



PRINCESS SUMAYA BE UNIVERSITY FOR TECHNOLOGY

KING HUSSEIN SCHOOL OF COMPUTING SCIENCES

Prepared by ABDULLAH BEANO AND HASAN SAMMOUR

MedLabs Specimen Tracking System

Supervised by Mohammad Abu-Snober

Project submitted in partial fulfillment for the degree of Bachelor of Science in Computer Science

2019 - 2020

Declaration of Originality

This document has been written entirely by the undersigned team members of the project. The source of every quoted text is clearly cited and there is no ambiguity in where the quoted text begins and ends. The source of any illustration, image or table that is not the work of the team members is also clearly cited. We are aware that using non-original text or material or paraphrasing or modifying it without proper citation is a violation of the university's regulations and is subject to legal actions.

Acknowledgments

Acknowledgments go here...

Abstract

Abstract goes here...

List of Abbreviations

Royal Scientific Society. (HTML)
Global Position System. (GPS)
Global System Module. (GSM)
Subscriber Identification Module. (SIM)
Graphical User Interface. (GUI)
Hypertext Markup Language. (HTML)
Cascading Style Sheets. (CSS)

Contents

Abstract	3
List of Abbreviations	4
List of Figures	6
List of Tables	7
1 Introduction	8
1.1 Overview	8
1.2 Problem Statement	8
1.3 Related Work	9
1.4 Document Outline	9
2 Project Plan	10
2.1 Project Deliverables	11
2.2 Project Tasks	11
2.3 Gantt Chart	12
2.4 Roles and Responsibilities	12
2.5 Cost Estimation	12
2.6 Project Management Tools	12
3 Requirements Specification	13
3.1 Stakeholders	13
3.2 Platform Requirements	14
3.2.1 Mobile Application and Website	14
3.2.2 Specimen Tracking System	14
3.3 Functional Requirements	14
3.3.1 Mobile Application and Website	14
3.3.2 Specimen Tracking System	17
3.4 Non-Functional Requirements	17
3.4.1 Mobile Application and Website	17
3.4.2 Specimen Tracking System	19
3.5 Other Requirements	19
3.5.1 Hardware Requirements	19
3.5.2 Software Requirements	19
3.5.3 Other Supplementary Requirements	22
4 System Design	23
4.1 Logical Model Design	23
4.2 Physical Model Design	23

List of Figures

List of Tables

2.1	Project Tasks	11
3.1	Stakeholders	13

Chapter 1

Introduction

1.1 Overview

In this project, a live tracking system of the different couriers working for MedLabs in Amman will be developed. These couriers' main job is to deliver varying medical samples from MedLabs branches to the main branch for further testing. A portion of these workers are also lab technicians who will be taking samples from the patients from their homes, the system will track their routes as well. Secondly, an application and a website will be developed to allow users to fill request-forms for their samples to be taken from home. by a capable lab technician. As the only way to achieve that currently is through a phone call to MedLabs call-center.

There's undeniable value when it comes to making improvements to the health-care system in any way. And a dynamic, modern and powerful health-care system is of utmost significance to help our community. As for us working to build this project, learning new software, hardware, and technologies is of absolute importance and growing an entrepreneurial sense delving into a new field such as this will greatly benefit our careers moving forward.

1.2 Problem Statement

The current system at MedLabs is based on users making phone calls directly to the call center to make an arrangement for their samples to be taken at home. Which regularly leads to heavy load on the call-center subsequently leading to customer dissatisfaction. This inefficient and often confusing - for the patients - system will be renovated by the software to be developed in this project. Which will make the process far simpler, faster, and more convenient for the patients while lifting sizeable load off the call-center. A live tracking system of the couriers will also greatly benefit MedLabs business as a whole, by helping to optimise their already rich testing, and boosting efficiency.

