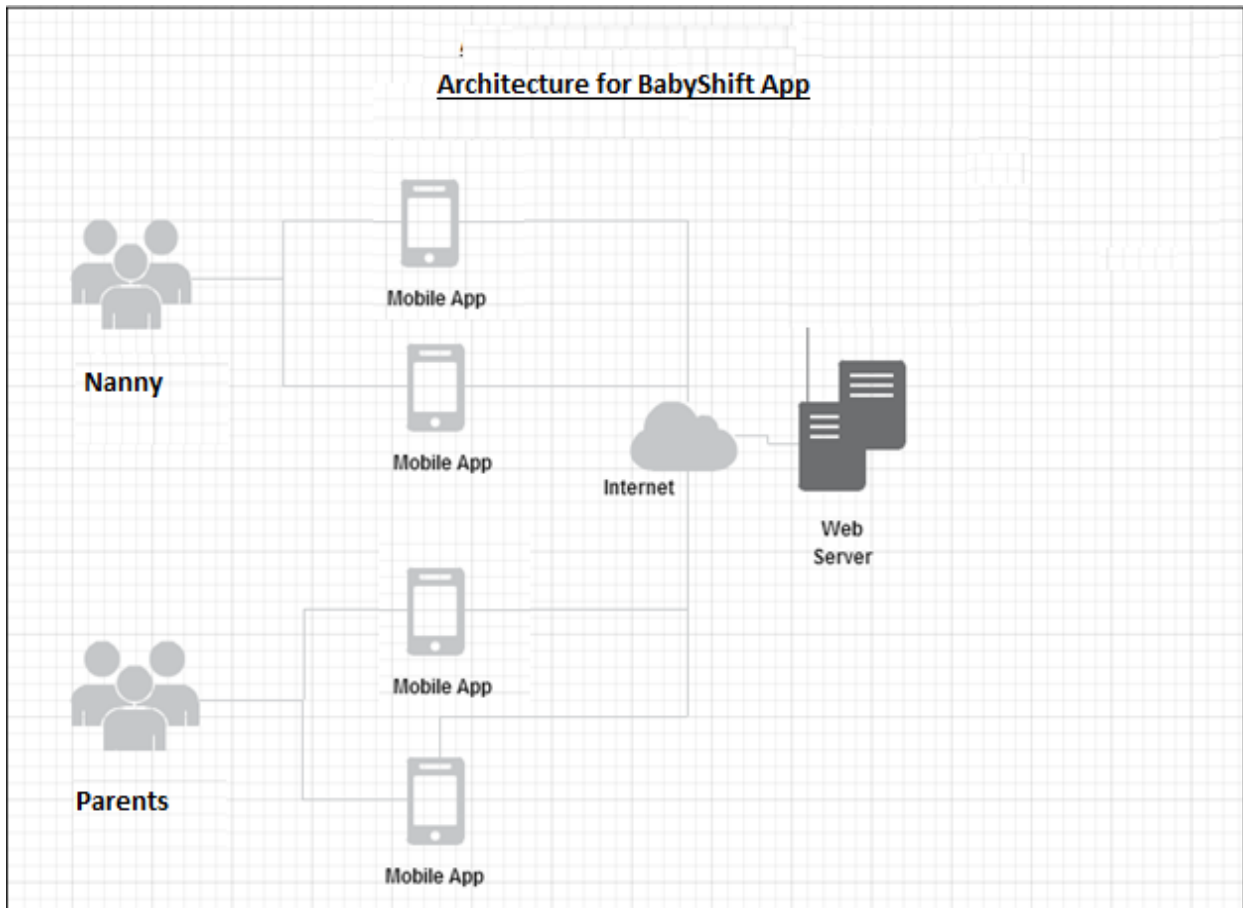


FIGURE 2. Software architecture for App

2.1.2 Logical view

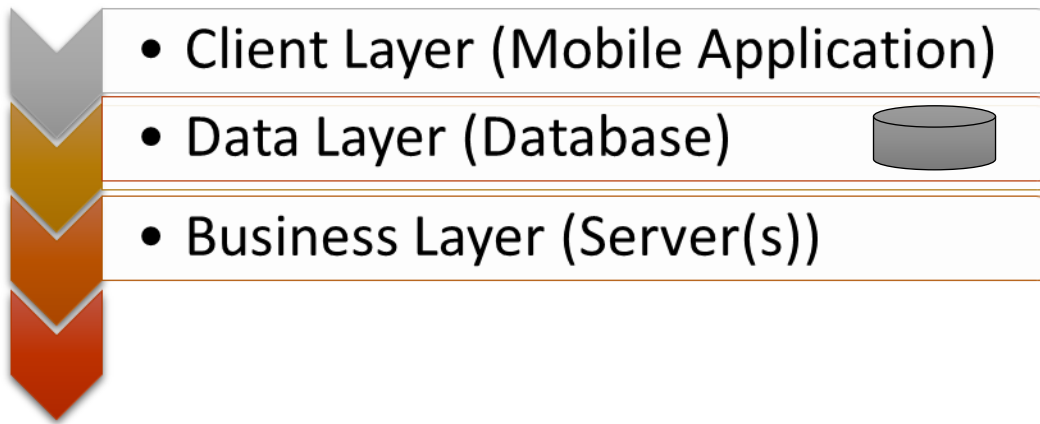


FIGURE 3. Logical View

The BabyShift app consists of a mobile based application which uses the Internet connectivity of the phone to communicate with the web server in order to install and get the app. The database (SQLite) of this application is local. It means that no Internet connectivity is required to get the application data. The Internet is only needed to get the application started. (See figure 3).

