

# **1. Software Design Description**

## **1.1 Overview**

This document ensures the software design descriptions for the purpose of AppDermis application project. This document is prepared according to the “IEEE Standard for Information Technology

– Systems Design – Software Design Descriptions – IEEE 1016 –2009”. This software design document provides the details of how the AppDermis software should have been done. The further particulars are offered by using graphical notations such as viewpoints, use case models, sequence diagrams, class diagrams, object behavior models and alternative supporting design knowledges.

### **1.1.1 Scope**

Software Design Document (SDD) is for a ground level system which will work as a evidence of notion for the usage of building a system the supplies a ground level of functionality to represent applicability for wide scale production usage. Software Design is centered on the ground level system and crucial sections of the system. For this certain Software Design Document, the center is placed on generation of the documents and changes of the documents. The system will be used in association by other pre-actual systems and will be formed usually of a document interplay appearance that abstracts document interactions and usage of the document objects.

### **1.1.2 Purpose**

Software Design Document (SDD) is for a ground level system which will work as a evidence of notion for the usage of building a system the supplies a ground level of functionality to represent applicability for wide scale production usage. Software Design is centered on the ground level system and crucial sections of the system. For this certain Software Design Document, the center is placed on generation of the documents and changes of the documents. The system will be used in association by other pre-actual systems and will be formed usually of a document interplay appearance that abstracts document interactions and usage of the document objects.

### **1.1.3 Intended Audience**

In contradicton of the Software Requirements Specification (that is written for the purpose of client and user), most of this Software Design Description is written for engineers and designers or researchers who want to change and/or extend the existing reference implementation. In this way the Client is not going to be among the intended audience for this document, which is:

- Team
- Supervisor
- Auditors and Reviewers

## 1.2 Definitions

Term	Definition
User	Person interacting with the app
Database	Gathering all the information to be used in the application
SDD	Software Design Description
SRS	A document that entirely defines whole functions of a proposed system and the limitations under which it must operate
Class Diagram	Describes the structure of a system
Use-case Diagrams	Illustrates the relationships between use cases
UML	Standardized modeling language that enables developers to deliver, visualize, configure and determine the structures of a software system
Function	Identifies data elements that form part of the internal entity
Application	Program running on computers or on the phone
IEEE	Institution of Electrical Engineers

### **1.3 Conceptual Model for Software Design Descriptions**

For that document, we are going to use conceptive pattern for the SDD. Conceptive pattern mostly clarifies the source in that SDD has prepared and how it is going to be used. We planned to use multi-layered system architecture. Thus, it is going to be much easier to apply to the project and join feasible future attributes.

#### **1.3.1 Software Design in Context**

We planned to use object-oriented approach and multi-layered system architecture. Thus, we will increase to project implementation, adaptability and portability. Layers will assistant the modularity, safety and adaptability of the software. Using object-oriented design and multi-layered architecture will improve portability and unity among supplementals.

#### **1.3.2 Software Design Descriptions Within the Life Cycle**

The key software life cycle product that drives this software design is the SRS we have prepared. The requirements in the SRS document (interface, functional, nonfunctional and logical database requirements) specify the design of the project.

### **1.4 Design Description Information Content**

#### **1.4.1 Introduction**

Software Design Description of the AppDermis gives information about how AppDermis will be designed and implemented. This chapter is going to be contain information about SDD definition, design partners identified and design concerns, design perspectives, design views, design overlays and design logic selected by type descriptions of allowed design elements and design languages.

#### **1.4.2 SDD Identification**

This document is the initial pattern of the System Design Description for this project. This SDD report was created using Github. The markup of Draw.io and Github is used to draw the diagrams. In the first section an overview of SDD is given. Scope of the SDD report refers to the section 1.1, Purpose of the SDD report refers to section 1.2 and Intended Audience of this document refers to section 1.3. For design conceptual model for software design descriptions refer to the section 3. Lastly, for the design viewpoints including context, composition, logical, information, patterns use, interface, interaction, state dynamics and resource viewpoints refer to the section 5.

### **1.4.3 Design Stakeholders And Their Concerns**

In AppDermis, design stakeholders are the developer team of AppDermis and their advisors. Our design stakeholders are the people who know and understand software development and our stakeholders are the part of the development. Our stakeholders' concerns are listed below:

- The implementation should be safe, secure, maintainability and open to future changes.
- The interface shall be easy to read and use.
- The desired results should be obtained from the developed system.
- Database should be simple and efficient.
- New features will be adapted into AppDermis, so software must be proper for it.

