



**JARAMOGI OGINGA ODINGA UNIVERSITY OF SCIENCE AND TECHNOLOGY**

**SCHOOL OF INFORMATICS AND INNOVATIVE SYSTEMS**

**DEPARTMENT OF COMPUTER SCIENCE & SOFTWARE ENGINEERING**

**TITLE: SECURE PHOTOGRAPHER LOCATOR APP**

**PROJECT PROPOSAL BY:**

**BRIAN MWANZA MBURA - REG NO. I231/002/2019**

**NDUNGU STEPHEN MUCHIRI - REG NO. I231/0030/2019**

**A Project Proposal Submitted to the Board of School of Informatics and Innovative Systems in Partial Fulfillment of the Requirements for the Award of Bachelor of Science in Information and Communication Technology at Jaramogi Oginga Odinga University of Science and Technology**

## **Acknowledgement**

First and foremost, we would like to thank The Almighty God for giving us life and keeping us healthy throughout the project period and enabling us to complete the project .Secondly, we would like to thank our project supervisor for guiding us through.

## **Declaration**

This thesis is my original work and has not been presented for an award of a diploma or conferment of degree in any other university or institution.

Signature: ..... Date: .....

Brian Mwanza Mbura

I231/0002/2019

Signature: ..... Date: .....

Ndungu Stephen Muchiri

I231/0030/2019

Signature: ..... Date: .....

Dr. Newton Masinde

School of Information and Innovative Systems

Jaramogi Oginga Odinga University of Science and Technology

## **Abstract**

Photography one of the widely and extensively used service available in the film and production/entertainment industry provided either by a production company or freelancers for commercial gains. During the past few decades, advancements in technology have been exceptionally fast, especially in computer and mobile technologies. This fast improvement resulted in a need to develop applications and services that enhance human lives

Of late security of photographers has been at risk having getting complains from most of the photography service providers on damage and loss of photography equipment by robbers, loss of money due to agreed or disagreed contracts, undefined scope of purchased services according to the event at hand.

In this project we develop a mobile application that tracks photography service providers' mobile phones which are nearby or within a certain distance from clients in need of photography services using GPS. A request made from a client after selecting a nearby photography service provider enables sending a notification to the service provider and communications made, afterwards personal information inclusive of nationality and national identity number of client is to be provided before making any transactions to safe guard the credentials in case of any malpractice activity from either parties.

Global Positioning System (GPS) is the technique that is used to determine the position of objects on the Earth surface. This technique has become widely used in many desktop and mobile applications and will be in use in this project.

## Contents

Acknowledgement .....	2
Declaration.....	3
Abstract.....	4
List of Tables .....	7
List of Figures .....	8
CHAPTER ONE: INTRODUCTION .....	9
1.0: Introduction.....	9
1.1: Background Information.....	9
1.2: Statement of the Problem.....	9
1.3: Objectives .....	9
<b>1.3.1: General Objective</b> .....	9
<b>1.3.2: Specific Objectives</b> .....	10
Scope.....	10
CHAPTER TWO: LITERATURE REVIEW .....	11
2.0: Literature Review .....	11
2.1: Introduction.....	11
2.2: Previous Studies.....	11
2.3: Snappr .....	11
2.4: Shoot My Travel .....	12
2.5: Airbnb Experiences.....	12
2.6: Kodakit.....	12
2.7: Thumbtack .....	12
2.8: Fash.....	12
2.9: Comparison between our photographer locator app and the current applications. ....	12
CHAPTER THREE: METHODOLOGY .....	14
3.0: Methodology .....	14
3.1: Types of System Development life Cycle (SDLC) Methodologies.....	14
Waterfall Model .....	14
3.2: Methodology Used for this Project .....	15
CHAPTER FOUR: ANALYSIS AND REQUIREMENTS .....	17
4.0: System Analysis.....	17
4.1: Requirement elicitations .....	17
4.2: System Requirements .....	18
<b>4.2.1: Software requirement Specification</b> .....	18
<b>4.2.2: Hardware requirement specification</b> .....	19

4.3: Functional requirements .....	19
4.4: Non-functional requirements .....	20
<b>4.4.1: Usability requirements</b> .....	20
<b>4.4.2: Reliability requirements</b> .....	20
<b>4.4.3 Implementation requirements</b> .....	20
4.5: Technologies used.....	20
<b>4.5.1: Android Operating System</b> .....	20
<b>4.5.2: Global Positioning System (GPS)</b> .....	21
<b>4.5.2.1: Definition and Composition</b> .....	21
<b>4.6: MongoDB</b> .....	21
4.7: UML (Unified Modeling Language) .....	21
<b>4.7.1: Activity Diagram</b> .....	22
<b>4.7.2: Use Case Model</b> .....	23
<b>4.7.3: Sequence Diagrams</b> .....	24
<b>4.7.3.1:</b> .....	24
<b>4.7.3.2: Signup for Application Sequence Diagram</b> .....	25
<b>4.7.3.3: Login Sequence Diagram</b> .....	26
<b>4.7.5: Model Diagram</b> .....	27
CHAPTER FIVE: DESIGN.....	28
5.0: System Design .....	28
5.1: Architectural Design .....	28
5.2: Detailed Design.....	29
<b>5.2.1: Functional Diagrams</b> .....	29
<b>Class Diagram</b> .....	29
<b>5.2.2: Behavioral Description</b> .....	30
<b>Data Flow</b> .....	30
<b>Context Level DFD</b> .....	30
<b>Top Level DFD</b> .....	31
<b>Interaction Diagram</b> .....	32
6.0 REFERENCE.....	33

## **List of Tables**

- (a) If the thesis contains many tables, a separate page should be devoted to the List of Tables specifying the title and the page where the Tables can be found in the thesis.
- (b) The List of Tables should be separate from the Table of Contents and should come immediately after the Table of Contents.
- (c) All entries must contain a corresponding page number with leading dots or dashes connecting the entry to the page number.
- (d) Entries should follow the respective chapters as follows::
  - a. Table 1.1 (1<sup>st</sup> table in chapter 1)
  - b. Table 1.2 (2<sup>nd</sup> table in chapter 1)
  - c. Table 2.1 (1<sup>st</sup> table in chapter 2)
  - d. Table 3.1 etc.

## **List of Figures**

- (a) This list is governed by the same rules as the List of Tables (see above).
- (b) Figures include graphs, photographic illustrations, maps, and drawings.
- (c) Entries should follow the order of respective chapters as follows:·
  - a. Figure 1.1 (1<sup>st</sup> Figure in chapter 1)
  - b. Figure 1.2 (2<sup>nd</sup> Figure in chapter 1)
  - c. Figure 2.1 (1<sup>st</sup> Figure in chapter 2)



# **CHAPTER ONE: INTRODUCTION**

## **1.0: Introduction**

### **1.1: Background Information**

In the science. And technology field, development of application has been altering and redefining business success for quite a while now. Statistics are showing clearly that there are over 7 billion mobile phones around the world currently. Due to the promising number of users connected to the internet there is a need to automate manually done processes to digital. As result this will propel to develop new improved service from variety of businesses.

Having Information Communication and Technology diversifying in many areas, in the film and production industry in Kenya, there is need to come up with a secure photographer locator app to securely ease accessibility of photography service provision to clients based on regional locations, i.e. between the service provider and the client.

Over the years photography services have been done manually and this poses a huge risk on security issues to photographers and their equipment and as well as scheduling of time and events.

### **1.2: Statement of the Problem**

Photography industry has of late been experiencing security issues. Reports from the media shows that photography service providers (photographers) of late complain about damage and loss of photography equipment by robbers, loss of money due to agreed or disagreed contracts, undefined scope of purchased services according to the event at hand.

Social media platforms, personal and cooperate film production companies develop websites that act as portfolios for their service provision and marketing of their services but unfortunately not every client has clear intentions, therefore it is difficult for photographers to recognize clients with pure intentions which otherwise, in case of any malicious activity from the opposite party, a photographer gets it difficult to trace the person in question.

## **1.3: Objectives**

### **1.3.1: General Objective**

Developing secure photographic locator app which is enabled by real time tracking.

### **1.3.2: Specific Objectives**

- a. To determine if real time tracking of an individual of a person can work out.
- b. To analyze the workability of the online photography booking system.
- c. To integrate current payment systems with the app.
- d. To create a system that gives system ticket to booked services.

### **Scope**

The project scope defines the description of the work that is required in delivering the photographer locator application. The following are the scopes of work during the course of the project:

Study and understand the requirement of this project.

Construct Software Requirement Specification document of the system.

Construct Software Design Document of the system.

# **CHAPTER TWO: LITERATURE REVIEW**

## **2.0: Literature Review**

### **2.1: Introduction**

This chapter summarizes and synthesizes the existing scholarly research on related current applications.

Since the invention of Photography back in the 1830s in France by Sir Joseph Nicephore after recording the 1st image that could not fade quickly by using a portable camera obscure, there has been a tremendous technological growth in the film and production industry in terms of service delivery and digital marketing. The growth of social media platform being a major technological growth in the industries, where professionals in the industry get to sell and market there services online.

Photographer locator app is a geolocation application based on the use of GPS (Global Positioning System) technique which allows clients to securely request photography services near them by help of google map API, make transactions securely through an encrypted wallet in the system which blocks certain amount of money deposited by client as “deposit” from withdrawals to ensure that the photographers commitment to provide the service before full payment in order to prevent money laundering by the photographer. In addition, before a client makes any transaction to the system’s wallet, the client is supposed to enter his/her personal details that may be used for later security purposes in case of any malicious practices from either parties.

### **2.2: Previous Studies**

Since most of the current applications based on our topic are geolocation applications and a number of mobile applications make use of GPS and mobile network for location information. The following, briefly describe a number of famous related mobile applications and conclude with a comparison between them and our photographer locator app.

### **2.3: Snappr**

Snappr is an on-demand photography service that connects customers with professional photographers in their area. The app allows users to browse photographer portfolios, book a shoot, and receive their edited photos within 48 hours.

This application lacks the security part of both the client and the photographer as compared to what we will be solving in application.

Unlike the traditional booking system, where a photographer can receive a service request from a client but there is no security in either parties i.e. none of the parties is known to have clear intentions of the other.

#### **2.4: Shoot My Travel**

Shoot My Travel is an app that connects travelers with local professional photographers in their destination city. Users can browse photographer portfolios, book a shoot, and receive their edited photos within five days.

#### **2.5: Airbnb Experiences**

Airbnb Experiences is a platform that allows users to book unique experiences led by local hosts. Many of these experiences include a photography component, and users can browse and book experiences that include professional photography services.

#### **2.6: Kodakit**

Kodakit is an on-demand photography service that connects businesses with professional photographers for commercial shoots. The app allows businesses to request a shoot, and Kodakit handles the logistics of finding and booking a photographer.

