Princess Sumaya 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

We'd like to offer our deepest feelings of gratitude to our supervisor, Dr. Mohammad Abo-Snober, for his great efforts throughout this project were of utmost signifiance. We'd also like to thank Dr. Abdullah Darweesh for the invaluable advice and direction he'd provided, Eng. Iyad Al-Dasouqi for the great deal of motivation he'd provided and continues to provide to everybody around him, Eng. Khalid Al-Samhouri for the tremendous work he's put forward, helping us understand the requirements we'd have to meet for a successful project, Eng. Islam Ahmad and the entire RSS team for the treasured knowledge and experience they'd helped us attain in this project. Not to mention our families, friends, and loved ones for their enchanting prayers and graceful wishes.

Abstract

The efficient use of new and upcoming technologies is significant when developing software. In this project we'll be making use of a real-time tracking system to maximise productivity in a courier-based system currently operated by MedLabs to process all the different medical samples. Overall, two applications will be developed, both of which for MedLabs. The first application will be used by patients and will provide all the necessary services typically present in similar applications. The patients will also be able to fill a form requesting a medical sample to be taken from them from their homes. This is currently done through the call-center at MedLabs. The second application will be used by the couriers and the admins with each being able to make use of functionalities relevant to their roles. A real-time tracking system that uses GPS technology developed by fellow students at PSUT's King Abdullah II School of Engineering will be used by the admins to assign tasks to the couriers as effectively and efficiently as possible.

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

List of Figures

List of Tables

Chapter 1

Introdution

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 nad 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 am 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 arragenment for their samples to be taken at home. Which regularly leads to heavy load on the call-center subsequently leading to customer dissastifcation. 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

The Ground Positioning System (GPS) is owned and operated by the United States of America. It's a powerful system of more than 30 satellites orbiting the earth and constantly keeping a steady flow of signals being sent. This allows for their position to be known at all times. Using signals from at least 4 of those satellites, a receiver can deduce its accurate position on the ground.
We'll be incorporating the project our fellow students from PSUT's King Abdullah II School of Engineering have been working on. The project consists of a real-time tracking system that uses GPS to provide the most reliable positioning possible, regardless of whether the user has an active internet connection or not. This system will be built into each of the couriers' vehicles. As long as a straight unobstructed line of sight is maintained between the vehicle and at least 4 satellites, the tracking will not be interrupted.

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

Analysis Phase

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 and hold brainstorming sessions to collect and document user requirements	Week 1	T01
T03	Identify system requirements	Deduce, document, and prioritise all system requirements	Week 2	T02
T04	Learn the required documentation tools	Learn the required tools and programming languages necessary for documentation (Latex)	Week 2	None
T05	Manage system requirements	Use a traceability matrix to manage system/user requirements	Week 2	T02, T03

ID	Task	Description	Duration	Dependencies
T06	Form a development strategy	Research and document hardware and software tools to be adopted in the project. Research and choose the best fitting design model	Week 2	T03

Design Phase

ID	Task	Description	Duration	Dependencies
T07	Design system requirements	Use a wide and applicable array of diagrams to manifest the system's requirements	Week 3	T03
T08	Design user interfaces	Use the appropriate software to desing user interface prototypes	Week 4	T03, T06
T09	Design the database	Define a proper database with apt attributes and constraints	Week 4	None
T10	Design the physical model	Adopt the best suitable software architecture	Week 5	T09, T07

Table 2.2: Project Tasks

2.3 Gantt Chart

Figure 2.1 below depicts the Gantt Chart detailing the timeline for the tasks in this project