### **1.4.4 Design Views**

In this SDD, for representing the diagrams of view, UML is used. Design views are design rational, contextual, combination, interface, sensible and coaction observes. That coaction observes have clarified in chapter 5.

### **1.4.5 Design Viewpoints**

There are three viewpoints that are used in this project which are context, composition and interaction viewpoints. These viewpoints are explained detailly with UML diagrams in section 5.

- Context perspective describes the relationships, dependencies and interactions between the system and its environment (users, systems, etc.).
- The composition perspective defines how the design subject is structured into the constituent parts and determines the roles of these parts.
- The interaction perspective defines the interaction strategies of why, where, how, and at what level actions take place.

### **1.4.6 Design Elements**

The main design elements are entities, attributes and some other member associated with communication and relations between modules and user of our project. These main design elements are defined inside the related viewpoints in detail in chapter 5.

### **1.4.7 Design Overlays**

The essential factors to explain design choices in appdermis is simplicity and sustainability. The features will add in the future so appDermis is designed with this vision. appDermis developers

document process of development so new developers can understand and modify or add features easily.

#### **1.4.8 Design Rationale**

The software must be designed in a way that future models and features can be added, and current models can be changed and updated independently. The object-oriented design supports classify the objects of the software thus a new object could be lightly added to the design or an existing component can be easily deleted. System developers have to use comments in their code frequently so that other developers can understand the code and the structure of the system in the future. Besides, when writing code, function and variable names are deliberately selected to indicate what they are doing. So, its functionality could be lightly understood in the future. When implementing the system database, tables are created considering both the ER Diagram displayed in Chapter 5 and the class diagram to concur the database and models. Design Languages

In this project, Unified Modeling Language (UML) is selected as a part of design viewpoint and it will be used for clarifying design viewpoints.

### **1.5 Design Viewpoints**

#### **1.5.1 Introduction**

In this section, the perspectives of appDermis are explained in detail. UML diagrams is going to be used (Case Diagrams, Class Diagrams, Sequence Diagrams, State Diagrams etc.). There are 13 opinions for our projects.

- Context viewpoint
- Composition viewpoint
- Logical viewpoint
- Dependency viewpoint
- Information viewpoint
- Patterns use viewpoint
- Interface viewpoint
- Structure viewpoint

- Interaction viewpoint
- State dynamics viewpoint
- Algorithm viewpoint
- Resource viewpoint