3 FUNCTIONAL AREA DECOMPOSITION OF BABYSHIFT APP

Functional Area decomposition refers to the use cases or the functions that a user can perform on the application as shown in figure 4 below.

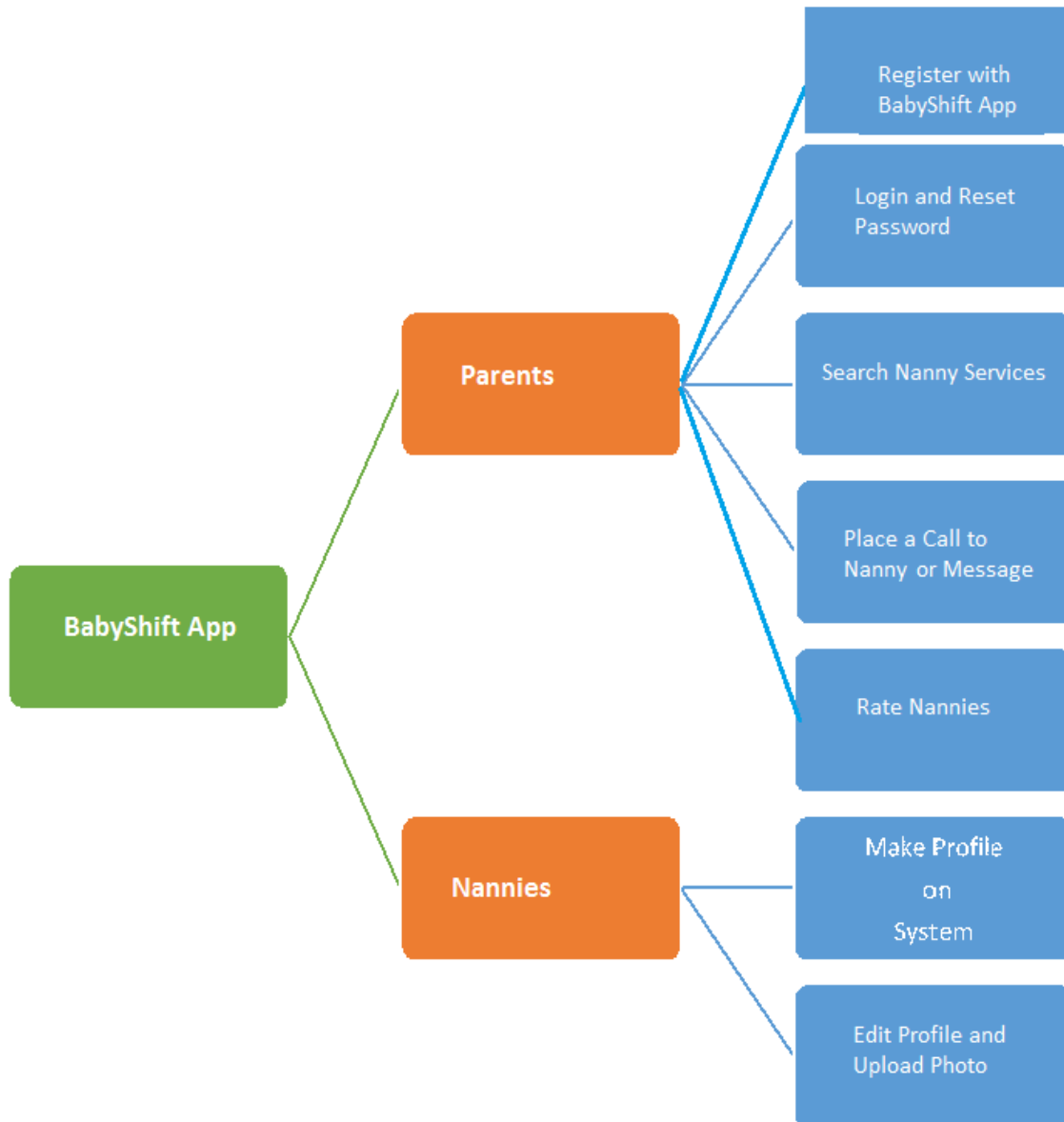


FIGURE 4. Use Cases of BabyShift App

3.1 System features and functional requirements

Expanded use case: Register with BabyShift

The primary actor for this use case is a parent.

Stakeholders and interests:

- **Parents:** If a parent wants to avail the services and view the nannies, then they have to register with the system.
- **Nanny:** A nanny would provide services to the parents who have registered with the system.

Preconditions:

- A parent must install the app to register with BabyShift.

Success guarantee (or post conditions):

- Services are offered by the nannies.
- Parents can rate the nannies.

TABLE 1. Scenario for a use case register with BabyShift.

MAIN SUCCESS SCENARIO	
Actor Action	System Action
1). Parents select the option to register as a parent on the mobile app of BabyShift.	2). The system shows a form to be filled in order to register with the system.

<p>3). A parent fills in the form and provides all the necessary details, which include username and password.</p> <p>5). Parents can see the services provided by nannies.</p>	<p>4). The system registers and authenticates the parent.</p>
---	---

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on a text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock

TABLE 2. Error triggers and messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No Internet connection. Please try again.
3a. Input fields not filled.	Please fill in all details.

Expanded Use Case: Login and reset a password

The primary actor for this use case is a parent and a nanny.