#### **2.7: Thumbtack**

Thumbtack is a platform that connects users with local professionals for a variety of services, including photography. Users can browse photographer portfolios and reviews, request a quote, and book a shoot directly through the app.

#### **2.8: Fash**

Fash is an app that connects users with local photographers for fashion and beauty shoots. Users can browse photographer portfolios and book a shoot directly through the app.

#### **2.9: Comparison between our photographer locator app and the current applications.**

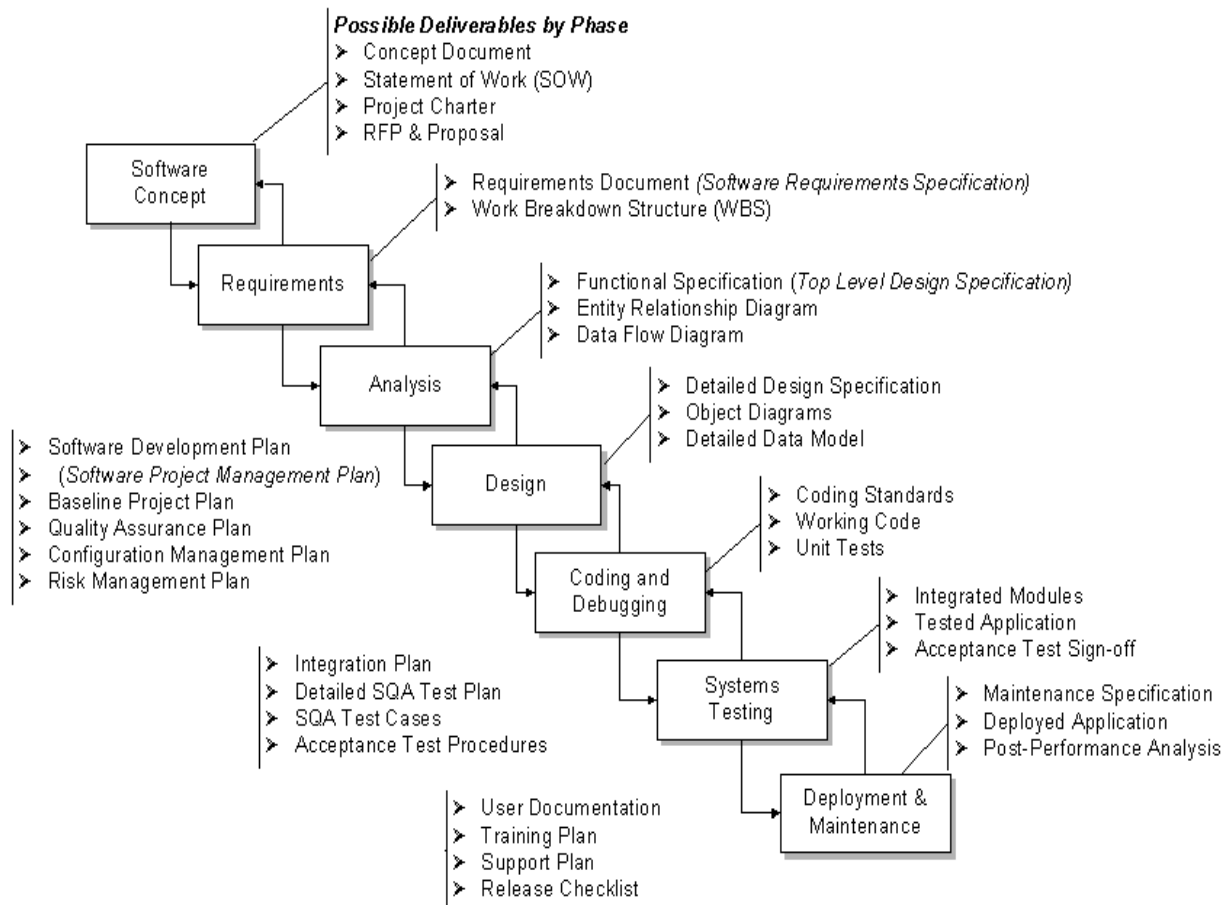
There are a number of mobile tracking apps in the market. However, they all provide continuous tracking in a one-to-one fashion. Meaning that one user is allowed to continuously monitor the location of another mobile. The photographer locator app only reveals location information to interested clients within a regional radius of the photography service providers. In addition, security of both the photographer and client is ensured when a request made from a client after selecting a nearby photography service provider, a notification is sent to the service provider and communications made, afterwards personal information inclusive of nationality and national

identity number of client is to be provided before making any transactions to safe guard the credentials in case of any malicious activity from either parties.

## CHAPTER THREE: METHODOLOGY

### 3.0: Methodology

The methodology used in this project is Waterfall model. This is a conceptual framework describing all activities in a software development project from planning to maintenance. This process is associated with several models, each including a variety of tasks and activities. Waterfall model is one of the models in the System development cycle. The phases of waterfall and their



potential deliverables are:

Figure1: System Development Life Cycle (SDLC) Diagram

### 3.1: Types of System Development life Cycle (SDLC) Methodologies

#### Waterfall Model

This is a linear sequential approach where each phase of the development cycle must be completed before moving onto the next.

This SDLC model is documentation-intensive, with earlier phases documenting what need be performed in the subsequent phases.

## **Agile Methodology**

This is an iterative approach that emphasizes collaboration, flexibility, and customer satisfaction. In the Agile method, the entire project is divided into small incremental builds. All of these builds are provided in iterations.

## **Rapid Application Development (RAD)**

This approach focuses on rapid prototyping and iterative development, with the goal of delivering a functional system in a short amount of time.

## **Structured Analysis and Design Technique (SADT)**

This methodology uses a graphical notation to represent a system's processes, data flows, and data structures.

## **Incremental Model in SDLC**

It is essentially a series of waterfall cycles. The requirements are divided into groups at the start of the project. For each group, the SDLC model is followed to develop software. The SDLC life cycle process is repeated, with each release adding more functionality until all requirements are met. In this method, every cycle act as the maintenance phase for the previous software release.

## **Spiral Model**

The spiral model is a risk-driven process model. This SDLC testing model helps the team to adopt elements of one or more process models like a waterfall, incremental, waterfall, etc.

### **3.2: Methodology Used for this Project**

Methodology used for this project is waterfall methodology. Made up of the below phases:

#### **Requirements**

In this phase we define and plan the project without mentioning specific processes. Potential requirements, deadlines and guidelines for the project are analyzed and placed into a formal requirements document, also called a functional specification.

## **Analysis**

In this phase, the system specifications are analyzed to generate the application's models to guide production. We also define the financial and technical resources in a feasibility study that we will carry out.

## **Design**

This stage will include the overall design of the system, that is, the User Interface and the database design. This stage will help in identify faults from the previous phase and its output, which is the design specification, will help in the next stage of implementation.

## **Coding and Implementation**

In this phase, we write down the source codes used to develop the application's models, logic and requirement specifications designated in the prior phase. The application is coded in smaller components, or units before being put together.

## **Testing**

Verification and validation of what we will have implemented is done at this stage to see if it meets the system requirements.

## **Maintenance**

During maintenance, designing strategies for updating and upgrading the system are availed.

Reason for use of waterfall methodology is because ensures specified goals and requirements are met by the system hence prevents confusion as well as the development of parallel goals that can derail complex promising projects.



## CHAPTER FOUR: ANALYSIS AND REQUIREMENTS

### 4.0: System Analysis

Upon the completion of the Photographer Locator Application, there are a number of things that will be expected of it not only by the prospected users, but also for the administrator of the system. These will therefore form the requirements of the Photographer Locator Application and will be broadly classified in to the system requirements, functional requirements and the Non-functional requirements.

First will have look at requirements elicitions. Requirement elicitions will describe various ways on how we came to our system requirements.

### 4.1: Requirement elicitions

Requirements elicitation is the process of gathering and defining the requirements for a software system. The goal of requirements elicitation is to ensure that the photographer locator application development process is based on a clear and comprehensive understanding of the customer needs and requirements.

We used the following requirement elicitation methods to come up with a solution to solve the problem in question:

#### *Revision of literature, procedures and forms*

By help of a reports from fstoppers.com Business news media, we were able to gather information on complains made by most of the photography service providers being robbed by clients.

This was a problem that we had to solve and came up with security requirements that were to be used in our application to safeguard the photography service providers from being robbed.

#### *Observation*

This is a method of fact finding where the analyst participates in an activity or watch closely as another person performs an activity in order to learn about the system.

We were able to make an observation on the behaviour of mysterious clients, some anonymous with bad intentions while others with pure intentions.

### *Advantages of observation*

- Cheap.
- Data gathered is slightly reliable and issued to validate data obtained through finding techniques.
- Concepts that are too difficult for non-technical staff to explain can be observed.

## **4.2: System Requirements**

### **4.2.1: Software requirement Specification**

Software requirements needed in developing of this application are:

- Android Operating System:

The application must be compatible with the version of Android operating system that it is designed for.

- Dart Programming Language:

Dart is an open-source general-purpose programming language developed by Google. It supports application development in both client and server-side.

- Flutter Plugin:

Flutter is an open source framework to create high quality, high performance mobile applications across mobile operating systems - Android and iOS. It provides a simple, powerful, efficient and easy to understand SDK to write mobile application in Google's own language, Dart.

- Integrated Development Environment (IDE):

Android Studio is the official IDE for Android development. It provides all the necessary tools for developing and testing Android applications.

- Android Software Development Kit (SDK):

The SDK includes libraries, tools, and APIs required for developing Android applications. It is typically included in the IDE.

- Application Programming Interfaces (APIs):

APIs provide a set of functions and methods that developers can use to interact with the Android system and its features. For example, the camera API can be used to take pictures from the camera.

- Third-party Libraries and Frameworks:

Developers can use third-party libraries and frameworks to speed up the development process and add additional features to the application.

- **User Interface (UI) Design Tools:**

The Android Studio IDE includes tools for designing user interfaces, such as layout editors and resource editors.

- **Back-end Services:** We will be using the MongoDB as our data base and

- **Quality Assurance (QA) Tools:**

To ensure that the application is bug-free and functions as intended, developers need to use QA tools such as emulators, testing frameworks, and debugging tools.

- **Deployment Tools:**

Once the application is ready, developers need to use deployment tools to publish the application on the Google Play Store or other app marketplaces.