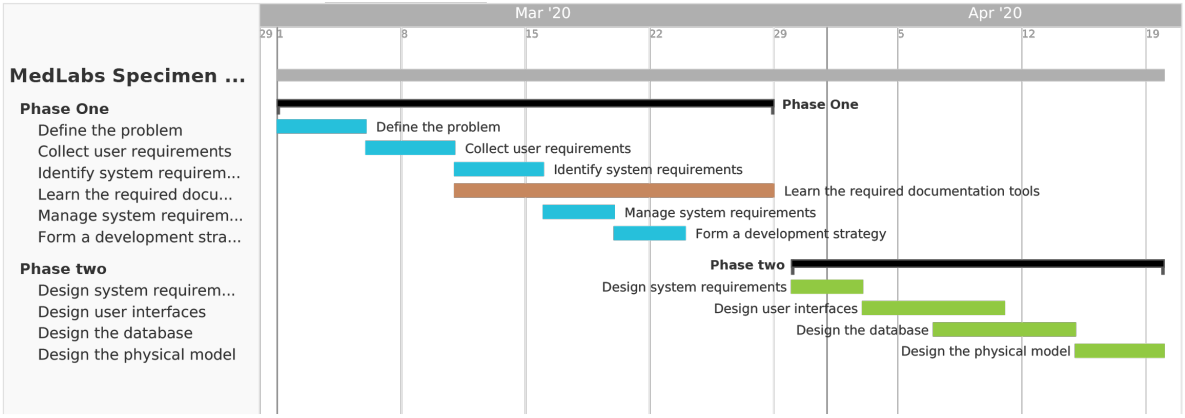

Figure 2.1: Gantt Chart

2.4 Roles and Responsibilities

Abdullah's Responsibilities

ID	Task	Description
01	Define the problem	Define the problem and write the problem statement
02	Collect user requirements	Interview personnel from MedLabs to collect user requirements
03	Documentation	Define all stakeholders, document project tasks and individual roles and responsibilities
04	Risk Assessment	Brainstorm possible risks, their impacts and plan to cope with each risk
05	Identify system requirements	Identify and document the system's functional, non-functional, and platform requirements

ID	Task	Description
06	Learn the required documentation tools	Learn and build a Latex template for the full documentation

Hasan's Responsibilities

ID	Task	Description
01	Define the problem	Define the problem
02	Collect user requirements	Interview personnel from MedLabs to collet user requirements
03	Identify system requirements	Identify and document the system's nonfunctional requirements
04	Identify system requirements	Identify and document the system's software and hardware specifications

Table 2.4: Roles and Responsibilities

2.5 Cost Estimation

2.6 Risk Assessment

ID	Task	Risk and Impact	Portability	Plan to Cope
T01	Define the problem	There's a risk of misspointing the real problem in a system. Accuracy is significant	Highly unlikely	Further interviewing and brainstorming
T02	Collect user requirements	Insufficient data might lead to miss identification of the user / customer's needs and wants	Unlikely	Further interviewing with the user
T03	Identify system requirements	An inaccurate understanding of the user's requirements will lead to incorrect system requirements	Unlikely	Careful reconsideration of the users's requirements

ID	Task	Risk and Impact	Portability	Plan to Cope
T04	Learn the required documentation tools	Time is an important resource that will be wasted if the wrong tools are used	Highly unlikely	Further research of available tools
T05	Manage system requirements	A mismatch between the user's requirements and the system's identifed requirements will cause design problems	Unlikely	A review and a possible edit of the functional requirements to accurately make sure every user requirement is met
T06	Form a development strategy	Choosing the wrong strategy to follow will definitely lead to wasted time and effort	Unlikely	A thorough study of the system's requirements. Research of different development strategies
T07	Design system requirements	The system's requirements must be illustrated as accurately as possible to prevent misunderstanding of the system	Slightly likely	A comprehensive study of the system's requirements
T08	Design user interfaces	The user's wants must be taken into account more than anything in this task	Likely	Further interviewing with the user to properly understand their wants
T09	Design the database	Any flaws or blunders in the design of the database will cause complications on the other levels	Slightly likely	An exahaustive study of the sytem's logical operations and requirements
T10	Design the physical model	The flow of data must be fluid as possible maximise the system's efficiency	Slightly likely	A meticulous study of the system's operations

2.7 Project Management Tools

Storage

A GitHub repository with separate branches was used to keep work synchronised and backed up. A master branch keeps the most important files from being edited improperly while working on separate development branches and merging them to the master branch after careful auditting makes sure everything stays secure and up to date. A Google Drive folder was also used as a back-up storage in case of any emergency. This folder was updated as frequently as possible to ensure further safety.

Planning

A shared Google Keep note was used to keep track of all the different sections of the project that had to be completed. The simple interface of Keep helps keep everything clear and easy to reach. A Google Tasks list was used to keep track of individual tasks assigned to each member of the team.

Charts and Diagrams

TeamGantt was used to create the Gantt Chart illustrated in figure 2.1.

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 vehile 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 admininstrator and the employee will be developed to track all the vehicles.

3.3 Functional Requirements

3.3.1 The Client's Application and Website

1. **Account Related Functional Requirements**

   (a) (1.1.a) The program must allow a new user to create a new account. he 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) (1.1.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) (1.1.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) (1.1.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) (1.1.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) (1.2.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) (1.2.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. item [-] Priority: Mo; must have.