Stakeholders and interests:

- **Parents:** If a parent wants to use the services and view the nannies, then they have to login the system.
- **Nanny:** A nanny would provide services to the parents who have registered within the system.

Preconditions:

- A parent and nanny must register within the application to log in and reset a password.
- Both must enter the correct user names to reset the password.

Success guarantee (or post conditions):

- Services are offered by the nannies
- Parents can view the nannies.

TABLE 3. Scenario for a use case log in and reset a password.

MAIN SUCCESS SCENARIO	
Actor Action	System Action
<p>1). A parent enters a username and password.</p> <p>3). Parents can now view and rate the nannies.</p> <p>5). A parent can reset the password by selecting a reset password.</p> <p>7). A user enters the details and selects to reset a password.</p>	<p>2). The system authenticates the parent and allows the parent to login.</p> <p>4). If an authentication fails, the system displays an error message.</p> <p>6). The system asks for a new password and confirms the password along with the username.</p> <p>8). The system updates the details into the backend database.</p>

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on a text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock

TABLE 4. Error Triggers and Messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No Internet connection. Please try again.
1a. Input fields not filled.	Please fill in all details.
2a. A wrong username password.	Login failed.
7a. A user enters the wrong details.	Display an error message.

Expanded Use Case: Search nanny services

The primary actor for this use case is a parent.

Stakeholders and interests:

- **Parents:** Enables the parents search the nannies by the location and name.

- **Nanny:** A nanny would provide services to the parents who have registered with the system.

Preconditions:

- A parent must log in the system to search for nannies.

Success guarantee (or post conditions):

- Services are offered by the nannies.
- Parents can rate the nannies.

TABLE 5. Scenario for a use case search nanny services.

MAIN SUCCESS SCENARIO	
Actor Action	System Action
<p>1). Parents log in the BabyShift app.</p> <p>3). A parent can view the nannies and search them by the name and location.</p> <p>5). Parents can see the services provided by nannies according to the search criteria provided by them.</p>	<p>2). The system displays a list of nannies.</p> <p>4). If the search results match the query of the user, the system displays the list of nannies. Otherwise, it displays a message “No results found”.</p>

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on a text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock.

TABLE 6. Error Triggers and Messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No Internet connection. Please try again.
4a. The search query not correct.	No results found.

Expanded Use Case: Call or send message to a nanny

The primary actor for this use case is a parent.

Stakeholders and interests:

- **Parents:** Parents view the nannies and select a particular nanny to call or to send a message and to acquire services.
- **Nanny:** A nanny would provide services to the parents who have registered with the system.

Preconditions:

- Parents must select a particular nanny to call or to send a message.

Success guarantee (or post conditions):

- Services are offered by the nannies.
- Parents can rate the nannies after using their services.

TABLE 7. Scenario for a use case call and send a message to nanny.

MAIN SUCCESS SCENARIO	
Actor Action	System Action
1). Parents select a nanny after logging into the system. 3). Parents select the call nanny option. 5). Parents can call the nanny to use their services. 6). The user selects the message option displayed at the bottom of the	2). The system displays the option to call to the nanny. 4). This option displays the user with a phone call app and displays the contact number of the nanny on a dialler.

<p>screen.</p> <p>8) The user can type a text message and send it to the nanny in order to use the services.</p>	<p>7) The system takes the user to a text message application and displays the contact number of a particular nanny in the recipient field.</p>
--	---

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock.

TABLE 8. Error Triggers and Messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No Internet connection. Please try again.

Expanded use case: Rate a nanny

The primary actor for this use case is a parent.

Stakeholders and interests:

- **Parents:** Parents rate the nannies after using their services. This is to ensure the quality of services.
- **Nanny:** A nanny who provides good services, receives a good rating, thus increasing the chances for more orders.

Preconditions:

- A parent must use the services of a nanny in order to rate it.

Success guarantee (or post conditions):

- Services are offered by the nannies.
- Parents can rate the nannies.

TABLE 9. Scenario for a use case rate a nanny.

MAIN SUCCESS SCENERIO	
Actor Action	System Action
1). Parents log in the system and select the profile of a particular nanny. 2). The system displays the nanny profile.	

3). A parent can simply click on 5-star rating to rate the nanny.	4). The system saves the rating for the nanny.
---	--

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on a text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock

TABLE 10. Error Triggers and Messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No Internet connection. Please try again.

Expanded Use Case: Make a profile on the system

The primary actor for this use case is a nanny.

Stakeholders and interests:

- **Parents:** Parents want to view a variety of nannies to choose from.
- **Nanny:** A nanny would provide services to the parents by making an attractive profile within the system.

Preconditions:

- A parent must install the app and register.

Success guarantee (or post conditions):

- Services are offered by the nannies.
- Parents can view, contact and rate the nannies.

TABLE 11. Scenario for a use case make a profile on the system.

MAIN SUCCESS SCENARIO	
Actor Action	System Action
1). A nanny opens the app and selects a register with a nanny option.	
2). The system asks for the essential details.	
3). A nanny fills in the form and provides all the necessary details, which include a username, a	

password, a phone number, a city, a state and a description of services.	
5). A nanny can maintain her profile and provide services through the application.	4). The system registers and authenticates the nanny.

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on a text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock.

TABLE 12. Error Triggers and Messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No internet connection. Please try again.
3a. Input fields not filled.	Please Fill in all details.

Expanded Use Case: Edit a profile and upload a photo

The primary actor for this use case is a nanny.

