



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 Sheetsage. (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	8
1.4 Document Outline	8
2 Project Plan	9
2.1 Project Deliverables	9
2.2 Project Tasks	9
2.3 Gantt Chart	9
2.4 Roles and Responsibilities	9
2.5 Cost Estimation	9
2.6 Project Management Tools	9
3 Requirements Specification	10
3.1 Stakeholders	10
3.2 Platform Requirements	10
3.2.1 Mobile Application and Website	10
3.2.2 Specimen Tracking System	11
3.3 Functional Requirements	12
3.3.1 Mobile Application and Website	12
3.3.2 Specimen Tracking System	14
3.4 Non-Functional Requirements	15
3.4.1 Mobile Application and Website	15
3.4.2 Specimen Tracking System	16
3.5 Other Requirements	16
3.5.1 Hardware Requirements	16
3.5.2 Software Requirements	16
3.5.3 Other Supplementary Requirements	16
4 System Design	17
4.1 Logical Model Design	17
4.2 Physical Model Design	17

List of Figures

List of Tables

3.1 Stakeholders	10
----------------------------	----

Chapter 1

Introduction

1.1 Overview

1.2 Problem Statement

1.3 Related Work

1.4 Document Outline

Chapter 2

Project Plan

2.1 Project Deliverables

2.2 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

3.5.2 Software Requirements

3.5.3 Other Supplementary Requirements

Chapter 4

System Design

4.1 Logical Model Design

4.2 Physical Model Design