

**بسم الله الرحمن الرحيم**

Islamic University – Gaza  
Faculty of Information Technology  
Department of Software Development  
and Computer Science



الجامعة الإسلامية – غزة  
كلية تكنولوجيا المعلومات  
قسم تطوير البرمجيات وعلم الحاسوب

**PalAST**  
**A Mobile Application for Automatic Antimicrobial Susceptibility Testing**

A Graduation Project Presented to the  
Faculty of Information Technology, the Islamic University of Gaza  
in Partial Fulfilment for the Degree of Bachelor of Information Technology

By

Ayah Abdelrahim Elshawa – 220191521  
Ghadeer Abedelazeem Alhayek – 220191207  
Malak Nabeel Ghabayen - 220190672  
Raghad Majed Alkhattib – 220190759

Advisor  
Prof. Iyad El Agha

*June 2023*

## Abstract

Antibiotic resistance is the ability of bacteria to resist the effects of antibiotics, making infections more difficult to treat and increasing the risk of complications and death. One way to fight antibiotic resistance is by identifying the most effective antibiotics for treating bacterial infections. This can be done through a laboratory test called AST, which is used to determine the susceptibility of bacteria to antibiotics. However, manual AST has several limitations that include time delay, limited accuracy, limited testing capacity, and subjective interpretation of results. Therefore, there is an emergent need for a more reliable and efficient alternative to manual AST.

Recently, few works have tried to automate disk diffusion AST through AI-based solutions and mobile applications. However, these works do not support advanced analysis and interpretation of results, do not present evaluation of detection performance, or are not publicly available to download and use.

This work proposes PalAST, a cross-platform mobile application that supports automated disk diffusion AST. The application enables biologists to take AST photos and analyse them in real time with minimal human intervention. It uses image processing and a pre-trained machine learning model to detect antibiotic disks in the agar plate and predict bounding circles for inhibition zones. Then, it provides an interpretation of results including the diameters of the inhibition zones, the labels on the antibiotic disks, and the rating of the bacteria as susceptible, intermediate, or resistant to each antibiotic. PalAST also stores the results of tests, allowing users to access and review past test results.

PalAST was tested using a number of real AST photos, and the detection performance was evaluated by using common metrics, i.e. precision, recall, and Intersection over Union. We also used expert evaluation through a questionnaire to assess the PalAST usability and ease of use.

## **DEDICATION**

*To our supportive parents,*

*To our beloved and generous families,*

*To our Palestinian People who suffer unknowingly,*

*To our motherland Palestine,*

*To all...*

*with our eternal love and gratitude*



## **ACKNOWLEDGEMENT**

First and foremost, all praise of gratitude and thankfulness are due to the Almighty Allah for enabling us to complete this work, and peace and blessing of Him be upon His Messenger Mohammad, who said, "Whoever does not thank people (for their favors) is not thankful to the Almighty God." This study would not have been possible without the support and assistance of several dedicated people. We would like to thank them, and We ask Allah to reward them on our behalf.

First, we would especially like to thank our supervisor, Dr. Iyad El Agha, for taking time to encourage, support and give suggestions that helped us finalize this study. We simply could not wish for more encouraging and understanding supervisor. Thank you for believing in us and for helping us to reach this point.

Our deepest thanks go to our families and especially to our parents, who supported us in every step of our academic journey. Their unwavering love and belief in us have sustained us through the challenges of this journey.

Lastly, we would like to thank the staff of our Information Technology department in our beloved IUG. It has been a wonderful experience through the past four years to meet new people, and learn new things that made this research come to life

# TABLE OF CONTENTS

	Page
<b>ABSTRACT</b>	<b>i</b>
<b>DEDICATION</b>	<b>ii</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>LIST OF TABLES</b>	<b>vi</b>
<b>LIST OF FIGURES</b>	<b>vi</b>
<b>CHAPTER 1: INTRODUCTION.....</b>	<b>1</b>
1.1 AST Background.....	2
1.2 Terminology .....	4
1.3 Statement of the problem .....	5
1.4 Objectives.....	5
1.4.1 Main objective .....	5
1.4.2 Specific objectives .....	5
1.5 Importance of the project .....	6
1.6 Scope and limitations of the project.....	6
1.6.1 Scope .....	6
1.6.2 Limitations .....	6
1.7 Methodology .....	7
1.7.1 Software Methodology .....	7
1.7.2 Procedure .....	7
1.8 Tools, equipment, and methods.....	9
1.8.1 Tools .....	9
1.8.2 Equipment .....	10
1.9 Timetable.....	10
<b>CHAPTER 2: RELATED WORKS .....</b>	<b>13</b>
2.1 AntibiogramJ .....	14
2.2 Kadlec, et al. .....	15
2.3 Baltekin, et al. .....	15
2.4 Burg, et al. .....	16
2.5 Pascucci, et al. .....	16
2.6 Panpradist, et al. .....	17
2.7 Croxatto, et al. .....	17
2.8 Antibiogram .....	18
2.9 Antibiotic Guide.....	18

<b>CHAPTER 3: ANALYSIS .....</b>	<b>21</b>
3.1 Requirements Specification.....	22
3.1.1 Functional Requirements .....	22
3.1.2 Non-Functional Requirements .....	24
3.2 Use Case Diagram.....	26
3.3 Use Cases' Description .....	26
3.4 Sequence Diagram.....	29
3.5 User Interface (UI) Sketches .....	31
<b>CHAPTER 4: DESIGN .....</b>	<b>34</b>
4.1 Architecture.....	35
4.2 API Documentation.....	36
4.3 DB Schema.....	37
4.4 Mobile App .....	38
<b>CHAPTER 5: IMPLEMENTATION .....</b>	<b>40</b>
5.1 The Server Side of PalAST .....	41
5.2 The Client Side of PalAST .....	43
<b>CHAPTER 6: TESTING AND EVALUATION .....</b>	<b>44</b>
6.1 Evaluation of the detection performance .....	45
6.1.1 Data Collection.....	45
6.1.2 Evaluation Metrics .....	46
6.1.3 Results and Discussion .....	47
6.2 Expert evaluation of PalAST usability and ease of use .....	50
<b>CHAPTER 7: CONCLUSION AND FUTURE WORK.....</b>	<b>52</b>
<b>BIBLIOGRAPHY .....</b>	<b>54</b>

## List of Figures

<b>Figure 1</b> - Antimicrobial Susceptibility Testing .....	3
<b>Figure 2</b> - The Development Stages .....	7
<b>Figure 3</b> - AntibiogramJ Program Flow.....	14
<b>Figure 4</b> - Microfluidic Chip Design.....	15
<b>Figure 5</b> - Multipath with Automated Analyzer.....	16
<b>Figure 6</b> - OLA-Simple Process.....	17
<b>Figure 7</b> - Use Case Scenario .....	26
<b>Figure 8</b> - Sequence Diagram .....	31
<b>Figure 9</b> - App UI Sketches .....	33
<b>Figure 10</b> - High-Level System Design.....	35
<b>Figure 11</b> - Database Schema .....	37
<b>Figure 12</b> - Screenshots of sample screens from PalAST .....	39
<b>Figure 13</b> - Image Analysis in PalAST.....	43
<b>Figure 14</b> - Photos from the Tested Dataset.....	46
<b>Figure 15</b> - Examples of inaccurate detection of inhibition zones .....	48

## List of Tables

<b>Table 1</b> - Stage Allocation .....	11
<b>Table 2</b> - Comparison between works that tackle automated AST.....	19
<b>Table 3</b> - Use Case Scenarios Explanation .....	26
<b>Table 4</b> - Register Use Case.....	27
<b>Table 5</b> - Login Use Case.....	27
<b>Table 6</b> - Create New Test Use Case .....	27
<b>Table 7</b> - Export Results Use Case.....	28
<b>Table 8</b> - View Past Results Use Case .....	28
<b>Table 9</b> - Confusion matrix of the zone detection process.....	49
<b>Table 10</b> - Evaluation results of the zone detection process .....	49
<b>Table 11</b> - Confusion matrix of the antibiotic label detection process .....	49
<b>Table 12</b> - Evaluation results of the antibiotic label detection process.....	49
<b>Table 13</b> - Feedback received from experts. ....	51



# **CHAPTER 1:**

# **INTRODUCTION**

# CHAPTER 1

## INTRODUCTION

The misuse of antibiotics contributes to the development of a global health concern known as antimicrobial resistance. Although AST is used to check for antibiotic resistance in bacterial infections, it faces strong criticism for inter-operator variability and the difficulty of the interpretive reading process. In this chapter, we will discuss AST, and the main reasons as to why we chose to make this application.

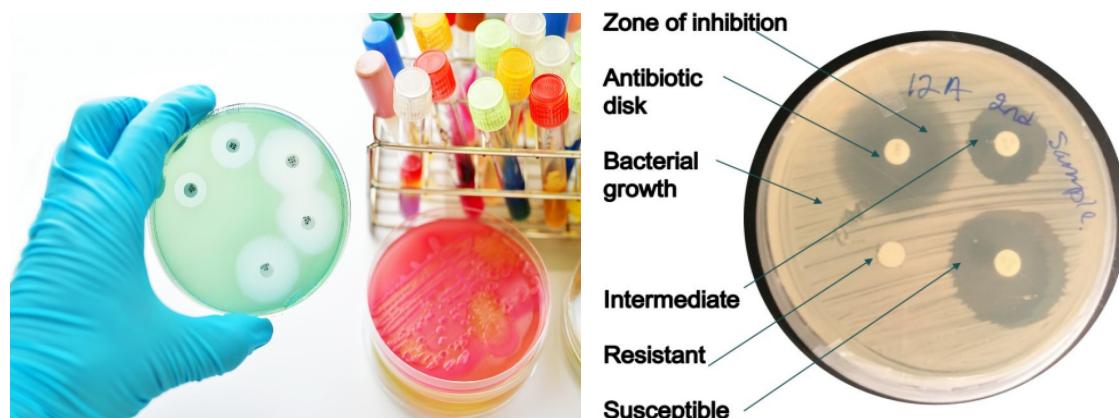
### 1.1. AST Background

Antibiotic resistance is a growing problem worldwide. It occurs when bacteria develop mechanisms to resist the effects of antibiotics, rendering these drugs ineffective in treating bacterial infections. This can lead to longer hospital stays, more expensive treatments, and even death in severe cases.

One way to combat antibiotic resistance is through antibiotic susceptibility testing (AST). AST is a laboratory technique used to determine the most effective antibiotic for treating a specific infection[16]. It involves growing bacteria in the presence of different antibiotics to see which drugs are most effective in killing the bacteria. By using AST to guide antibiotic treatment, healthcare providers can avoid prescribing antibiotics that are unlikely to work, reducing the likelihood of antibiotic resistance. This approach can also help to ensure that patients receive the most effective treatment for their infection, leading to better outcomes and reduced healthcare costs.

One of the most common methods for performing AST is the disk diffusion method [5], which involves placing small disks containing different antibiotics onto a culture plate inoculated with a bacterial isolate. The plate is then incubated, and the resulting growth or inhibition of bacteria around the disks is measured to determine the susceptibility of the organism to antibiotics. Figure 1. Illustrates the AST test. It shows the antibiotic disks with the resulting inhibition zones in an agar plate. The inhibition zone refers to the area of bacterial growth inhibition around an antibiotic disk. This zone is measured to determine the susceptibility or resistance of the bacterial isolate to the antimicrobial agent being tested. Based on comparisons of

inhibition zones on the plate and standard values, the bacteria tested were determined to be resistant, intermediate, and susceptible to different antibiotics present. AST can be done manually or automatically, depending on the laboratory's resources and capabilities. Manual AST methods involve the use of agar plates such as the one shown in Figure 1. Plates are inoculated with bacterial samples and incubated overnight. The plates are then examined to determine the minimum inhibitory concentration (MIC)[13] of antibiotics required to inhibit bacterial growth. The results are interpreted by measuring the diameter of the zone of inhibition around each antibiotic disk. However, manual methods are often less expensive and can be performed in resource-limited settings, but are more time-consuming and prone to human error [8].



**Figure1 - Antimicrobial Susceptibility Testing**

On the other hand, automated AST methods involve the use of specialized equipment that uses predefined protocols to perform the testing process [15, 20]. The equipment can process a large number of samples quickly and accurately, and the results can be interpreted automatically or with minimal human intervention. Automated methods are faster and more accurate, but they require specialized equipment and can be more expensive to set up and maintain.

In the Gaza Strip, the AST test is often performed manually in medical laboratories and hospitals, due to the lack and high cost of the equipment needed for automated tests. In light of increasing antimicrobial resistance around the world, there

is an emerging need for an alternative solution to improve the speed, accuracy, and standardization of AST, while also reducing costs.

Driven by the aforementioned needs, this work proposes PalAST, a mobile application that enables to perform disk diffusion AST automatically. PalAST uses image processing and ML techniques to analyze the photo of the agar plate captured by the mobile camera. Then it provides measurements and interpretations that include the diameters of the inhibition zones, the antibiotic labels, and the rating of the bacteria as susceptible, intermediate, or resistant to each antibiotic.

The following sections are organized as follows: The next section discusses related works and compares them with PalAST. Then, the PalAST usage scenario is presented along with snapshots to highlight its key features and capabilities. Afterwards, the architecture and implementation of PalAST are described, focusing on the system components and the role of each component in the analysis process. The preliminary evaluation of PalAST using both quantitative and qualitative approaches is then presented, and the evaluation results are discussed. Finally, conclusion and future works are presented.