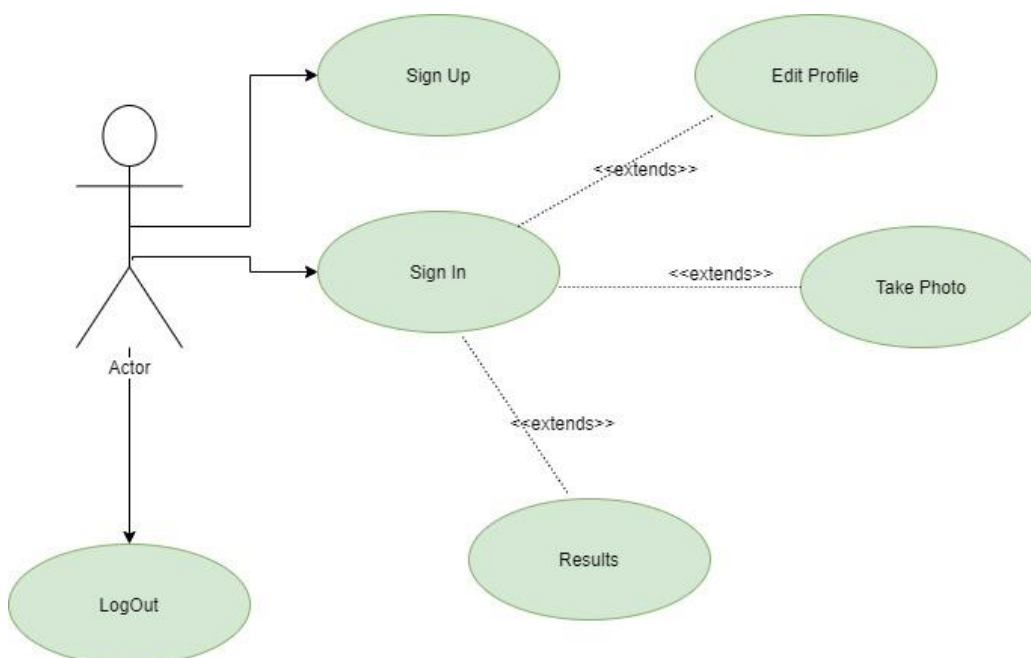
## 1.5.2 Context Viewpoint

### 4.5.2.1 Design Concerns

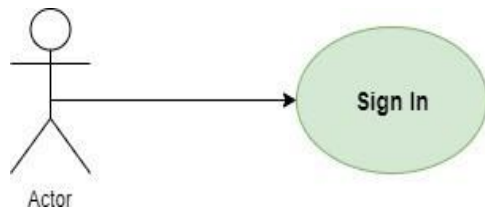
The main service category related to system which is User. Users are people who will use the application.

### 4.5.2.2 Design Elements

User and actions are design entities for application. As you can see, use case diagram is showed with whole functions.



#### 4.5.2.2.1 Sign In

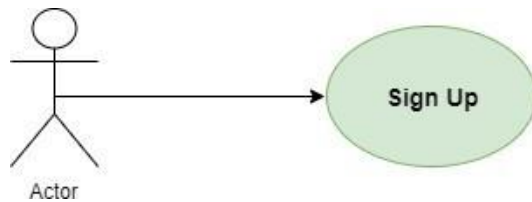


<b>Use Case Number</b>	<b>1</b>
<b>Use Case</b>	Sign In
<b>Summary</b>	User can sign into the application.
<b>Actor</b>	User
<b>Trigger</b>	Sign In Button
<b>Primary Scenario</b>	User will enter the application. If user didn't sign in before, opening page will be SignIn page. After entering email and password, user will be redirected to profile page.
<b>Alternative Scenario</b>	No alternative scenarios.
<b>Exceptional Scenario</b>	Wrong e-mail and password combination or wrong e-mail format.
<b>Pre-Conditions</b>	User must register before trying to sign in to the system.
<b>Post-Conditions</b>	User will be redirected to her/his profile. User can use every function of the system after signing in.



<b>Assumptions</b>	User must be connected to the internet.
--------------------	-----------------------------------------

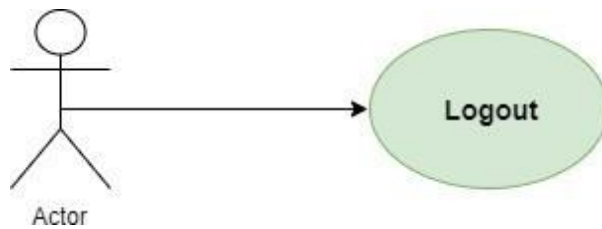
#### 4.5.2.2.2 Sign Up



<b>Use Case Number</b>	2
<b>Use Case</b>	Sign Up
<b>Summary</b>	User can sign up to the application
<b>Actor</b>	User
<b>Trigger</b>	Sign Up Button
<b>Primary Scenario</b>	User will enter the application. In the sign in page there will be a button for sign up page. After filling his/her information, user can sign up if he/she didn't sign up before.
<b>Alternative Scenario</b>	No alternative scenarios.
<b>Exceptional Scenario</b>	Email address is already taken or password doesn't meet the standards.
<b>Pre-Conditions</b>	No pre-conditions.
<b>Post-Conditions</b>	User can sign in to the application.

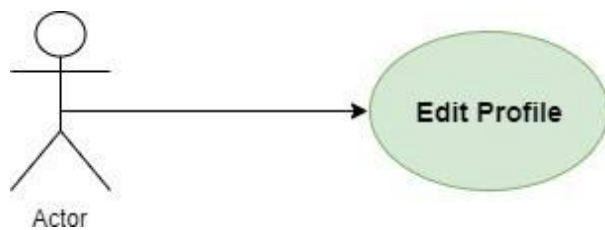
<b>Assumptions</b>	Valid e-mail address.
--------------------	-----------------------

#### 4.5.2.2.3 Logout



<b>Use Case Number</b>	<b>3</b>
<b>Use Case</b>	Logout
<b>Summary</b>	User can logout from the application.
<b>Actor</b>	User
<b>Trigger</b>	Logout button in the Profile.
<b>Primary Scenario</b>	After signing in, user can log out from her/his profile.
<b>Alternative Scenario</b>	No alternative scenarios.
<b>Exceptional Scenario</b>	No exceptional scenarios.
<b>Pre-Conditions</b>	User must sign in to the system.
<b>Post-Conditions</b>	User will be redirected to Sign In page.
<b>Assumptions</b>	User has sign in to the system.

