# **App for Diagnosing Melanoma**

# **Christopher G. Tscheschlog, Reeve W. Blake, Kaden C. Carr, Ayleen Roque**

## **Draft: 2.71**

## **2022-11-2**

## **EGS 3045C Perspectives on Engineering Challenges**

## **Instructor: Dr. Janusz Zalewski**

## **Whitaker College of Engineering**

## **Florida Gulf Coast University**

## **Ft. Myers, FL 33965**

1. **Introduction**

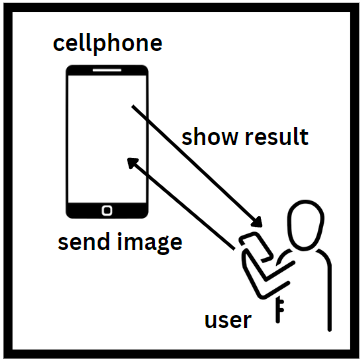
**1.1 - Project Objectives**

The purpose of this application is to simplify the process for worried individuals to check if they’re at risk of Melanoma.

**1.2 - Discussion**

Melanoma is one of the most common cancers in young adults under thirty, yet it goes undiagnosed until after their fifties [1]. The longer it is undiagnosed, the more it spreads, and the more difficult it is to treat. In an effort to help individuals at risk detect melanoma earlier to increase their survival rate, we have created an app to help them detect it with simple steps from their phone.

**1.3 - Physical Diagram**



User sends image into the app which will then calculate and show the melanoma possibility result

**1.4 - Technologies**

The technologies used in this project include Python [3], Javascript [4], CSS [6], HTML [5], Node.js [7], Computer Vision [8], SQLAlchemy [9] and SQLite [10], Flask[13], React native for the frontend of the app [11], AWS Lambda [12] and API Gateway [14] for hosting.

1. **Requirements Specification**

**2.1 - Project Objectives**

* Shall have a cross-platform mobile app.
* Shall utilize machine learning to determine melanoma results.
* Shall present the user with positive or negative results for melanoma.
* Shall have a diagnostics server to process the image.
* Shall return a percent confidence level.

**2.2 - Input / Output Requirement**

1. Input
   1. The mobile app(user) shall be able to take pictures from the camera.
   2. The mobile app(user) shall have access to camera roll.
   3. The mobile app(user) shall post images to the diagnostics server.
2. Output
   1. The diagnostics server shall respond to the mobile app with the probability of melanoma.
   2. The diagnostics server shall respond to the mobile app with a positive or negative result based on the model confidence.
   3. The diagnostics server shall store information related to past test results.
   4. The diagnostics server shall send past results to the mobile app.

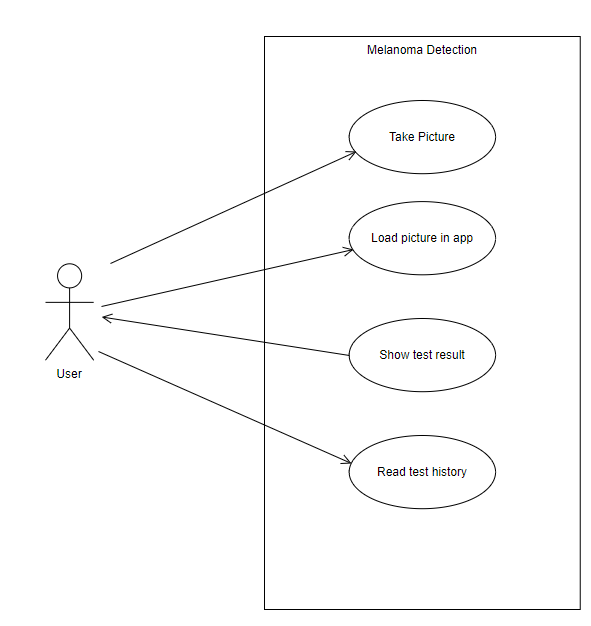
**2.3 - Functional Requirements**

* The mobile app shall have a navigation menu which displays the Test/Submission page, the Results History page
* The mobile app shall have a page for submitting pictures to the diagnostics server.
* The mobile app’s submission page shall have a button for taking pictures.
* The mobile app’s submission page shall have a button for searching camera roll.
* The mobile app’s submission page shall have a display of the current picture.
* The mobile app’s submission page shall have a button for sending a picture to the diagnostics test.
* The mobile app shall have a page for viewing the latest result of the diagnostics test (Result Page).
* When the test result comes from the diagnostics server the user shall be redirected to the latest result page.
* The mobile app’s results page shall have a button for viewing previous test results(redirects to Result’s History page).
* The mobile app’s results page shall display the current result and a list of past results(React component).
* The mobile app shall have a Result’s History Page.
* The mobile app’s result history page shall contain a scrollable list of previous results with a date.
* The diagnostics server shall contain latest test result on https://address/api/result
* The diagnostics server shall contain test results on https://address/api/testresults
* The mobile app(user) shall be able to upload photos to the diagnostics server.
* The mobile app(user) shall be able to view results when their image is processed.
* The mobile app(user) shall be able to view the history of diagnostics results.

