Ambulance Management System

Submitted By:

UMAIR AHMED, 2020-CS-3
KASHIR SAEED, 2020-CS-27
M. Farukh Haider, 2020-CS-45

Submitted To:

Mr. Samyan Qayyum Wahla

For Fulfillment of:

CS 261 Data Structure and Algorithm

Department of Computer Science

University of Engineering and Technology, Lahore

Contents

Introduction

Purpose

Ambulance management system will help us to automate the ambulance service provide by the Health Care Department. This will help us to manage the record and schedule the ambulances for the emergency scenarios. Using this, we can manage drivers and ambulances effectively and can provide basic life support facilities in a much better way.

Scope

Scope of Ambulance Management System is very vast. With the help of this system we can provide the services of ambulance at any corner of the city. This ambulance management system provides a solid plate form to all the patients for the treatment so that all the patients can be cured as soon as possible. This is an organized system which is very disciplined in performing its duties i.e such that providing general health facilities to the local public.

Project Overview

Table1: Project overview

Project Overview	Detail
Project Name	Ambulance Management System
Project Type	Desktop Application
Project Language	C#

System Overview

In this project, we will be able to add the new employees and new buses. We are also able to register the hospitals to know the number of beds available in the region. When new shift will start, we first role call the attendance that will tell us the number of drivers, co-drivers and buses that are in the working condition. Then, we will make the groups of all the present staff and each group contains one Driver, one Co-Driver and one Bus. When we will receive the call, we take some basic information from the caller i.e the point of arrival and the current situation of the patient. We will also automatically get the current location of the caller by the assistance of the PTA. After getting this information we will check whether a group is available for the rescue. If yes, then we search the availability of the bed from the registered hospitals. If bed is also available, then we assign the current group to that patient. After that we will find the shortest path to reach the patient by the assistance of GPS. Then, the rescue group will receive the patient and dispatch him to the current available hospital. At the end we will upload the ECF form on the rescue web server so that the warden of the patient can search and trace their patient on the website.

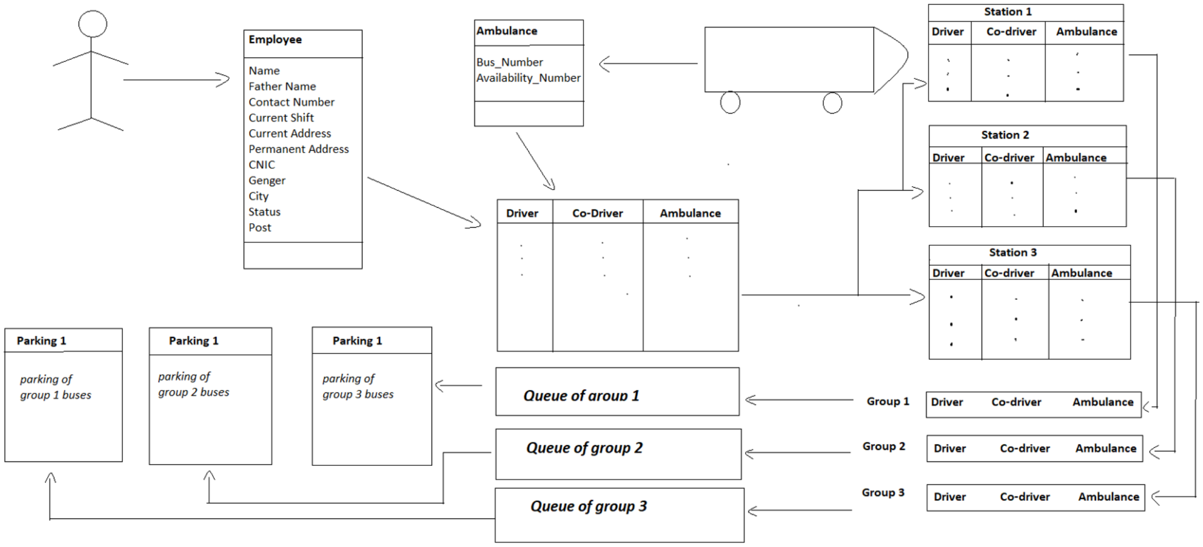


Figure 1: Flow of project

Acronyms and Abbreviations

Table2: Acronyms

Acronym/Abbreviation	Meaning
CTWO	Computer Wireless Operator
WO	Wireless Operator
PTA	Pakistan Telecommunication Authority
GPS	Global Positioning System
ECF	Emergency Control Form
FIFO	First In First Out

Operational Details

Employee

Add

Admin of the application will be able to add the new employee. The attribute of the new employee are:

- Name
- Father Name
- Contact Number
- Current Address
- Permanent Address
- CNIC
- Gender
- City
- Status
- Post
- Shift
- ID

Admin will get the data and automatically the objects of the employee class will be created and saved.

Update

Admin will also be able to update the record of the employee. Firstly, he will search the employee whose data he wants to update. Then, he will update the attributes of the employee.

Delete

Admin will also able to delete the employee from the application. In the same way, as in the update he first, search the employee and then delete it.

BUS

Add

Admin of the application will be able to add the new bus. The attribute of the new bus are:

- Name
- Model
- Status
- Number

Admin will get the data and automatically the objects of the BUS class will create and added in the tree.

Update

Admin will also be able to update the record of the bus. Firstly, he will search the bus whose data he wants to update. Then, he will update the attributes of the bus.

Delete

Admin will also able to delete the bus from the application. In the same way, as in the update he first, search the bus and then delete it.

HOSPITAL

Add

Admin of the application will be able to register the hospital. The attributes of the hospital are:

- Name
- Total Beds
- Address
- Occupied Beds

Admin will get the data and automatically the objects of the Hospital class will create and saved.

Update

The incharge of the hospital emergency ward will be able to update the record of the hospital ie: update the number of occupied beds in the ward.

Delete

Admin of the application will be able to delete the hospital. Firstly, the desired hospital will be search then it will be deleted.

Attendance

Employee

When it is going to be the time for any shift we first take the attendance of all the employees and place the present employees in the different queue according to the job type i.e Driver and Compoder.

Buses

In this operation we check the status of the buses whether the bus is in the working condition or not. If bus is in the working condition then, we place it in the queue.

Prepare Shift

After the operation of the attendance we prepare the shift group in order to rescue the people. Each working group has following attributes

- Group Id
- Compoder
- Driver
- Bus

When all the groups are done then we place them in the form of the queue and the buses are also parked in the in same sequence of the queue in the active position.

Update Shift

When it is going to be the time of a new shift admin will update the shift and load the record of the hospitals in the stack that are available for receiving the patient.

Call from Patient:

Know moving towards the main operation of the application, when caller calls the rescue number, it is first directed to the CTWO. He will check the authentication of the emergency and fill the ECF form then direct the call to the WO. The WO will get the location of the caller by the assistance of PTA and calculate the shortest path to reach the required destination. This task will be done by the assistance of the GPS. In the meanwhile, he will check the availability of the bed in the stack of the hospitals. Then, the rescue group is dequeue from the queue of the shift group and the information is given to the current group i.e pickup location, address of the hospital and the shortest path to reach the pickup point and the hospital. After dispatching the patient to the hospital, the rescue group will give the complete information to the WO relating to the ECF form. Then, WO report to the CTWO that the patient is successfully rescued and ECF form will be uploaded on the web server.