## 1.2. Terminology

- **Antimicrobial resistance (AMR)** happens when bacteria (microbials in general) develop the ability to resist antibiotics [21].
- **Zone of inhibition (ZOI)** is the immediate circular area around the antibiotic disc where there's no bacteria growth -if it's inhibited by the antibiotic. [22]
- **Minimal Inhibitory Concentration (MIC)** is the lowest concentration of an antimicrobial that will inhibit the visible growth of a microorganism after overnight incubation. [23]
- **Mueller-Hinton (MH) agar** is a microbiological growth medium used in AST tests. [24]
- **Antibiotic disc (pellets)** are small circular paper discs that are impregnated with antibiotic (labelled with the antibiotic's name). [25]
- **Petri dish** is a clear shallow cylindrical glass (or plastic) dish with a lid, used to hold a thin layer of agar. [26]

### **1.3. Statement of the problem**

Antimicrobial resistance and antibiotic misuse have hugely increased and therefore caused a lot of deaths. In addition, more AST is needed, and it became harder to analyse them manually such as in hospitals with limited hardware and infrastructure due to the complexity of analytical reading and irregularity between operators. Moreover, it is labour-intensive, time-consuming, and requires an advanced level of expertise for interpretation, especially in places where AST is difficult or impossible to implement.

### **1.4. Objectives**

#### **1.4.1. Main objective**

Our objective is to build a mobile application that automatically analyses disc diffusion ASTs and provides fast, reliable, and comprehensible results.

#### **1.4.2. Specific objectives**

The specific objectives of the project are:

- Study the AST process in detail by meeting with microbiologists.
- Study any solutions used for computerizing the AST.
- Identify the image processing and AI techniques needed to capture, process, extract features of AST Petri dishes.
- Identify the main functions of the proposed mobile app by interviewing the potential end users (microbiologists who will carry out the AST).
- Develop the backend image processing solution that will perform the required measurements.
- Develop the client-side mobile application.
- Develop the API connecting the mobile app with the backend.
- Test the app with realistic images from the laboratory.
- Tune and optimize the app components to resolve potential pitfalls.

## **1.5. Importance of the project**

Antimicrobial resistance is a challenging issue that makes it difficult to treat diseases. A high-profile review forecasts ten million deaths worldwide by 2050.[27]

Identifying a range of resistance is the first step in finding an appropriate way to fight it.

Disk diffusion antibiotic susceptibility testing is widely used in laboratories and hospitals. Despite its strengths, one potential downside to using it as a routine AST method is that the measurements are relatively imprecise, slow, and often cannot standardize values which increases variability among laboratory technicians who interpret results before communicating a report to a patient's physician. Automatic reading systems have been introduced to reduce these drawbacks. But because of their price, hardware, and infrastructure requirements, these systems are not suited to environments such as dispensaries or hospitals in resource-limited settings.

Hence the importance of this project is that our mobile application provides a new free efficient way to get more accurate results and saves time, effort, and money by lower methods costs and reducing labour requirements. laboratory technicians only need to install the application, take images of ASTs then follow the guidance to get interpreted results, all work was done entirely on the same device used to acquire the picture.

We truly believe that the App will have a great impact in our country, where a high degree of antibiotic resistance in hospitals is observed against the most commonly used antibiotics in the health sectors promoted by antibiotic misuse.[28]

## **1.6. Scope and limitations of the project**

### **1.6.1. Scope**

- The project is an Android and iOS mobile application.
- The application's targeted people include biologists and laboratory specialists.
- The application is only concerned with AST.

### **1.6.2. Limitations**

- The application requires internet connection because the image processing will be performed at the server side.

- Lighting and camera resolution may affect the accuracy of the testing results.
- The application was tested using a few sample images taken at the IUG laboratory. We could not find any existing dataset of AST images to test the app.

## 1.7. Methodology

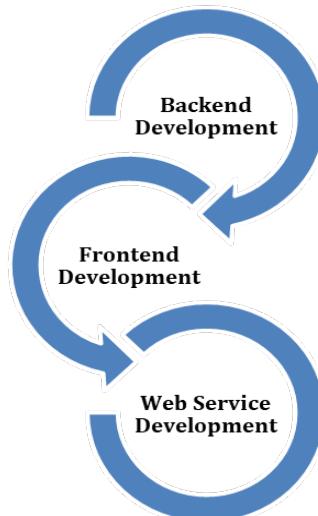
In the section, we will first talk about the software methodology that we adopted, then we will describe the technical procedure to develop our app.

### 1.7.1. Software Methodology

In terms of software methodology, we decided to adopt the Agile software methodology. Agile software development is a set of principles and practices that guide the development and delivery of software using an iterative and incremental approach. It is designed to be flexible, responsive to change, and focused on delivering value to customers.[29] Agile methodology seems to be the most suitable for our project mainly because the requirements of the application we are working on is still unclear, as it is not similar to commercial or common applications that can be relied upon. Therefore, we must consult the target stakeholders (clinical laboratory scientists or employees) about what they want in the application. For example, we need to determine how the user will interact with the photo of the Petri dish, how he will request the services (i.e., measurement of inhibition zones and identification of antibiotic pellets), as well as how the results will be displayed either numerically, visually, or both.

### 1.7.2. Procedure

The development of our app went through the following stages which are depicted in Figure 2 and are briefly illustrated as follows:



**Figure 2 - The Development Stages**

- **Backend development:** Photos of Petri dishes captured by lab analysts were automatically transferred to the server side for the image processing and analysis tasks. This is because a computationally expensive pre-trained machine learning model is required for the image analysis, which is difficult to run on mobile devices. At the backend side, we used a special library called AST-Image-Processing that is built for processing Petri dishes' photos and involves a pretrained model for this purpose. Besides, we decided to use the following algorithms for processing those images:
  - GrabCut algorithm, that was used in pre-processing the AST picture, which consists of the Petri dish to be cropped from the remaining background. [45]
  - SWITCH (Spatial Weighted Intensity Threshold CHangepoint) algorithm is the algorithm that was used for automatic diameter measurement. SWITCH operates a k-means clustering of the pixel intensity (around each antibiotic pellet) to classify inhibition and bacteria pixels (with k = 2). [46]
- **Frontend development:** A mobile app was built for the use of the lab analyst. Through the envisaged app, the end user will take a photo of the Petri dish, and request the service (e.g., analyse the dish and measure the inhibition zones and view the results). The application will interact with the backend through a web service and retrieve the results to be displayed to the end user. The design of the user interface of the application will be determined based on the requirements that were collected from the stakeholders.
- **Web service development:** A web service was built which acted as a mediator between the frontend and the backend, and whose task was to send the images captured by the lab analysts to the server side for processing and extraction of results. The results were returned to the user in JSON format to be overlaid over the image as annotations.

## 1.8. Tools, equipment, and methods

### 1.8.1. Tools

1. **CMake:** An open-source, cross-platform family of tools designed to build, test and package software. It is used to control the software compilation process using simple platform and compiler independent configuration files and generate native makefiles and workspaces that can be used in the compiler environment. [30]
2. **GTest:** A unit testing library for the C++ programming language, based on the xUnit architecture. [31]
3. **OpenCV:** An open-source computer vision and machine learning software library. OpenCV provides a common infrastructure for computer vision applications to accelerate the use of machine perception in the commercial products. [32]
4. **Python:** An interpreted, object-oriented, high-level programming language with dynamic semantics. It is high-level built in data structures, combined with dynamic typing and dynamic binding, making it very attractive for Rapid Application Development. [33]
5. **PyCharm:** PyCharm is a dedicated Python Integrated Development Environment (IDE) providing a wide range of essential tools for Python developers, tightly integrated to create a convenient environment for productive Python, web, and data science development. [34]
6. **JupyterLab:** The Jupyter Notebook is a web-based interactive development environment for code and data. It is used for creating and sharing computational documents to configure and arrange workflows in data science, and machine learning. [35]
7. **Flask:** A web application framework written in Python, based on the Werkzeug WSGI toolkit and the Jinja2 template engine. [36]
8. **Flutter:** An open-source framework by Google for building beautiful, natively compiled, multi-platform applications from a single codebase. [37]

- 9. Figma:** A cloud-based design tool that combines the accessibility of the web with the functionality of a native app. It is used to build an iterative design flow with live collaborations between team members. [38]
- 10. Android Studio:** An Integrated Development Environment (IDE) for Android app development, based on IntelliJ IDEA. [39]
- 11. XCode:** Apple's integrated development environment (IDE) for macOS, used to develop software for macOS, iOS, iPadOS, watchOS, and tvOS.[40]
- 12. Microsoft Word:** A word processor application designed by Microsoft, for document and text processing. [41]
- 13. GitHub:** GitHub is a for-profit company offering a cloud-based Git repository that helps developers store, manage, track and control changes to their code. [42]

### **1.8.2. Equipment**

The equipment essential to use the application:

- **Any basic smartphone:** Android or iOS, with a camera's minimal resolution of 12 megapixels.
- **A prepared and incubated Petri dish** to be pictured and analysed.
- **A simple acquisition setup** made up of cardboard and any two containers available in the laboratory as stands.

### **1.9. Timetable**

The app development has been sectioned into stages as shown in Table1. Each stage has its own activities which were allocated a period of time as shown below.

**Table 1 - Stage Allocation**

Activity	Estimated duration
<b>Stage 1: Requirement elicitation.</b>	
Information collection	5 days
Image collection	5 days
Total	10 days
<b>Stage 2: Building the server-side.</b>	
Requirement Analysis	5 days
Design	2 days
Code	15 days
Test	3 days
Total	25 days
<b>Stage 3: Building the client-side app.</b>	
Requirement Analysis	3 days
Design	3 days
Code	15 days
Test	3 days
Total	24 days

<b>Stage 4: Building the web service.</b>	
Requirement Analysis	3 days
Design	2 days
Code	10 days
Test	2 days
Total	17 days
<b>Final Presentation of Project</b>	
Finalization of the final project report.	12 days
Uploading content and reports.	2 days
Preparing and practicing presentation.	2 days
Total	16 days
<b>Total effort</b>	<b>92 days</b> <b>(Approximately 13 weeks)</b>

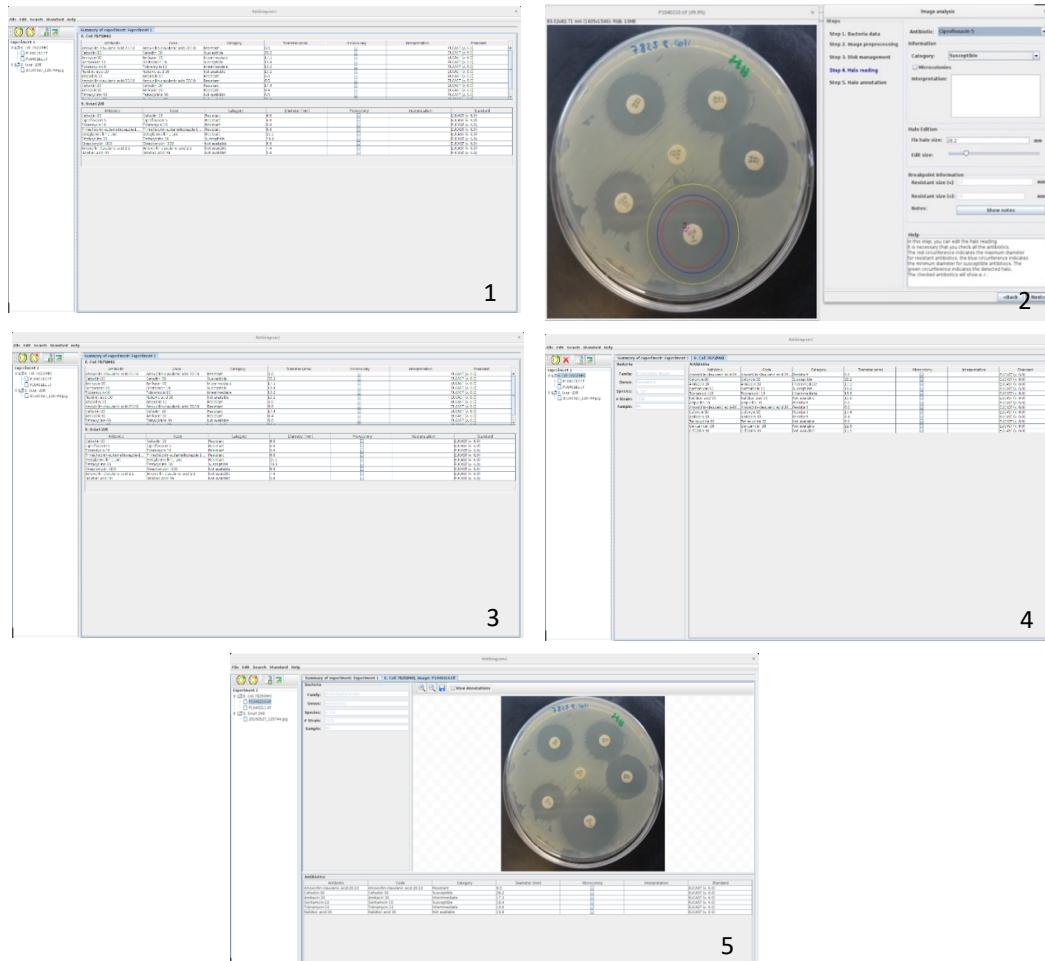
# **CHAPTER 2: RELATED WORKS**

# CHAPTER 2

## RELATED WORKS

In recent years, increasing interest has been given to the use of AI and image processing to support medical diagnosis. These technologies have proven to improve diagnosis accuracy and speed and reduce healthcare costs. Recently, a large number of mobile applications have leveraged AI and image processing techniques to support a variety of medical tests, such as the detection of dermatological diseases [12, 14, 17] and breast cancer [1, 7]. However, a few works have been proposed to aid in antibiotic resistance and AST. In what follows, we give an overview of these works and highlight the differences between them and PalAST.