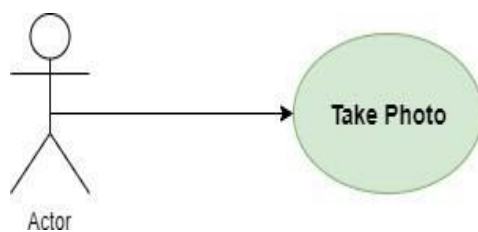
#### 4.5.2.2.4 Edit Profile



Use Case Number	4
Use Case	Edit Profile
Summary	User can edit his/her informations.
Actor	User
Trigger	Edit button
Primary Scenario	User must register then login to system.
Alternative Scenario	None.

<b>Exceptional Scenario</b>	1. Not registered user.
<b>Pre-Conditions</b>	User must login to the system.
<b>Post-Conditions</b>	User will edit the information. User Profile will be updated.
<b>Asumptions</b>	User must be connected to the internet.

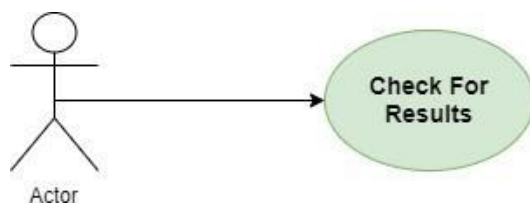
#### 4.5.2.2.5 Take Photo of Lesion



<b>Use Case Number</b>	<b>5</b>
<b>Use Case</b>	Take Photo of Lesion
<b>Summary</b>	User can take photo of suspicious lesion then see the risk range about lesion.
<b>Actor</b>	User
<b>Trigger</b>	Camera button
<b>Primary Scenario</b>	User must login to system.
<b>Alternative Scenario</b>	None.
<b>Exceptional Scenario</b>	1. Not entered user.

<b>Pre-Conditions</b>	User must login to the system and push camera button.
<b>Post-Conditions</b>	Photo is saved in the album then system calculates the cancer risk of the lesion.
<b>Asumptions</b>	None

#### 4.5.2.2.6 Check For Results



<b>Use Case Number</b>	8
<b>Use Case</b>	Check For Results
<b>Summary</b>	User can check for results.
<b>Actor</b>	User
<b>Trigger</b>	User enters the check for results tab and enters the check for results button.
<b>Primary Scenario</b>	User enters the check for results tab.

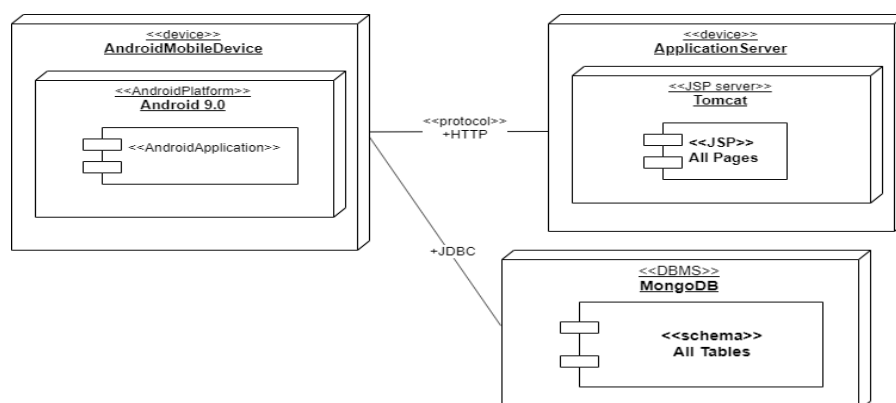
<b>Alternative Scenario</b>	No alternative scenarios.
<b>Exceptional Scenario</b>	If no taken photo taken before, user cannot check for any result.
<b>Pre-Conditions</b>	Result
<b>Post-Conditions</b>	No post conditions.
<b>Assumptions</b>	User checked photo to see for results.