#### **4.2.2: Hardware requirement specification**

The collection of internal electronic circuits and external physical devices used in building a computer is called the Hardware. The minimum hardware requirement specifications for developing this project are as follows:

Processor: Standard processor with a speed of 2.4 GHz or more

RAM: 8GB MB RAM or more

Hard Disk: 100 GB or more

Monitor: Standard color monitor

Keyboard: Standard keyboard

Mouse: Standard mouse

#### **4.3: Functional requirements**

- i. Security; to ensure confidentiality, integrity and availability
- ii. Administrative functions
- iii. Authentication
- iv. Authorization levels
- v. External interfaces

- vi. Historical data
- vii. Legal and regulatory requirement
- viii. Transaction corrections, adjustments and cancellations

#### **4.4: Non-functional requirements**

##### **4.4.1: Usability requirements**

This system should be user-friendly and easy to use so that users can perform their tasks easily.

##### **4.4.2: Reliability requirements**

The system must perform accurately towards the administrator request. For example, when the administrator saves the edited details of the user, after he reviews the details later, they must be changed according to the latest details that was updated. Moreover, the client is not allowed to view the details that the administrator has. Besides that, the login form will have validity check to ensure that only the authorized users gain access to the system.

##### **4.4.3 Implementation requirements**

In implementing the system, it uses Android Studio as the main tool. This forms the front-end.

At the back-end, the Firebase and MongoDB will be synchronized and be used to maintain the information in the database. This is formed by the databases and other data stores.

Integrating the system with mobile money API

#### **4.5: Technologies used**

##### **4.5.1: Android Operating System**

Android is a mobile OS (operating system), it is open source system based on the

Linux kernel and currently developed by Google. With a user interface based on direct manipulation, Android is designed primarily for touchscreen mobile devices such as smartphones and tablet computers.

## **4.5.2: Global Positioning System (GPS)**

### **4.5.2.1: Definition and Composition**

The Global Positioning System (GPS) is a satellite-based navigation system made up of a network of 24 satellites placed into orbit by the U.S. Department of Defense(DoD). GPS was originally intended for military applications, but in the 1980s, the government made the system available for civilian use. GPS works in any weather conditions, anywhere in the world, 24 hours a day. There are no subscription fees or setup charges to use GPS.

GPS is a system. It's made up of three parts:

- Satellites: act like the stars in constellations—we know where they are supposed to be at any given time.
- The ground stations: use radar to make sure they are actually where we think they are.
- A receiver: it is the equipment found in phones or cars, and is constantly listening for a signal from these satellites. The receiver figures out how far away they are from some of satellites.

## **4.6: MongoDB**

NoSQL databases are more scalable and provide superior performance. MongoDB is such a NoSQL database that scales by adding more and more servers and increases productivity with its flexible document model.

## **4.7: UML (Unified Modeling Language)**

It is an industry-standard graphical language for specifying, visualizing, constructing and documenting the artifacts of an object-oriented system under development. It provides a number of models. In this project we use the following:

#### 4.7.1: Activity Diagram

The figure below represents an activity diagram that illustrates the general functionality of the photographer locator app.

Having stipulated the main actors in the system and main functions carried out in and by all actors in the photographer locator app.

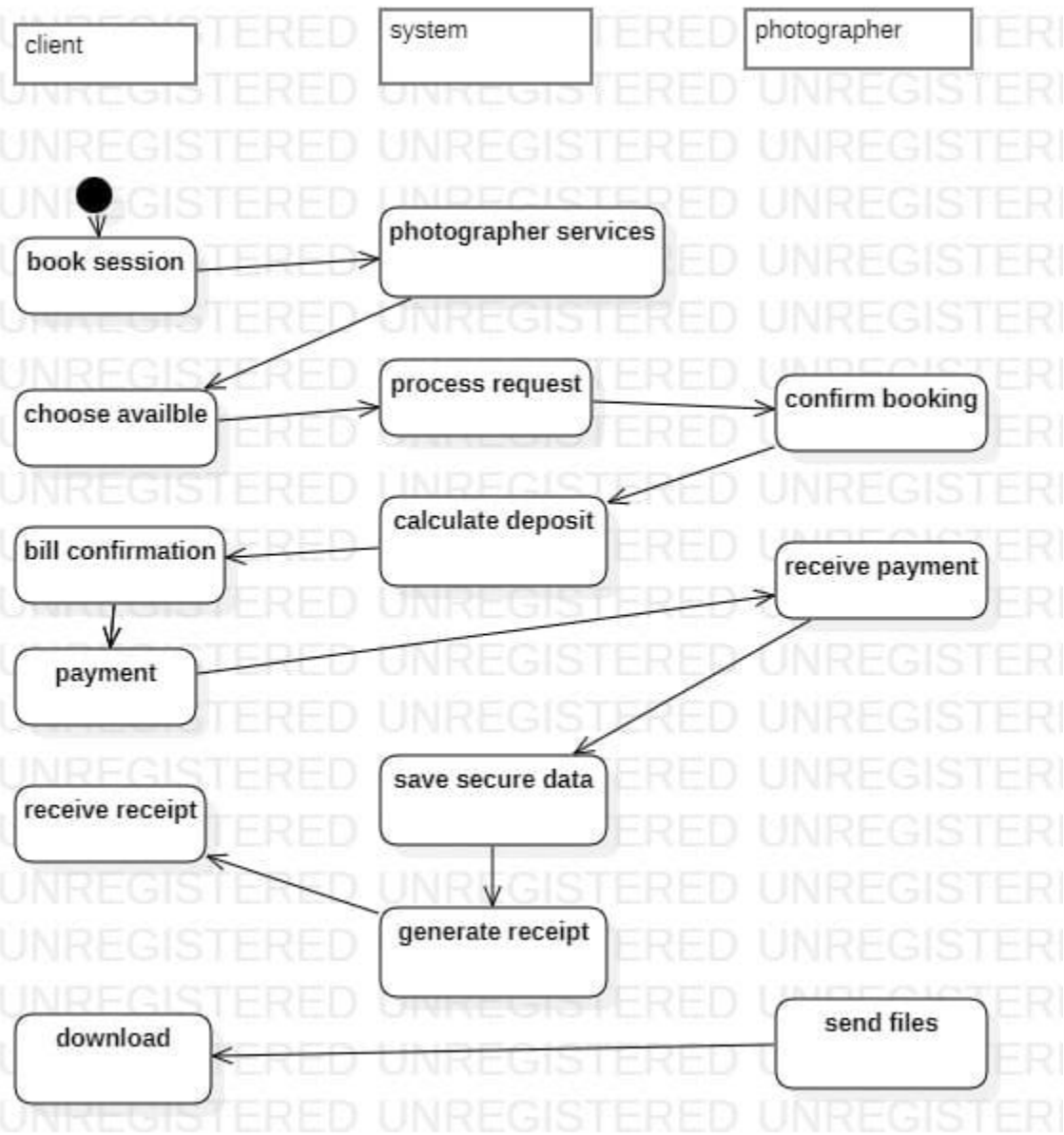


Figure 3: Photographer Locator App Activity Diagram

#### 4.7.2: Use Case Model

The use case model captures the requirements of a system. Use cases are a means

of communication among users and other stakeholders what the system is intended to do.

The below use case diagram illustrates the general functionality of the photographer locator app.

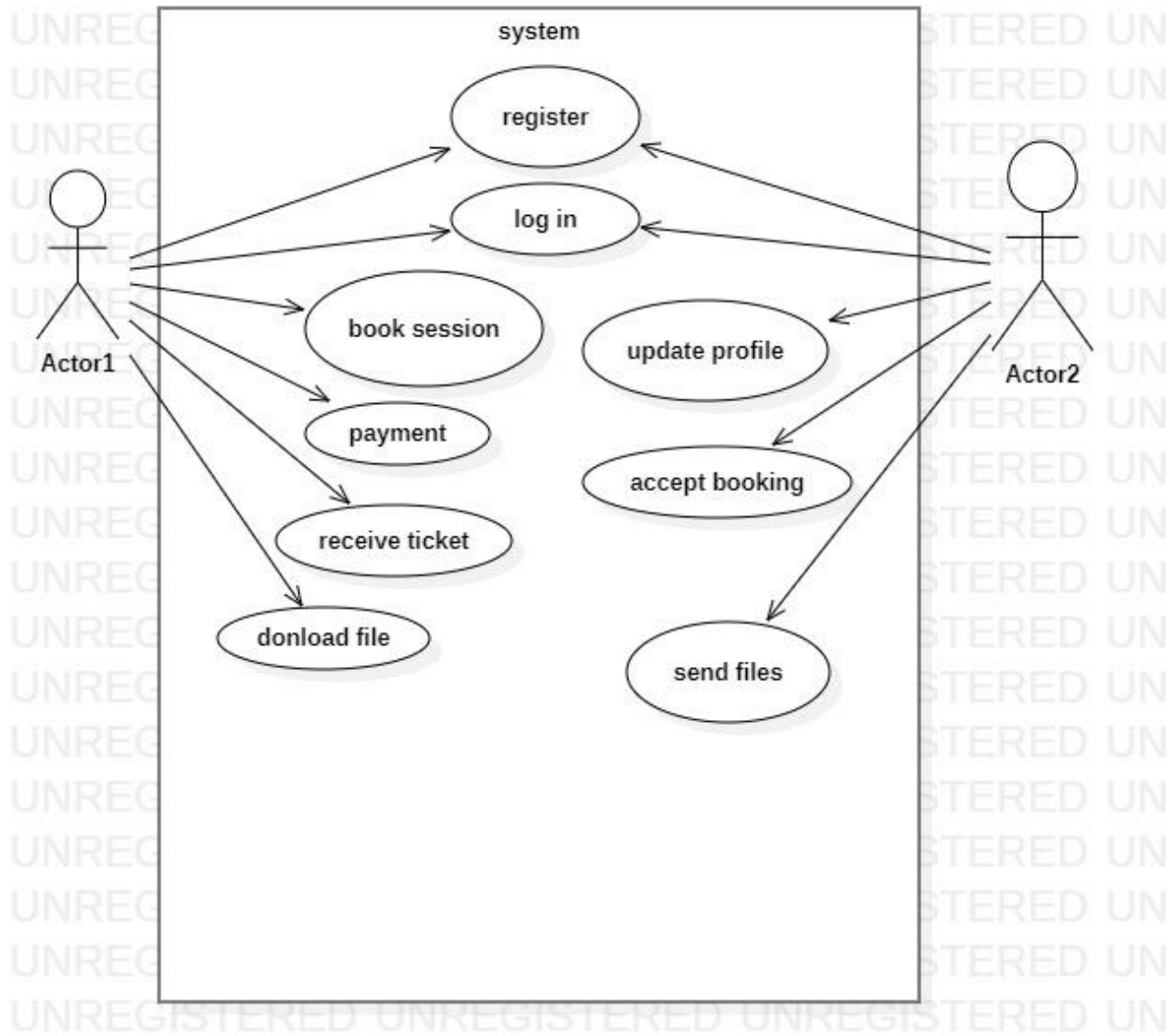


Figure2: Photographer Locator App Use Case Diagram

### 4.7.3: Sequence Diagrams

Used to represent the flow of messages, events and actions between the objects or components of a system.