**2.1 AntibiogramJ [2]** is a desktop Java-based application that takes antibiogram images as input, then identifies and measures inhibition zones as output. However, it has not been produced as a mobile application. In addition, it does not recognize

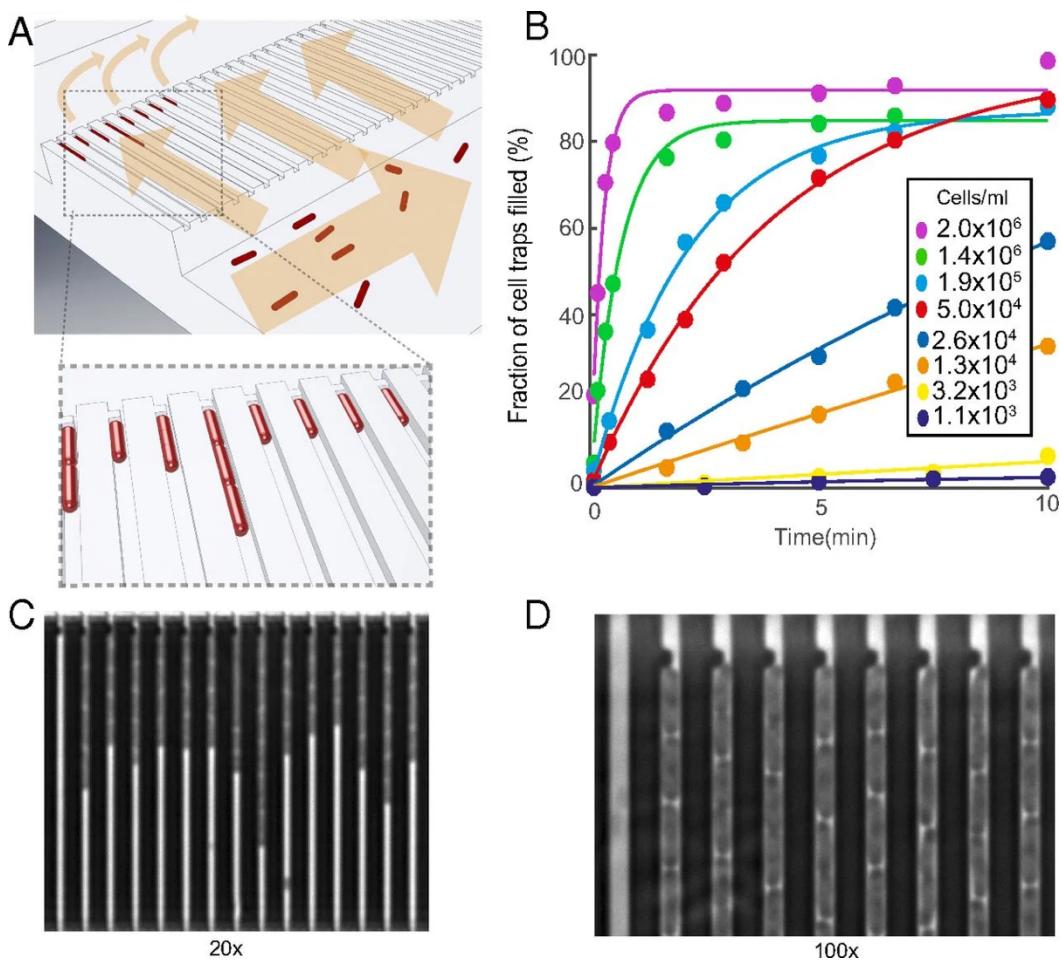


**Figure 3 – AntibiogramJ Program Flow**

antibiotic labels from captured image, and does not provide interpretations of measurements as we do PalAST. Figure 3 - illustrates AntibioGramJ program flow.

**2.2 Kadlec, et al. [10]** proposed a mobile application that offers an automated approach to AST by using a method called gas-permeable microwell arrays, a colorimetric cell viability reagent. Although it uses image processing to analyze bacterial growth, this approach is less common when compared to the disk diffusion method, which is currently the most common, less expensive and most accurate approach for AST.

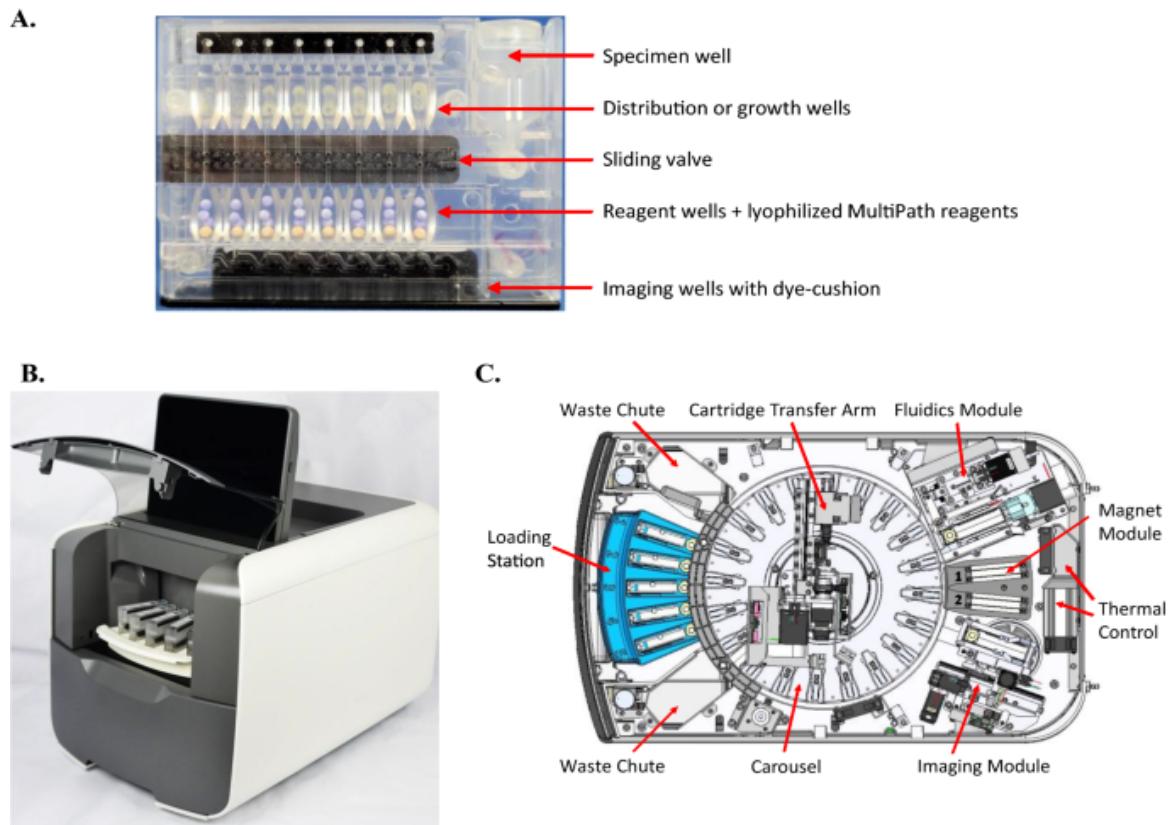
**2.3 Baltekin, et al. [3]** developed a point-of-care susceptibility test for urinary tract infection. Bacterial cells are directly captured from samples using a custom designed microfluidic chip. This test requires loading samples to the chip and then diagnostic readout, which is done in 30 minutes. This work is restricted to AST for



**Figure 4 - Microfluidic Chip Design**

urinary tract infection. In addition, the production of the microfluidic chip needed for this test could be a costly solution, especially in areas with limited resources. Figure 4 shows the design of the chip that performs the test.

**2.4 Burg, et al. [4]** proposed an automated platform that identifies urinary tract infection pathogens in 45 min and provides phenotypic AST results in less than 5



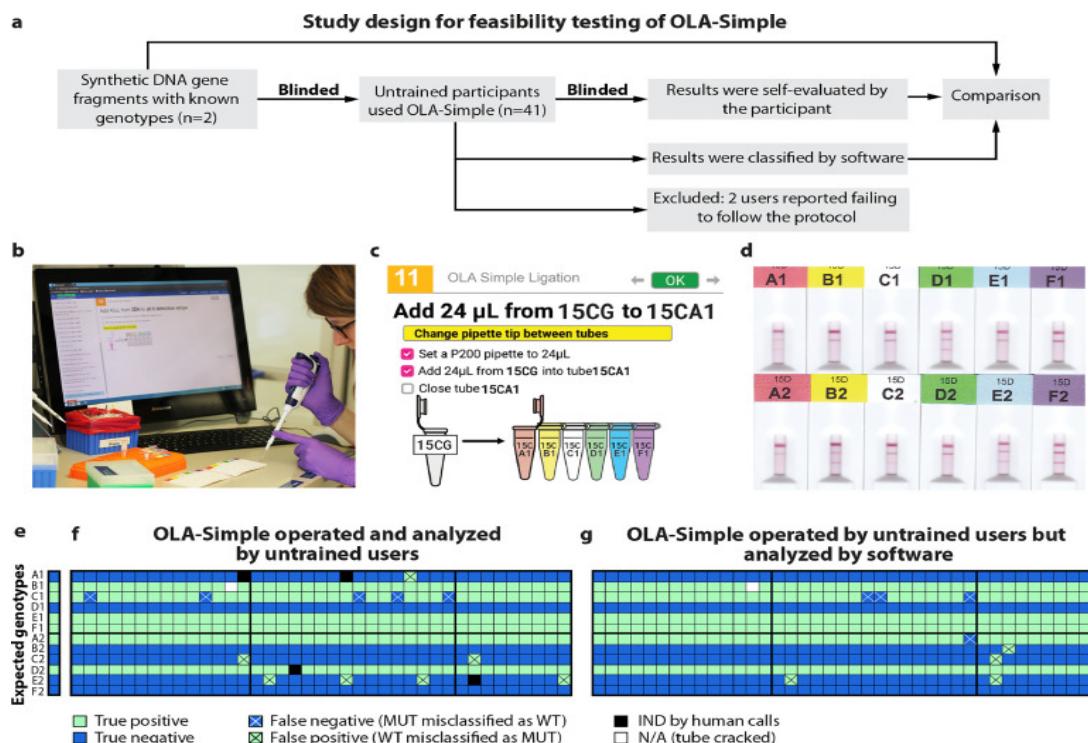
**Figure 5 - Multipath with Automated Analyzer**

hours from urine specimens without colony isolation. Again, this work is limited to AST for urinary infections, and requires special equipment. PalAST does not require any hardware other than the mobile application.

**2.5 Pascucci, et al. [19]** presented an AI-based mobile app for antimicrobial analysis. To the best of our knowledge, this is the most recent and only work that offers an automatic approach for AST by using the disk diffusion method. They also offered an image processing library to analyze images of plates. However, their

mobile application works only on Android devices. They also do not provide their mobile application for public use. This has motivated us to build a similar application that performs similar functionality but is freely available for public use. In fact, we employed the pre-trained ML model from their library to analyze the captured images of plates in PalAST. PalAST offers additional services such as local storage of test results and user authentication. Furthermore, our app is developed to be cross-platform, and thus can work on both Android and iOS devices. Our work also evaluates the usability of the mobile application and the accuracy of the underlying detection methods.

**2.6 Panpradist, et al. [18]** proposed OLA (oligonucleotide ligation assay), which is a kit developed for detection of HIVDR (HIV Drug Resistance) against non-nucleoside/nucleoside reverse transcriptase inhibitors. However, this work does not consider disk diffusion AST. Figure 6 demonstrates the process



**Figure 6 - OLA-Simple Process**

**2.7 Croxatto, et al. [6]** presented a mobile application designed specifically for interpreting AST results from urine cultures. It provides guidance on which antibiotics

are likely to be effective for treating urinary tract infections caused by specific bacteria. However, this application is used particularly for automated inoculation of urine samples to generate isolated colonies. It is not tailored for disk diffusion AST.

**2.8 Antibiogram**<sup>1</sup> is a mobile application that allows users to create custom antibiograms based on local susceptibility data. It also includes tools for tracking resistance patterns over time and comparing data across different healthcare facilities. However, this app does not employ image processing or AI. The laboratory technician still needs to perform AST manually, and then input the measures to Antibiogram in numeric format. Then, it interprets the input measurements to categorize the susceptibility of the bacteria.

**2.9 Antibiotic Guide**<sup>2</sup> is an application that offers a comprehensive database of antibiotics and their indications, dosages, and adverse effects. It also includes information on resistance patterns and susceptibility testing methods. However, this app offers information and guidance, but does not perform AST or offer any AI-based functionalities.

It can be concluded from the above review is that many works that tackled AST have focused on specific diseases, such as HIV or urinary infections, but a few of them have targeted disk diffusion AST. Some of them required special equipment or devices that are difficult to secure in areas with limited resources. Works that support disk diffusion AST through mobile applications either do not support advanced analysis and interpretation of results or are not publicly available to download and use. Table 2 summarizes the differences between the aforementioned works and PalAST.