Stakeholders and interests:

- **Parents:** Parents can view the nannies and find a good nanny for their children.
- **Nanny:** Nannies can make their business profiles attractive by editing them and uploading their business profile photo.

Preconditions:

- A nanny must register and log in the system in order to edit the profile.

Success guarantee (or post conditions):

- Services are offered by the nannies.
- Parents can view the nannies.

TABLE 13. Scenario for a use case edit a profile and upload a photo.

MAIN SUCCESS SCENARIO	
Actor Action	System Action
1). A nanny logs in the system. 3). The nanny selects an “edit a profile” option and edits the fields name, city, state, contact and description of	2). The system displays the profile of the nanny that they made during the registration with a default photo.

<p>services.</p> <p>5). The nanny selects upload photo option.</p> <p>7). The nanny browses the photo and uploads it.</p>	<p>4). The system saves the changes.</p> <p>6). The system displays the option to a browse photo.</p> <p>8). The system saves all the changes into the database.</p>
---	--

Special requirements:

- A user friendly mobile app having a text that is visible from 12 Inches.
- Data validation checks must be applied on application forms.
- A language internationalization on a text displayed.
- A robust recovery of data when the system fails.

Technology and data validation list:

- A keypad is used for typing and entering data.
- Information must be in the specified format when the user is entering data.

Frequency of occurrence

- Round the clock

TABLE 14. Error Triggers and Messages (mapped on the use case).

Trigger	Error Message
1a. The Internet connection not available.	No Internet connection. Please try again.
3a. Input fields not filled.	Please fill in all details.

4 USER INTERFACES

This chapter contains images and description of the graphical user interface of the BabyShift application.

Installation permission



FIGURE 5. App installation



FIGURE 6. App seeking permission to call the contact of nannies through a phone app.

Login

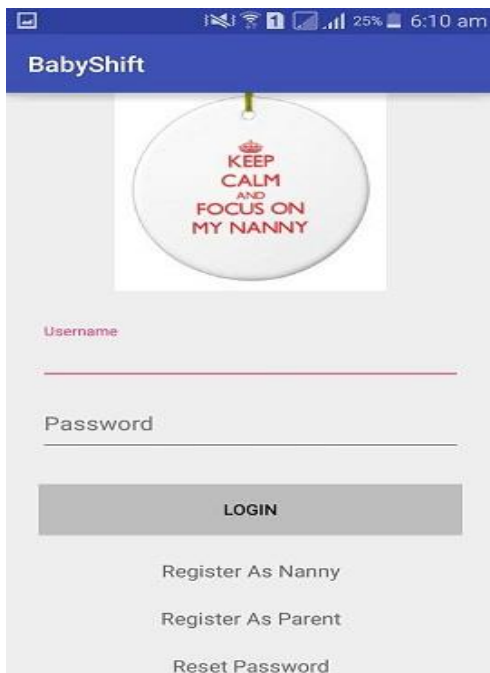


FIGURE 7. Login screen

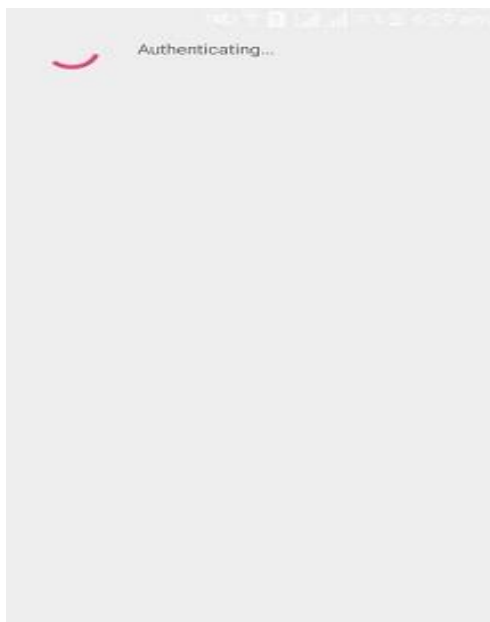


FIGURE 8. Login authentication. (Afterwards the account of nannies or parents are logged in accordingly).

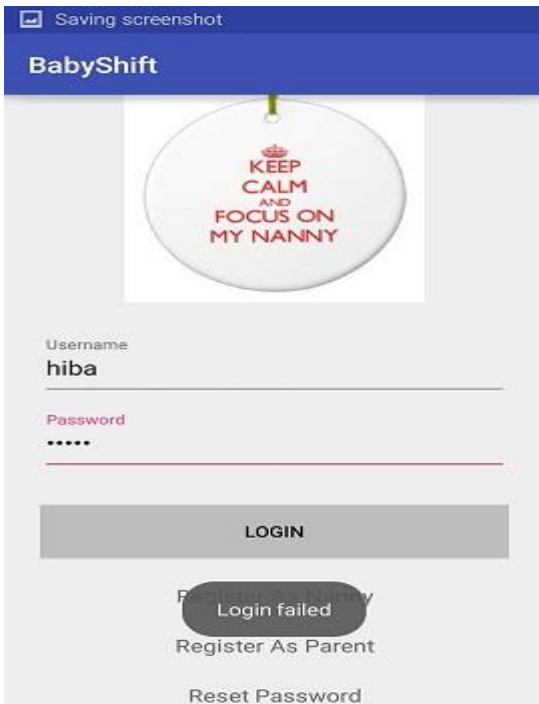


FIGURE. 9. If authentication failed

Resetting a password



FIGURE 10. If a user selects reset password option on the login screen.