1.3 Related Work

1.4 Document Outline

The remainder of this document is arranged as follows:

- Chapter 2 details the project's plan and the analysis that went into it.
- Chapter 3 describes the system's requirements in detail.
- Chapter 4 illustrates the system's design using a plethora of diagrams and the model's description.

Chapter 2

Project Plan

2.1 Project Deliverables

2.2 Project Tasks

ID	Task	Description	Duration	Dependencies
T01	Define the problem	Precisely describe the problem to be solved in a problem statement	Week 1	None
T02	Collect user requirements	Interview personnel from MedLabs to collect and document user requirements	Week 1	T01
T03	Identify system requirements	Deduce, document, and prioritise all system requirements	Week 2	T02
T04	Manage system requirements	Use a traceability matrix to manage system/user requirements	Week 2	T02, T03
T05	Form a development strategy	Research and document hardware and software tools to be adopted in the project	Week 2	T03
T06	Design system requirements	Use a wide and applicable array of diagrams to manifest the system's requirements	Week 3	T04
T07	Design user interfaces	Use the appropriate software to design user interface prototypes	Week 4	T03, T06

Table 2.1: Project Tasks

2.3 Gantt Chart

2.4 Roles and Responsibilities

2.5 Cost Estimation

2.6 Project Management Tools

Chapter 3

Requirements Specification

3.1 Stakeholders

Stakeholder	Description	Importance of Role
MedLabs	The tracking system, the application, and the website developed in this project were made to benefit MedLabs' business and to optimise their courier system	High
Royal Scientific Society (RSS)	The project's idea was made available and the project's database was built by the RSS. The RSS also specified the system's requirements	Medium
The developers	The students who built the tracking device and the students who designed and developed the tracking system, the application, and the website. All of the aforementioned will greatly benefit the students' learning process and careers in the future	Low

Table 3.1: Stakeholders

3.2 Platform Requirements

3.2.1 Mobile Application and Website

The mobile application will be developed using the React JavaScript library for Android and iOS. The application is rather simple and not at all taxing and will run without trouble on almost any mobile device with an active internet connection.

The website will be accessible and should run effectively on any desktop or laptop with an active internet connection.

3.2.2 Specimen Tracking System

The hardware used in this project is an Arduino based vehicle tracking system. Arduino is a single-board microcontroller that - in this case - uses a global position system (GPS), and a global system module (GSM) to achieve real-time tracking of any vehicle or object that bears and makes use of a SIM card even if it's offline.

The system will be assembled inside of the vehicle. Two applications, for the administrator and the employee will be developed to track all the vehicles.

3.3 Functional Requirements

3.3.1 Mobile Application and Website

1. Account Related Functional Requirements

- (a) The program must allow a new user to create a new account. The user will have to provide a valid username, a valid password and a valid personal phone number along with their full name.
 - Input: A valid username, a valid password, a valid phone number, a first name, and a last name.
 - Output: Creation of a new account.
 - Constraints: The username should not be longer than 20 characters. The password should contain at least: One number, one lowercase letter, and one uppercase letter. The phone number should be valid.
 - Priority: Mo; must have.
- (b) The program must allow a user who entered a password / username combination that matches one in the database to access the account.
 - Input: A valid username and a valid password.
 - Output: The user will be directed to his home page in the application.
 - Constraints: The combination of the username and the password entered must match the combination stored in the database.

- Priority: Mo; must have.
- (c) The program must allow a user who forgot their account's password to reset it using the phone number linked to that account. The user will have to provide a valid username, and a valid personal phone number.
- Input: A valid username, and a valid phone number.
 - Output: A randomly-generated new password sent to the user's phone number. The user can change this password once they log in.
 - Constraints: A risk of a lost phone being accessed by other users.
 - Priority: Mo; must have.
- (d) The program should allow a user who forgot their account's username to reset it using the phone number linked to that account.
- Input: A valid phone number and a valid password.
 - Output: A randomly-generated new username sent to the user's phone number. The user can change this username once he logs in.
 - Constraints: A risk of a lost phone being accessed by other users.
 - Priority: S; should have.
- (e) The program could allow the user to login using their phone number and their password instead of the username and password.
- Input: A valid phone number and a valid password.
 - Output: The user will be directed to his home page in the application.
 - Constraints: The combination of the phone number and the password entered must match the combination stored in the database.
 - Priority: Co; could have.