### 1.5.3 Composition Viewpoint

#### 4.5.3.1 Design Concerns

In this part, composition viewpoint will assist software process easily. There are three essential components in this software as you can see 5.3.2 Design Elements Part.

#### 4.5.3.2 Design Elements



- Design Entities: The main components in system; database, application server and android application.
- Application Server: Establishing connection between application and database.
- Design Attributes will discuss next chapters.

## **1.5.4 Logical Viewpoints**

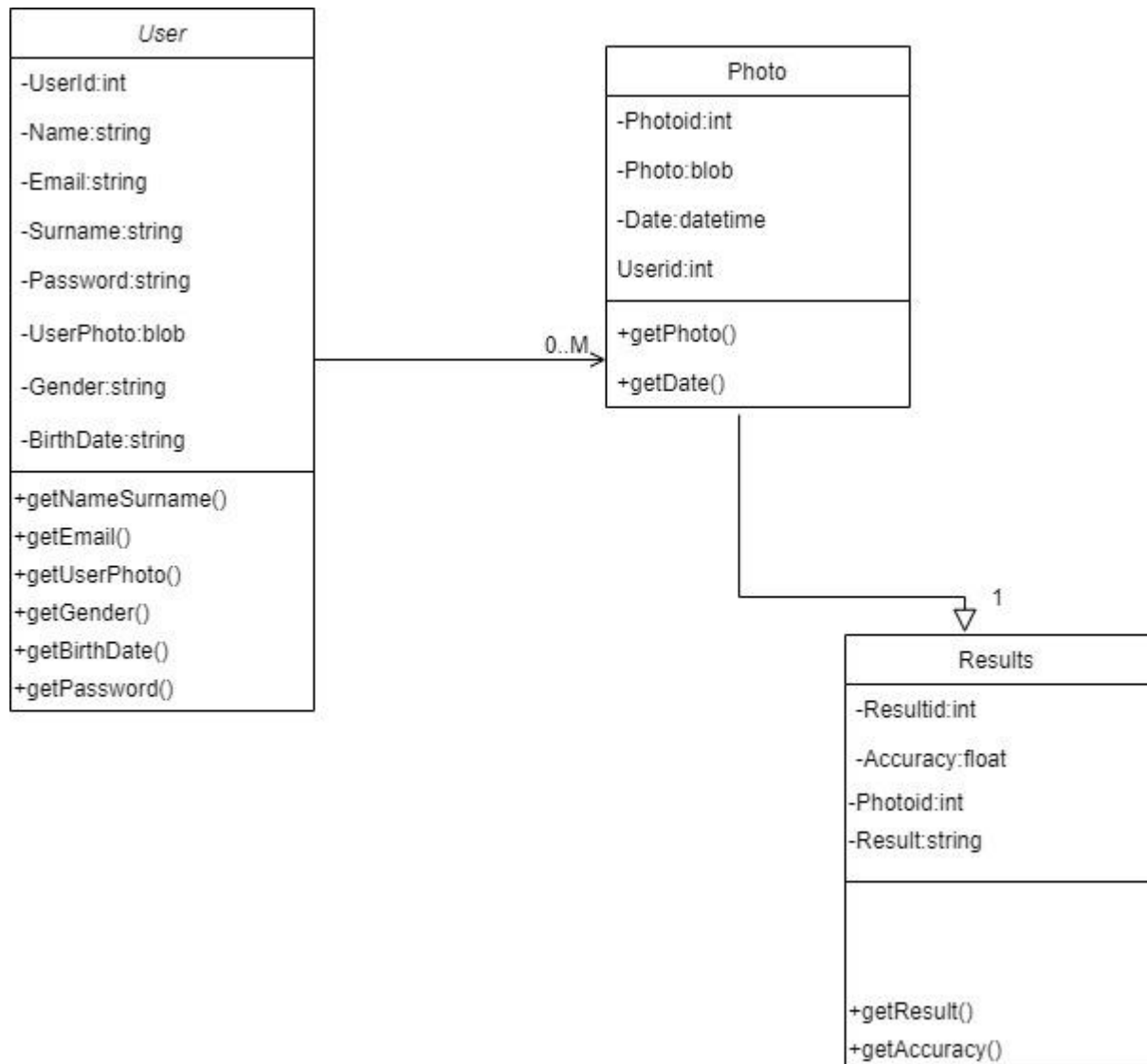
### **4.5.4.1 Design Concerns**

In this part, entire classes and relations are identified in detail. The main purpose is to explain and clarify appDermis system design. Complete class diagram includes classes and their relations. Entire classes and methods/fields are explained.