(c) (1.2.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) (1.3.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) (1.4.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) (1.5.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 Admin and Courier Application

1. **System Related Requirements**

   (a) (2.1.a) 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.

(b) (2.1.b) The program must allow couriers and admins who entered a password / username combination that matches one in the database to access their accounts and be directed to the proper interface depending on their job title.

- Input: A valid username and a valid password.
- Output: The user will be directed to the interface related to their job title.
- Constraints: The combination of the username and the password entered must match the combination stored in the database.
- Priority: Mo; must have.

2. **Admin Related Requirements**

(a) (2.2.a) The admin can allocate tasks to the couriers on his dashboard which displays all tasks in detail.

- Input: Input by the admin on the GUI.
- Output: Task allocation and editting.
- Constraints: Need for an active internet connection to send and receive information.
- Priority: Mo; must have.

(b) (2.2.b) 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 on admin's map.
- Constraints: Connection to the GPS device being lost.
- Priority: Mo; must have.

(c) (2.2.c) The system must allow the admin to create a new account using a username and a password.

- Input: A valid username, and a valid password.
- 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.

3. **Courier Related Requirements**

(a) (2.3.a) The courier can view tasks assigned to him in detail on his dashboard.

- Input: None
- Output: Tasks allocated to the courier.
- Constraints: Need for an active internet connection to send and receive notifications.
- Priority: Mo; must have.

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-reversable 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 Specification

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**

  – React Javascript

    React JS Library will be used to develop the interactive UIs and Functionality for the website

  – Redux

    Maintain the state of the application in a single tree object

  – React-Bootstrap

    Bootstrap will be used as the main front-end framework to build the UI components

(b) **Back-End**

- NodeJS

  The server side JavaScript environment will be the core back-end for the application

- REST API

  HTTP Requests made by the application will be handled by a RESTful API

- SQL (MySQL, MONGO, or Firebase)

  Real-time tracking information will be stored on a database such as MySQL, or Firebase to be of use

- **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**

  - React Native

    React JS Library will be used to develop the interactive UIs and Functionality for the mobile application cross platform

  - Redux

    Maintain the state of the application in a single tree object

  - React Navigation

    Handling the routing and navigation for the react application

  - React-Bootstrap

    Bootstrap will be used as the main front-end framework to build the UI components

(b) **Back-End**

- NodeJS

  The server side JavaScript environment will be the core back-end for the application

- REST API

  HTTP Requests made by the application will be handled by a RESTful API

- SQL (MySQL, MONGO, or Firebase)

  Real-time tracking information will be stored on a database such as MySQL, or Firebase to be of use

3. **Tools**

- AdobeXD

  AdobeXD is a powerful platform that helps to create designs for website, and mobile apps

- UI Component Libraries

  A variety of other libraries will be used for components

- React Native Tools

  An extension offering a helpful environment to develop React Native Projects

- React Developer Tools

  Chrome DevTools extension for the open-source React JavaScript library. It allows you to inspect the React component hierarchies

- Storybook

  Storybook is an open source tool for developing UI components in isolation for React