---

<sup>1</sup> <https://play.google.com/store/apps/details?id=co.inergia.antiBiograma2&hl=en&gl=US>

<sup>2</sup> <https://play.google.com/store/apps/details?id=com.emra.AntibioticGuide>

**Table 2 - Comparison between works that tackle automated AST**

Work	Measurement of inhibition zones	Identification of antibiotic labels	Interpretation of measurements	Mobile app
Kadlec, et al. [10]	✓ (Using gas-permeable microwell arrays)			✓
AntibiogramJ [2]	✓			
Baltekin, et al. [3]	✓ (For urinary tract infection only, requires custom-designed chip)	✓	✓	
Burg, et al. [4]	✓ (For urinary tract infection only, requires special devices)	✓	✓	
Panpradist, et al. [18]	✓ (a kit for detection of HIV Drug Resistance)	✓	✓	

Pascucci, et al. [19]	✓	✓	✓	✓ (Not available for public use, only for Android device)
Croxatto, et al. [6]	✓ (From urine cultures only)	✓		✓
Antibiotic Guide <sup>3</sup>				✓
Antibiogram <sup>4</sup>			✓ (Measurements should be performed manually and inputted by the user)	
PalAST	✓	✓	✓	✓ (Cross-platform)

<sup>3</sup> <https://play.google.com/store/apps/details?id=com.emra.AntibioticGuide>

<sup>4</sup> <https://play.google.com/store/apps/details?id=co.inergia.antiBiograma2&hl=en&gl=US>

# **CHAPTER 3:**

# **ANALYSIS**

## CHAPTER 3

### ANALYSIS

This chapter provides a concise summary of the system requirements, including functional and non-functional aspects, as well as the dynamic behaviour depicted through the sequence diagram and the high-level functionality presented in the use case diagram. It serves as a quick reference to understand the scope and behaviour of the system.

### 3.1 Requirements Specification

This section outlines the functional and non-functional requirements that define the expected behaviour and qualities of the system. It encompasses the specific functionalities to be implemented, as well as the performance, security, usability, and compatibility aspects that the system should adhere to.

#### 3.1.1 Functional Requirements

##### 1. User Registration

- The app should provide a signup functionality for new users to create an account.
- Users should be able to enter their personal details, such as username, email, and password, during the registration process.
- The backend should validate the entered information and store the user's data securely.

##### 2. User Login

- The app should allow registered users to log in using their credentials.
- The backend should authenticate the user's login credentials and provide access to the app's features upon successful authentication.

##### 3. Retrieve All Tests

- The app should retrieve a list of all available tests that belong to a specific user.
- The backend should provide an API endpoint to fetch the list of tests that belong to a specific user from the database.

##### 4. Create a Test

- The app should allow users to create a new test.
- Users should be able to enter relevant details for the test, such as sample type, bacteria name, and test image.
- Users should be able to choose sample type from a list of supported samples.
- Users should be able to choose bacteria from a list of supported bacteria types.
- The backend should validate the entered information and store the test details in the database.

## **5. Capture Test Image**

- The app should provide functionality to capture an image for a test.
- Users should be able to use their device's camera to take a picture and associate it with a specific test.
- The app should handle image capturing and storage securely.

## **6. Import Test Image**

- The app should allow users to import a test image from their device's gallery.
- Users should be able to select an image and associate it with a specific test.
- The app should handle image importing and storage securely.

## **7. Analysis of Results**

- The app should facilitate the submission of test results for analysis by sending a request to the backend server.
- The backend server should have an API endpoint to receive and process the analysis request.
- The backend server should implement the necessary algorithms or integrations to analyze the test results.
- Once the analysis is completed, the app should display the analysed results to the user.

## **8. Modification of Test Results**

- The app should receive the test results from the backend server.
- The app should provide a user interface for the modification of test results, in a clear and organized manner.
- The user interface should allow the user to correct the antibiotic label if necessary.
- The user should be able to make changes to the antibiotic label and save the updated label.
- The user interface should allow the user to correct the radius of inhibition zone if necessary.
- The app should provide a slider to enable the user to modify the radius of inhibition zone.
- The user should be able to make changes to the radius of inhibition zone and save the updated value.

## **9. Annotation and Overlay of Test Results on Images.**

- The app should send a request to the backend server to draw on a test image.
- The backend server should implement the drawing feature on the test image.
- Using appropriate image processing techniques or libraries, the server should add labels and draw inhibition zones around the antibiotics as instructed.
- The server should generate a modified image that includes the drawn labels and inhibition zones.
- The server should send the modified image back to the app.
- The app should receive the modified image data from the server.
- The modified image should be rendered and presented to the user, showing the added labels, and drawn inhibition zones.

- The app's user interface should include controls or gestures that allow the user to zoom in and out on the displayed image.

## **10. Show results**

- The app should display the analysed test results in a tabular format.
- Users should be able to view the results in a structured and organized manner.
- The app's UI should present the results clearly and allow easy navigation and interaction.
- The app should enable users to export the analysed results to a PDF format.
- The exported PDF file should contain the processed image as well as the table of results.

### **3.1.2 Non-Functional Requirements**

#### **1. Security**

- The app should implement appropriate security measures to protect user data and ensure secure communication between the client and server.
- User passwords should be securely stored using encryption techniques.
- API endpoints should be protected against common security vulnerabilities, such as SQL injection and cross-site scripting (XSS)
- The app should enforce secure authentication using JWT (JSON Web Tokens).
- The app should include functionality for user registration, login, and token expiration.
- Upon successful login, the app should receive a JWT token from the backend server.
- The app should securely store the JWT token to maintain the user's authenticated session.
- The app should implement appropriate security measures to prevent unauthorized access or tampering of the stored JWT token.
- The app should send the JWT token with each subsequent request to the backend server.
- The app should include the JWT token in the request headers for authentication and authorization purposes.
- The backend server should validate the JWT token for each incoming request to ensure the authenticity and permissions of the user.

- The backend server should implement proper validation and verification of JWT tokens.
- The server should validate the token's signature, expiration, and other relevant claims to ensure its validity.

## **2. Performance**

- The app should be responsive and provide a smooth user experience.
- The backend should be optimized to handle concurrent requests efficiently.
- Image processing and analysis should be performed in a timely manner to minimize wait times for the user.

## **3. Usability**

- The app's user interface should be intuitive and user-friendly.
- Users should be able to navigate through the app's features easily and understand how to perform various actions.

## **4. Compatibility**

- The app should be compatible with multiple platforms, such as Android and iOS, to reach a wider user base.
- The app's UI should be responsive and adapt to different screen sizes and orientations.

## **5. Reliability:**

- The app should be stable and reliable, with minimal downtime.
- The backend should handle errors and exceptions gracefully and provide appropriate error messages to the user when necessary.

## **6. Maintainability:**

- The codebase for both the server-side and client-side components should be well-structured and maintainable.
- Proper documentation and comments should be provided to assist future development and maintenance tasks.

## 3.2 Use Case Diagram

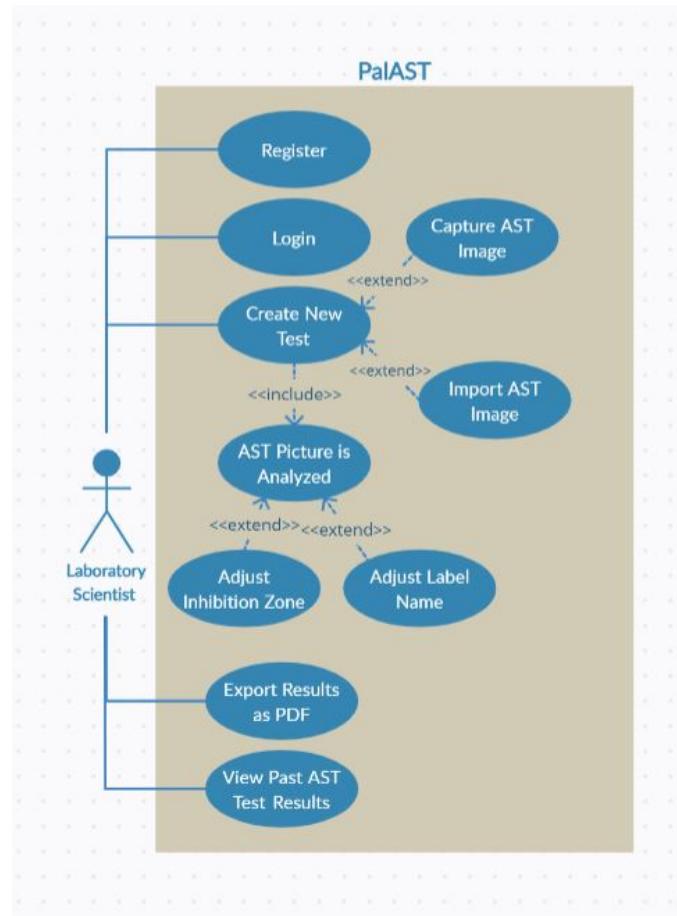


Figure 7 - Use Case Scenario

## 3.3 Use Cases' Description

Table 3 - Use Case Scenarios Explanation

Name	Name of the Use Case
Actor	The stakeholder who initiated the use case
Pre-condition	The condition that must be met to invoke this use case
Post-condition	The condition that must be met after the use case is executed
Basic Flow	The details of the use case in which nothing goes wrong
Alternate Flow	Any alternate scenario of the basic flow that might occur

**Table 4 - Register Use Case**

Name	Register
Actor	Any laboratory scientist who installed the app
Pre-condition	None
Post-condition	The user creates an account to use the app
Basic Flow	<ol style="list-style-type: none"> <li>1. The user opens the register page.</li> <li>2. The system prompts the user to enter his email and password.</li> <li>3. The user's data get validated.</li> <li>4. The user is forwarded to the dashboard</li> </ol>
Alternate Flow	The user's data doesn't match the specified input requirements (duplicate values, incorrect email format)

**Table 5 - Login Use Case**

Name	Log In
Actor	Any laboratory scientist who installed the app
Pre-condition	The user must be registered in the system
Post-condition	The user is logged into the system successfully
Basic Flow	<ol style="list-style-type: none"> <li>1. The user opens the login page.</li> <li>2. The system prompts the user to enter his email and password.</li> <li>3. The system validates the entered data and logs the user into the system</li> </ol>
Alternate Flow	<ul style="list-style-type: none"> <li>- The user doesn't have an account.</li> <li>- The user entered invalid data</li> </ul>

**Table 6 - Create New Test Use Case**

Name	Create New Test
Actor	Any laboratory scientist who installed the app
Pre-condition	The user is logged into the system
Post-condition	The user successfully chose a method for creating a new AST Test (importing an AST image or capturing one)
Basic Flow	<ol style="list-style-type: none"> <li>1. The user is forwarded to the dashboard page after successful login.</li> <li>2. The user clicks on the (Create Test) button.</li> <li>3. The user chooses in which way the petri dish image of the AST will be uploaded</li> </ol>
Alternate Flow	None