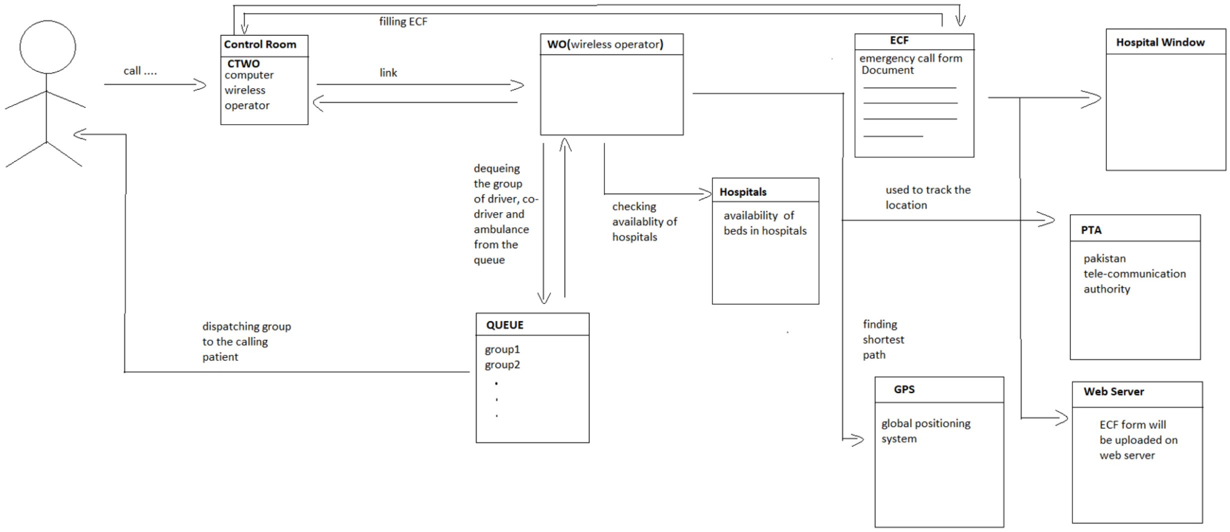


Figure2: Flow of rescue operation

Use Cases

Use Case 1

Operations: Add Employee, Add Bus, Add Hospital.

Actor: Admin
Scenario: When admin needs to add new data in the application i.e Employee, Bus and Hospital. He will select the option to add the new data like

- Add Employee
- Add Bus
- Add Hospital

The input field form will be shown on the screen, admin will input the required data and press the add button in this way new data will be added in the application.

Use Case 2

Operations: Update Employee, Update Bus, Update Hospital.
Actor: Admin**.**
Scenario: When admin needs to update the data in the application i.e Employee, Bus and Hospital. He will select the option to update the data like

- Update Employee
- Update Bus
- Update Hospital

The update form will be shown on the screen, admin will search the Employee, Bus or Hospital whom data he wants to update then all the attributes of the searched data will be shown on the screen and he can change them and then press the update button in this way data will be updated in the application.

Use Case 3

Operations: Delete Employee, Delete Bus, Delete Hospital.
Actor: Admin.
Scenario: When admin needs to delete the data from the application i.e Employee, Bus and Hospital. He will select the option to delete the data like

- Delete Employee
- Delete Bus
- Delete Hospital

The delete form will be shown on the screen, admin will search the Employee, Bus or Hospital whom data he wants to delete then searched data will be shown on the screen and he can delete the data by pressing the delete button in this w` ay data will be deleted from the application.

Use Case 4

Operations: Shift Attendance, Shift Prepare.
Actor: Admin.
Scenario: When it is going to be the time of the shift update, admin will roll call the attendance of the employees and all the buses. Then, he will save the data in the queue. After roll calling the attendance admin will make the groups. Each group contains Driver, Compoder and Bus. Then, we place all these groups in the queue and dequeue the group at the emergency time.

Use Case 5

Operations: Call from Patient.
Actor: CTWO.

Scenario: When caller dial the rescue number, the call will be directed to the CTWO. He checks the authentication of the emergency and direct the call to WO. He will fill the ECF, get the address of the caller from PTA, check the availability of the bed in the registered hospitals and get the shortest path by the assistance of the GPS system to reach the location of the caller. After all this procedure, he will dequeue the rescue group from the queue and dispatch it to the patient.

Use Case 6

Operations: ECF uploading.
Actor: CTWO.
Scenario: When the patient is rescued then in order to maintain the record we will concatenate the data of the patient i.e ECF form and the group who rescued the patient and the information of the hospital in which we dispatch the patient and save this information and also upload it on the webserver for the convenient of the warden of the patient.

Data Structure along Use Cases

Use Case 1

Tree
In order to save the data of the employees we will use the tree data structure.
Queue
In order to save the data of vans we will use queue data structure.
Linked list
In order to save the data of hospitals we will use the doubly linked list.

Use Case 2

Tree
In order to update the data of the employee we will use DFS to search the data from the tree.

Use Case 4

Queue
After taking the attendance we will put the present employees and he vans in the queues and then dequeue one by one in order to make the groups then we also place each group in the queue.

Use Case 5

Queue
When more than one calls are received simultaneously we hold them in the queue and get the query in FIFO order.
Dijkstra’s Algorithm
In order to find the shortest path, we will use dijkstra’s algorithm.
Stack
We put the hospital in the stack in decreasing order of the distance from the patient. Then we pop it one by one to check the availability of the bed in the hospital and assign it to patient.

Use Case 6

Tree
In order to save the record for the rescue we will use the tree data structure.

Project Plan

Summary

We have planned to implement the project by dividing each task in three equal parts so that, none of our group member will be overloaded. Our aim is to get the contribution of each member in every task so that, every group member will be familiar with the implementation of each and every portion of the project.

Steps to Complete Project

- Generate Data
- UI Implementation
- Codes of data structures
- Integration of code with UI
- Testing
- Remove Bugs
- Final Report

Generate Data

Till now we have decided to generate the random data to proof test the project. We will make different functions that will randomly generate the different attributes of the data i.e patient name, phone number and address.

UI Implementation

User interface provides a user means what he wants from the application. We have planned to make our application more user friendly and easy to interact. So, our wireframes are easy to understand. We decided to make the interface in the VS using .Net Framework and our every group member will try his best to equally contribute in this implementation. User interface of our project is according to the above given wireframes. There are round about our 15-20 wireframes we divide them equally and each group member will implement 6 of them.

Code of Data Structures

Next we decided to implement the data structures. The data structures we are using in our project are

- Doubly Linked List
- Stack
- Queue
- Tree
- Graph

We implement all these data structure and the contribution of every group member in this section must be required because this will be the main section of our project and every group member must know the complexities of this section. The implementation scheme is given below.

M. Farukh Haider: Graph
Kashir Saeed: Doubly Linked List, Stack
Umair Ahmed: Queue, Tree

Integration of Code with UI

This is a last step to bring the project in the working position. Actually this section is about to concatenate the front end implementation and the back end implementation and we have decided that everyone will concatenate the data structure implemented by him.

Testing

Our next section is about the testing of the application for this purpose we will generate the dummy data. This section is very important for the project and we have decided that all the group members will test the application separately. Our testing section will examine the functionality, security, usability and stability of the application. We will try to test the project at the end conditions.

