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 usege of building a system the supplies a ground level of functionality to represent applicability for wide scale production usege. 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 usege of building a system the supplies a ground level of functionality to represent applicability for wide scale production usege. 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 delivers, 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 easiliy.

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 cold 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

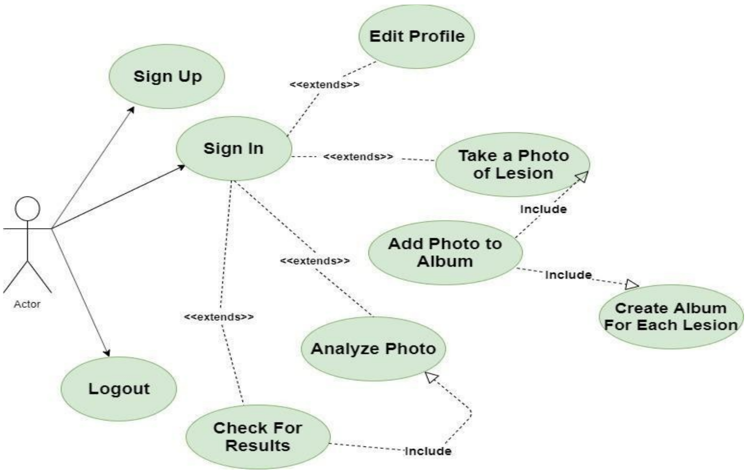
1.5.2 Context Viewpoint

4.5.2.1 Design Concerns

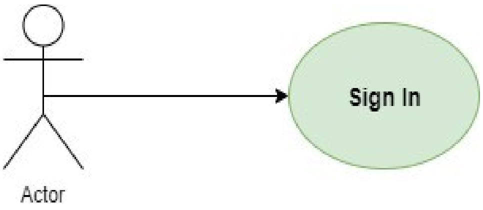
The main service category releated to system which is User. Users are people who will use the application.

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.



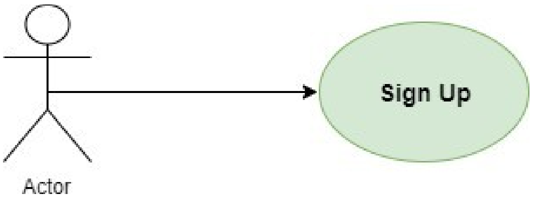
4.5.2.2.1 Sign In



Use Case Number	**1**
Use Case	Sign In
Summary	User can sign in to the application.
Actor	User
Trigger	Sign In Button
Primary Scenario	User will enter the appplication. 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.
Alternative Scenario	No alternative scenarios.
Exceptional Scenario	Wrong e-mail and password combination or wrong e-mail format.
Pre-Conditions	User must register before trying to signing in to the system.
Post-Conditions	User will be redirected to her/his profile. User can use every function of the system after signing in.

Assumptions	User must be connected to the internet.

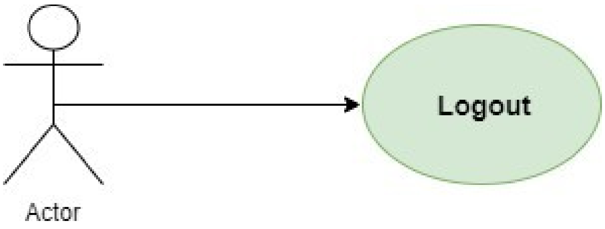
4.5.2.2.2 Sign Up



Use Case Number	**2**
Use Case	Sign Up
Summary	User can sign up to the application
Actor	User
Trigger	Sign Up Button
Primary Scenario	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.
Alternative Scenario	No alternative scenarios.
Exceptional Scenario	Email address is already taken or password dosen't meet the standards.
Pre-Conditions	No pre-conditions.
Post-Conditions	User can sign in to the application.

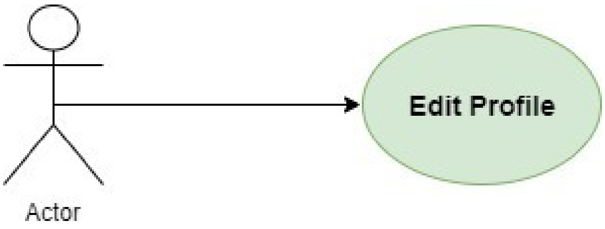
Assumptions	Valid e-mail address.