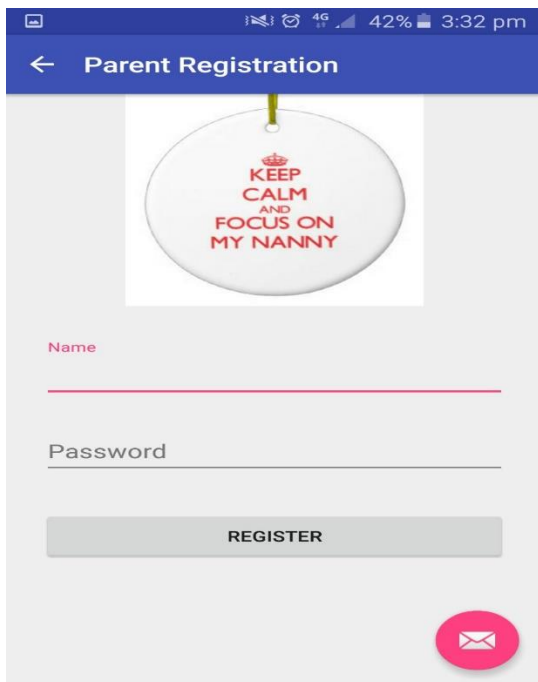


FIGURE 11. Password successfully changed.



FIGURE 12. If the user changing the password does not exist in the database.

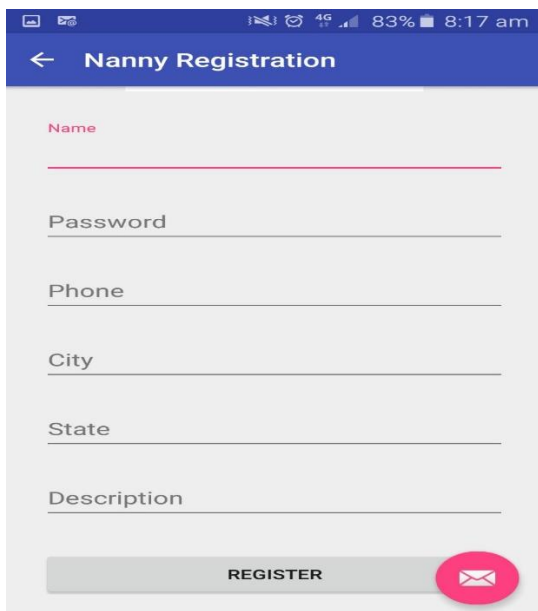
Parent registration



The screenshot shows a mobile application interface for "Parent Registration". At the top, there is a blue header with a back arrow and the text "Parent Registration". Below the header is a white circular image of a keychain with the text "KEEP CALM AND FOCUS ON MY NANNY". Underneath the image are two input fields: "Name" and "Password". A grey "REGISTER" button is positioned below the password field. In the bottom right corner, there is a red circular button with a white envelope icon.

FIGURE 13. Parent registration.

Making a nanny profile



The screenshot shows a mobile application interface for "Nanny Registration". At the top, there is a blue header with a back arrow and the text "Nanny Registration". Below the header are several input fields: "Name", "Password", "Phone", "City", "State", and "Description". A grey "REGISTER" button is located at the bottom left, and a red circular button with a white envelope icon is at the bottom right.

FIGURE 14. Nanny registration.

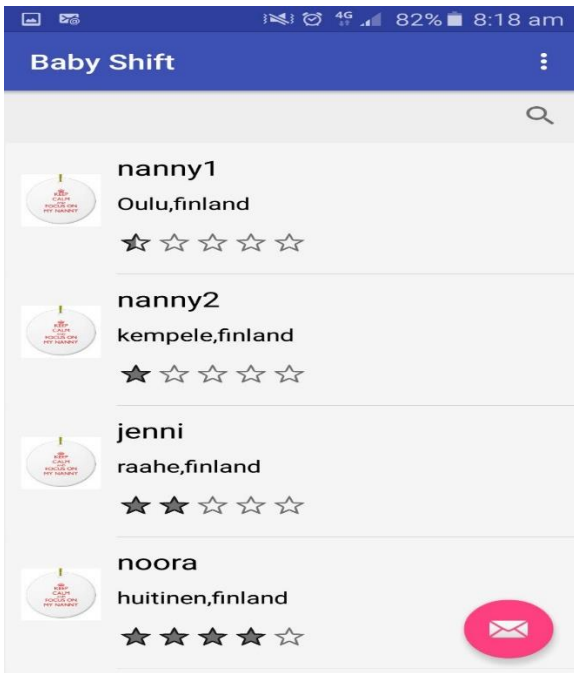


FIGURE 15. A home screen displayed after a successful registration.

Searching nanny services

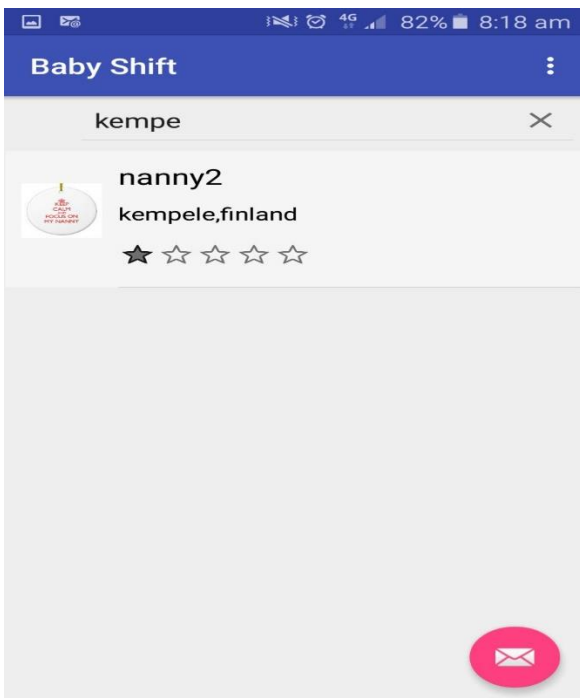


FIGURE 16. Search services.