Remove Bugs

After the test section we will be familiar with bugs, errors and the limitations of our project. Then collectively we will debug the application, everyone will debug the portion of the code implemented by him and try to remove the errors and bring the application the maximum perfection level.

Final Report

At the end come to the report section of the project and our every group member is going to take part in this section because everyone will report about his own implementation in the project then at the completion of the report every group member will proof read the report and then our project will be ready to submit.

Object Oriented Model

Employee Class

This class contains the data related to the two types of employees.ie: the driver and the compoder. This class has many relations with other classes. It has inheritance relation with Driver class and compoder class. It also has an aggregation relation with Bus class and composition relation with Rescue Group class. There is a separate Employee CRUD class which contains, all the CRUD operations.

Bus Class

Basically, this class contains all the data related to the bus. It has an aggregation relationship with the employee class. This class also has a composition relation with the Rescue Group class. There is a separate Bus CRUD class which contains all the CRUD operations.

Rescue Group Class

Basically, this class contains the objects of other classes. It contains an employee and an ambulance as an attribute. It has a composition relation with both Employee class and Bus class. This class contains several groups of employees and ambulances. Whenever some emergency call is received by the CTWO, the WO assigns some of the duties to one of these groups. This group rescues the patients and the patients are dispatch to the hospitals.

Hospital Class

This class contains all the attributes of the hospitals. It has a separate Hospital CRUD class which contains all the CRUD operations ie: add, remove, search, update. These operations will be implemented on all the objects of the hospitals.

Driver and Compoder Classes

Both of these classes has an inheritance relation with the employee class. Employee class is the parent of both of these classes. As we know that these classes are the child classes of employee class so they inherit all the properties of the employee class and futher extends the employee class. These classes do not contain any operations because they inherit all the CRUD operations from their parent class.

Rescue Case Class

This class contains an object of ECF class, Bus class and Hospital class as an attribute. It also has an attribute of case ID. This ID contains the case number which was rescued by the rescue group.

ECF Class

This class contains the information of the patient. For example, it contains all the personal information of the patient. It also contains the attribute of the pickup address from where the patients will be received and also destination address where the patients will be delivered.

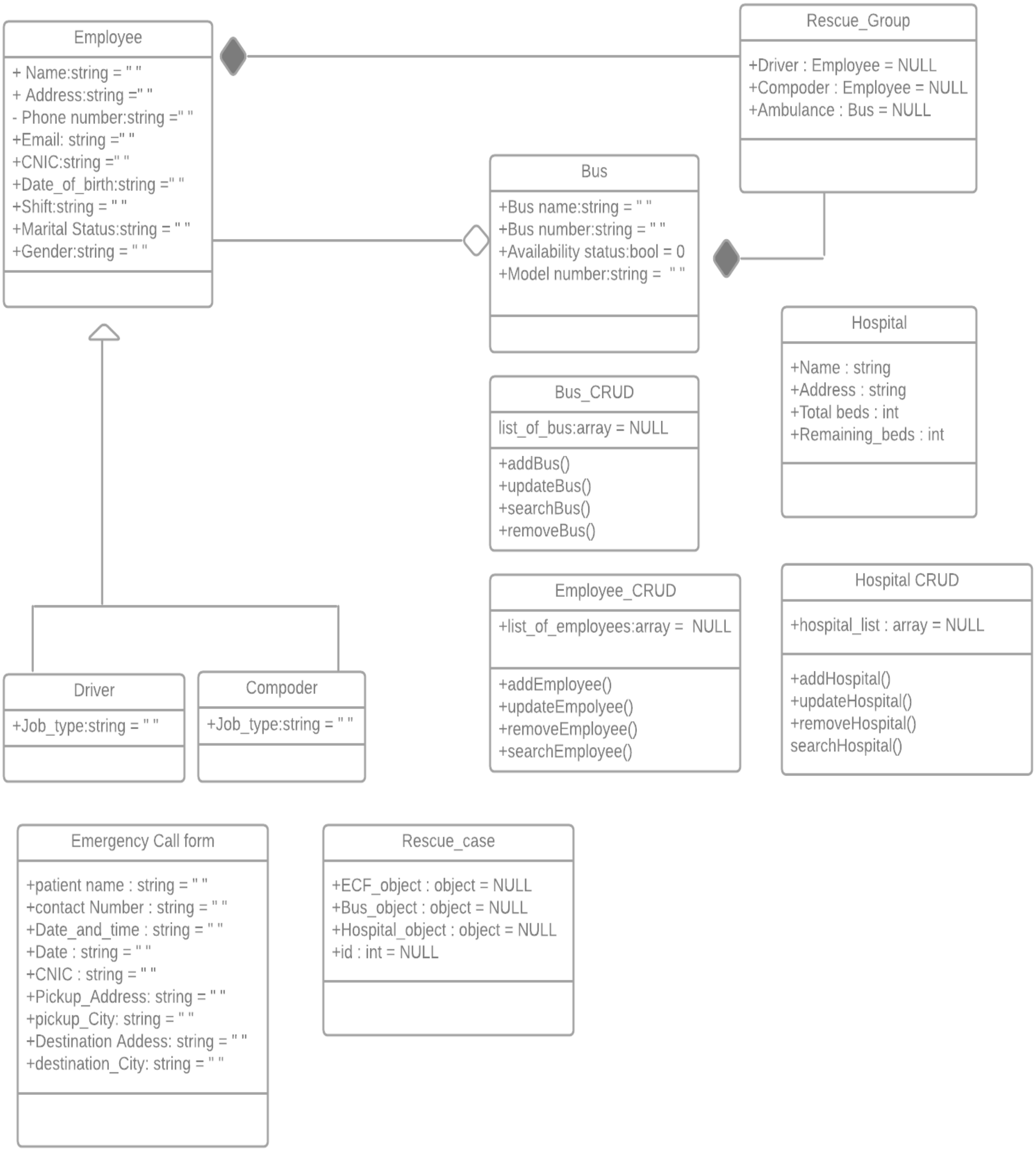
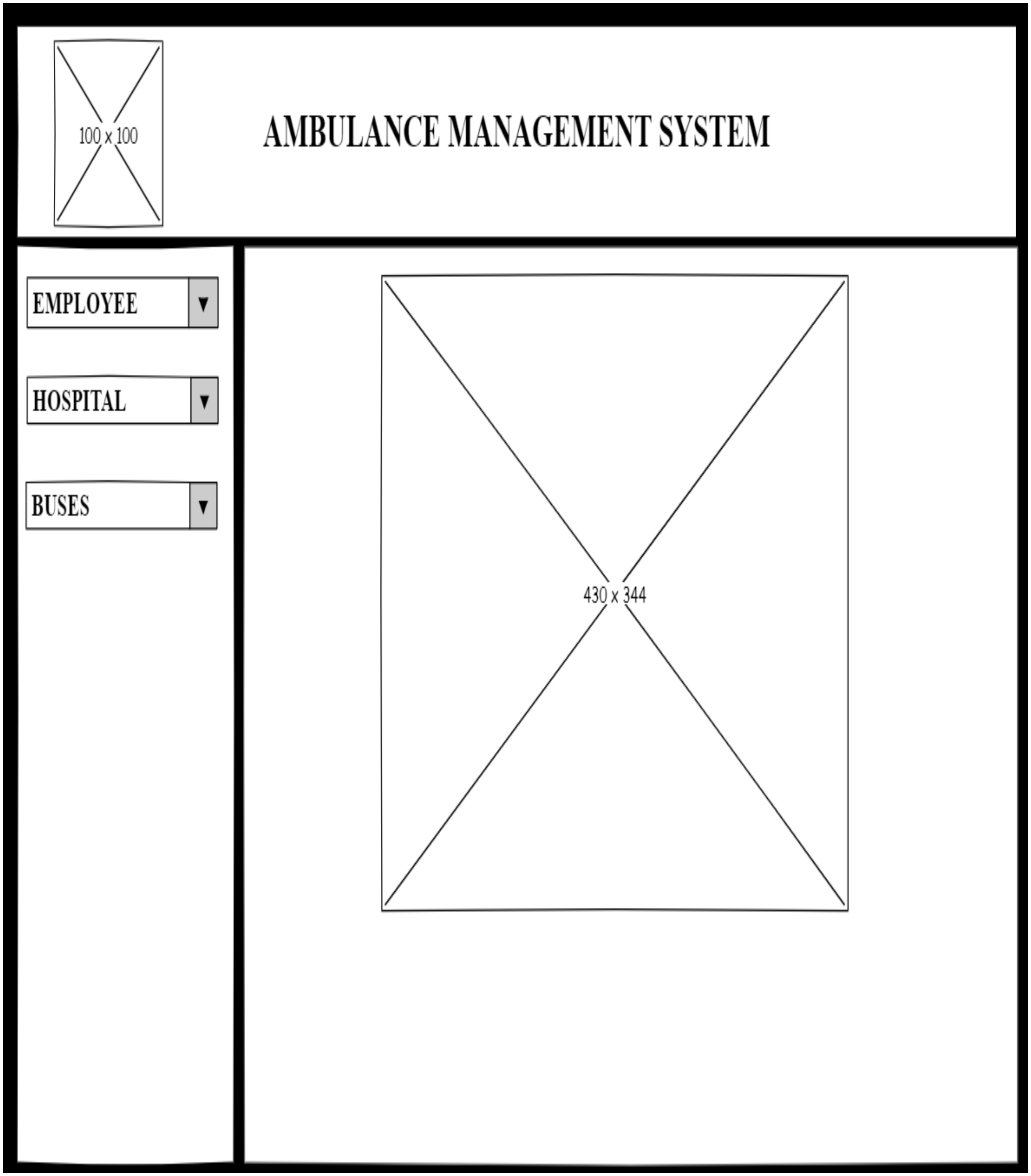