4.5.2.2.3 Logout



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

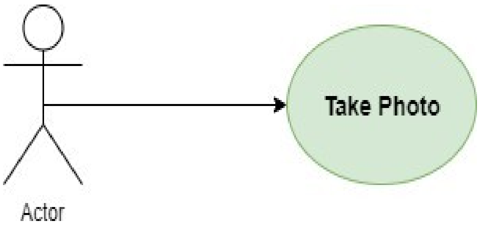
4.5.2.2.4 Edit Profile



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.

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

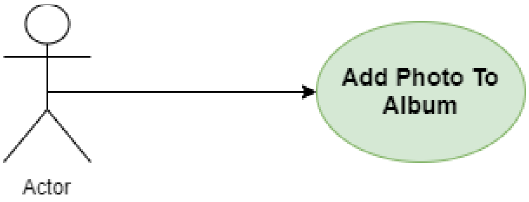
4.5.2.2.5 Take Photo of Lesion



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

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

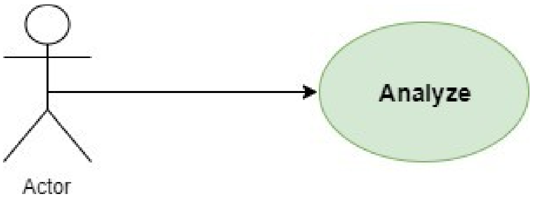
4.5.2.2.6 Add Photo To Album



Use Case Number	**6**
Use Case	Add Photo To Album
Summary	Every lesion has its own album. User can photos to album.
Actor	User
Trigger	Add To The Album button appears when a new photo is taken.
Primary Scenario	User signs in to the application. Every lesion has its own album. After taking a new photo, user can add photo to its album.
Alternative Scenario	No alternative scenarios.

Exceptional Scenario	No albums was created before for that lesion. In this case application will create a new album.
Pre-Conditions	User must take the photo of the lesion. There must be an album to save the photo.
Post-Conditions	User will be redirected to the camera.
Assumptions	User added right photo to right album.

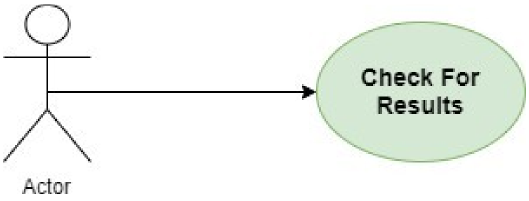
4.5.2.2.7 Analyze Lesions



Use Case Number	**7**
Use Case	Analyze Lesions
Summary	If user wants to make a comparison, user must have previous photos of the lesion in the album or must take a new photo to analyze.
Actor	User
Trigger	Analyze button
Primary Scenario	User must take photo or have previous photos.
Alternative Scenario	User accesses the album then select a photo to analyze.

Exceptional Scenario	None.
Pre-Conditions	1. Login is required. 2. User must access analyze lesion part by button.
Post-Conditions	None.
Asumptions	None.

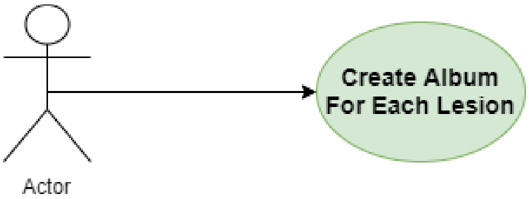
4.5.2.2.8 Check For Results



Use Case Number	**8**
Use Case	Check For Results
Summary	User can check for analyze results.
Actor	User
Trigger	User enters the check for results tab and enters the check for results button.
Primary Scenario	User enters the check for results tab. Results will be showed in 2 different ways. User can check for album results to see the development of a lesion or check for only one photo results. Also user can see the history of album.

Alternative Scenario	No alternative scenarios.
Exceptional Scenario	If no analysis done before,user can not check for any result.
Pre-Conditions	Analyze
Post-Conditions	No post conditions.
Assumptions	User analyzed at least 1 photo to check for results.

4.5.2.2.9 Create Album For Each Lesion



Use Case Number	**9**
Use Case	Create Album For Each Lesion
Summary	User must create an album for each lesion to see the development of that lesion.
Actor	User
Trigger	Create an album button at the Albums tab.
Primary Scenario	User enters the Albums tab. Pushes the Create an Album button.
Alternative Scenario	User can create a new album for a new photo after taking the photo if there is no album created for that lesion.