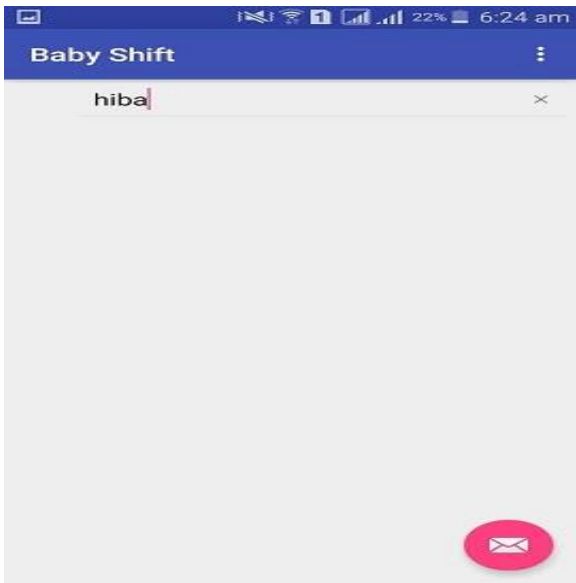


FIGURE 17. If a search query does not match to any nanny.

Call or message nanny



FIGURE 18. View nanny and place a call by clicking on phone icon.

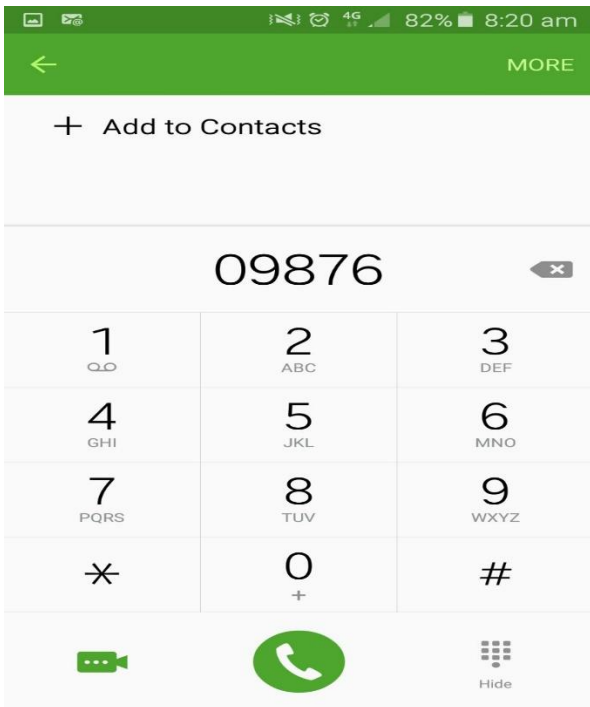


FIGURE 19. A nanny contact displayed on a dialler (The same for a message).

Rating a nanny

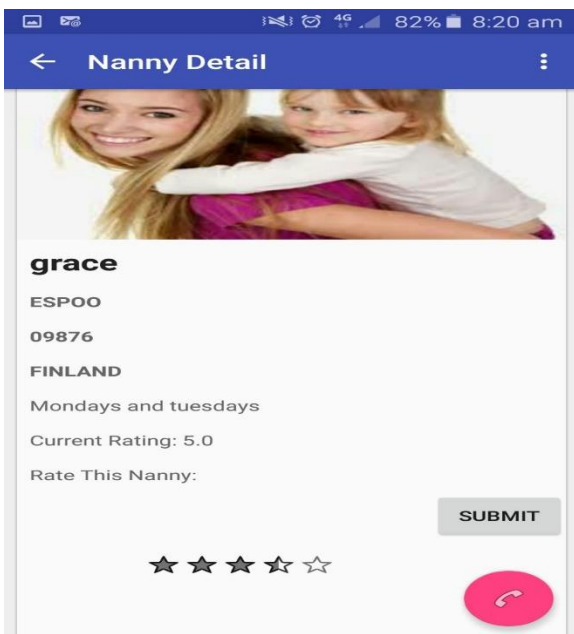


FIGURE 20. Rate a nanny just by clicking on stars and clicking a submit button.

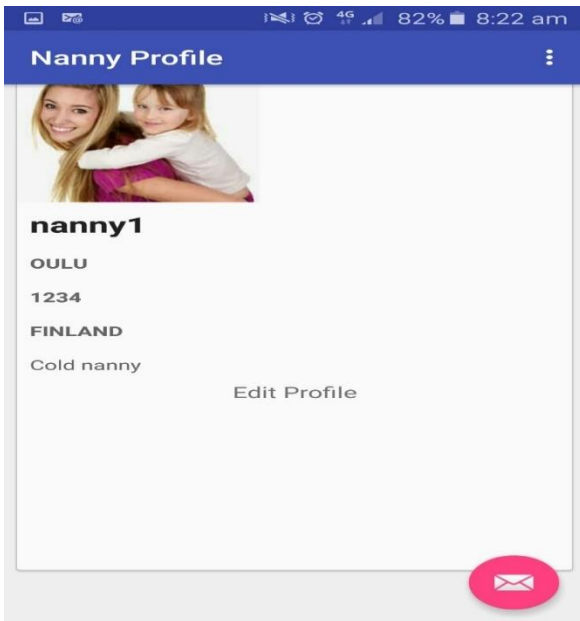


FIGURE 21. A default photo and registration information of a nanny displayed after a successful authentication.