Figure 3: Object Oriented Design

Wireframes

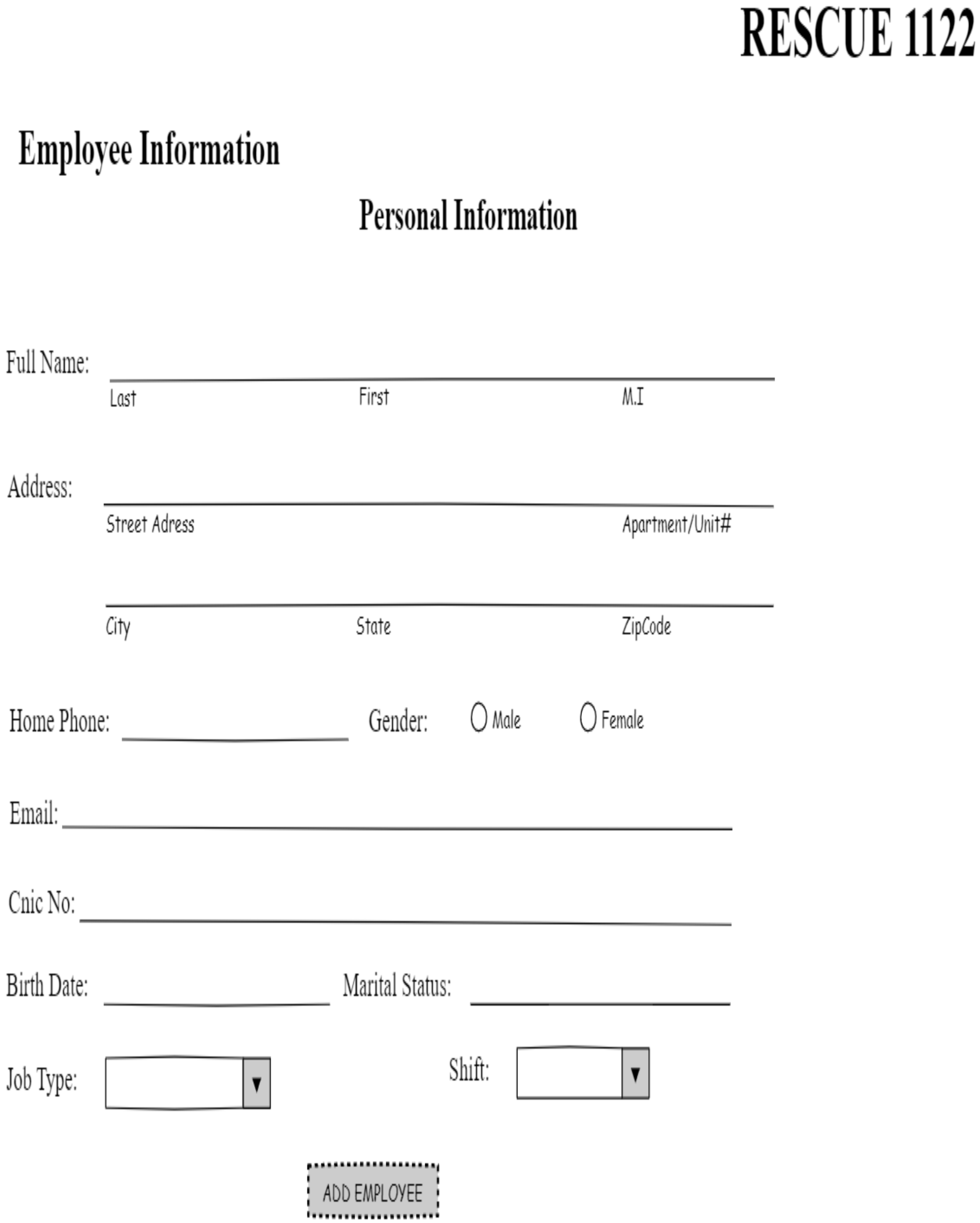
Ambulance Management System Window:

At the top left corner, there will be the logo of our project. There is a navigation bar on the left side which contains the three combo boxes. These three combo boxes are of buses, employees, and hospitals. Each of these combo boxes contains all the CRUD operations ie: add, delete, search, retrieve etc.



Employee Information Window:

This window contains all the information of the employee. This form is useful for only a new employee. As the new employee is hired for this job, he has to provide all his personal information which is mentioned in the window. As the **Add Employee** button is pressed, then the data of the new employee is added in the tree.