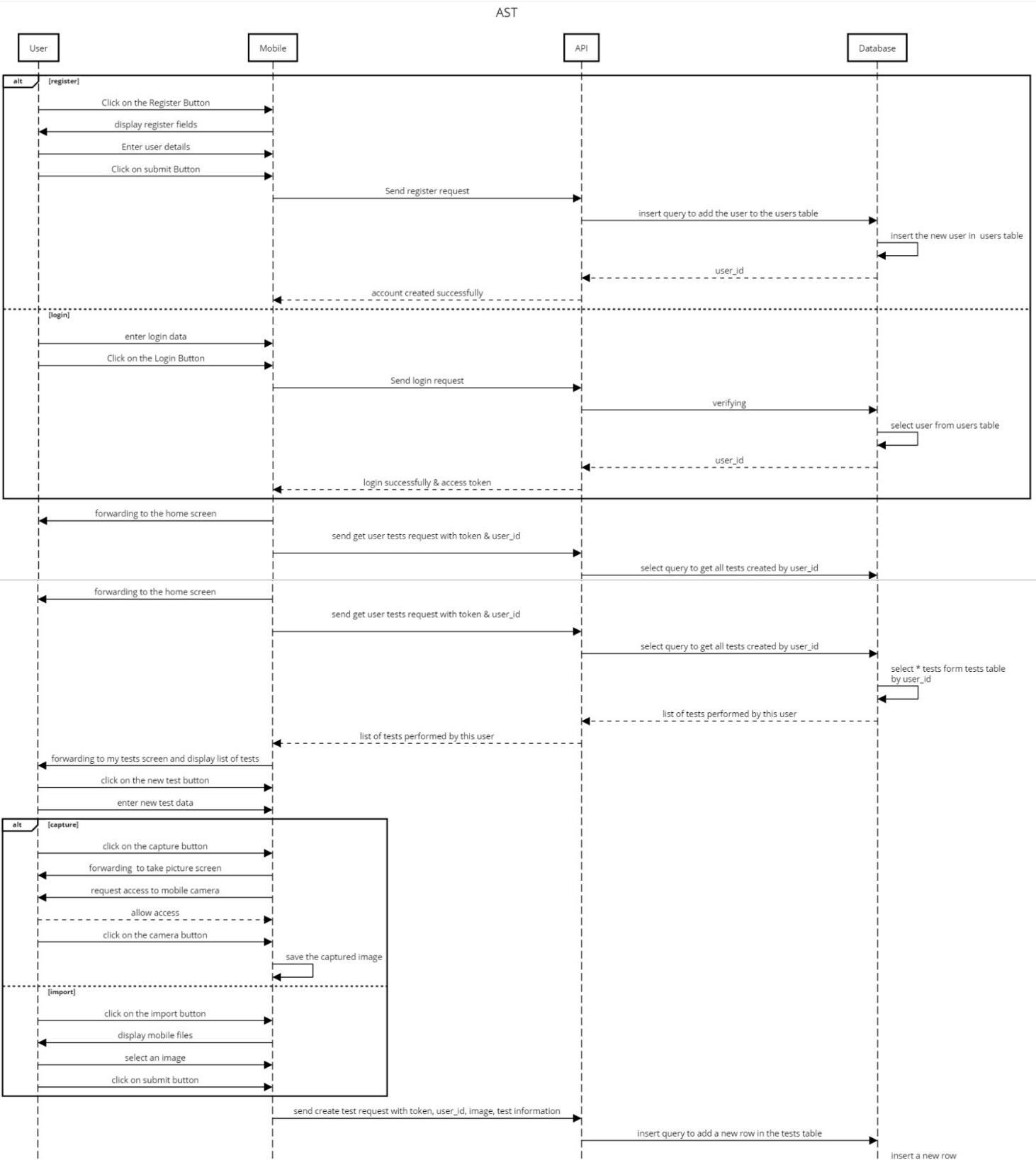
**Table 7 - Export Results Use Case**

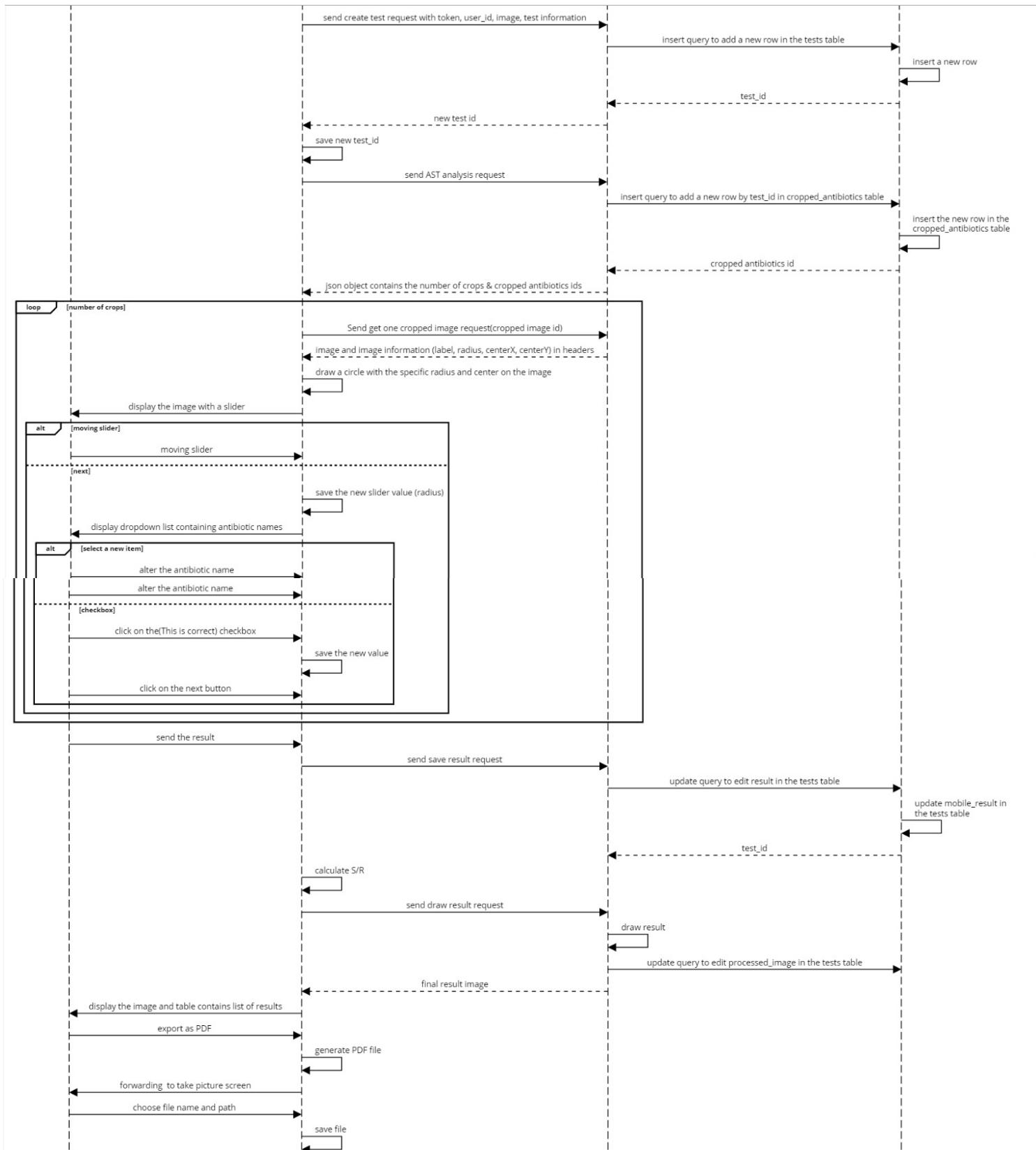
Name	Export Results as PDF
Actor	Any laboratory scientist who installed the app
Pre-condition	The system successfully analysed the AST image
Post-condition	An analysed AST image along with the result table is successfully saved as a PDF file
Basic Flow	<ol style="list-style-type: none"><li>1. The system prompts the user to upload an AST image.</li><li>2. The user uploads the AST image.</li><li>3. The system validates the AST image</li></ol>
Alternate Flow	<ul style="list-style-type: none"><li>- The uploaded image isn't an AST image.</li><li>- The uploaded image doesn't meet the AST image acquisition protocol</li></ul>

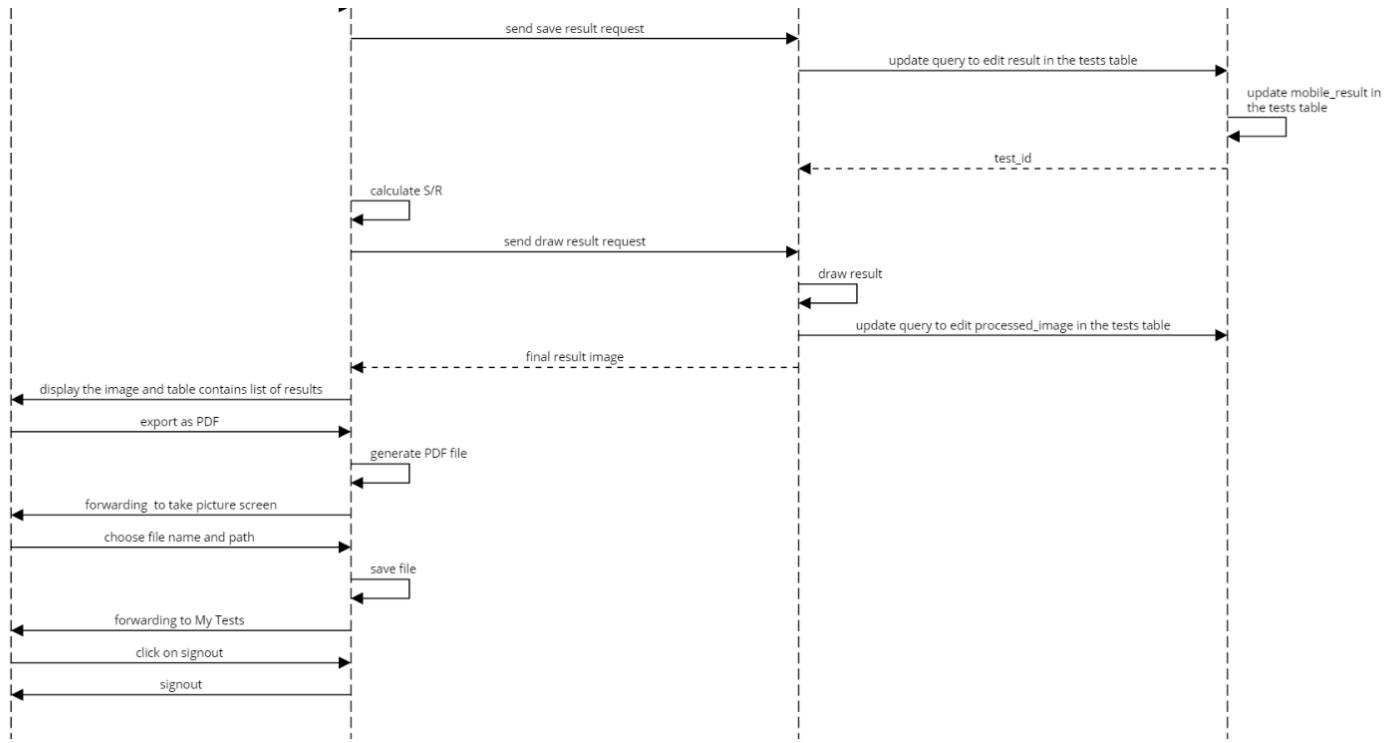
**Table 8 - View Past Results Use Case**

Name	View Past AST Results
Actor	Any laboratory scientist who installed the app
Pre-condition	An AST image is previously analysed and saved by the user
Post-condition	A previously analysed test is displayed to the user
Basic Flow	<ol style="list-style-type: none"><li>1. The user clicks on a previously analysed test.</li><li>2. The system displays the analysed AST image along with the results table</li></ol>
Alternate Flow	None

### 3.4. Sequence Diagram







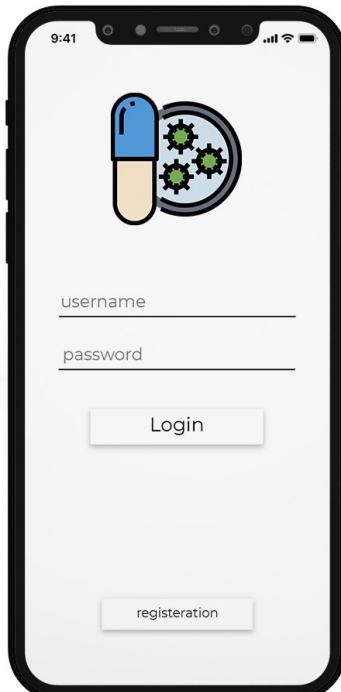
**Figure 8 - Sequence Diagram**

### 3.5. User Interface (UI) Sketches

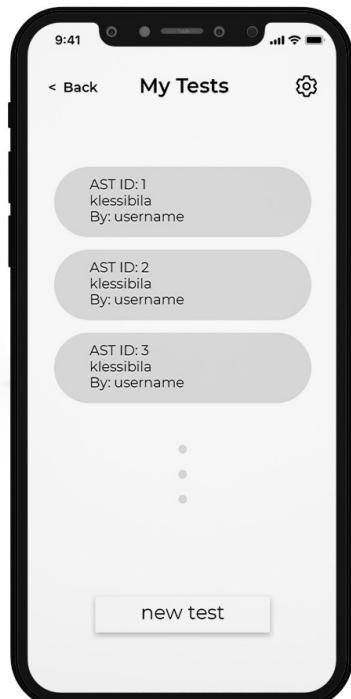
This section showcases visual UI sketches that provide an initial representation of the app's screens and layout.



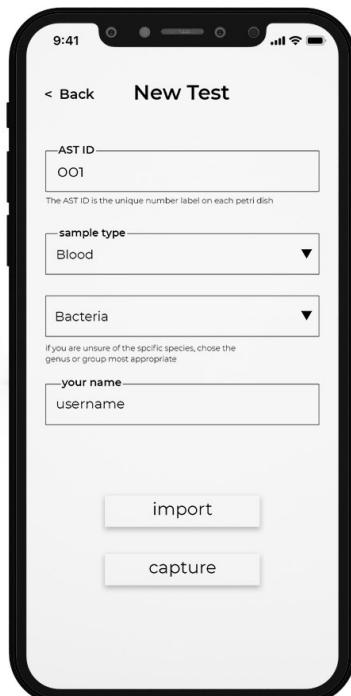
**Figure 9A – Registration Screen Sketch.**



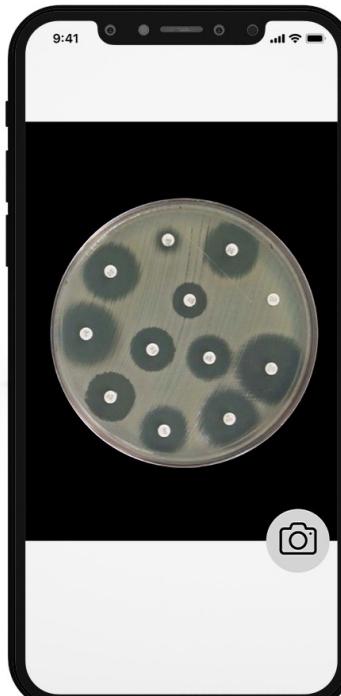
**Figure 9B – Login Screen Sketch.**



**Figure 9C – All Tests Screen Sketch.**



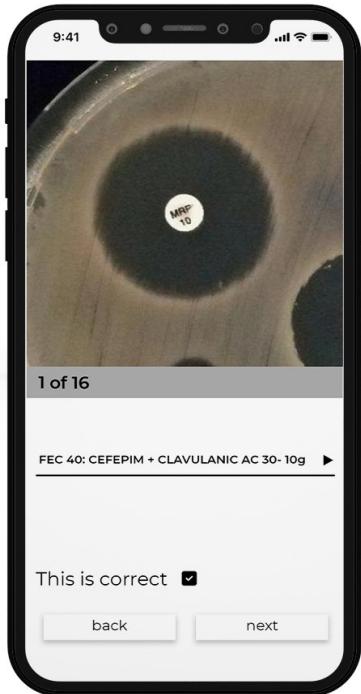
**Figure 9D – Create Test Screen Sketch.**