2. User Related Functional Requirements

- (a) The program must allow the user to edit information related to their account. Which includes: The username, password, phone number, first name, last name, personal avatar / image, and blood type.
- Input: Bits of information entered by the user in their respective fields on the GUI.
 - Output: Updated user information being stored in the database.
 - Constraints: The profile editing interface is only accessible to the user after logging in successfully to make sure no user modifies another's information.
 - Priority: Mo; must have.
- (b) The program must allow the user to view their history of tests done, and lists of scheduled and ongoing tests.
- Input: Bits of information entered by the user in their respective fields on the GUI.
 - Output: Updated user information being stored in the database.

- Constraints: The profile editing interface is only accessible to the user after logging in successfully to make sure no user modifies another's information.
 - Priority: Mo; must have.
- (c) The program must notify the user with the following: a) Confirmation of the time, location and type of sample to be taken - specified by the user - once the request has been processed by the system. b) A sample has been tested and the results are ready. In which case the results will be shown in the list of history of tests.
- Input: None.
 - Output: A Notification from the system to the user.
 - Constraints: Requires an active internet connection.
 - Priority: Mo; must have.

3. Collection of data from the form

- (a) The program must allow the user to fill a form detailing the test he'd like to have done and a request for the sample to be taken at home.
- Input: Bits of information entered by the user in their respective fields on the GUI.
 - Output: Data formatted in JSON.
 - Constraints: Insufficient data entered by the user.
 - Priority: Mo; must have.

4. Processing of collected data

- (a) The program must send an API data request with the correct permission, and parameter in order to process it in the database.
- Input: Collected data formatted in JSON.
 - Output: Processed data to be stored in the database.
 - Constraints: The data should be secured with its designated user. A Token/Key technique is used to represent the claim between the two parties.
 - Priority: Mo; must have.

5. Storage of data

- (a) The program must properly store the data received in the database.
- Input: Processed data.
 - Output: Data stored in the database.
 - Constraints: The data should be accurately stored.
 - Priority: Mo; must have.

3.3.2 Specimen Tracking System

1. System Related Requirements

- (a) The system must provide real-time tracking of the couriers' vehicles.
 - Input: Latitude and longitude provided by a GPS device.
 - Output: An accurate position of the courier.
 - Constraints: Connection to the GPS device being lost.
 - Priority: Mo; must have.
- (b) In case of an emergency, the system must notify the nearest courier to the impaired one. In order to take the best measures.
 - Input: Notification from an impaired courier.
 - Output: Notification to the nearest courier to the incident.
 - Constraints: Need for an active internet connection to send and receive notifications.
 - Priority: Mo; must have.

2. User Related Requirements

3. Admin Related Requirements

4. Courier Related Requirements

3.4 Non-Functional Requirements

3.4.1 Mobile Application and Website

1. Usability

All system features shall be simple, transparent, and user-friendly to all archetypes of users. A clean GUI and clear indicators and labels on all text-fields in the forms will allow users to fill them as conveniently as possible.

2. Performance

The program will perform swiftly and smoothly to make sure the forms are submitted and processed in a timely manner so appointments can be made while delivering results as soon as they're ready.

3. Security

There will be multiple functionalities in place to provide a secure program that assures the users of their information's safety. And they're as follows

- No user can access any information unless they login. This will prevent users from gaining access to other users' personal information.

- All passwords will be hashed and stored in a protected database to prevent any leaks of information. The un-reversible hashing will provide additional security in case of a security breach.