RESCUE 1122

Employee Information

Personal Information

Full Name: ______________________________
Last First M.I

Address: ______________________________
Street Adress Apartment/Unit#

____
City State ZipCode

Home Phone: ____________ Gender: ○ Male ○ Female

Email: ______________________________

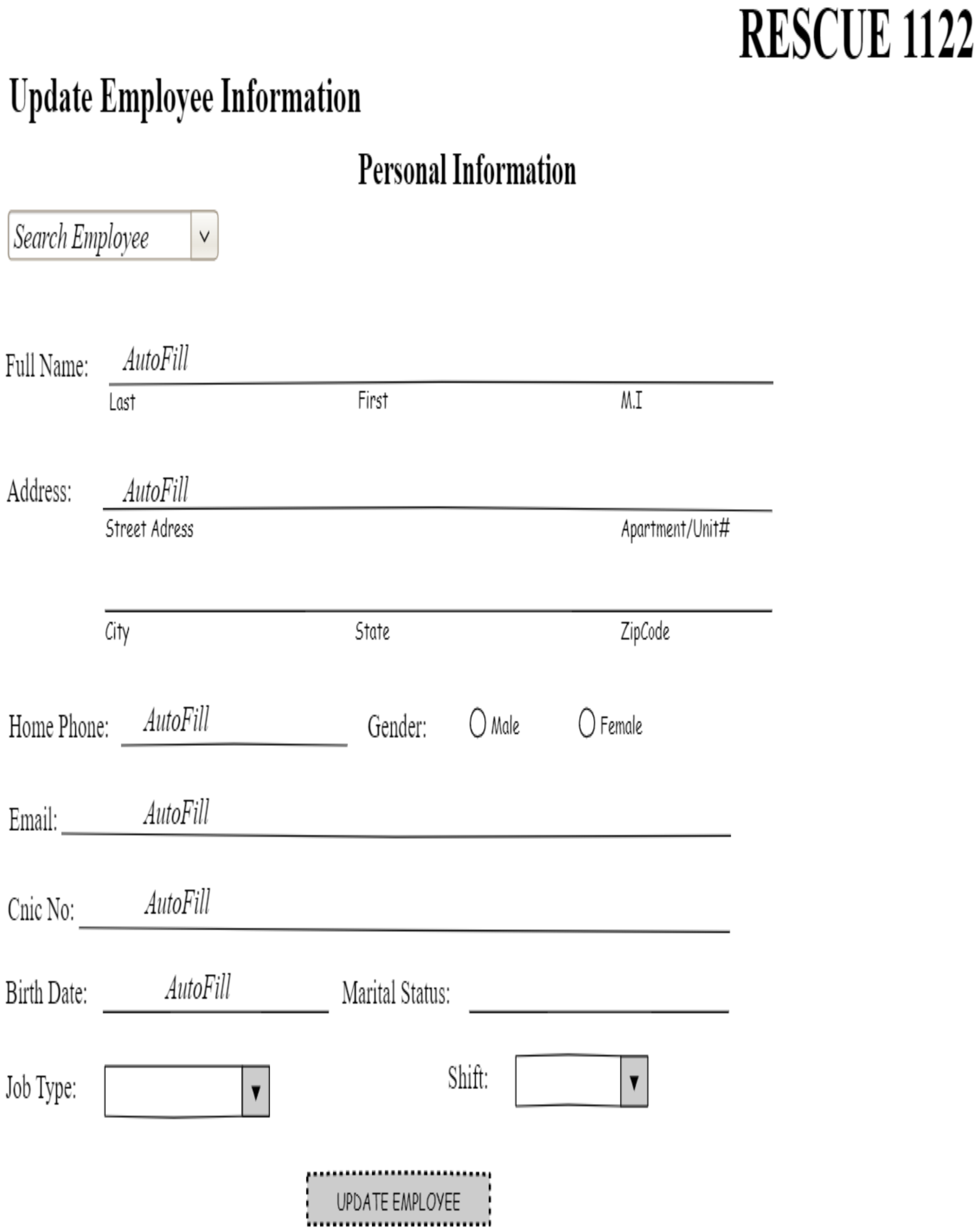
Cnic No: ______________________________

Birth Date: ____________ Marital Status: ____________

Job Type: ▼ Shift: ▼

Update Employee Window:

This window also contains the search panel at the top of the page. The admin will search the Employee which he wants to update. As the Employee is selected, then rest of the information is automatically filled. Then, the admin will change the information which he wants to change. As the **Update Employee** button is pressed, then as a result the data of the employees in the tree is updated.



RESCUE 1122

Update Employee Information

Personal Information

Search Employee v

Full Name: AutoFill
Last First M.I

Address: AutoFill
Street Adress Apartment/Unit#

City State ZipCode

Home Phone: AutoFill Gender: ○ Male ○ Female

Email: AutoFill

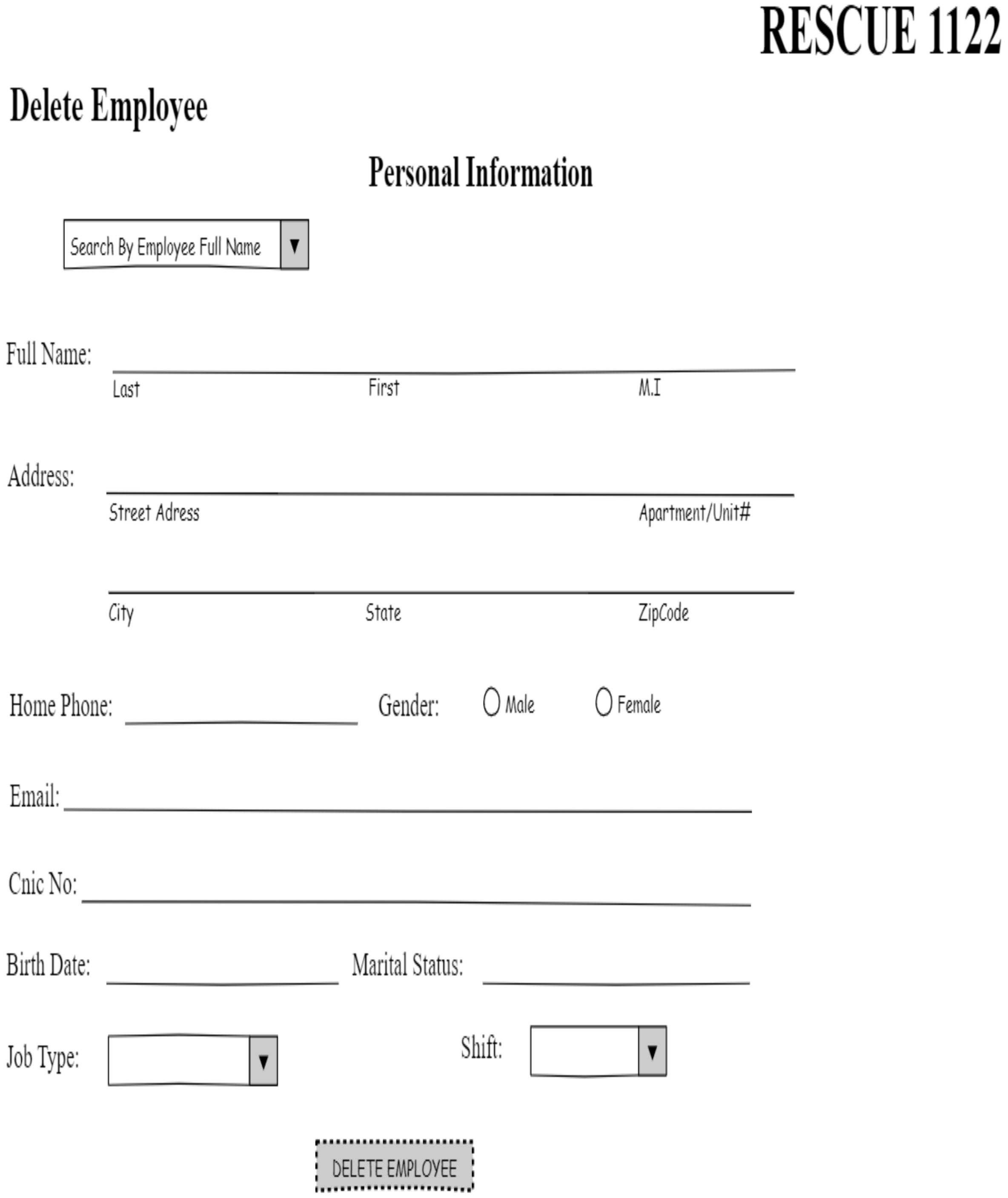
Cnic No: AutoFill

Birth Date: AutoFill Marital Status:

Job Type: ▾ Shift: ▾

Delete Employee Window:

This window also contains a search panel. The admin searches the employees which he wants to delete. When he will select the employee, rest of all the information regarding the employee will be automatically filled. The employees will be deleted from the tree as the **Delete employee button** is pressed.



RESCUE 1122

Delete Employee

Personal Information

Search By Employee Full Name ▼

Full Name: ______________________
Last First M.I

Address: ______________________
Street Adress Apartment/Unit#

____
City State ZipCode

Home Phone: __________ Gender: ○ Male ○ Female

Email: ______________________

Cnic No: ______________________

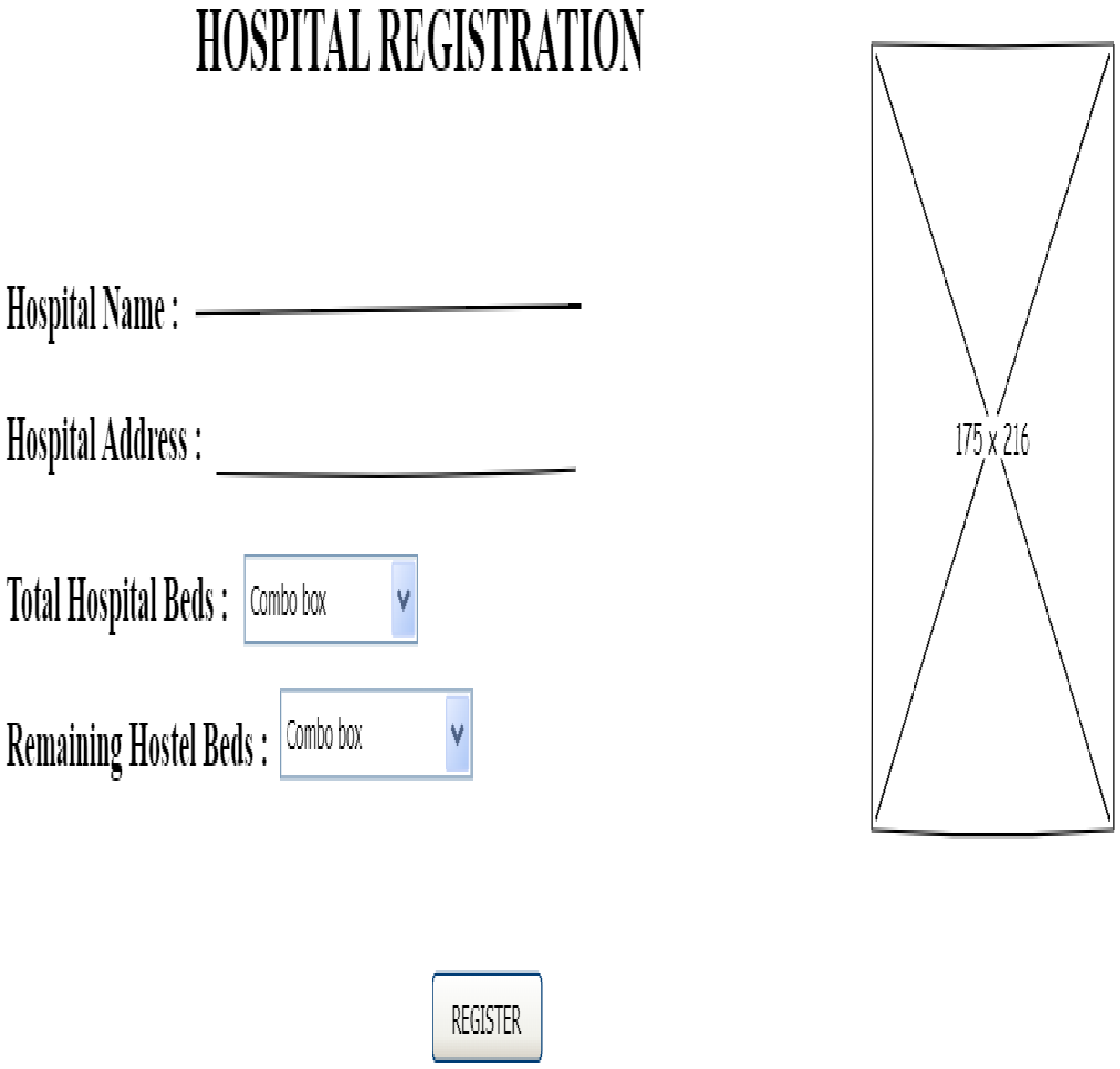
Birth Date: __________ Marital Status: __________