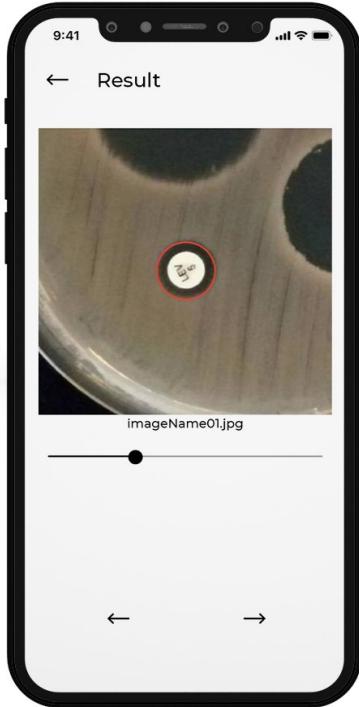
**Figure 9E – Capture Image Screen Sketch.**



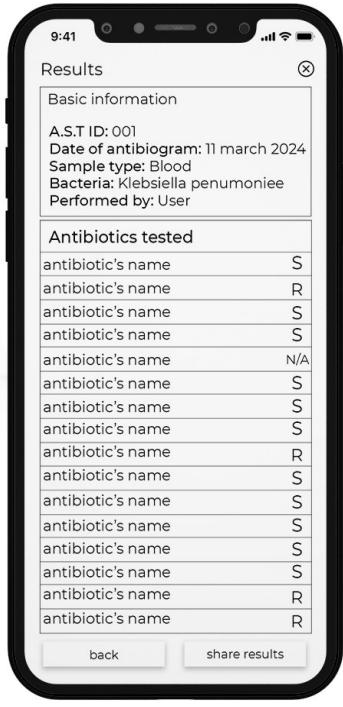
**Figure 9F – Import Image Screen Sketch.**



**Figure 9G – Antibiotic Label Modification Screen Sketch.**



**Figure 9H – Inhibition Zone Radius Modification Screen Sketch.**



**Figure 9I – Test Results Screen Sketch.**

**Figure 9 - App UI Sketches**

# **CHAPTER 4:**

# **DESIGN**

# CHAPTER 4

## DESIGN

In this chapter, we'll discuss the architecture and methodology that we followed in developing PalAST. We will go into details about every technical aspect of PalAST, including the API documentation and database schema.

### 4.1. Architecture

The system handles the problem of developing a mobile application that enables users to examine disc diffusion AST images and offers quick and accurate results. The user, mobile application, server, and database make up the system's components. For the mobile application to communicate with the web server, the interfaces between those elements include HTTP APIs. To arrange the various system components, a three-tier architecture has been implemented; these three levels are:

#### a. Presentation Layer

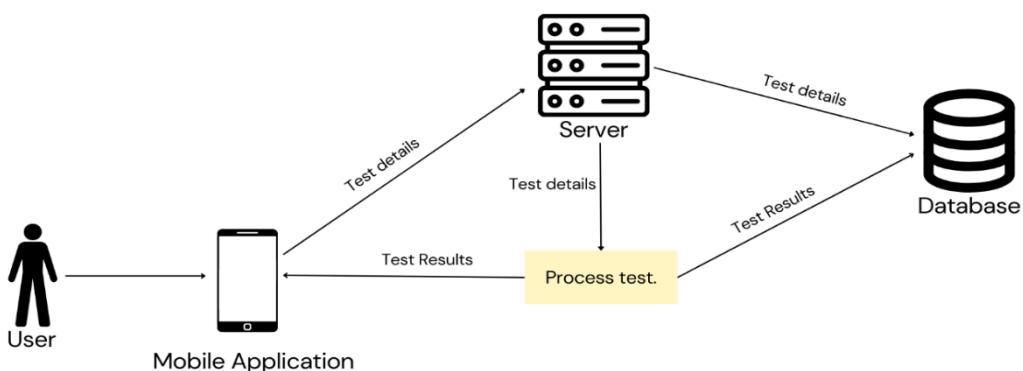
The mobile application interacts with the web server through a REST API. The web server provides the necessary data and functionalities to the mobile application.

#### b. Application Layer

The web server handles the processing of requests and responses from the mobile application. It interacts with the database server to retrieve or update data as needed.

#### c. Data Layer

The database server stores the system data and provides APIs for the web server to interact with it. Figure (10) demonstrates the high-level system design.



**Figure 10 - High-Level System Design**

## 4.2 API Documentation

- **Signup API “/auth/signup”.**
  - Input: username, email, and password.
  - Output: the newly created user id
  - Method: HTTP POST
- **Login API “/auth/login”.**
  - Input: email, password
  - Output: authentication token
  - Method: HTTP POST
- **Get Users Tests API “/user/tests”.**
  - Input: authentication token
  - Output: list of tests that were performed by this user
  - Method: HTTP GET
- **Create Test API “/test/create”.**
  - Input: Bacteria, Sample type, image file, authentication token
  - Output: the newly created test id
  - Method: HTTP POST
- **Confirm Test API “/test/confirmation”.**
  - Input: authentication token, test id, image information
  - Output: number of tests updated
  - Method: HTTP POST
- **Crop Image API “/process/crops”.**
  - Input: authentication token, test id
  - Output: JSON response that contains the number of antibiotic disks that were identified, and a list of the cropped images to be fetched later.
  - Method: HTTP POST
- **Send One Image API “/fetch/crop”.**
  - Input: authentication token, cropped image id
  - Output: image with the specified id along with its information.
  - Method: HTTP POST.

- **Send Drawn Image API “/fetch/draw”.**
  - Input: authentication token, test id
  - Output: test image that illustrates the test results drawn directly on.
  - Method: HTTP POST.

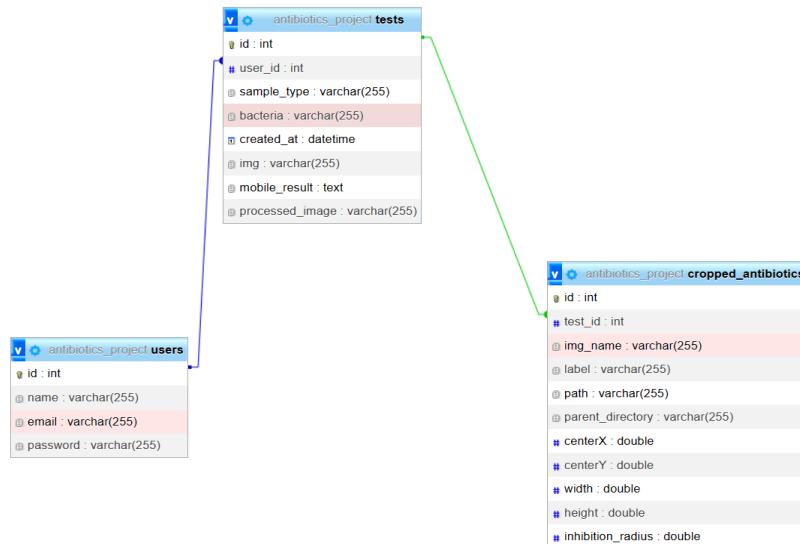
### 4.3 Database Schema

Our flask web service database is designed to store and manage information about users, tests performed by users, as well as information about antibiotic images that were generated from users' test. The schema consists of three tables: the 'users' table, the 'tests' table, and the 'cropped\_antibiotics' table.

The 'users' table stores basic information about each user who uses our mobile app. It contains columns for the name, email, and password of the user. The 'tests' table has the details of each test's data. It contains columns that describe the sample type of the test, the bacteria being tested, the path of the AST image, the test's creation date, the modified test result, and the path of the drawn image. The 'cropped\_antibiotics' table stores information about each antibiotic image that was identified from the AST image. It contains columns that describe the test that this image belongs to.

Overall, our database schema is designed to provide a centralized location for managing our users and their tests data.

**Figure 11 demonstrates the database schema.**



**Figure 11 - Database Schema**

## 4.4 Mobile App

First, the user needs to register for the app. User registration is necessary to create a profile for each user, where all his/her conducted tests and results can be saved so that they can be retrieved and accessed at a later time from anywhere.

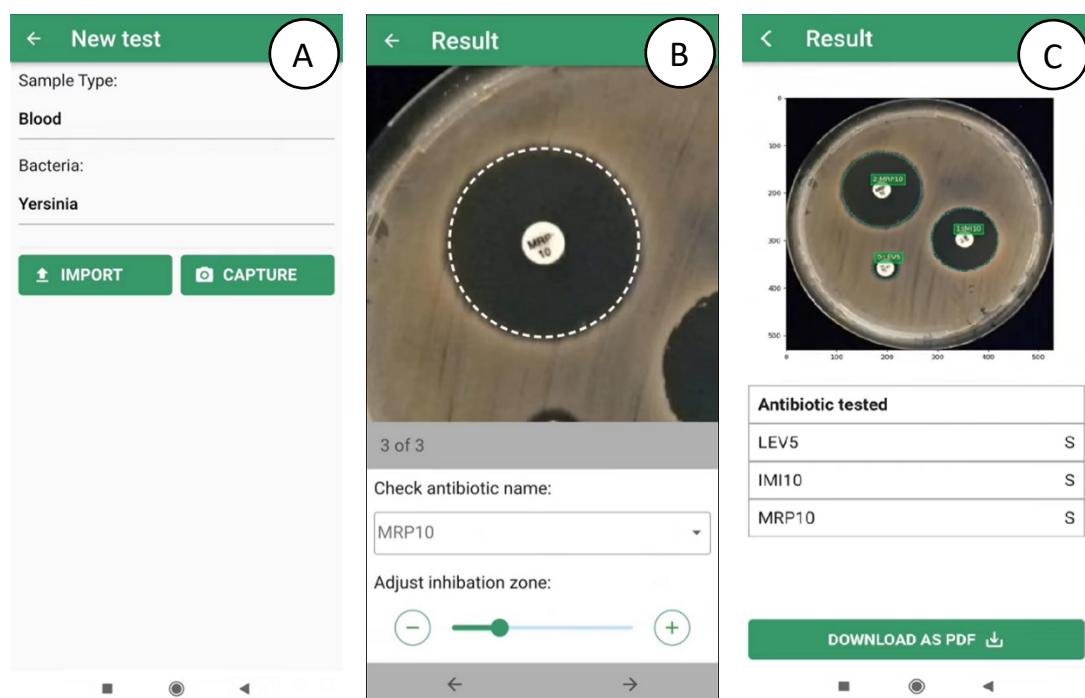
The user then creates a new AST test by specifying its settings that include the nutrition medium and the type of bacteria as shown in Figure 12.A, and then taking a high-quality photo of the plate. For the analysis to give good results, the captured photo should fulfil the following requirements: 1) the plate should be entirely included in the photo with no perspective distortion and should not touch the borders of the photo. 2) the photo is not blurred. 2) There are no shadows or light reflection on the plate.

After taking the photo, PalAST will send it to the server side for analysis. The server side analyses the photo to detect the antibiotic disks, identify the antibiotic labels, and measure the diameters of inhibition zone. The entire analysis takes a few seconds to complete, and the results are returned to the client to be presented and interpreted on the mobile device.

After the analysis completes, the photo of each antibiotic disk with the surrounding inhibition zone is cropped out and displayed to the user one by one, as shown in Figure 12.B. Each cropped photo will be augmented with annotations showing the detected antibiotic label and the bounding circle of inhibition zone. Although the underlying ML model used in PalAST has high prediction accuracy as will be revealed from the evaluation results, some bounding circles may not match the boundaries of the inhibition zones perfectly. Therefore, PalAST gives the user the ability to manually adjust the size of the predicted bounding circle or choose a different antibiotic label from a predefined list of labels as shown in Figure 12.B. This feature enables the user to correct any potential prediction errors. After verifying the measurements related to one antibiotic disk, the user clicks on the next button to view the photo of the next antibiotic.

The above process continues until the user verifies all results. At this point, the application will display the entire photo of the plate annotated with the predicted/adjusted bounding circles and antibiotic labels (see Figure 12.C). It also

shows a final report summarizing the susceptibility categories of the bacteria isolate as resistant (R), intermediate (I), or susceptible (S) to each antibiotic used in the plate. These categories are determined based on the comparison between the diameters of detected bounding circles and establish standards such as those developed by the Clinical and Laboratory Standards Institute (CLSI)<sup>5</sup>. These standards define what so called breakpoints, which are the diameters of the zone that separates susceptible, intermediate, and resistant categories for each bacteria type. For example, a bacterial isolate is considered susceptible if the size of the inhibition zone is larger than the established breakpoint for susceptibility. The final results report can be exported in PDF format and shared via email or social networks. In addition, test results are stored in the user's profile so that the user can access and view past tests at any time.



**Figure 12 - Screenshots of sample screens from PalAST**

**A)** The user creates a new AST and specifies its settings. **B)** Detected zones are presented one by one in sequence to the user for verification and correction. **C)** Final results showing the entire plate with annotations, the susceptibility categories of the bacteria.

<sup>5</sup> <https://clsit.org/> (Clinical and Laboratory Standards Institute)

# **CHAPTER 5:**

# **IMPLEMENTATION**

# CHAPTER 5

## IMPLEMENTATION

The architecture of PalAST consists of two components: the server side, where all the AI and image processing functionalities are performed, and the client side where results are interpreted and displayed to the user. The two parts of PalAST communicate through a RESTful web service. The codes of both server and client sides are freely available for download<sup>6</sup> <sup>7</sup>. These components are explained in what follows.