It has two dimensions:

- The horizontal dimension shows the objects participating in the interaction.
- The vertical arrangement of messages indicates their order.

#### 4.7.3.1:

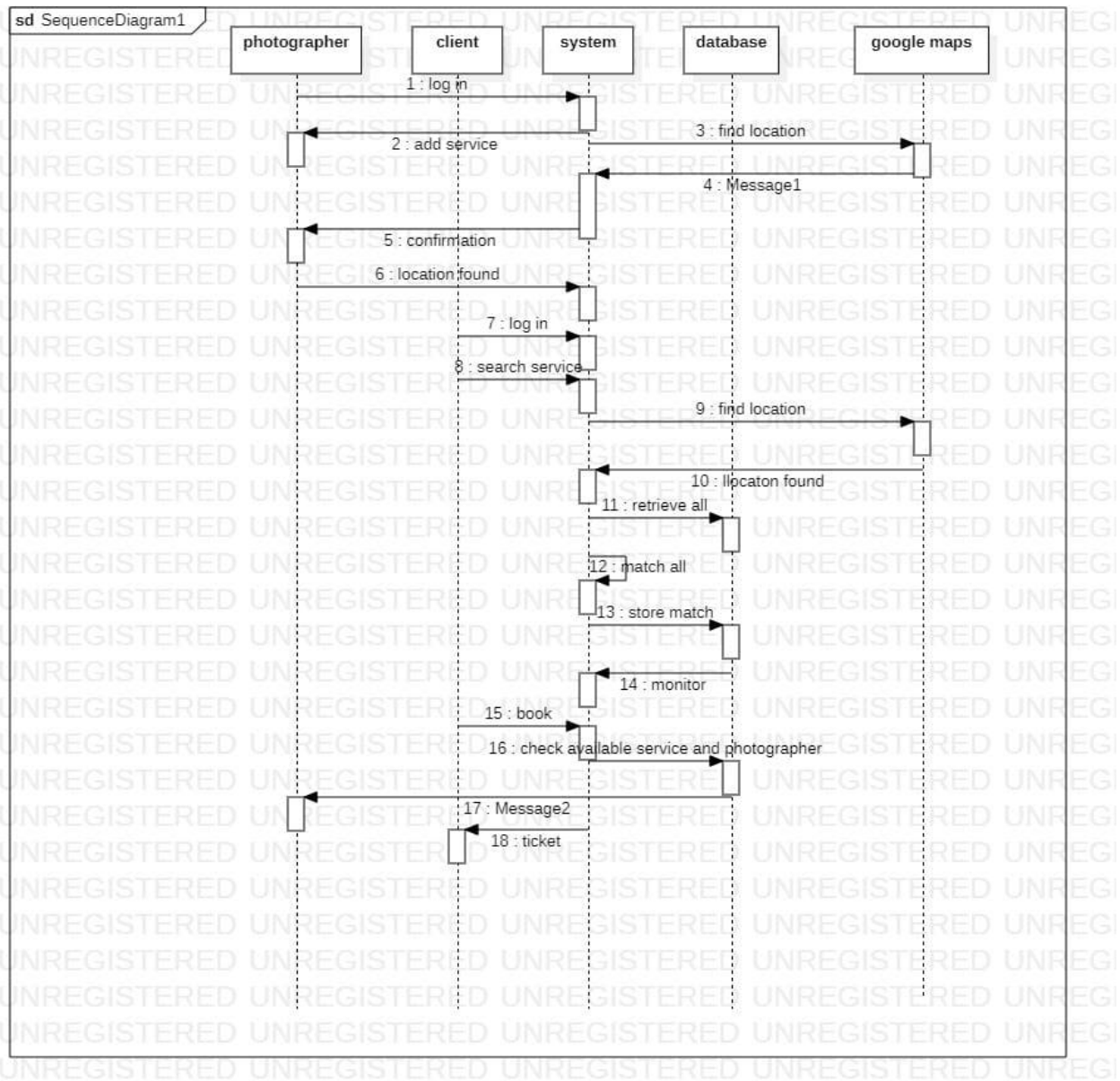


Figure 4: Photographer locator app activity diagram.



#### 4.7.3.2: Signup for Application Sequence Diagram

The system asks the user to enter username and password to login to the applications shown in Table 1 and Figure 4.

Use case name	Signup
Actors	Photographer, Client
Precondition	No
Main flow of events	<ul style="list-style-type: none"><li>• The system asks the user to enter email, username, password and phone number.</li><li>• The user enters this information.</li><li>• The system saves the input information in the database.</li><li>• It is signed up to the system.</li></ul>
Post-condition	The system displays a graphical interface containing successful text.

Table 1: Signup for application

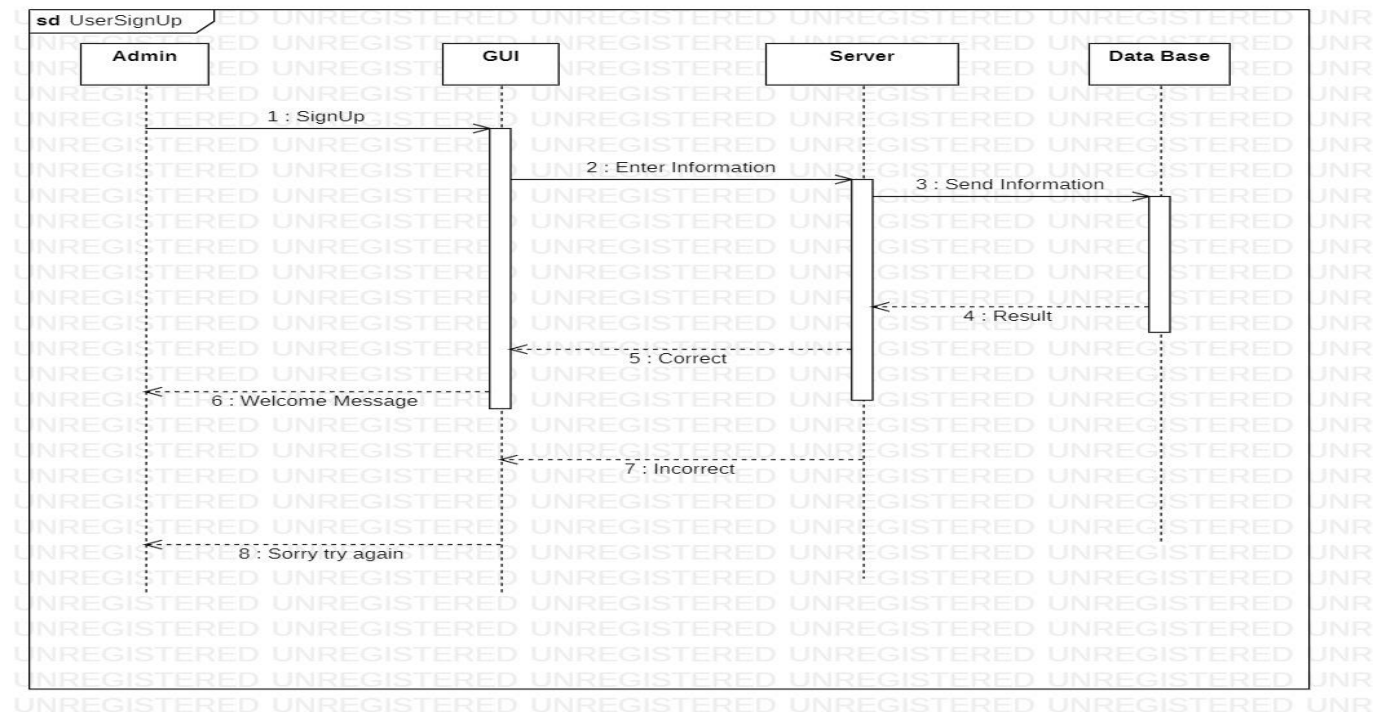


Figure 5: Signup for application activity diagram

#### 4.7.3.3: Login Sequence Diagram

The system asks the user to enter username and password to login to the application as shown in Table 2 and Figure 5.

Use case name	Signup
Actors	Photographer, Client
Precondition	No
Main flow of events	<ul style="list-style-type: none"><li>• The system asks the user to enter username and password.</li><li>• The user enters username and password.</li><li>• The system checks the validity of the input information.</li><li>• It is logged on to the system.</li></ul>
Post-condition	The system displays a graphical interface containing successful text.

Table 2: Login

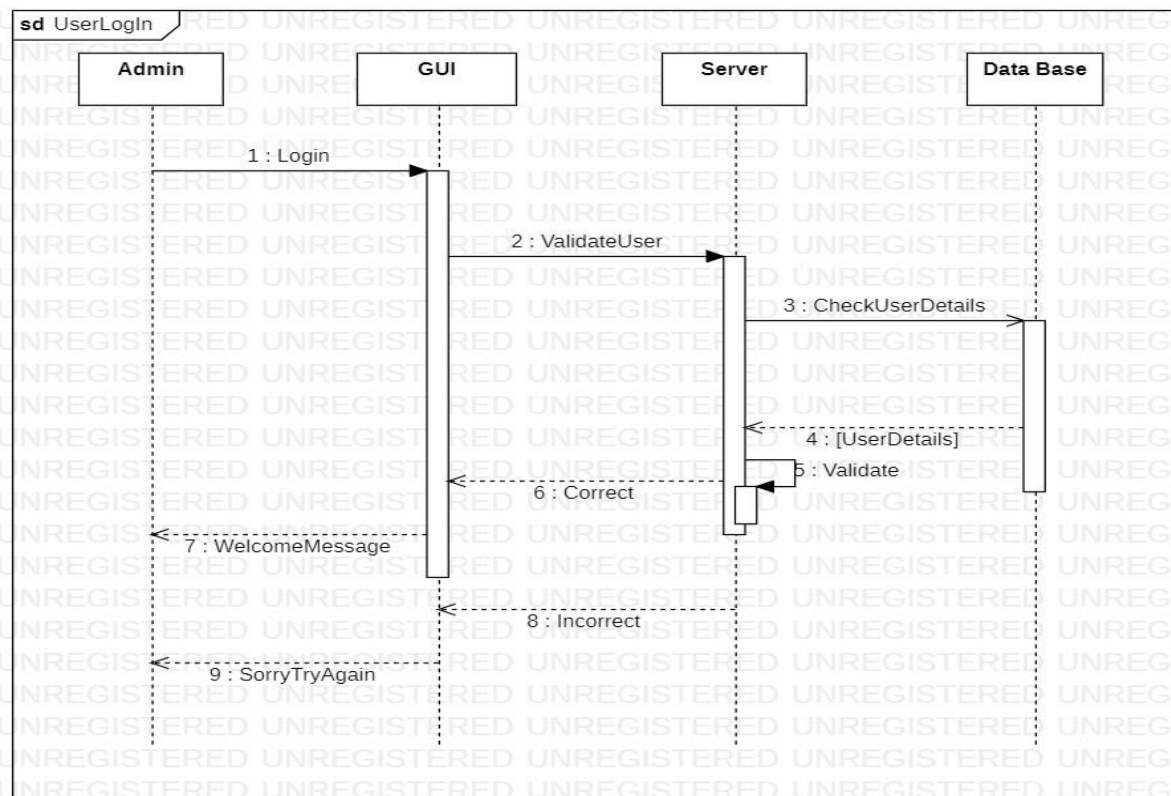


Figure 6: Login Sequence diagram

#### 4.7.5: Model Diagram

A domain class diagram is a type of UML diagram that shows the relationships between various classes in a software system. It is used to model the static structure of a system.