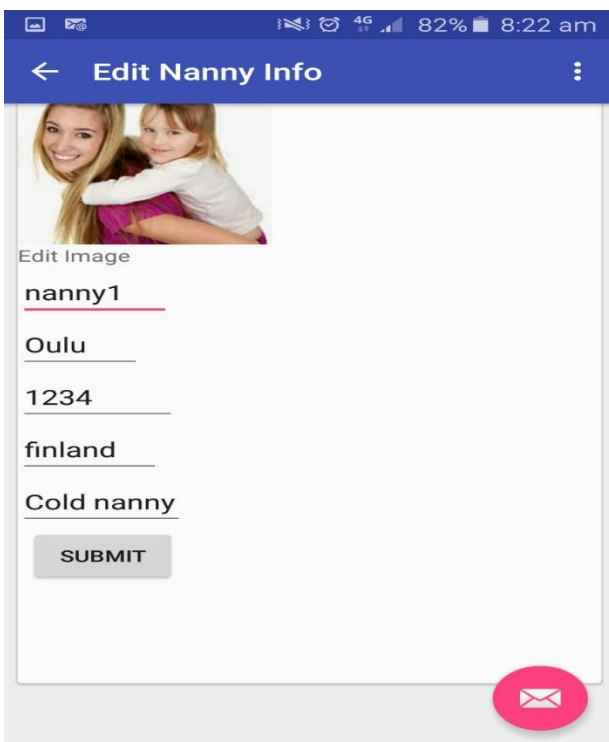


FIGURE 22. Edit a nanny profile screen is displayed.

Uploading a photo

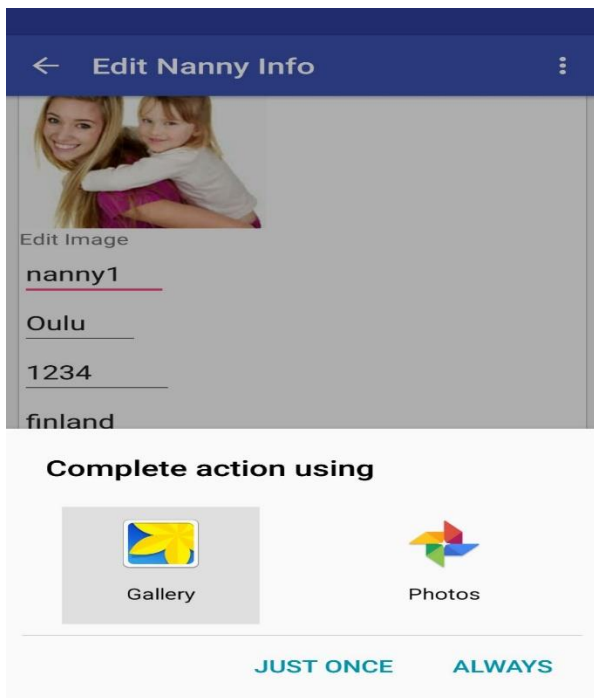


FIGURE 23. If the user selects an edit image option.

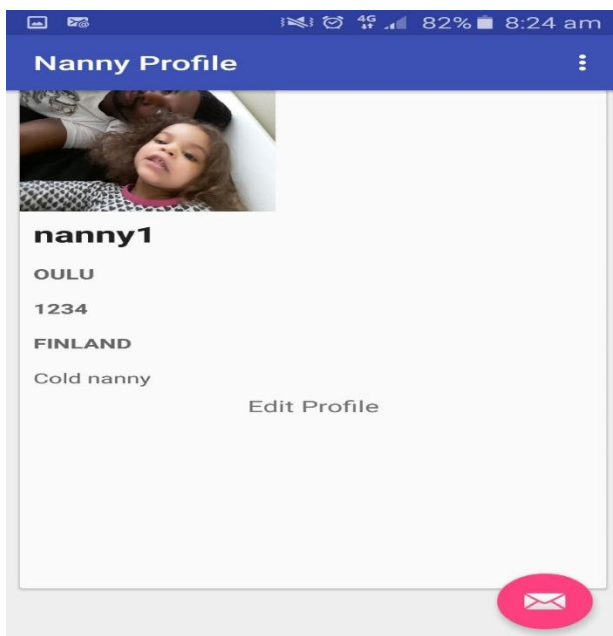


FIGURE 24. After submitting the edited information, it is displayed on the profile.