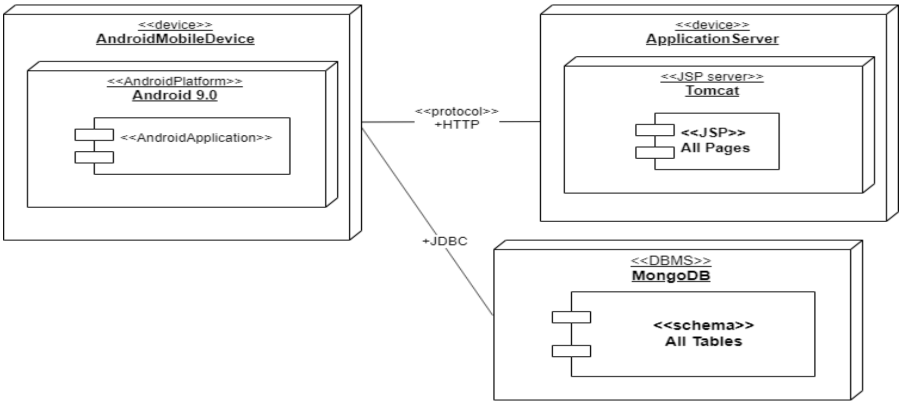
Exceptional Scenario	No exceptional scenarios.
Pre-Conditions	User must login.
Post-Conditions	After creating the album, user can add photos to the album.
Assumptions	Individual albums created for same lesion's photos.