### **4.5.4.2 Design Elements**

- **5.4.2.1 Class Relations**

There are four classes; user, photo, result and album and they are related to each other. The class diagram includes only essential classes, on the other hand there are trivial classes. However these classes are not shown because of clarity.





#### 4.5.4.2.2 User Class

Method/Field	Definition
<b>int user-id</b>	Unique id for each user.
<b>string name</b>	User's name.

<b>string surname</b>	User's surname.
<b>string email</b>	Email of the user.
<b>string gender</b>	Gender of the user.
<b>string skincolor</b>	User's skin color.
<b>date birthdate</b>	Birthdate of the user.
<b>string getNameSurname()</b>	Returns name and surname
<b>string getSkinColor()</b>	Returns the skin color
<b>string getGender()</b>	Returns the gender.

#### 4.5.4.2.3 Photo Class

Method/Field	Definition
<b>int photoid</b>	Unique id for each photo
<b>blob photo</b>	Photo of a lesion.

#### 4.5.4.2.4 Result Class

Method/Field	Definition
<b>int resultid</b>	Unique id for each result
<b>float accuracy</b>	Accuracy of the results
<b>int photoid</b>	Results are belong to this photo
<b>string algorithmresults</b>	When the all machine learning algorithms run, results will be here.
<b>string getAlgorithmResult()</b>	Returns the results.
<b>float getAccuracy()</b>	Returns the accuracy.

### **1.5.5 Dependency Viewpoint**

Dependency viewpoint explains dependencies and relationships between system design component.

#### **4.5.5.1 Design Concerns**

Defining dependencies of application and determining which subsystems are depends on other subsystems. It helps deciding the priorities in design entities.

#### **4.5.5.2 Design Elements**

Design Entities: Application server, database server and the client. Design Relationships: Each entities are related each other, application server is between the database and client.

#### **4.5.5.3 Example Languages**

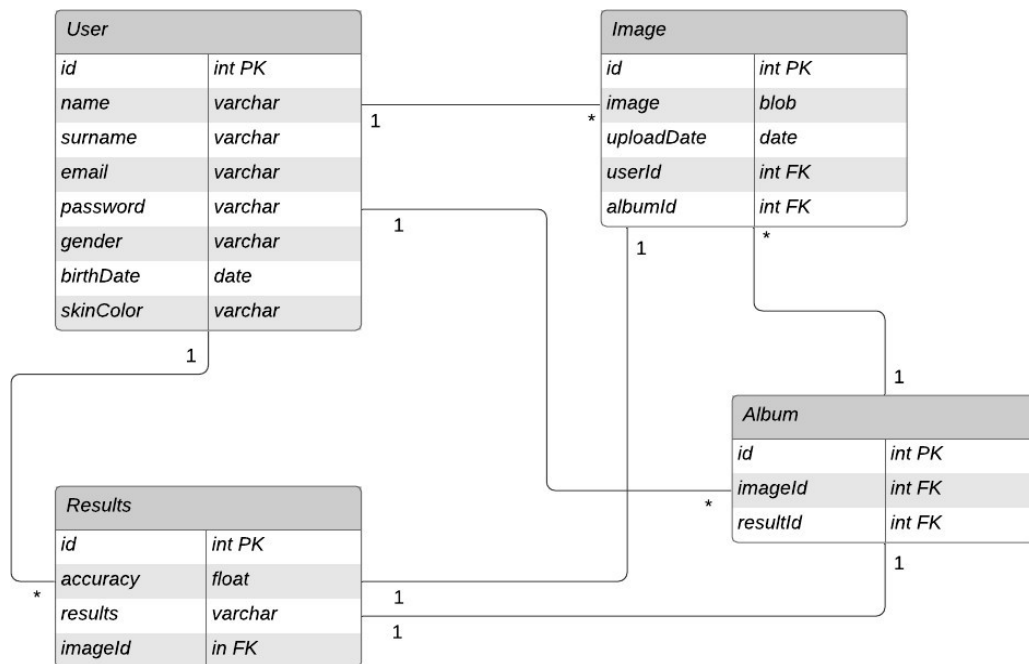
You can see the diagram in section 5.3.2.