### 5.1 The Server Side of PalAST

The image analysis is carried out on the server side and goes through the steps shown in Figure 13 as follows:

1. **Image cropping:** The image of the plate that was captured and sent by the client side is first pre-processed to detect and crop the plate. An edge detection algorithm from OpenCV is used to detect the edges of the plate and then crop it to remove the surrounding empty spaces.
2. **Detection of inhibition zones:** The cropped image is then passed as input to a pre-trained ML model from the AST library [19]. The model is pre-trained to automatically recognize inhibition zones and predict bounding circles around these zones as shown in Figure 13. For each inhibition zone, the model returns the diameter in pixels, the centre coordinates and a confidence score representing how confident the model is about the zone measurements.

Note that the library measures the diameter in pixels. The scale in pixels cannot be directly converted to real length in millimetres because the image can be captured from different perspectives or zoom ratios. To determine the real diameter from the image of the plate, it is necessary to have a reference to scale the zone in image. Therefore, we chose the antibiotic disk as a reference object because it typically has a fixed size of 6 mm [9]. We then calculated the scaling factor by dividing the diameter of the antibiotic disk by its diameter in

---

<sup>6</sup> Client side code (PalAST Flutter App) can be downloaded from <https://github.com/malak271/AST>

<sup>7</sup> Server side code can be downloaded from <https://github.com/GhadeerHayek/AST-Flask>

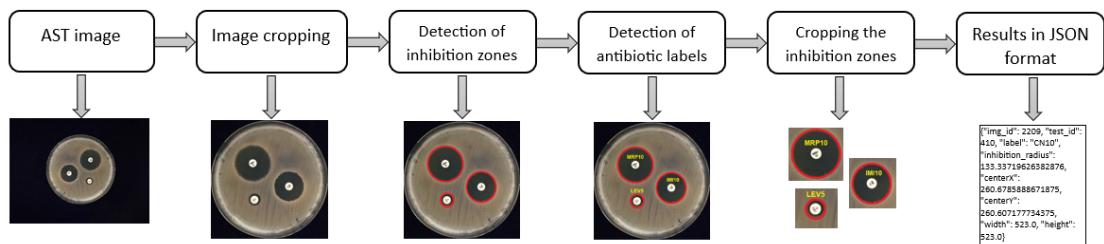
pixels as in Equation 1. This will give a scaling factor, expressed in millimetres per pixel.

$$\text{Scaling factor (mm/pixel)} = \frac{\text{Diameter of antibiotic disk (6 mm)}}{\text{Measured diameter of antibiotic disk in pixels}} \quad (1)$$

Finally, we multiply the diameter of the inhibition zone in pixels by the scaling factor to obtain the real diameter in millimetres as in Equation 2.

$$\begin{aligned} \text{Diameter of inhibition zone (mm)} &= \\ \text{Diameter of inhibition zone in pixels} * \text{Scaling factor} &\quad (2) \end{aligned}$$

3. **Detection of antibiotic labels:** The AST library also offers an optical character recognition (OCR) function that enables recognizing and reading the label written on the antibiotic disk.
4. **Cropping the inhibition zones:** After detecting the antibiotic disks and the associated zones, the next step is to crop out each zone in a separate image. Therefore, the image of plate is split into multiple images, each of which shows a single antibiotic disk with the surrounding inhibition zone. These images are stored on the server and will be retrieved later by the client side to be presented to the user on the mobile for verification as explained in Section 3.
5. **Creating JSON representation of analysis results:** At this point, the analysis process should have been completed, and the analysis results are grouped together and presented in JSON format that will be sent back to the client side. The JSON output contains a sequence of JSON objects, each of which corresponds to an antibiotic disk and its cropped image. The Information of each detected antibiotic includes:
  - The detected antibiotic label.
  - Centre coordinates (position of the disk)
  - Diameter of the inhibition zone.
  - ID of the corresponding cropped image (to be accessed from the mobile device).



**Figure 13 - Image Analysis in PalAST**

## 5.2 The Client Side of PalAST

The client side of PalAST is a mobile application that enables the user to initiate the AST test and get results in real time through a user-friendly interface. The application was developed using Flutter, which allows it to run on both the Android and iOS platforms. The application has five main modules:

- 1- **The connector:** This module is responsible for submitting the captured image of the agar plate to the server side, and then receiving the analysis results in JSON format. Then it extracts information from JSON and sends it to the annotator module.
- 2- **The annotator:** It creates and displays annotations over the image of the plate. These annotations include textual labels showing the antibiotic label and the diameter of the inhibition zone. It also draws a bounding circle around the inhibition zone.
- 3- **The verifier:** The verifier is a UI module that displays the cropped inhibition zones of detected antibiotic disks one by one in sequence. It enables the user to correct potential errors by adjusting the bounding circle and the detected label of the antibiotic as illustrated in Section 3.
- 4- **The storage manager:** This module is responsible for storing results of all AST tests performed by the user and enabling access to these results at any time.
- 5- **The interpreter:** Using the measured diameters of detected inhibition zones and the establish standards for breakpoints in AST, this module provides an interpretation of the results. It categorizes the tested bacteria as susceptible, intermediate, or resistant to each antibiotic disk on the plate.

# **CHAPTER 6:**

# **TESTING AND**

# **EVALUATION**

# CHAPTER 6

## TESTING AND EVALUATION

We evaluated PalAST with these objectives in mind:

- Assessing the ability of PalAST to accurately create bounding circles that capture the full extent of the inhibition zones.
- Assessing the ability of PalAST to recognize the antibiotic labels written on disks.
- Assessing the usability and ease of use of PalAST.

And to achieve the latter, we used two evaluation methods:

- Evaluation of the detection performance by using common metrics used for evaluating object detection.
- Expert evaluation of usability and ease of use through a user-centred questionnaire.

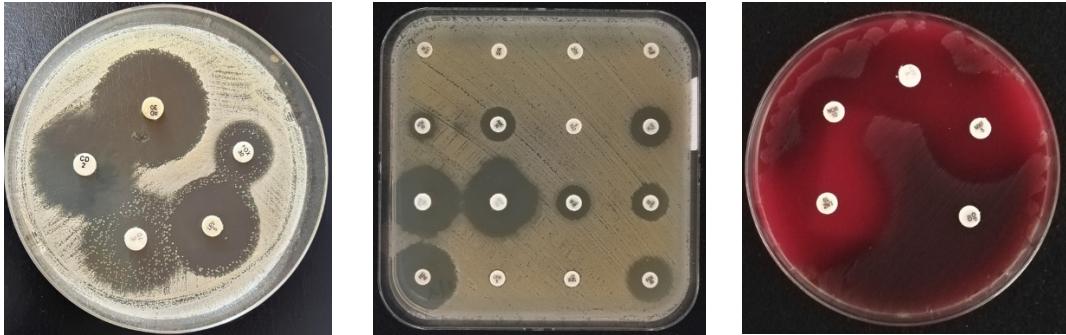
In the following subsections, we explain each evaluation method in detail.

### **6.1 Evaluation of the detection performance**

#### **6.1.1 Data Collection**

- To evaluate the object detection performance, we need to experiment with a number of AST images using disk diffusion method, and then use common metrics for evaluating object detection. However, we could not find any freely available dataset of AST images that we could use for evaluation. Therefore, we contacted biologists from our local university to provide us with real photos of disk diffusion AST. We asked them to take photos of agar plates that use various types of bacteria and antibiotics. By using their mobile phones. We gave them instructions on how to capture good quality photos. In addition, we collected several photos of AST plates from the Internet. In total, we collected 14 photos of agar plates. These plates contain 138 antibiotic disks in total, around which zones of inhibition are present. The number of unique antibiotics in these photos is 12. AST plates were selected to have different shapes, i.e., circular, and square, and colours of nutrition medium so that the detection results can be assessed under

different conditions. Figure 14 shows sample photos. The dataset can be accessed from ([here](#)).



**Figure 14 - Photos from the Tested Dataset**

### 6.1.2 Evaluation Metrics

We use several metrics to evaluate the detection performance. In the following, we present these metrics and explain how we adapt them to our evaluation:

1. **Precision:** Precision measures the percentage of true positive detections in relation to the total number of detections. It is calculated as  $TP / (TP + FP)$ , where TP in our case denotes the number of truly detected antibiotic labels or bounding circles. FP is the number of false positives. In our case, false positives refer to antibiotic disks that are detected, but are mistakenly identified as of another type, or are localized with misaligned bounding circles that are too small or large.
2. **Recall:** Recall measures the percentage of true positive detections in relation to the total number of ground truth objects. It is calculated as  $TP / (TP + FN)$ . FN refers the number of undetected antibiotic labels or inhibition zones.
3. **Intersection over Union (IoU):** IoU measures the overlap between the predicted bounding circle and the actual boundaries of the zone. It is calculated as the area of intersection between the two circles divided by the area of their union.

Since PalAST performs two different detection tasks: the detection of the inhibition zones and the detection of antibiotic labels, we calculated the precision and

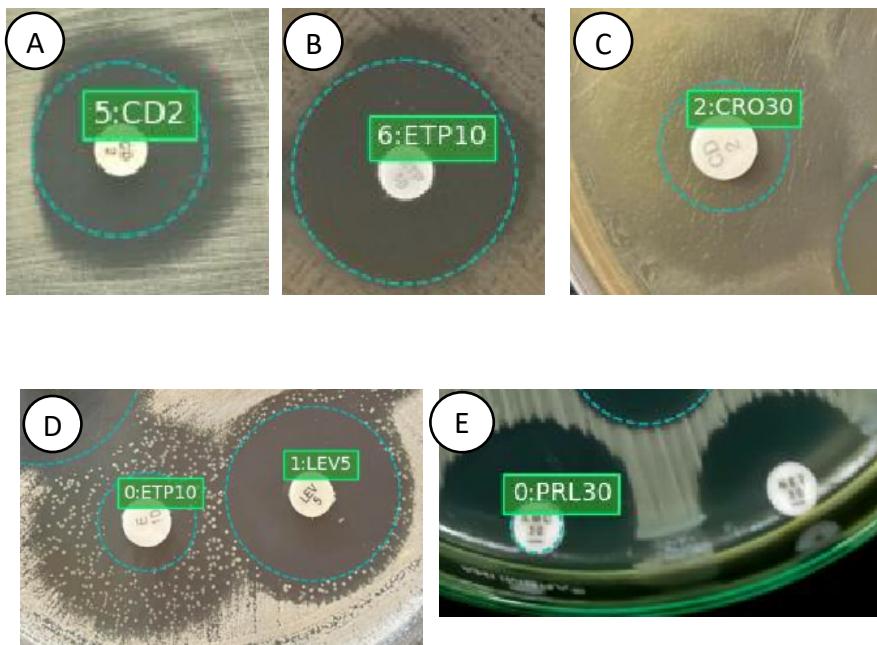
recall values for each detection method separately. The IoU is calculated only for the zone detection task.

### 6.1.3 Results and Discussion

The plate images were processed by PalAST to obtain new images annotated with the predicted bounding circles and antibiotic labels similar to the image in Figure 6.C. The annotated images were then presented to a human expert (me) to evaluate the predicted bounding circles and antibiotic labels with reference to actual values. Tables 9 and 10 presents the evaluation results of the zone detection task. In total, 97 out of 138 zones were accurately detected and localized, and the predicted bounding circles matched the zones as indicated by the evaluator. In general, PalAST achieved 0.858 precision and 0.795 recall.

Note that some inhibition zones grew on the plate to be not perfectly circular. In such a case, the bounding circles may fully include the zones of inhibition as in Figures 15.A and 15.B. These bounding circles may have different diameters in different directions. For these zones, the detection algorithm automatically picks the inner diameter that makes a full circle, excluding areas beyond this radius.

The number of zones that were detected but with misaligned bounding circles were 16 out of 138. When inspected, we found that the diameters of these zones were incorrectly measured due to reasons such as poor lighting and light reflection that made the inhibition area imperceptible and hard to distinguish, as in Figure 15.C. In addition, some detection errors resulted from inhibition zones that are not completely empty and may show some bacterial growth such as the one shown in Figure 15.D. Experts identified these as zones of partial inhibition and indicated that such cases occurred due to factors related to the concentration of the antibiotic and the incubation conditions. These cases can be resolved by repeating the test with different settings. Other detection errors were due to antibiotic disks placed closed to edges of the plate, and thus their zones grew as strips instead of complete circles (see Figure 15.E). In general, many of the detection errors can be resolved by capturing high quality images with no blurring or light reflection, or by allowing enough spaces between antibiotic disks and the plate borders.



**Figure 15 - Examples of inaccurate detection of inhibition zones**

**A,B) A inhibition zone with non-circular shape. C) Zone is obscure due to poor lighting. D) Zones of partial inhibitions. E) Zones colliding with the edge of the plate**

The average IoU value is 0.706. This indicates about 70% overlap between the actual zones and the predicted bounding circles. For the true positive cases only (accurately bounded zones), the average IoU is 0.901. For the false-positive cases (zones bounded with misaligned circles), the average IoU is 0.702. The IoU is considered to be 0 for false negative cases (undetected zones).

Tables 11 and 12 show the result of the label detection task. PalAST achieved 0.733 precision and 0.81 recall for this task. When the false predictions were inspected, we found that most of them resulted from issues related to blurred images and poor lighting that made it impossible to read labels. By comparing results from Tables 12 and 13, it can be concluded that PalAST performed slightly better with the zone detection task than with the label detection task.