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

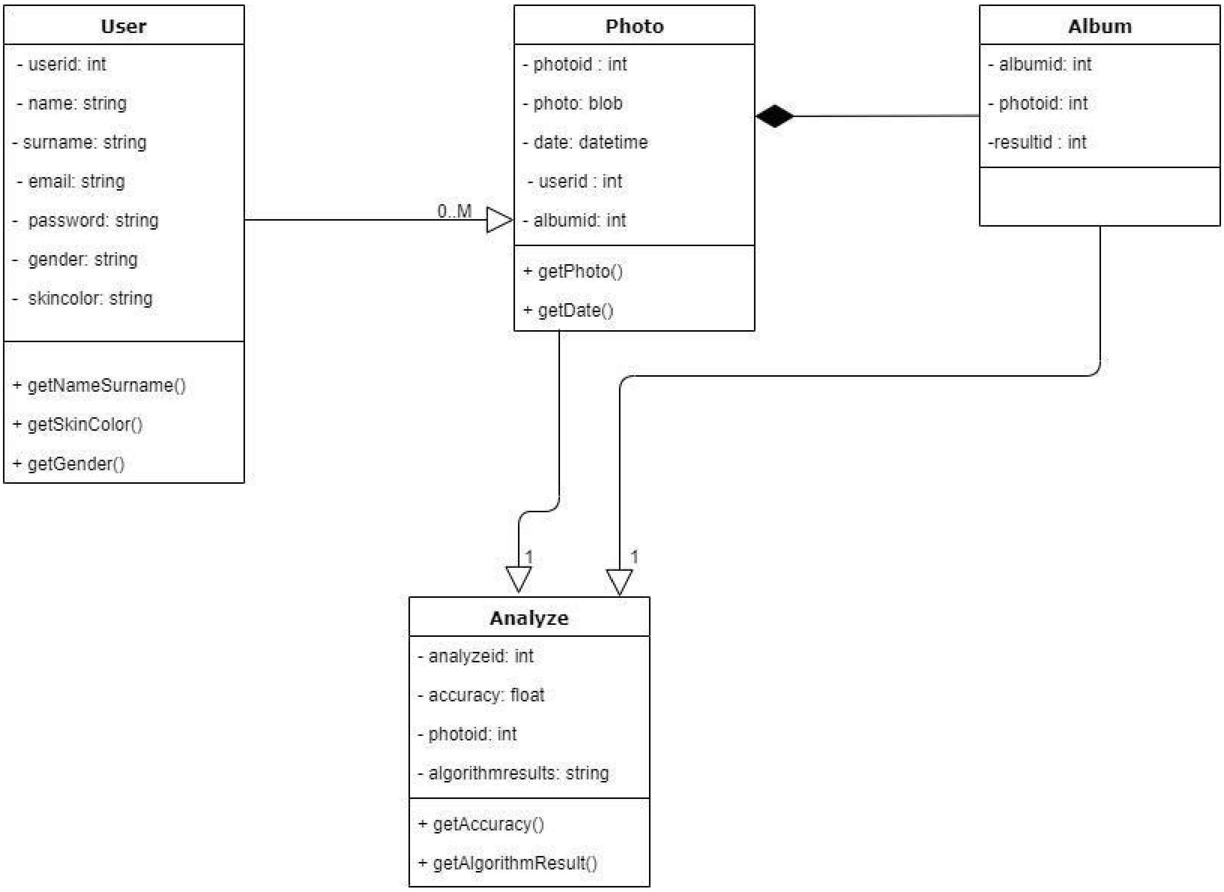
4.5.4.1 Design Concerns

In this part, enire 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, analyze 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
int user-id	Unique id for each user.
string name	User's name.

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

4.5.4.2.3 Photo Class

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

date datetime	Date of the photo.
int userid	Photo belongs to this user.
int albumid	Photo belongs to this album.
blob getPhoto()	Returns the photo.
date getDate()	Returns the date for that photo.

4.5.4.2.4 Analyze Class

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

4.5.4.2.5 Album Class

Method/Field	**Definition**
int albumid	Unique id for each album

int photoid	Photo id of the photo that belongs to this album
int resultid	Result id of the result that belongs to this album

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

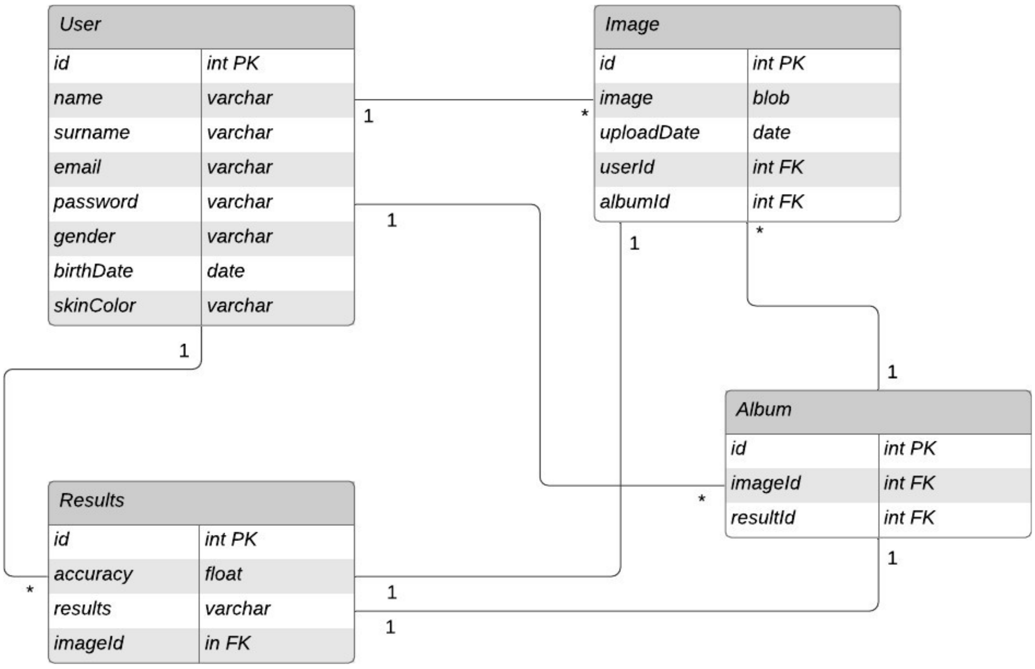
You can see the diagram in section 5.3.2.

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



User	
id	int PK
name	varchar
surname	varchar
email	varchar
password	varchar
gender	varchar
birthDate	date
skinColor	varchar

Image	
id	int PK
image	blob
uploadDate	date
userId	int FK
albumId	int FK

Album	
id	int PK
imageId	int FK
resultId	int FK

Results	
id	int PK
accuracy	float
results	varchar
imageId	in FK

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

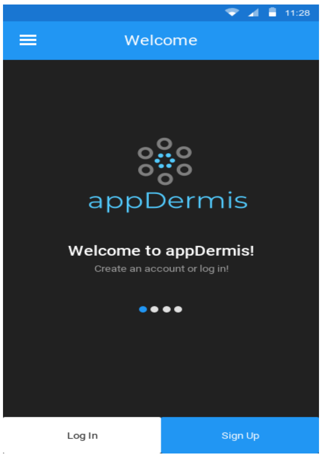
4.5.8.1 Design Concerns

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.