### **1.5.6 Information Viewpoint**

#### **4.5.6.1 Design Concerns**

In this section, data management of the application and data structure will be shown. ER diagram is used. Design entities in this diagram are our tables. There are data types, data items, and extra information about data in every table. Design relationships are displayed with connections. Connections between tables show us how these tables are related to each other (one to one or one to many).

## 4.5.6.2 Design Elements



## 1.5.7 Patterns User Viewpoint

In appDermis server-client design pattern is used so the software reusable. The server-client design pattern is consisted two component client and server. The server listens to request from client and provide services.

## 1.5.8 Interface Viewpoint

### 4.5.8.1 Design Concerns

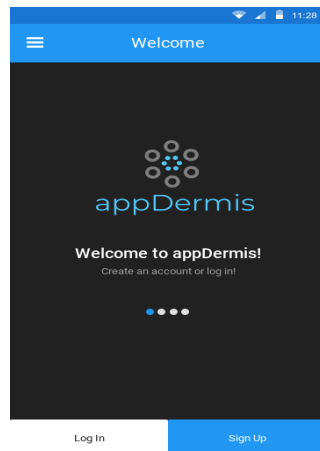
This viewpoint shows users how to use appDermis application. Every interface of the application has its own prototype and description. This will be very helpful for users and developers.

## 4.5.8.2 Design Elements

### 4.5.8.2.1 Welcome Page

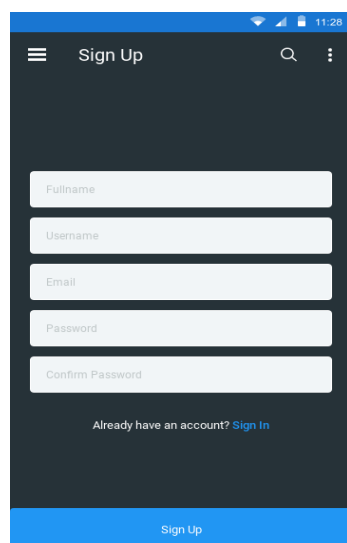
After downloading and entering the application, the user will see this tab first. This page will consists of information about the application. And for the users who want to use the application,

there will be 2 options: Sign up and Sign in. It will be redirected to the next page, depending on the user's choice.



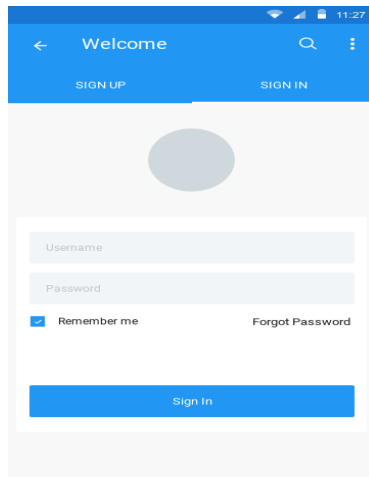
#### 4.5.8.2.2 Sign Up Page

In this page, the user can sign up to the system by entering some personal information. After signing up, the user can edit, change or add information to his/her profile page.



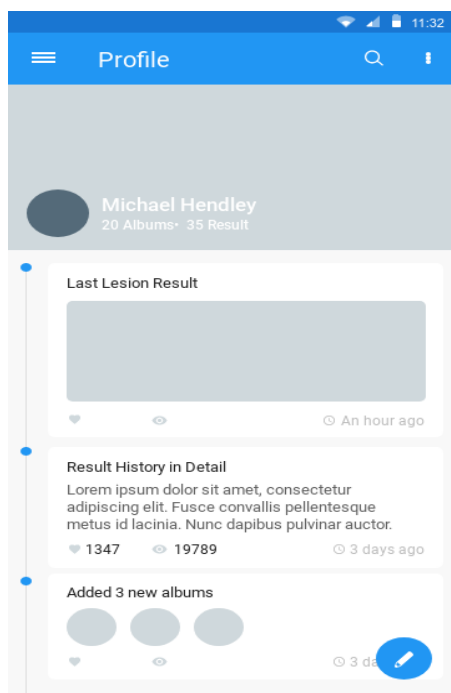
#### 4.5.8.2.3 Sign In Page

In this page, the user can enter the application by entering his/her information if she/he is already signed up.