- Visual Studio Code

  Source code editor tool used for building projects

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 integal GPS

Chapter 4

System Design

4.1 Use-Case Diagrams

4.1.1 Employee App

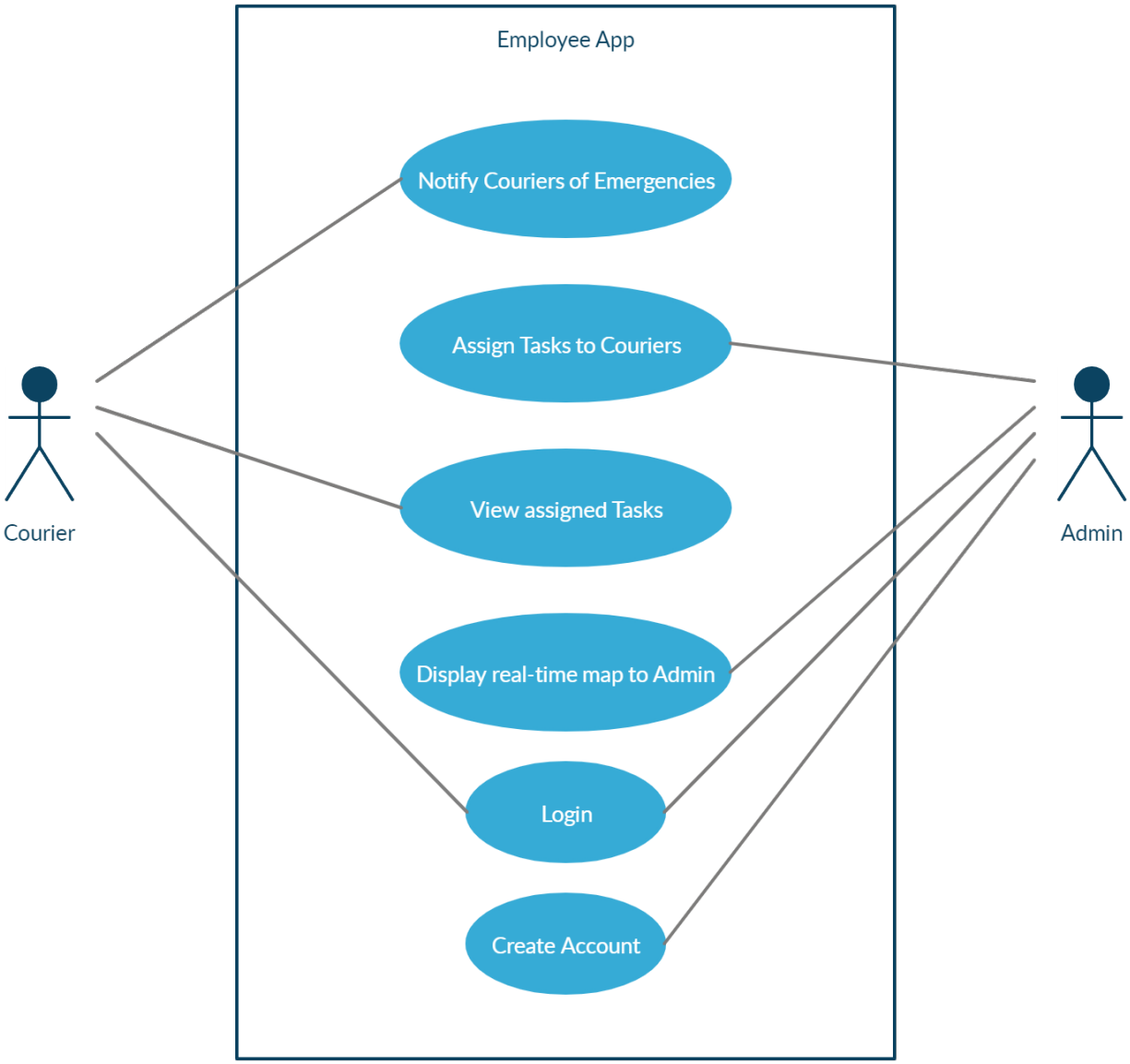

Figure 4.1: Employee App Use Case Diagram

1. Use Case: Notify Couriers of Emergencies
    - Actor: The Admin and the Courier.
    - Purpose: Notify Couriers of emergencies to take the proper action.
    - Success Scenario: In case of an emergency occuring with a certain courier. That courier till notify the system of said emergency. The system will determine the best course of action. E.g. a nearby courier takes the samples from the impaired courier. The system then notifies that courier of the actions to be taken.
    - Type: Primary
    - Cross References: Functional Requirement 2.1.b
2. Use Case: Assign Tasks to Couriers
    - Actor: The Admin and the Courier.
    - Purpose: Assign different tasks to the couriers.
    - Success Scenario: Using the real-time tracking map provided to the admin. They assign tasks of varying types to the couriers available by using the admins' dashboard.
    - Type: Primary
    - Cross References: Functional Requirement 2.2.a
3. Use Case: Display Real-time Map to the Admin
    - Actor: The Admin.
    - Purpose: Display real-time map to the admin
    - Success Scenario: The system will provide the admins with a real-time tracking map. This will aid the admin when assigning tasks to available couriers.
    - Type: Primary
    - Cross References: Functional Requirement 2.2.b

4.1.2 Client App

- Use Case: Send notifications
    - Actor: The System and the Patient.
    - Purpose: Notify patients with important information.
    - Success Scenario: The system will also send 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. When a test the patient had requested has been completed and the results are ready to be viewed. The system here will notify the user of the particular test that has been completed. The notification will successfully reach the patient if they have an active internet connection.
    - Type: Primary
    - Cross References: Functional Requirement 1.2.a