4. Availability

The program will be available for use to fill forms at all times. Exceptions may occur during system maintenance. Which shall be conducted during times of minimum traffic.

5. Portability

The program will be available as an application downloadable for Android and iOS. The program will conjointly be available as a website in like manner.

3.4.2 Specimen Tracking System

1. Usability

The system shall use an unambiguous and user-friendly map to track the couriers distinctly. While sending automatic notifications to couriers in case of an emergency being reported.

2. Performance

The system shall run in real-time to track the couriers with no delays as long as the GPS hardware device is functioning properly.

3. Availability

Unavailability might occur during system maintenance. This should not cause significant hindrance to any operations since the maintenance shall be concluded during off-times for couriers.

4. Maintainability

Any new couriers being added should not cause any complications to the system as a whole. And maintenance shall conclude during couriers' off-time.

3.5 Other Requirements

3.5.1 Hardware Requirements

An arduino based device will be used within the system. Integrating it with the web service developed. The device revolves around the usage of GPS, and GSM, a circuit is designed to track the location of the vehicles using these technologies. The tracking system sends geographical coordinates back to the web server, feeding the data into the cloud. By using this system, we are able to share real time information about transportation. This will also aid the means of transportation by providing shortest paths to a certain coordinate to the user.

GSM module is used in this scenario for sending the coordinates of the vehicle to the server. GPS handles the continuous results of coordinates as a form of string data, known as longitude and latitude. Integrating it with the graphical user interface application, gives us the capability to showcase the location of a vehicle on a map dashboard in real-time.

3.5.2 Software Requirements

1. Recognition Model Requirements

With the addition of an Arduino based tracking device, tracking would be effortless. An API would be handling all the traffic between the server, and the application, hosting all the data on the cloud for ease of access and efficiency.

Python

A high-level programming language which serves a general purpose, dynamic, and is widely used within the scope of data scientists. Python allows the use of a variety of functions included for the utilization of data analysis and machine learning

- **Attributes**

- (a) **Create a CSV output to easily read, and manipulate data in a spreadsheet**
- (b) **Integration of SQL (Read, Write)**
- (c) **Using PySpark to build a machine learning model**
- (d) **NumPy, anmd Pandas Tool for Data Analytics**

2. User Interface Requirements

- **Web Requirements**

The purpose of a website is to engage, inform, and give ease of access for users with just an internet connection. Investing in a website helps invest in the idea of the project and give room for growth. Variety of tools will be used to develop the application such as:

- (a) **Front-End**

- i. **React Javascript**
- ii. **Visual Studio Code**
- iii. **Redux**
- iv. **ESLint**
- v. **Storybook**
- vi. **React Developer Tools**
- vii. **React-Bootstrap**
- viii. **UI Component Libraries**

- (b) **Back-End**

- i. **NodeJS**
- ii. **REST API**
- iii. **SQL (MySQL, or MONGO)**
- iv. **PHP**

- **Application Requirements**

The purpose of a website is to engage, inform, and give ease of access for users with just an internet connection. Investing in a website helps invest in the idea of the project and give room for growth. Variety of tools will be used to develop the application such as:

- (a) **Front-End**

- i. **React Native**
- ii. **Visual Studio Code**
- iii. **Nuclide**
- iv. **Redux**
- v. **ESLint**
- vi. **React Native Tools**
- vii. **React Navigation**
- viii. **UI Component A**
- ix. **AbodeXD**
- x. **Flow**

- (b) **Back-End**

- i. **NodeJS**
- ii. **REST API**
- iii. **SQL (MySQL, or MONGO)**
- iv. **PHP**

3.5.3 Other Supplementary Requirements

- 1. **Internet Connection**
- 2. **Android 5.0+, or iPhone 5s+**
- 3. **RAM of 1GB or larger**
- 4. **Must have an integral GPS**

Chapter 4

System Design

4.1 Logical Model Design

4.2 Physical Model Design