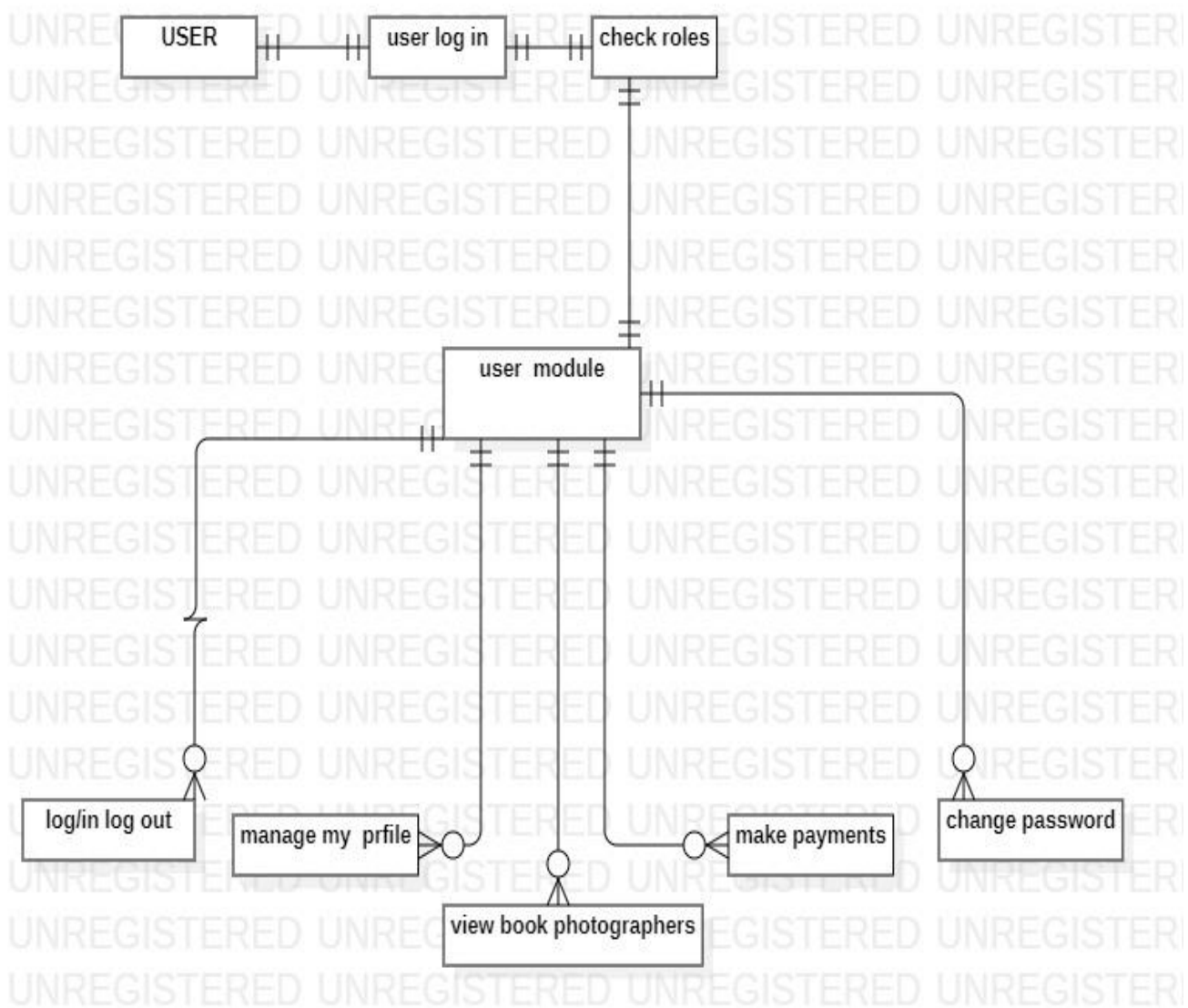


Figure 7: Domain model diagram

# CHAPTER FIVE: DESIGN

## 5.0: System Design

### 5.1: Architectural Design

Photographer locator app uses three-tier architecture. Three-tier architecture is a client-server architecture in which the functional process logic, data access, computer data storage and user interface are developed and maintained as independent modules on separate platforms. Three-tier architecture is a software design pattern and well-established software architecture.

The three tiers in photographer locator app are:

- i. **Presentation Tier:** Occupies the top level and displays information related to services available on a website. This tier communicates with other tiers by sending results to the browser and other tiers in the network.
- ii. **Application Tier:** Also called the middle tier, logic tier, business logic or logic tier, this tier is pulled from the presentation tier. It controls application functionality by performing detailed processing.
- iii. **Data Tier:** Houses database servers where information is stored and retrieved. Data in this tier is kept independent of application servers or business logic.

## 5.2: Detailed Design

### 5.2.1: Functional Diagrams

#### Class Diagram

Defines and provides the overview and structure of the whole system in terms of classes, attributes and methods, and the relationships between different classes.

Figure 9 illustrates and creates a functional diagram of the system classes and serves as a system development resource within the software development life cycle.

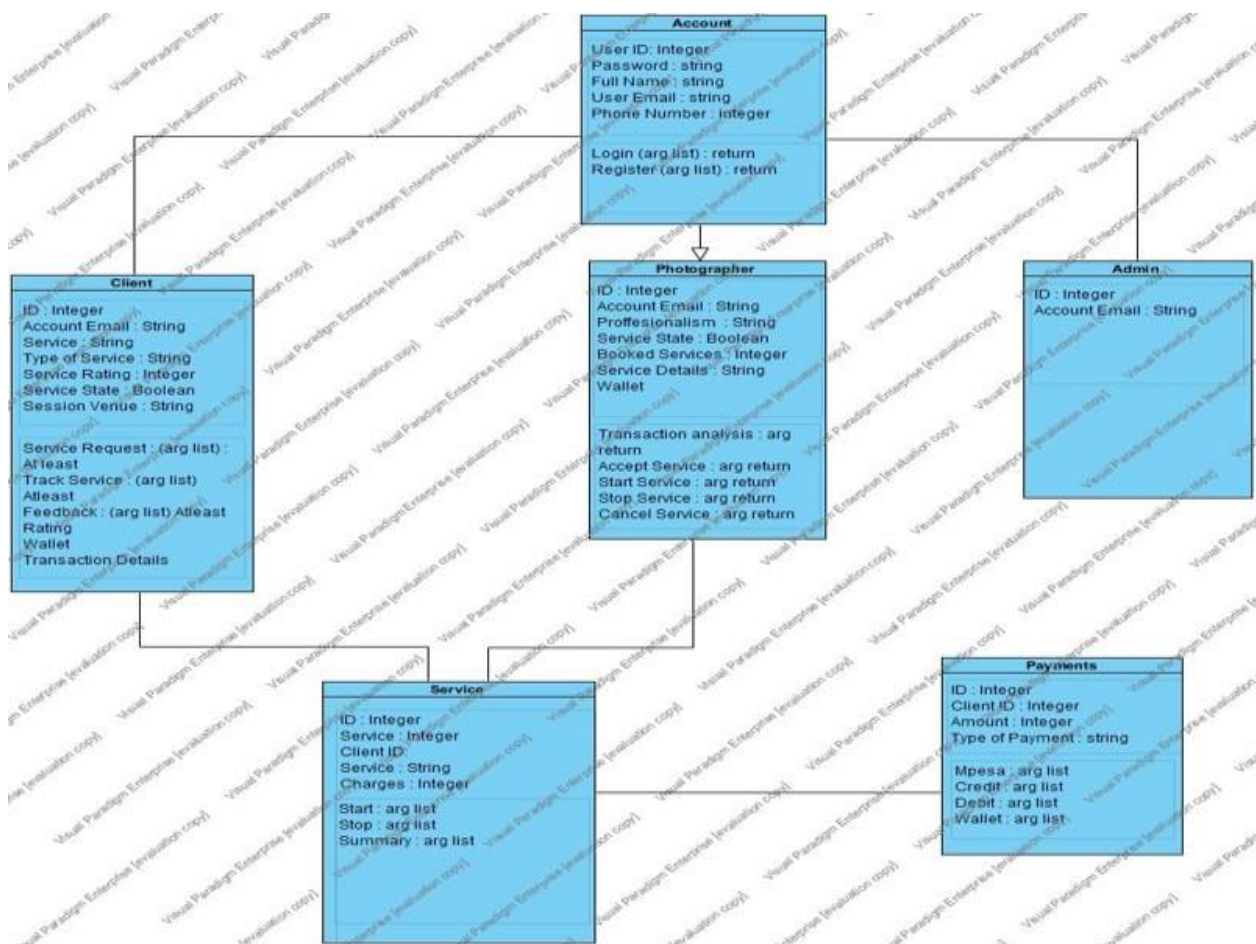


Figure 8: Photographer Locator App System Class Diagram

## 5.2.2: Behavioral Description

### Data Flow

#### Context Level DFD

In the Context Level, the whole system is shown as a single process.

Figure 10 below illustrates Photographer Locator App context level DFD showing Inputs to the overall system and outputs from the overall system shown together with their destinations.