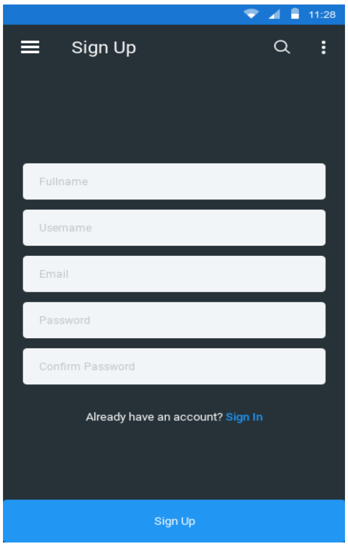


Welcome to appDermis!

Create an account or log in!

4.5.8.2.2 Sign Up Page

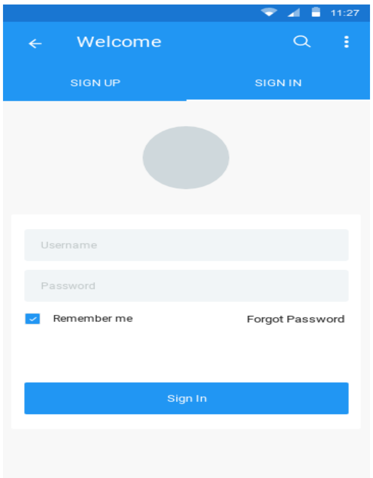
In this page, the user can sign up to the system by entering some personal information. After signing up, the user can edit, change or add information to his/her profile page.



Already have an account? Sign In

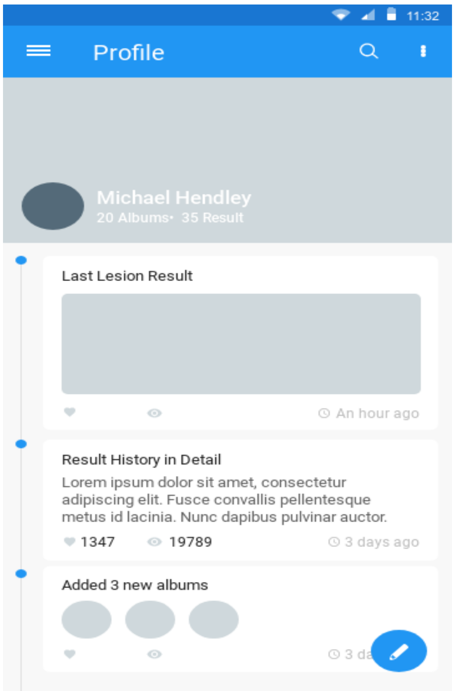
4.5.8.2.3 Sign In Page

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

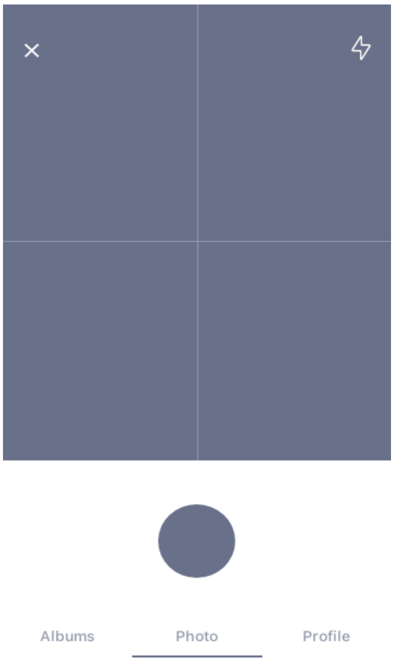
In this page, the user can see personal information, edit information or add more information to make our analysis better. Also, there will be an analysis history for the user's lesions.



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Fusce convallis pellentesque metus id lacinia. Nunc dapibus pulvinar auctor.