Logging out

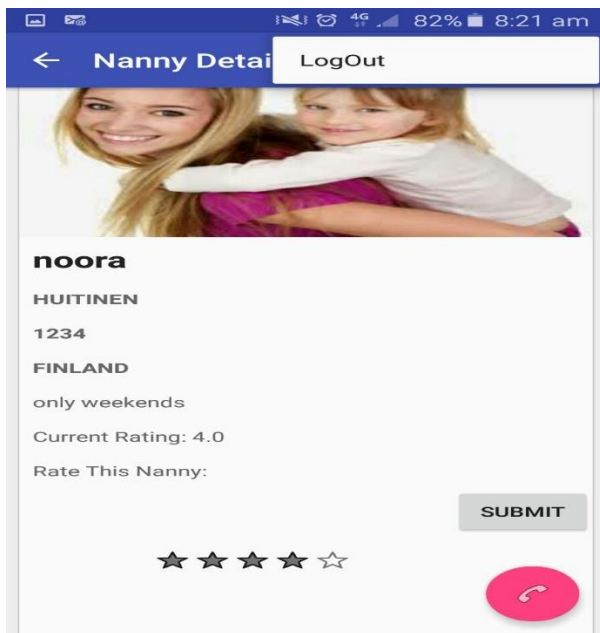


FIGURE 25. Logout Screen for a nanny (the same logout menu works for a parent).

4.1 Incorporation with the knowledge of HCI principles

The application was developed keeping in mind the following HCI design principles:

The application had to be flexible, simple and intuitive to use. It also had to have perceptible information, a low physical effort, size, and space for the approach to use.

4.2 Themes

Keeping in mind the principles of HCI, this application uses simplistic approach in mind when the app was designed. The system utilizes a theme similar to Android's default holo dark theme, however, it uses a different background to make the user feel at ease but still be astounded by a different view un-parallel to most other apps.

4.3 Non-functional requirements

Performance requirements

There is not any specific requirement regarding to speed, throughput and response time constraints on the system. The access to a server or to the Internet must be proficient enough, in order to provide a usage efficiency. It will be ensured that performance engineering has been properly applied at each iteration and phase of the project, enabling the system to receive a performance certification.

Safety requirements

It must be known how the system should respond to extreme conditions. IEEE STD-1228-1994 Software Safety Plans prescribe the best practices of the industry for conducting a software safety hazard analysis to help to ensure the safety requirements and attributes which are defined and specified for an inclusion in the software that commands, controls or monitors critical functions. The above mentioned standards were considered to ensure the safety of this software.

Availability

The system must be available 24/7, i.e. round the clock. The maximum downtime should not be more than 30 minutes.

Reliability

The system should be reliable enough, i.e. the frequency of a system failure should be very little.

Maintainability

The maintainability should be ensured in the system, i.e. the useful life of the project should be made the maximum. The system should be designed in such a way that the future amendments would be easier to meet the new requirements.

Portability

The system should be designed so that it would be usable in all environments. The system should facilitate several computing platforms. For this system, all the environments would be made useful, such as e.g. mobiles and notepads.

Security requirements

Only authorized users with user accounts will be granted access to the system. All security loopholes will be covered as this will prevent hackers from stealing and destroying data and the software will have regular security updates and patches.

Business rules

- Identity check will be done whenever the system is logged in.
- Services offered by nannies will be saved in specific locations as specified by nannies.

5 TESTING

Testing was carried out after the application had been developed and before it was deployed on the server. A black box testing was carried out to ensure that the system follows specifications. Black box testing means testing the system manually for correct specifications.

Inspection was used for the white box testing. WBT ensures that the system follows the correct programming procedures and implementation. Therefore for the implementation check, a technique of inspection, in which the developer themselves inspects the code was followed. The application was tested before deploying and numerous possible errors were removed to a great extent.

6 CONCLUSION

Implementing the BabyShift application using Android studio was great and very challenging, I was able to experience how to build an Android application from front-end to back-end. I used the Version Control systems on Android studio for the first time and it proved to be really useful. Moreover, learning about Android will give me an opportunity in the labour market as the current trend is the application development.

As far as time table is concerned, the project was carried out according to the plan. The aim of developing an Android app was successfully met. Android studio was used to develop the application along with a local database called SQLite. The GitHub version control system was used to keep track of the versions of the app and to collaborate with my team member.

Being new to the app development field, this field of technology enthralled and captivated me. I am pleased that this project assisted me to acquire more knowledge in the computer science field.

7 REFERENCES

1. About Git. 2016. Git. Date of retrieval 7.10.2016
<https://git-scm.com/>
2. About SQLite. 2014. Date of retrieval 6.10.2016
<https://www.sqlite.org/quickstart.html>
3. Developing an app using java. 2016. Date of retrieval 9.10.2016
<https://developer.android.com/training/index.html>
4. Facebook Adverts. 2016. Facebook Business. Date of retrieval 04.10.2016
<https://www.facebook.com/business/products/ads>
5. How people build software. 2016. GitHub. Date of retrieval 7.10.2016
<https://github.com/>
6. Advertising on Instagram. 2016. Instagram Business. Date of retrieval 04.10.2016
<https://business.instagram.com/advertising/>
7. Meet Android Studio. Date of retrieval 03.10.2016
<https://developer.android.com/studio/intro/index.html>
8. Ernest Michael. 2012. Version Control. Date of retrieval 05.10.2016
<https://homes.cs.washington.edu/~mernst/advice/version-control.html>
9. Programming language for Android. 2016. Date of retrieval 08.10.2016
<http://www.androidauthority.com/apps/>
10. Stack overflow. 2015. How to use GitHub with android. Date of retrieval 08.10.2016
<http://stackoverflow.com/questions/16644946/how-do-you-synchronise-projects-to-github-with-android-studio>
11. Why Android use Java. 2010. Date of retrieval 09.10.2016
<https://developer.android.com/training/index.html>
12. Google AdWords. Date of retrieval 04.10.2016
<https://adwords.google.com/home/>