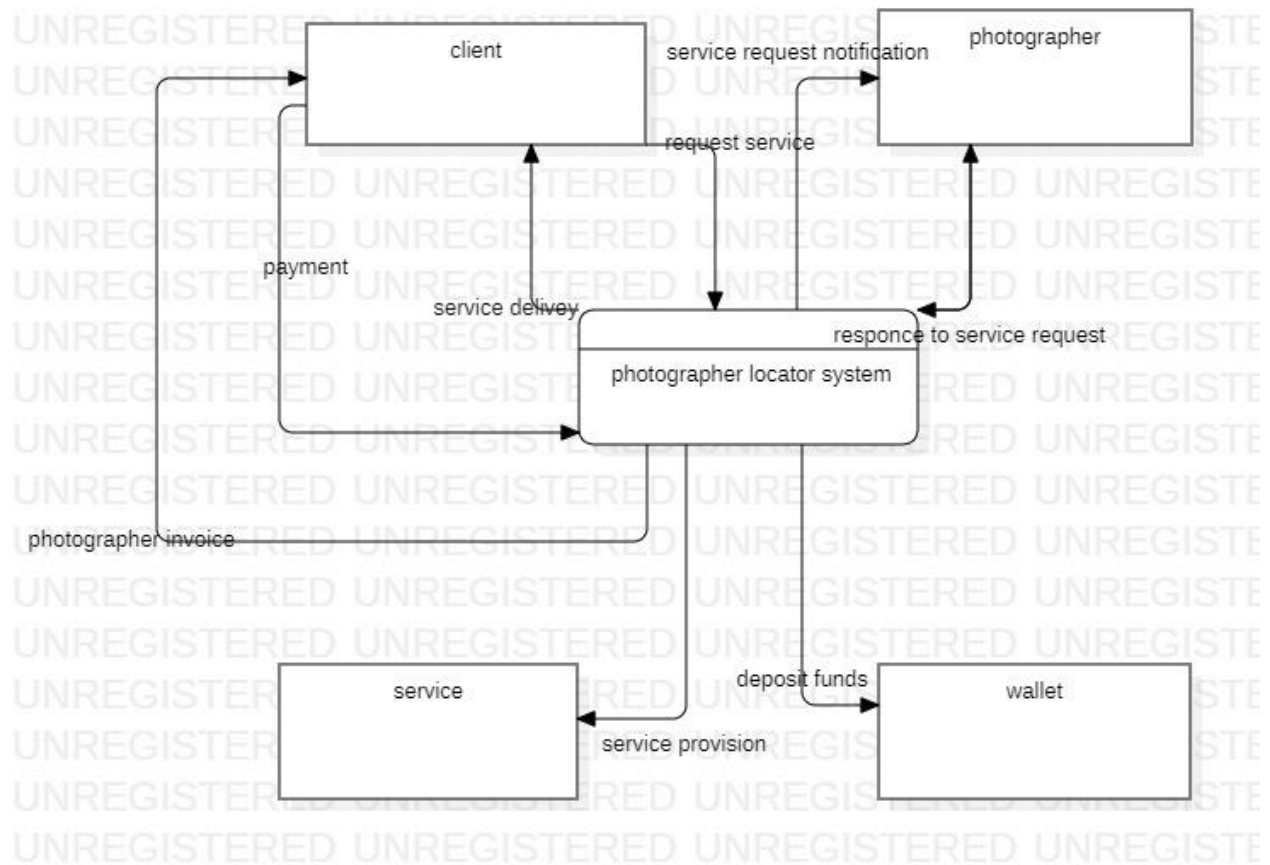


Figure 9: Photographer Locator App Context Level DFD

## Top Level DFD

The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. Figure 12 below shows the Photographer Locator App Top level DFD illustrating what the system does in more detail.

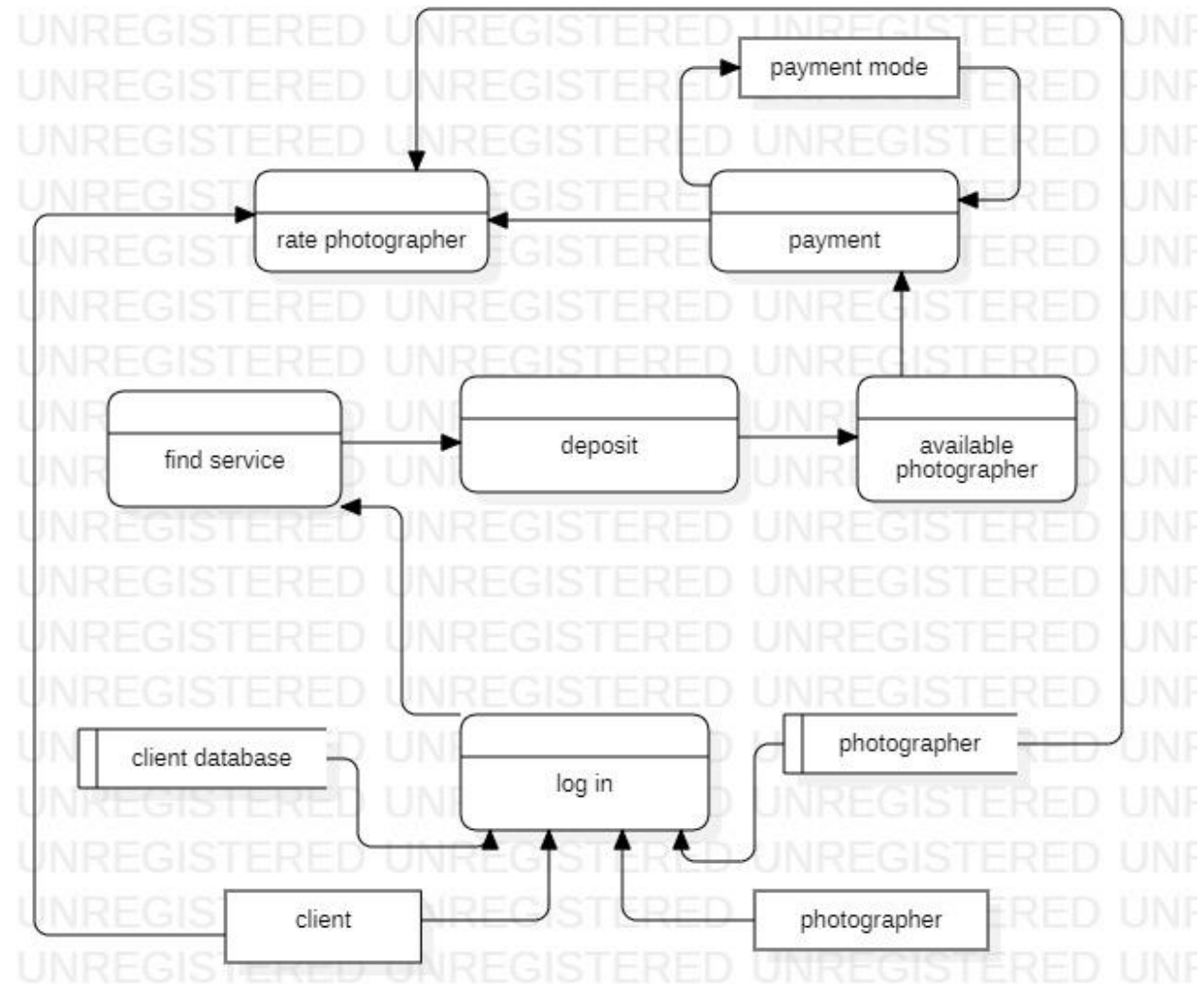


Figure 12: Photographer Locator App Top Level DFD

## Interaction Diagram

System interaction diagram initiates the interaction between the objects utilizing message passing. It is a specialization of both behavior and of interaction fragment which represents a unit of behavior that focuses on the observable exchange of information between connectable elements.

Figure 13 below shows how the system entities interact together to represent a unit of behavior.