Job Type: ▼ Shift: ▼

DELETE EMPLOYEE

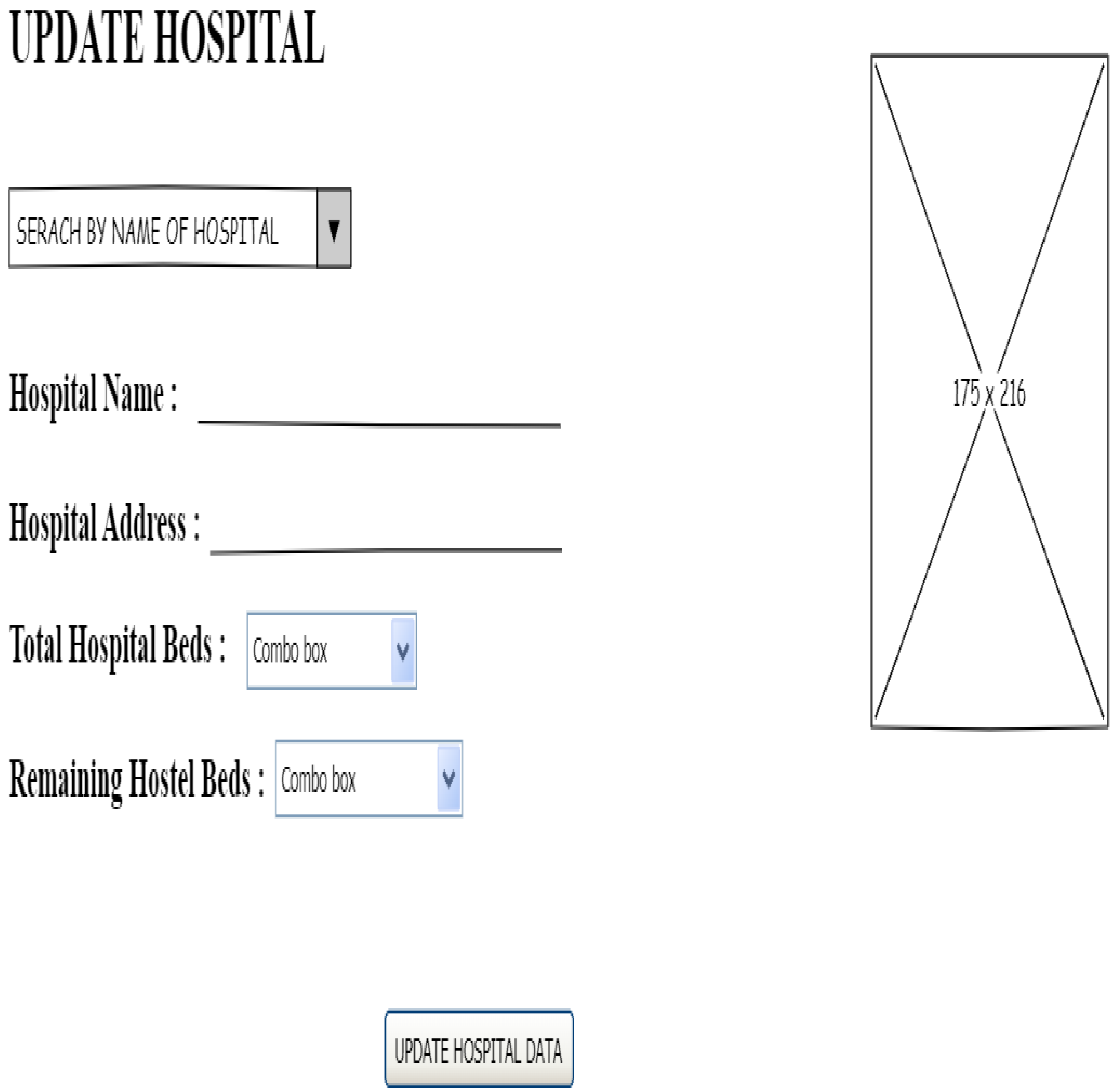
Hospital Registration Window:

Whenever some new hospital will be built in some area, the admin will fill all the information regarding that particular hospital which is mentioned in this window. As the **Register** button is pressed after filling all the information, than, a new hospital will be added is in their record.



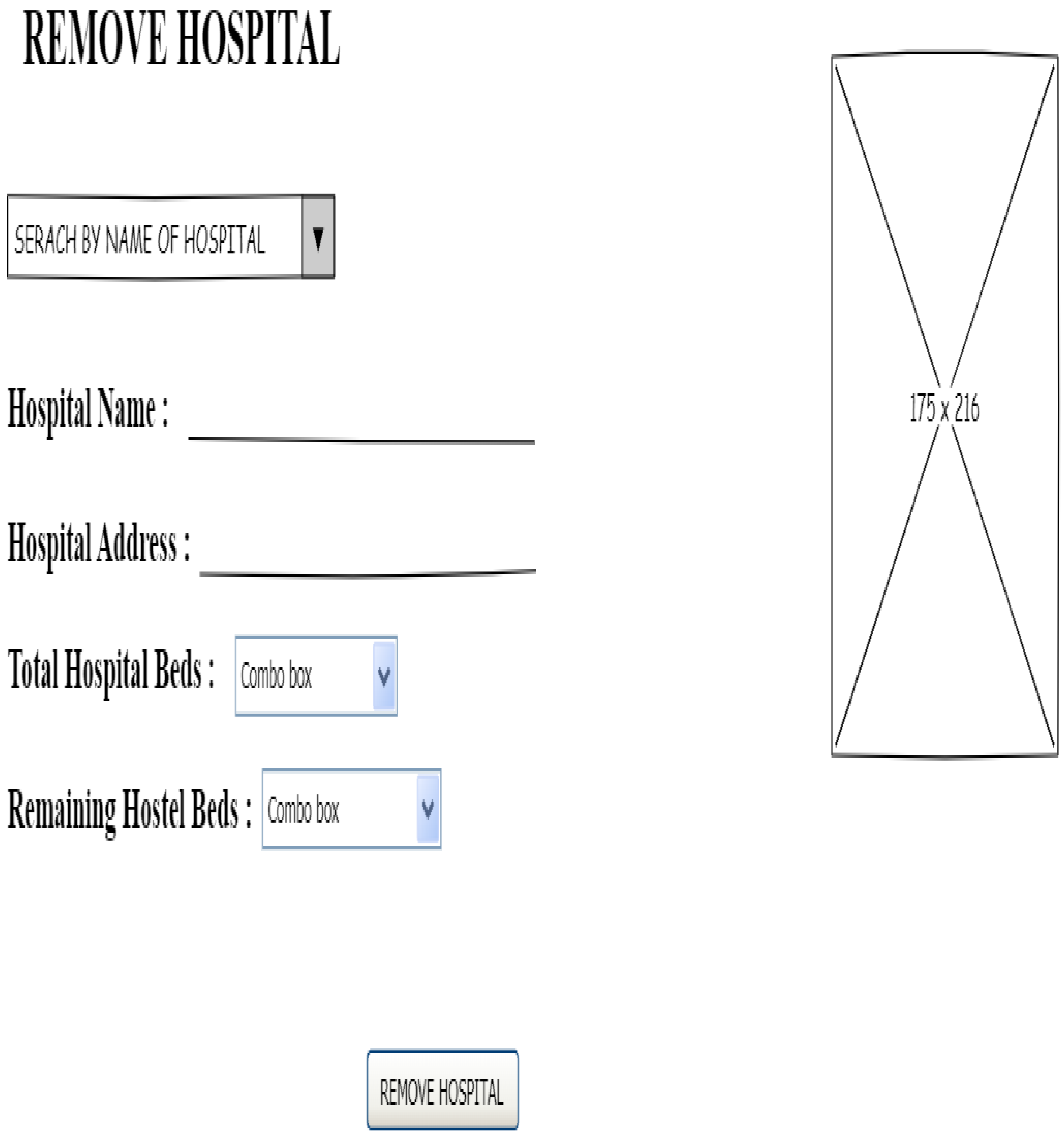
Update Hospital Window:

This window contains the search panel at the top of the page. The admin will search the hospital which he wants to update. As the hospital is selected, then rest of the information is automatically filled. Then, the admin will change the information which he wants to change. As the **Update Hospital Data** button is pressed, then as a result the data of the hospital is updated.