Note that in both detection tasks, TN (true-negative) represents the number of locations in an image that are correctly classified as not containing inhibition zones or antibiotic disks. Therefore, the TN value is considered to be zero.

**Table 9 - Confusion matrix of the zone detection process**

		Actual	
		Positive	Negative
Prediction	Positive	97	16
	Negative	25	0

**Table 10 - Evaluation results of the zone detection process**

Precision	0.858
Recall	0.795
F-measure	0.825
IoU	0.706

**Table 11 - Confusion matrix of the antibiotic label detection process**

		Actual	
		Positive	Negative
Prediction	Positive	85	31
	Negative	20	0

**Table 12 - Evaluation results of the antibiotic label detection process**

Precision	0.733
Recall	0.810
F-measure	0.77

## **6.2 Expert evaluation of PalAST usability and ease of use**

We prepared the questionnaire shown in Table 13, which includes questions covering key aspects of usability, such as ease of use, efficiency, and user satisfaction. Each question should be answered using a 5-point Likert scale, where 1 denotes “Very weak” and 5 denotes “Very good”. The questionnaire also includes an open question about any further comments or suggestions for improvement. We presented PalAST to two experts from our university and asked them to fill out the questionnaire. Both experts are medical professionals with expertise in microbiology and infectious diseases.

Table 13 also shows the feedback received from the two experts. In general, both experts provided very positive feedback, giving “Good” or “Very good” ratings for all the functions of PalAST. Both experts highly agreed that the application automates AST in a way that saves time and efforts. When asked about the functions that they liked most, they both mentioned the fast detection of zones and the automatic rating of bacteria. They also indicated that the ability to adjust the detected bounding circles is extremely helpful. Even in manual measurement, experts may measure or interpret results differently due to incomplete inhibition or the overlapping between zones.

When asked about suggestions for improvement, they both suggested the need to add new types of antibiotics and bacteria species, as well as update the interpretive criteria of AST which is likely to change over time. They also highlighted the need to run the application offline because the internet connection may not be available in some areas.

**Table 13 - Feedback received from experts.**

<b>Question</b>	<b>Expert 1 responses</b>	<b>Expert 2 responses</b>
PalAST saves time and effort by automating the AST	Very good	Very good
PalAST enables me to easily start a new AST test and capture AST photo	Very good	Good
Inhibition zones are accurately detected and measured by PalAST	Good	Good
Antibiotic names are accurately detected and recognized by PalAST	Good	Good
PalAST responds quickly to AST requests and presents results without incurring significant delay	Very good	Very good
PalAST enables me to easily verify the analysis results and correct any wrong detections or measurements.	Good	Good
PalAST correctly rates bacteria as susceptible, intermediate, or resistant to each antibiotic	Good	Good
PalAST enables me to access and review past test results	Very good	Very good
PalAST is easy to use and navigate	Very good	Very good

# **CHAPTER 7:**

# **CONCLUSION AND**

# **FUTURE WORK**

## CHAPTER 7

### CONCLUSION AND FUTURE WORK

This work proposes PalAST, a cross-platform mobile application that supports automated AST by using disk diffusion method. The application enables biologists to take AST images and analyze them in real time and with minimal human intervention. PalAST also provides an interpretation of the results including the measured diameters of the inhibition zones, the recognized antibiotic labels, and the categorization of the bacteria isolate as susceptible, intermediate, or resistant to each antibiotic.

PalAST was tested using a number of real AST images, and the detection performance was evaluated by using common metrics, i.e., precision, recall and IoU. We also used expert evaluation through a questionnaire to assess the usability and ease of use of PalAST.

We believe that PalAST can have important implications for patient care and public health in Palestine and other less advantaged areas in the world. More importantly, we think that PalAST contributes to the global efforts and priorities of the World Health Organization, which has recognized the antibiotic resistance as one of the most health threats facing the world in the coming years. Nevertheless, it should be noted that mobile applications should not be used as a substitute for professional medical advice or laboratory testing. They can be a useful tool to supplement clinical decision-making but should be used in conjunction with other sources of information and guidance from healthcare professionals.

In the future, we aim to upgrade PalAST to work offline, as it currently works online only. This is particularly important in areas that have limited or interrupted access to the Internet. Second, we aim to conduct a large-scale evaluation of PalAST by testing it with a larger number of AST images and getting feedback from a larger number of users and experts. This will help us to better identify potential defects or additional functionalities.

# BIBLIOGRAPHY

## BIBLIOGRAPHY

- [1] M. A. S. Al Husaini, M. Hadi Habaebi, T. S. Gunawan, and M. R. J. E. Islam, "Self-detection of early breast cancer application with infrared camera and deep learning," vol. 10, no. 20, p. 2538, 2021.
- [2] C. Alonso et al., "Antibiogramj: A tool for analysing images from disk diffusion tests," vol. 143, pp. 159-169, 2017.
- [3] Ö. Baltekin, A. Boucharin, E. Tano, D. I. Andersson, and J. J. P. o. t. N. A. o. S. Elf, "Antibiotic susceptibility testing in less than 30 min using direct single-cell imaging," vol. 114, no. 34, pp. 9170-9175, 2017.
- [4] L. Burg et al., "Rapid pathogen identification and phenotypic antimicrobial susceptibility directly from urine specimens," vol. 12, no. 1, p. 18315, 2022.
- [5] L. Coorevits, J. Boelens, G. J. E. J. o. C. M. Claeys, and I. Diseases, "Direct susceptibility testing by disk diffusion on clinical samples: a rapid and accurate tool for antibiotic stewardship," vol. 34, pp. 1207-1212, 2015.
- [6] A. Croxatto, K. Dijkstra, G. Prod'hom, and G. J. J. o. c. m. Greub, "Comparison of inoculation with the InoqlA and WASP automated systems with manual inoculation," vol. 53, no. 7, pp. 2298-2307, 2015.
- [7] C. Hou et al., "Predicting breast cancer in Chinese women using machine learning techniques: algorithm development," JMIR medical informatics, vol. 8, no. 6, p. e17364, 2020.
- [8] R. M. Humphries et al., "The continued value of disk diffusion for assessing antimicrobial susceptibility in clinical laboratories: report from the clinical and laboratory standards institute methods development and standardization working group," vol. 56, no. 8, pp. e00437-18, 2018.
- [9] B. E. Igere, A. I. Okoh, U. U. J. I. j. o. e. r. Nwodo, and p. health, "Antibiotic susceptibility testing (AST) reports: A basis for environmental/epidemiological surveillance and infection control amongst environmental vibrio cholerae," vol. 17, no. 16, p. 5685, 2020.
- [10] M. W. Kadlec, D. You, J. C. Liao, and P. K. J. S. T. Wong, "A cell phone-based microphotometric system for rapid antimicrobial susceptibility testing," vol. 19, no. 3, pp. 258-266, 2014.
- [11] A. Kohli, "Antimicrobial Sensitivity Testing: The Search for Antibiotics," 2021.
- [12] I. Kousis, I. Perikos, I. Hatzilygeroudis, and M. Virvou, "Deep learning methods for accurate skin cancer recognition and mobile application," J Electronics, vol. 11, no. 9, p. 1294, 2022.
- [13] B. Kowalska-Krochmal and R. J. P. Dudek-Wicher, "The minimum inhibitory concentration of antibiotics: Methods, interpretation, clinical relevance," vol. 10, no. 2, p. 165, 2021.
- [14] V. B. Kumar, S. S. Kumar, and V. Saboo, "Dermatological disease detection using image processing and machine learning," in 2016 Third International

- Conference on Artificial Intelligence and Pattern Recognition (AIPR), 2016, pp. 1-6: IEEE.
- [15] Y. Li, X. Yang, and W. Zhao, "Emerging microtechnologies and automated systems for rapid bacterial identification and antibiotic susceptibility testing," *SLAS TECHNOLOGY: Translating Life Sciences Innovation*, vol. 22, no. 6, pp. 585-608, 2017.
  - [16] M. Luc, "A comparison of disc diffusion and microbroth dilution methods for the detection of antibiotic resistant subpopulations in Gram negative bacilli," 2015.
  - [17] R. Pangti et al., "A machine learning-based, decision support, mobile phone application for diagnosis of common dermatological diseases," vol. 35, no. 2, pp. 536-545, 2021.
  - [18] N. Panpradist et al., "OLA-Simple: a software-guided HIV-1 drug resistance test for low-resource laboratories," vol. 50, pp. 34-44, 2019.
  - [19] M. Pascucci et al., "AI-based mobile application to fight antibiotic resistance," vol. 12, no. 1, p. 1173, 2021.
  - [20] S. Puttaswamy, S. K. Gupta, H. Regunath, L. P. Smith, and S. J. A. C. M. Sengupta, "A comprehensive review of the present and future antibiotic susceptibility testing (AST) systems," vol. 9, no. 3, 2018.
  - [21] Antimicrobial resistance. Accessed December 24, 2022.  
<https://www.who.int/news-room/fact-sheets/detail/antimicrobial-resistance>
    - [22] GUIDE\_2.1\_ANTIMICROBIAL.pdf. Accessed December 24, 2022.  
[https://www.woah.org/fileadmin/Home/eng/Our\\_scientific\\_expertise/docs/pdf/GUIDE\\_2.1\\_ANTIMICROBIAL.pdf](https://www.woah.org/fileadmin/Home/eng/Our_scientific_expertise/docs/pdf/GUIDE_2.1_ANTIMICROBIAL.pdf)
  - [23] Andrews JM. Determination of minimum inhibitory concentrations. *J Antimicrob Chemother*. 2001;48 Suppl 1:5-16. doi:10.1093/jac/48.suppl\_1.5
  - [24] 9: Kirby-Bauer (Antibiotic Sensitivity). Biology LibreTexts. Published March 26, 2016. Accessed December 24, 2022.  
[https://bio.libretexts.org/Learning\\_Objects/Laboratory\\_Experiments/Microbiology\\_Labs/Microbiology\\_Labs\\_I/09%3A\\_Kirby-Bauer\\_\(Antibiotic\\_Sensitivity\)](https://bio.libretexts.org/Learning_Objects/Laboratory_Experiments/Microbiology_Labs/Microbiology_Labs_I/09%3A_Kirby-Bauer_(Antibiotic_Sensitivity))
  - [25] 20-Antibiotic-Disc.pdf. Accessed December 24, 2022.  
<https://www.rcpath.org/uploads/assets/f180bd2d-8189-4b4c-813e9c1c795fc5f4/20-Antibiotic-Disc.pdf>
  - [26] Petri dish. Accessed December 24, 2022. <https://microbiologie-clinique.com/petri-dish.html>
  - [27] no-time-to-wait-securing-the-future-from-drug-resistant-infections-en.pdf. Accessed December 24, 2022. <https://www.who.int/docs/default-source/documents/no-time-to-wait-securing-the-future-from-drug-resistant-infections-en.pdf>
  - [28] دائرة المكتبات... المكتبة المركزية (جواهر لال نهرو) جامعة الأزهر - غزة... Accessed December 24, 2022. <http://www.alazhar.edu.ps/library/>

- [29] What is Agile Development? And how does it work? - Starttech Ventures. Accessed December 24, 2022. <https://www.starttech.vc/library/what-is-agile-development/>
- [30] CMake. Accessed December 24, 2022. <https://cmake.org/>
- [31] GoogleTest User's Guide. GoogleTest. Accessed December 24, 2022. <http://google.github.io/googletest/>
- [32] About - OpenCV. Accessed December 24, 2022. <https://opencv.org/about/>
- [33] What is Python? Executive Summary. Python.org. Accessed December 24, 2022. <https://www.python.org/doc/essays/blurb/>
- [34] Get started | PyCharm Documentation. Accessed December 24, 2022. <https://www.jetbrains.com/help/pycharm/quick-start-guide.html>
- [35] Project Jupyter | Home. Accessed December 24, 2022. <https://jupyter.org/>
- [36] Welcome to Flask — Flask Documentation (2.2.x). Accessed December 24, 2022. <https://flask.palletsprojects.com/en/2.2.x/>
- [37] Flutter - Build apps for any screen. Accessed December 24, 2022. <https://flutter.dev/>
- [38] Figma: the collaborative interface design tool. Accessed December 24, 2022. <https://www.figma.com/>
- [39] What is Android Studio? | Definition from TechTarget. Mobile Computing. Accessed December 24, 2022. <https://www.techtarget.com/searchmobilecomputing/definition/Android-Studio>
- [40] Xcode 14 Overview - Apple Developer. Accessed December 24, 2022. <https://developer.apple.com/xcode/>
- [41] What is Microsoft Word? - Definition from Techopedia. Techopedia.com. Accessed December 24, 2022. <http://www.techopedia.com/definition/3840/microsoft-word>
- [42] What Is GitHub? A Beginner's Introduction to GitHub. Accessed December 24, 2022. <https://kinsta.com/knowledgebase/what-is-github/>
- [43] Rother C, Kolmogorov V, Blake A. "GrabCut": interactive foreground extraction using iterated graph cuts. In: ACM SIGGRAPH 2004 Papers. SIGGRAPH '04. Association for Computing Machinery; 2004:309-314. doi:10.1145/1186562.1015720
- [44] Pascucci et al. - 2021 - AI-based mobile application to fight antibiotic re.pdf. Accessed December 24, 2022. [https://static-content.springer.com/esm/art%3A10.1038%2Fs41467-021-21187-3/MediaObjects/41467\\_2021\\_21187\\_MOESM1\\_ESM.pdf](https://static-content.springer.com/esm/art%3A10.1038%2Fs41467-021-21187-3/MediaObjects/41467_2021_21187_MOESM1_ESM.pdf)