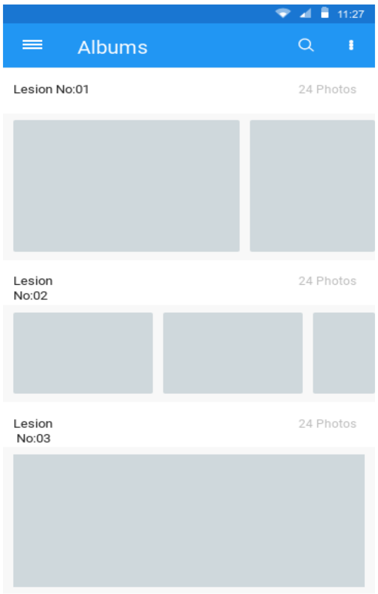
4.5.8.2.5 Camera

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 which album to save it.



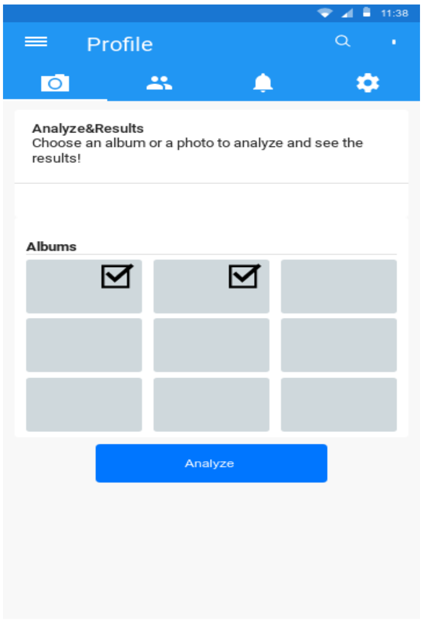
4.5.8.2.6 Albums

Every lesion will have a specific album. So we can check how much it has changed. And every album will store statistical information about the lesion. When you click the album, you will be able to see information and lesion photos.



4.5.8.2.7 Analysis

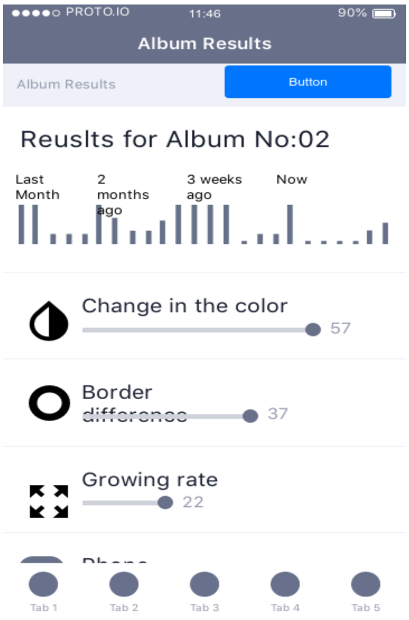
In this page, users can choose one of the albums or one of the lesion photos and analyze them. After clicking the analyze button, user will be redirected to the results page.



Analyze&Results
Choose an album or a photo to analyze and see the results!