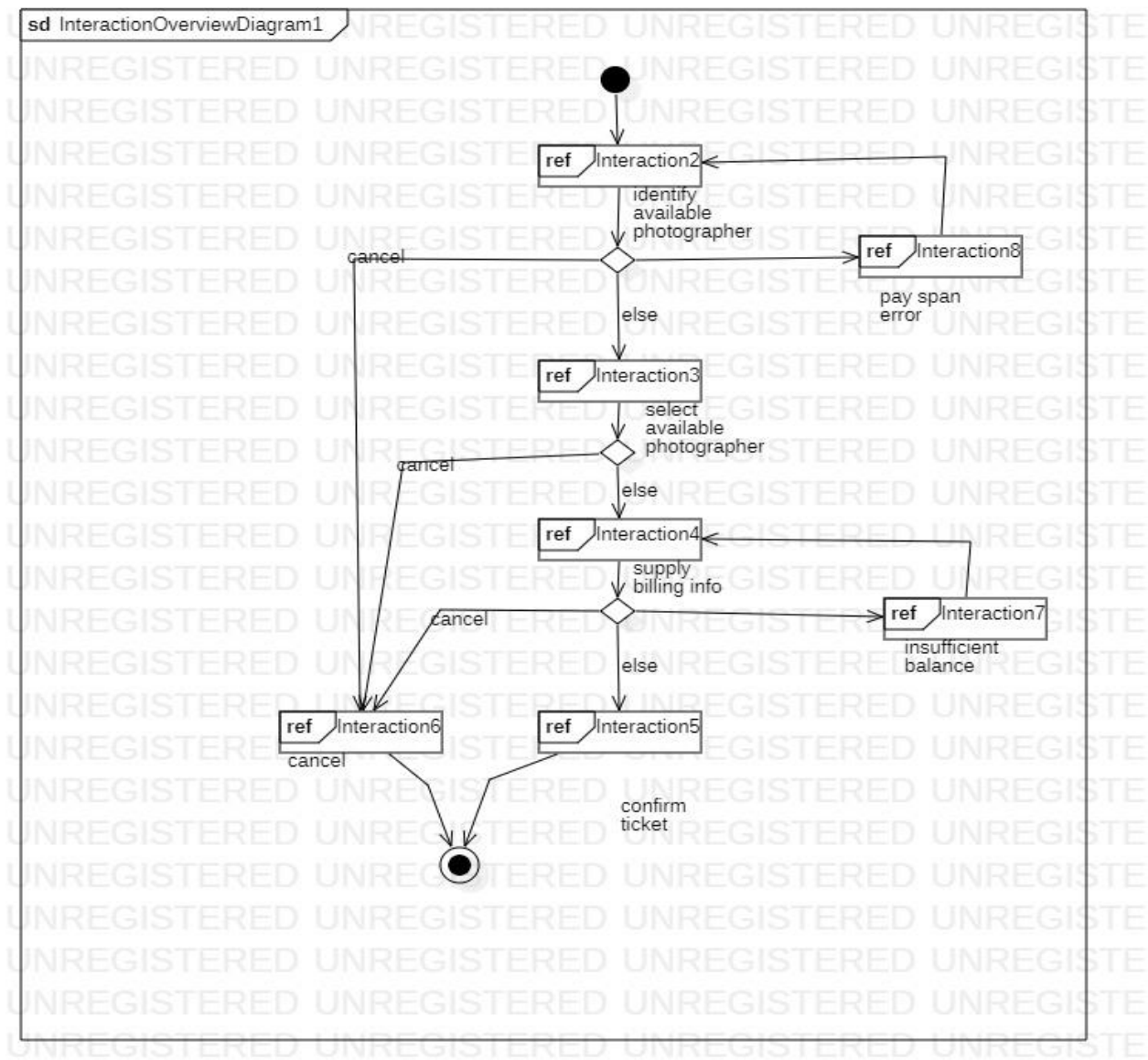


Figure 13: Interaction Diagram



## CHAPTER SIX: IMPLEMENTATION

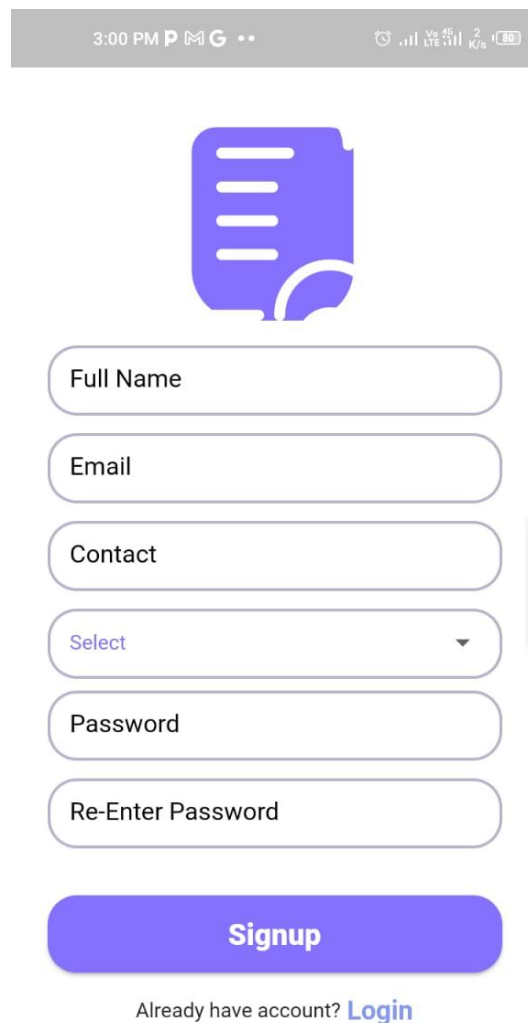
### 6.1: Introduction

This chapter shows the implementation of our mobile application. It demonstrates graphical interfaces for the service provider(photographer) and the client and examples of the execution of the application that has been done.

### 6.2: Application Interfaces

#### 6.2.1: Sign up

First the service provider(photographer) or the client must sign up to the system to be saved in the database and then login normally to the application as shown in Figure (14).



3:00 PM P M G ..

Full Name

Email

Contact

Select

Password

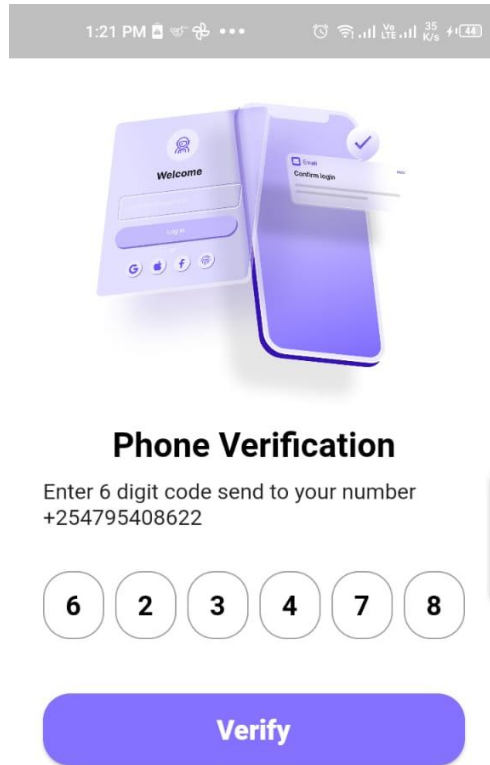
Re-Enter Password

**Signup**

Already have account? [Login](#)

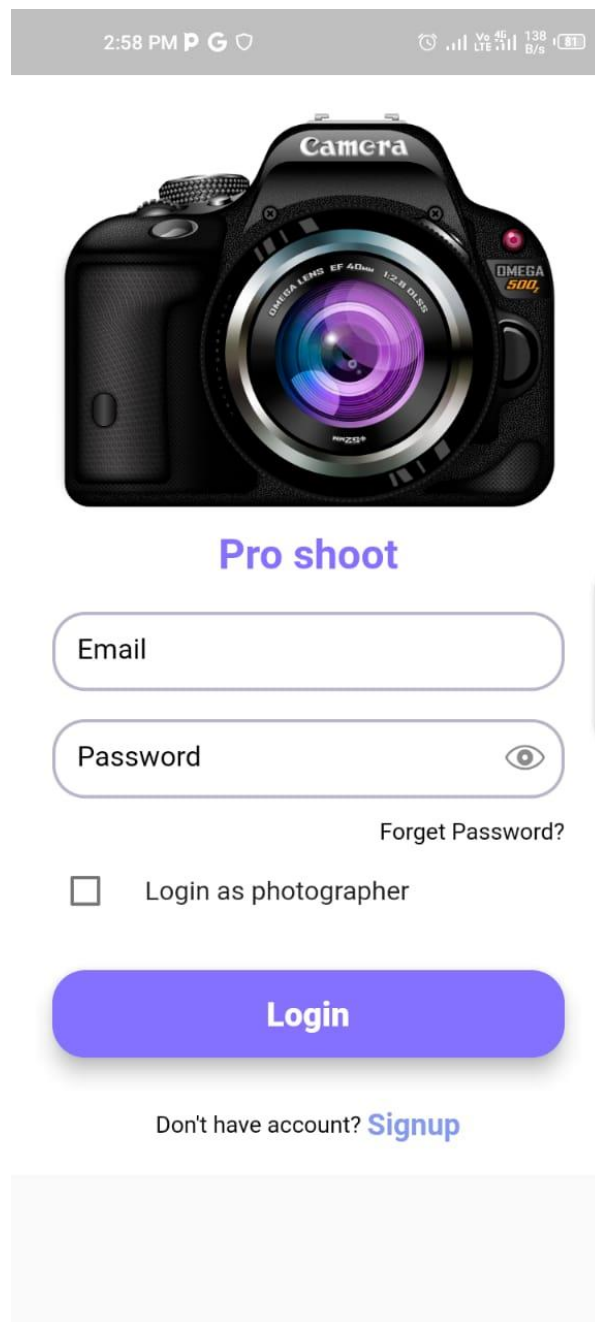
### 6.2.2: OTP Verification

The application send a verification code to a user for security functionalities between the users and the application as shown in figure (15).



### 6.2.3: Login

The service provider(photographer) or the client must login to the system to be able to use the application as shown in Figure (16). The login page allows a user to either login as a photographer or as a client or customer.



The image shows a mobile application login screen. At the top, a status bar displays the time as 2:58 PM, signal strength, 4G LTE, and battery level at 81%. Below the status bar is a large illustration of a black Canon EOS 50D DSLR camera with a purple-tinted lens. The word "Camera" is written above the camera, and "OMEGA 500" is on the right side. Below the camera, the text "Pro shoot" is displayed in a purple font. Underneath, there are two input fields: "Email" and "Password". The "Password" field has an eye icon to toggle visibility. To the right of the "Password" field is a link that says "Forget Password?". Below these fields is a checkbox labeled "Login as photographer". At the bottom of the form is a large purple button with the text "Login". Below the button is a link that says "Don't have account? Signup". The entire form is set against a light gray background.

2:58 PM P G

Camera

OMEGA 500

Pro shoot

Email

Password

Forget Password?

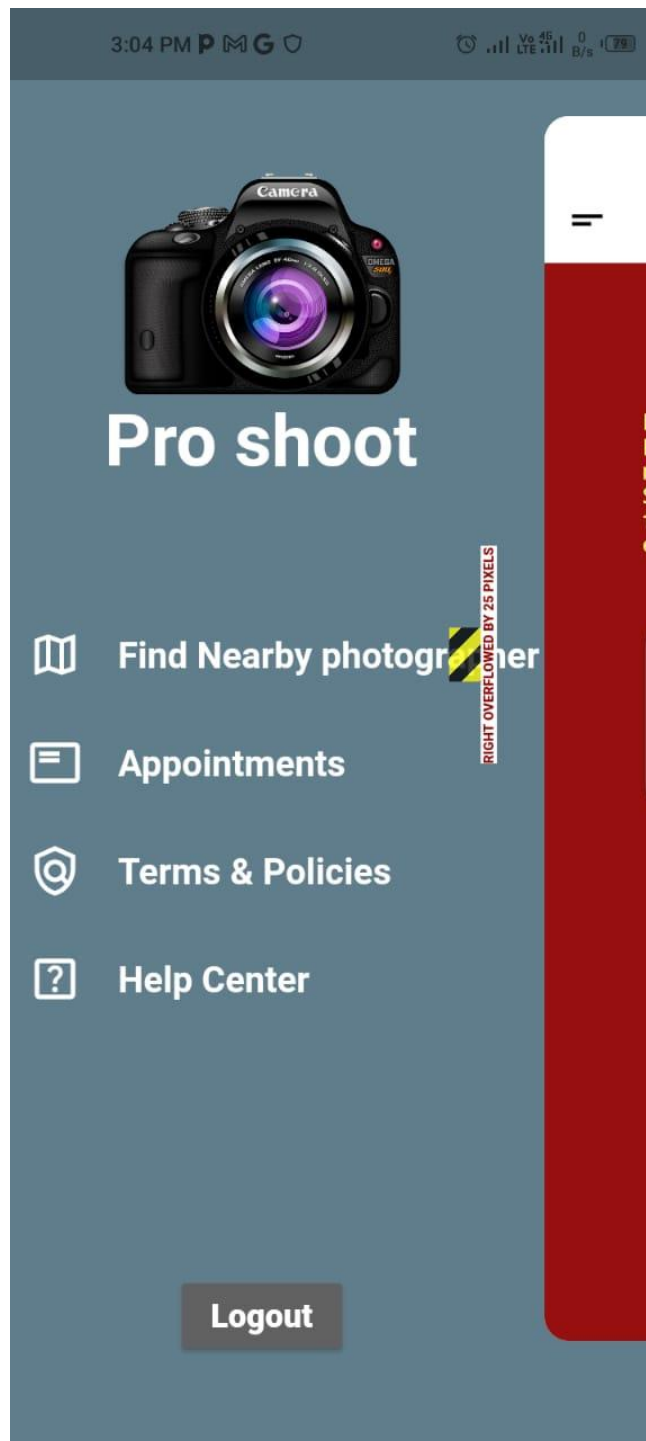
☐ Login as photographer

Login

Don't have account? [Signup](#)