#### 4.5.8.2.4 Profile

In this page, the user can see personal information, edit information, or add more information to make our analysis better.



#### 4.5.8.2.5 Camera

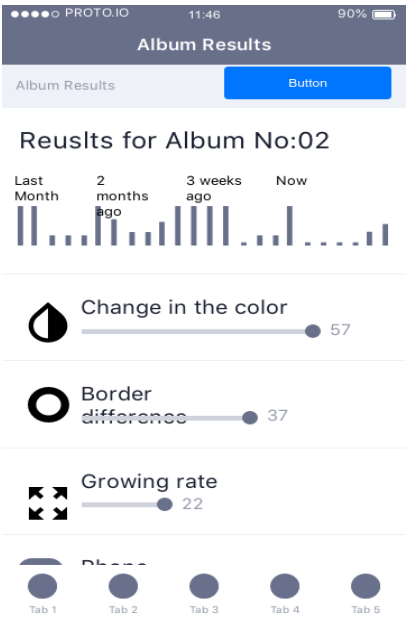
The user will take his/lesion photos in this page. The camera will be activated when the user wants to take a new lesion photo. After every photo is taken, the application will ask user for album to save it.





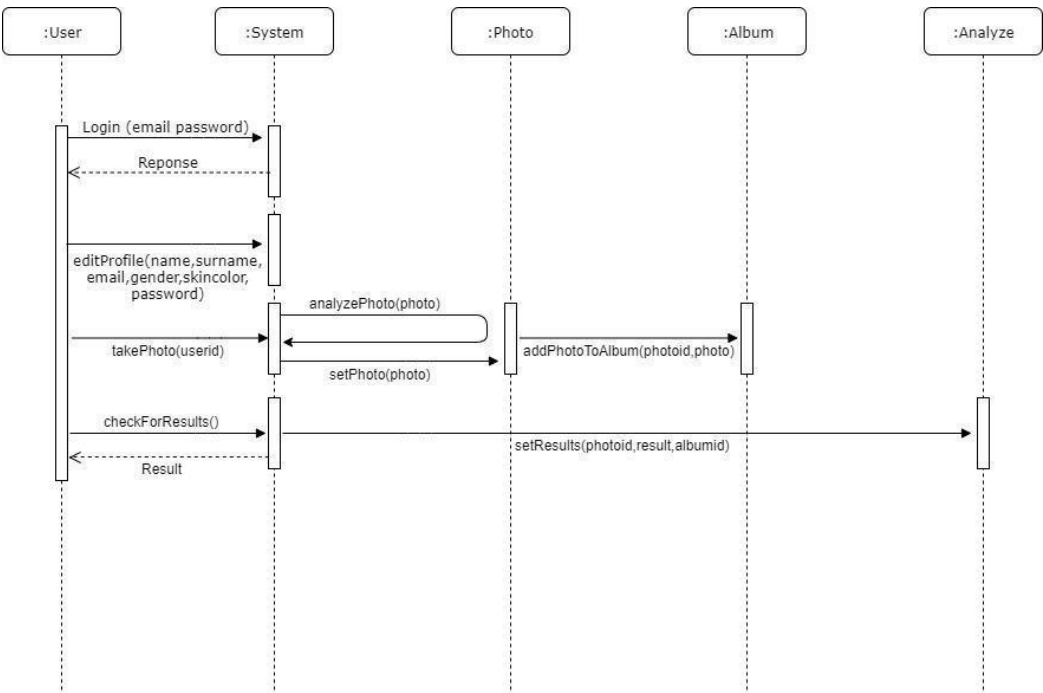
Check for Results

This page contains information about all photos and lesions. Results will be showed in graphs, diagrams and there will be a written explanation. If there is a risky situation, the application will notify the user for a doctor appointment.



1.5.9 Interaction Viewpoint

4.5.9.1 Design Elements



1.5.10 State Dynamics Viewpoint

4.5.10.1 Design Concern

The user starts the application, user is redirected sign up page. User can sign up if he/she has an account. When the user signs up the application, he/she can reach various feature of application.

#### 4.5.102 Design Elements

Design entities and design relationships are observed with the state transition diagram particularly in the section bellow.