**2.4 - Non-Functional Requirements**

* The melanoma prediction model shall be higher than 70% accuracy.
* Users shall receive a passing test score at above a 70% accuracy.
* The user shall receive their percentage confidence back from the model.
* The server shall be hosted on AWS Lambda and API Gateway.
* The API dependencies shall be compatible with Python version 3.9.

**2.5 - Use case diagram**



**2.6 - Design Constraints**

Prototype: <https://www.figma.com/file/lkDe6lec1KBYrJ60Zhkt5X/Design-for-Melanoma-Mobile-App?node-id=0%3A1>

Constraints include having access to a mobile phone with a working camera or, at the bare minimum, stored images in a photo gallery, and an internet connection. API should be created with the Python 3.9 programming language(lambda compatibility).

* Programming Language
  + The API shall use Python version 3.9 [3]. The mobile app shall use JavaScript [4], Css [6], Html [5], JSX [11].
* Operating System
  + The user's iOS device shall require iOS 7.0 or newer.
  + The user’s Android device shall require Android 4.1 (API 16) or newer.
* Hardware Platform
  + Apple A6 chip or newer
  + Processor
    - x86 or x64
  + RAM
    - 512 MB (minimum)
  + Hard Disk
    - 3 GB of free space

1. **References**

[1] American Cancer Society, Key Stats for Melanoma Skin Cancer. American Cancer Society. Atlanta, Ga. 2022, <https://www.cancer.org/cancer/melanoma-skin-cancer/about/key-statistics>

[2] Noel C. F Codella, David Gutman, M. Emre Celebi, Brian Helba, Michael A. Marchetti, Stephen W. Dusza, Aadi Kalloo, Konstantinos Liopyris, Nabin Mishra, Harald Kittler, Allan Halpern, Melanoma Detection Dataset, 2020, [https://www.kaggle.com/datasets/wanderdust/skin-lesion-analysis-toward-melanoma-detection](https://www.kaggle.com/datasets/wanderdust/skin-lesion-analysis-toward-melanoma-detection/code)

[3] Python Software Foundation, Python Overview, 2019-09-18, <https://wiki.python.org/moin/BeginnersGuide/Overview>

[4] Mozilla Foundation, Javascript, 2022-09-13, <https://developer.mozilla.org/en-US/docs/Web/javascript>

[5] Mozilla Foundation, HTML: HyperText Markup Language, 2022-09-13 <https://developer.mozilla.org/en-US/docs/Web/HTML>

[6] Mozilla Foundation, CSS: Cascading Style Sheets, 2022-09-13 <https://developer.mozilla.org/en-US/docs/Web/CSS>

[7] OpenJS Foundation, About Node.JS, <https://nodejs.org/en/about/>

[8] The International Business Machines Corporation, What is Computer Vision?, <https://www.ibm.com/topics/computer-vision>

[9] Michael Bayer, SQLAlchemy, <https://www.sqlalchemy.org/>

[10] Hipp, Wyrick & Company, Inc, What is SQLite?, 2022-09-29, <https://www.sqlite.org/index.html>

[11] Meta, Introduction, 2022-09-22, <https://reactnative.dev/docs/getting-started>

[12] Amazon Web Services Inc, What is Lambda?, <https://docs.aws.amazon.com/lambda/latest/dg/welcome.html>

[13] Armin Ronacher, Flask, <https://palletsprojects.com/p/flask/>

[14] Amazon Web Services Inc, What is Amazon API Gateway?, <https://docs.aws.amazon.com/apigateway/latest/developerguide/welcome.html>

| Student Name | Percentage | Remarks (NOT optional) |
| --- | --- | --- |
| Kaden C. Carr | 25% | Great Assignment |
| Reeve Blake | 25% |  |
| Ayleen Roque | 25% |  |
| Christopher G. Tscheschlog | 25% |  |