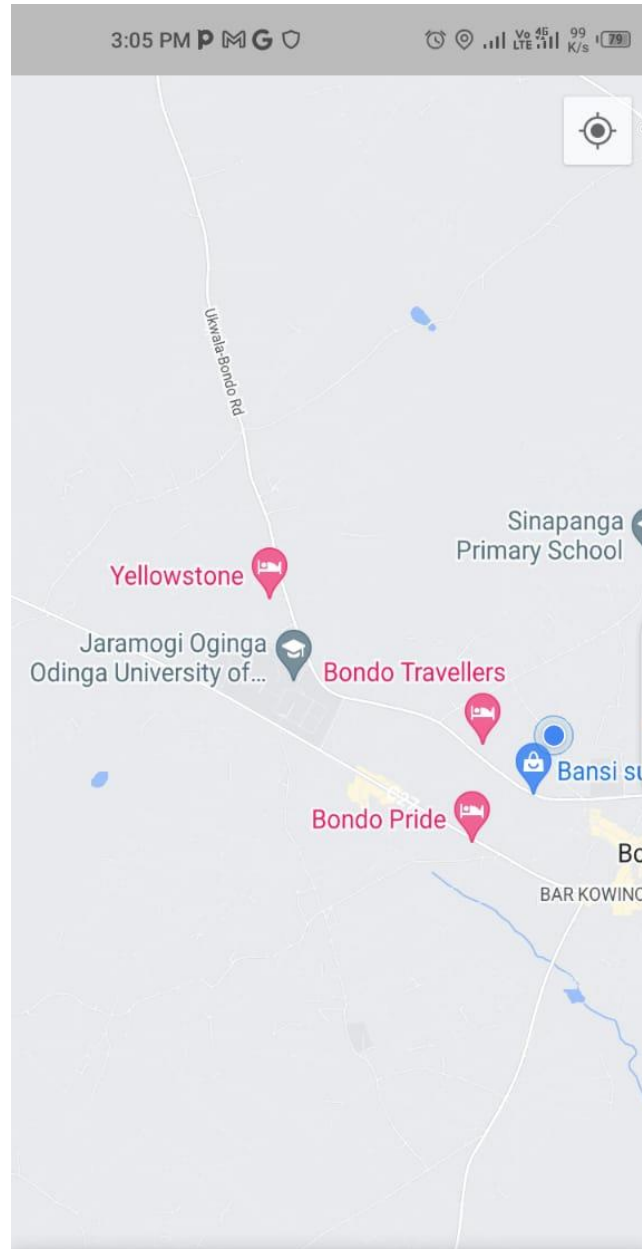
#### 6.2.4: Finding nearby photographer

Once the client or customer is logged into the application he or she can request a photoshoot from any nearby photographer by clicking on the text or flat button “*Find Nearby photographer*” as shown in figure (17).



#### 6.2.4: Searching For Nearby Photographer

The application automatically searches for the nearby photography upon clicking the text button “*Find Nearby photographer*” by the client/customer/user. The application does automatic search for nearby photographers by help of geolocator technology as shown in figure (18).



Finding near by Photographers

### 6.2.5: Photographer's screen

Upon registration of the photographer and receipt of service request from the client, the photographer schedules an appointment with the client as shown in figure (19).

The screenshot shows a mobile application interface for scheduling an appointment. At the top is a status bar with the time 12:08 PM, signal strength, LTE, and battery level at 43%. Below the status bar are three input fields: the first contains 'likoni', the second contains '5', and the third contains 'cool and nice'. Below these is a 'Timings' section with 'From' and 'To' labels. The 'From' field shows '8:00 PM' with a clock icon, and the 'To' field shows '4:05 PM' with a clock icon. Below the timings is a section titled 'Choose your shop location' with a right-pointing arrow. Under this title is a large rectangular box with the text 'Tap and Pick' inside. At the bottom of the screen is a large blue button with the text 'Next' in white.

12:08 PM △ P ...

likoni

5

cool and nice

**Timings**

From To

8:00 PM ⌚ 4:05 PM ⌚

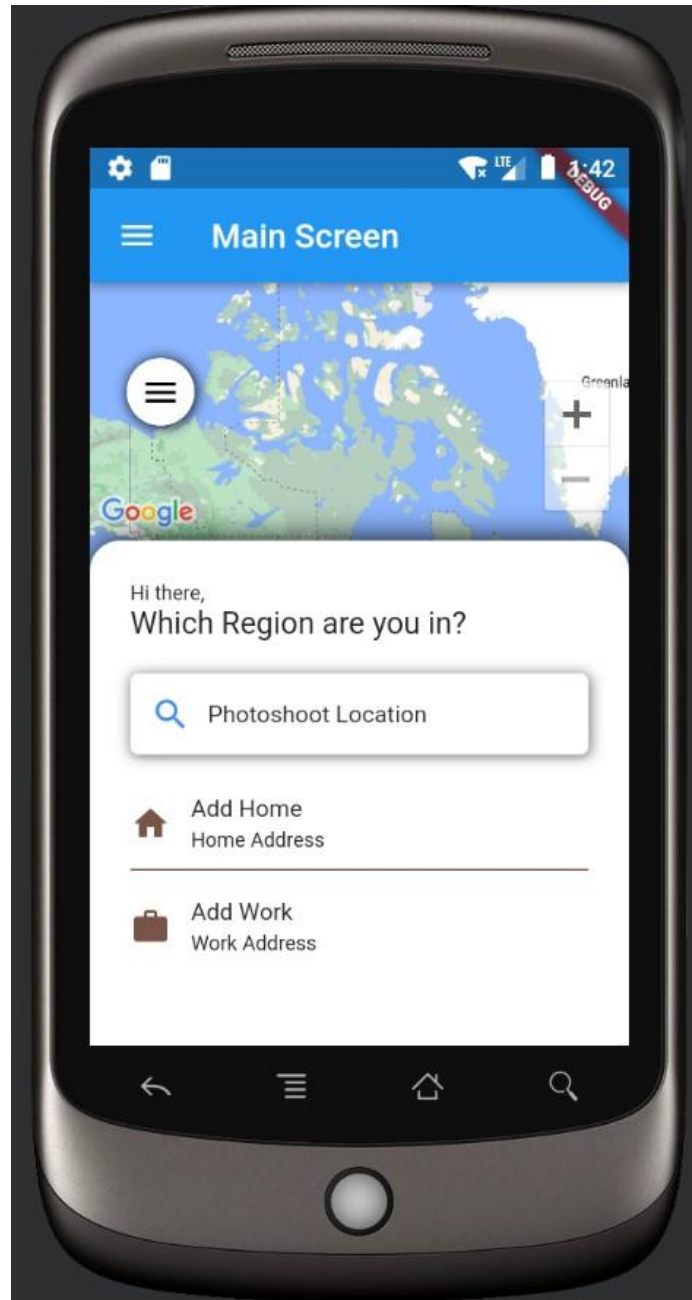
**Choose your shop location** →

Tap and Pick

**Next**

### 6.2.6: Selecting and locating of photoshoot location

The client is able to give his or her location address to the photographer and the suitable photoshoot location where both the service provider and client will meet as shown in figure (20).



### **6.3: Summary**

This chapter has represented the implementation of the mobile application and its interfaces which belong to administrator and user.



# **CHAPTER SEVEN: CONCLUTIONS AND RECOMMENDATIONS**

## **7.1: Introduction**

This chapter states conclusions and suggests recommendations for future research and studies.

## **7.2: Results**

In this project, the mobile application developed,has achieved results which:

- Located nearby photographers to clients in a given region.
- Allows photographers to schedule appointments with a client once a service is requested.
- Sends a notification to the photographer once a request is made by a client.
- Display maps.

## **7.3: Recommendations**




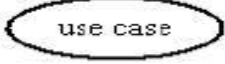
The following is recommended:

- Deal with the security side in a wide area and focus on it.
- Improve the system by allowing it to work in the global or international region for more users and efficiency across the glob.






## **7.4: Conclutions**

This chapter has represented the results from this research and it also shows the recommendations.

## APPENDICES

 Actor			 use case
User	connect	Connect task that depends on other	task

Explain the symbols used in (Use case Diagram)

				
End	Begin	Decision	Activity	Connect

Explain the symbols used in (Activity Diagram)

## REFERENCE

Mobile phones in ourlives,[online], url:

<http://myessaypoint.com/importance-of-mobile-phone-technology>, date accessed: 1/9/2016

Introduction to Mobile Apps,[online], url:

<https://www.reference.com/history/mobile-phones-important-a182c74764871316>, date accessed: 1/9/2016

Mobile Apps,[online], url:

<https://www.consumer.ftc.gov/articles/0018-understanding-mobile-apps>, date accessed: 1/9/2016

Understanding Mobile Apps, [online],

url: <http://www.makeuseof.com/tag/finding-your-friends-via-gps-9-free-mobileapps/>, date accessed: 1/6/2016

Android Operating System,[online], url:

[https://www.tutorialspoint.com/android/android\\_overview.htm](https://www.tutorialspoint.com/android/android_overview.htm), date accessed: 1/8/2016

GPS definition,[online], url:

<http://spaceplace.nasa.gov/gps/en/>, date accessed: 1/8/2016

UML the Unified Modeling Language, William H. Mitchell (whm), date accessed: 2/11/2016

Use case Diagram, [online], url:

<http://www.uml-diagrams.org/use-case-diagrams.html>, date accessed: 1/8/2016

Sequence Diagram, [online], url:

<http://www.uml-diagrams.org/sequence-diagrams.html>, date accessed: 1/8/2016

Activity Diagram, [online], url:

[https://www.tutorialspoint.com/uml/uml\\_activity\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_activity_diagram.htm), date accessed: 1/8/2016

Class Diagram, [online], url:

[https://www.tutorialspoint.com/uml/uml\\_class\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_class_diagram.htm), date accessed: 1/8/2016

Object Diagram, [online], url:

[https://www.tutorialspoint.com/uml/uml\\_object\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_object_diagram.htm), date accessed:

1/8/2016

Component Diagram, [online], url:

[https://www.tutorialspoint.com/uml/uml\\_component\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_component_diagram.htm), date accessed:  
1/8/2016

Deployment Diagram, [online], url:

[https://www.tutorialspoint.com/uml/uml\\_deployment\\_diagram.htm](https://www.tutorialspoint.com/uml/uml_deployment_diagram.htm), date accessed:  
1/8/2016

[https://fstoppers.com/business/why-photographers-don't-get-paid-367687](https://fstoppers.com/business/why-photographers-don-t-get-paid-367687)

<https://marututech.com/7-trends-of-mobile-application-development/>

Dart Programming for beginners, [online], url:

[Introduction to Dart Programming Language - GeeksforGeeks](#)