4.5.8.2.8 Check For Results

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



Reuslts for Album No:02

Change in the color 57

Border difference 37

Growing rate 22

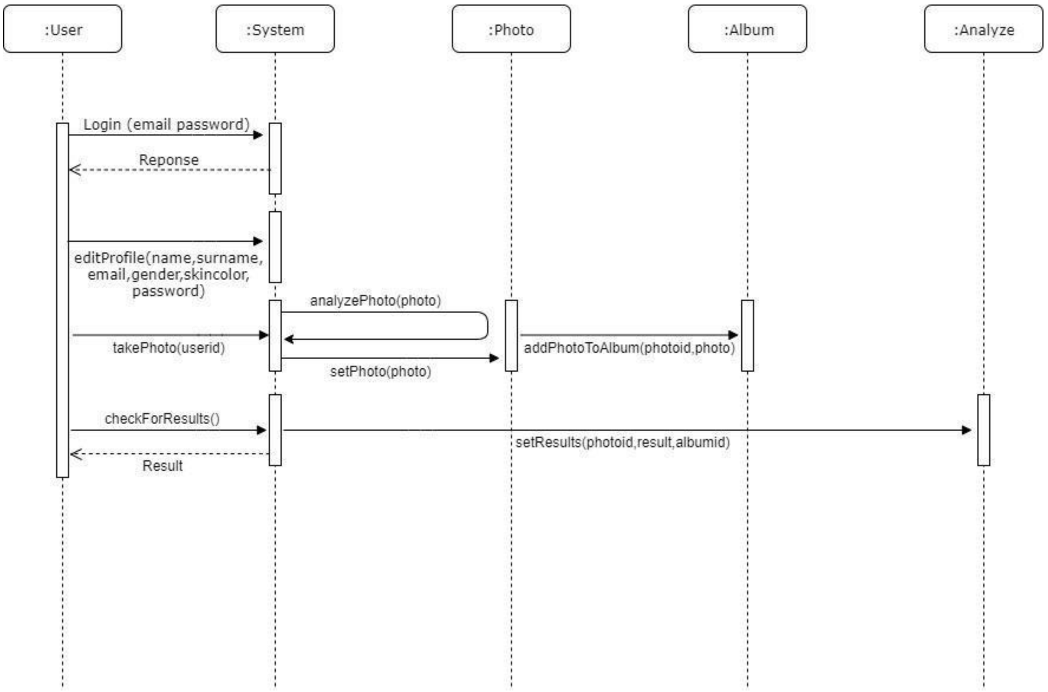
Phone

1.5.9 Interaction Viewpoint

4.5.9.1 Design Concerns

Sequence Diagram is used to show communication between objects.

4.5.9.2 Design Elements



1.5.10 State Dynamics Viewpoint

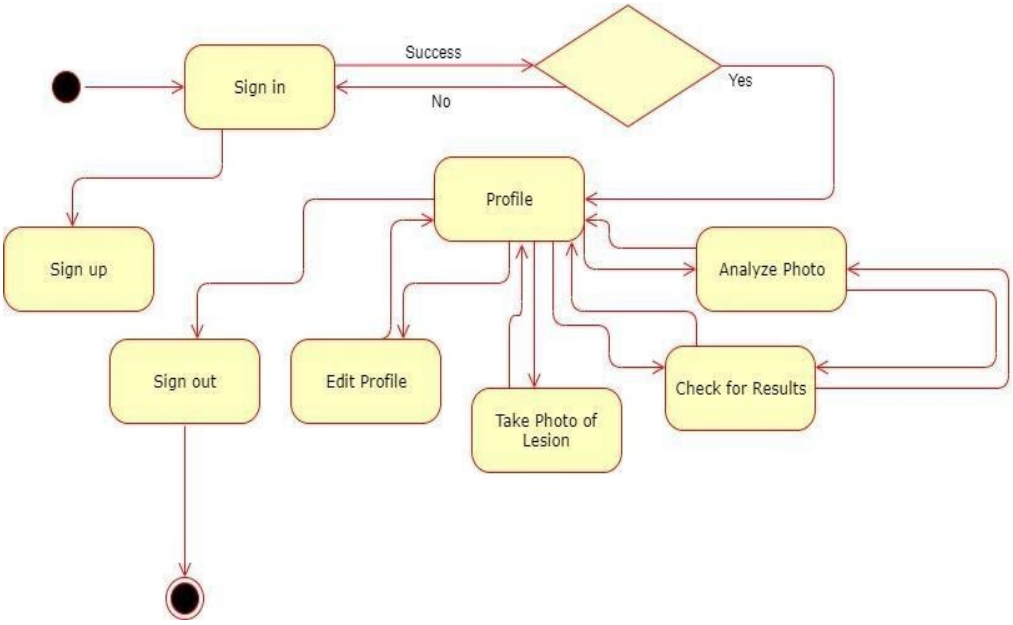
4.5.10.1 Design Concern

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.

4.5.10.2 Design Elements

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

4.5.10.3 Example Languages



1.5.11 Algorithm Viewpoint

Algorithm viewpoint is not available.

1.5.12 Resource Viewpoint

Resource viewpoint is not available.

2. Conclusions

In this 407 project, we have tried to explain what appDermis will be and how we will help the people who has suspicious about melanoma skin cancer. Our aim is building an app for being guide to people who has think that its mole can be melanoma risk so this way we can reduce the risk of deadly results and of course helping to create conscious society. Our researches showed us we can take best results with using SVM classification and CNN for deep learning algorithm. We will take a picture of mole and our algorithm will show to user the risk of melanoma of that mole and user can be able to create his or her own album about specific mole so when user wants to check is there anything change, like being bigger or changing colour of that mole, user will observe it.