- Use Case: Fill Form
  - Actor: The Patient
  - Purpose: Fill forms to request samples to be taken at home.
  - Success Scenario: The patient fills a form on the application. The form includes a wide range of necessary information regarding the test to be taken. Such as time, location, and surely the type of test to be taken. The form is submitted, processed and stored properly in the database.
  - Type: Primary
  - Cross References: Functional Requirement 1.3.a

4.2 Logical Model Design

4.3 Physical Model Design

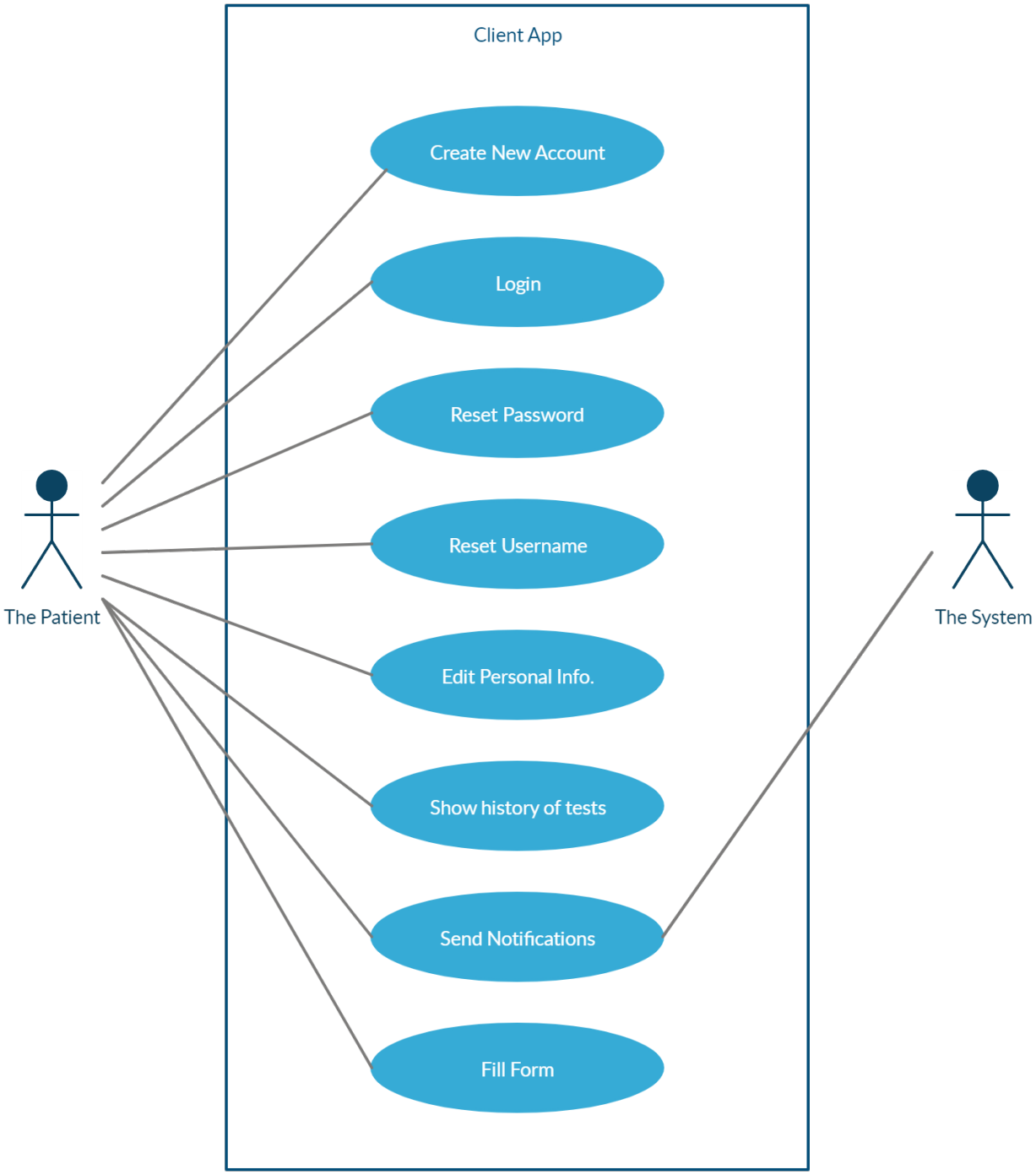

Figure 4.2: Client App Use Case Diagram