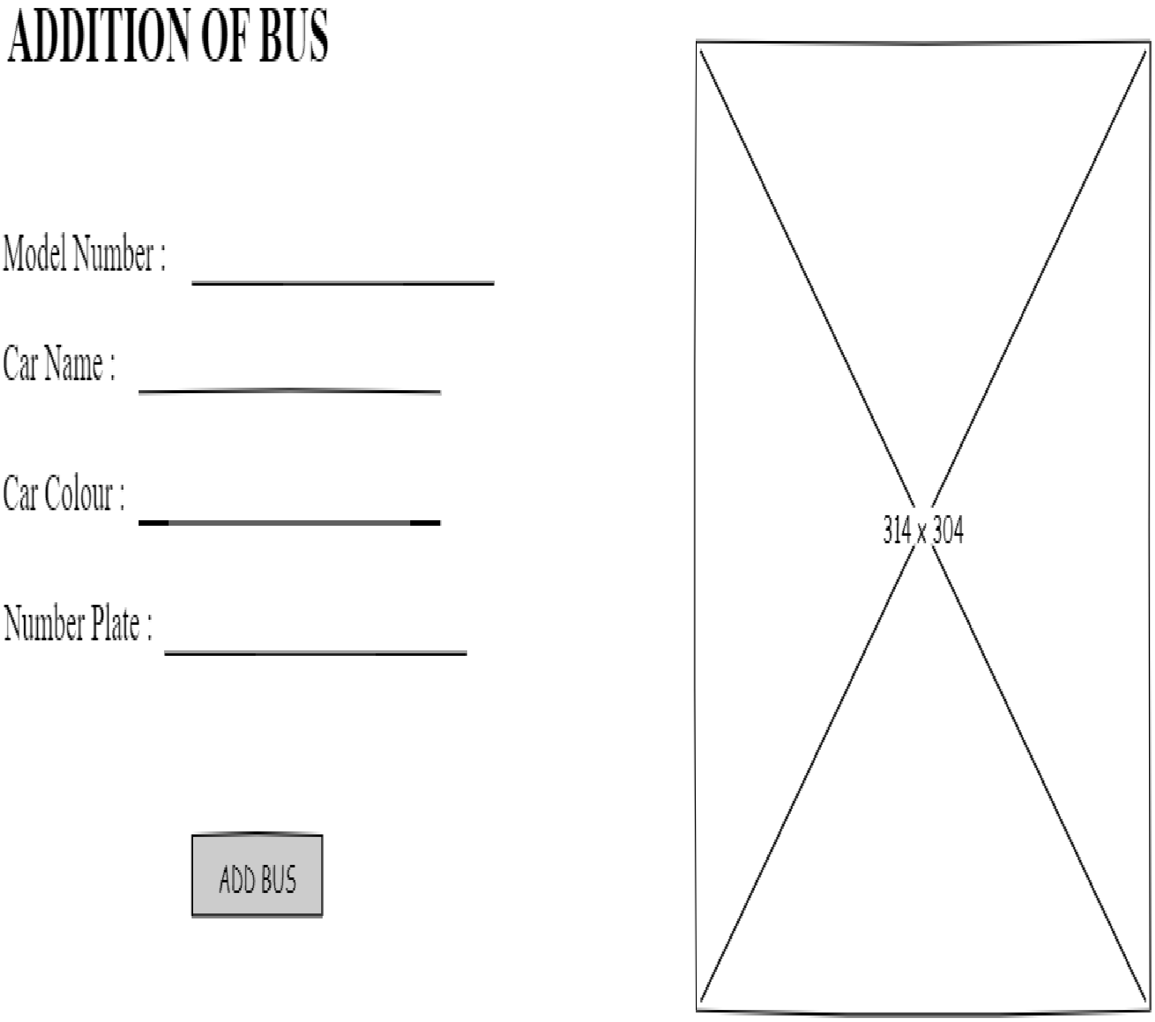
Remove Hospital Window:

In this window, there is a search panel. The admin will search the hospital from this panel which he wants to delete. As the hospital is selected, then all the next information will be will be autofilled. As the **Remove hospital** button is pressed, then the hospital is poped from the stack.



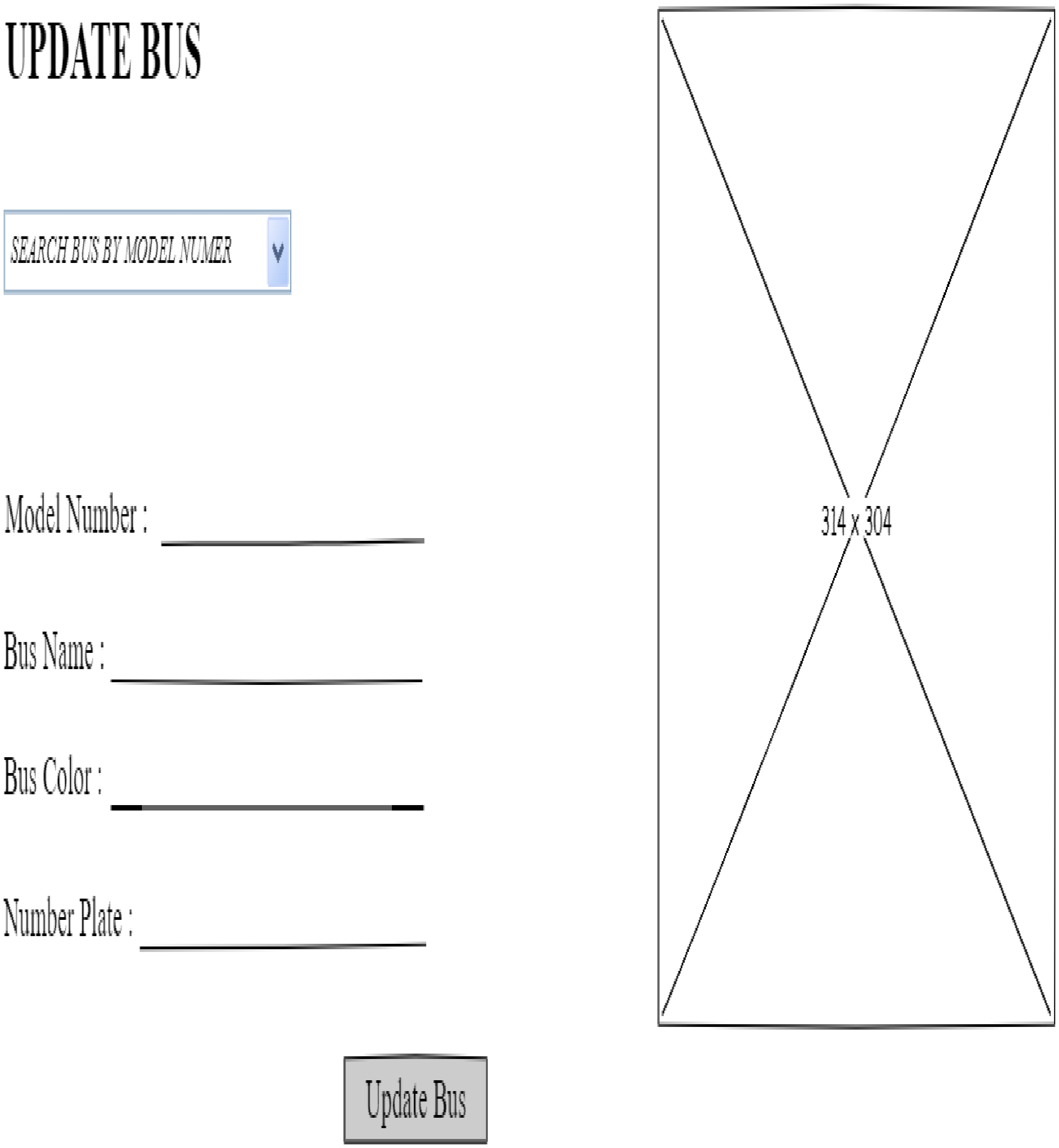
Add Bus Window:

Whenever a new bus is available for this service, the admin of the buses fills all the information related to the bus which is mentioned in the above window. after filling this form, as the **Add Bus** button is pressed, then the object of this bus will be added in the queue.



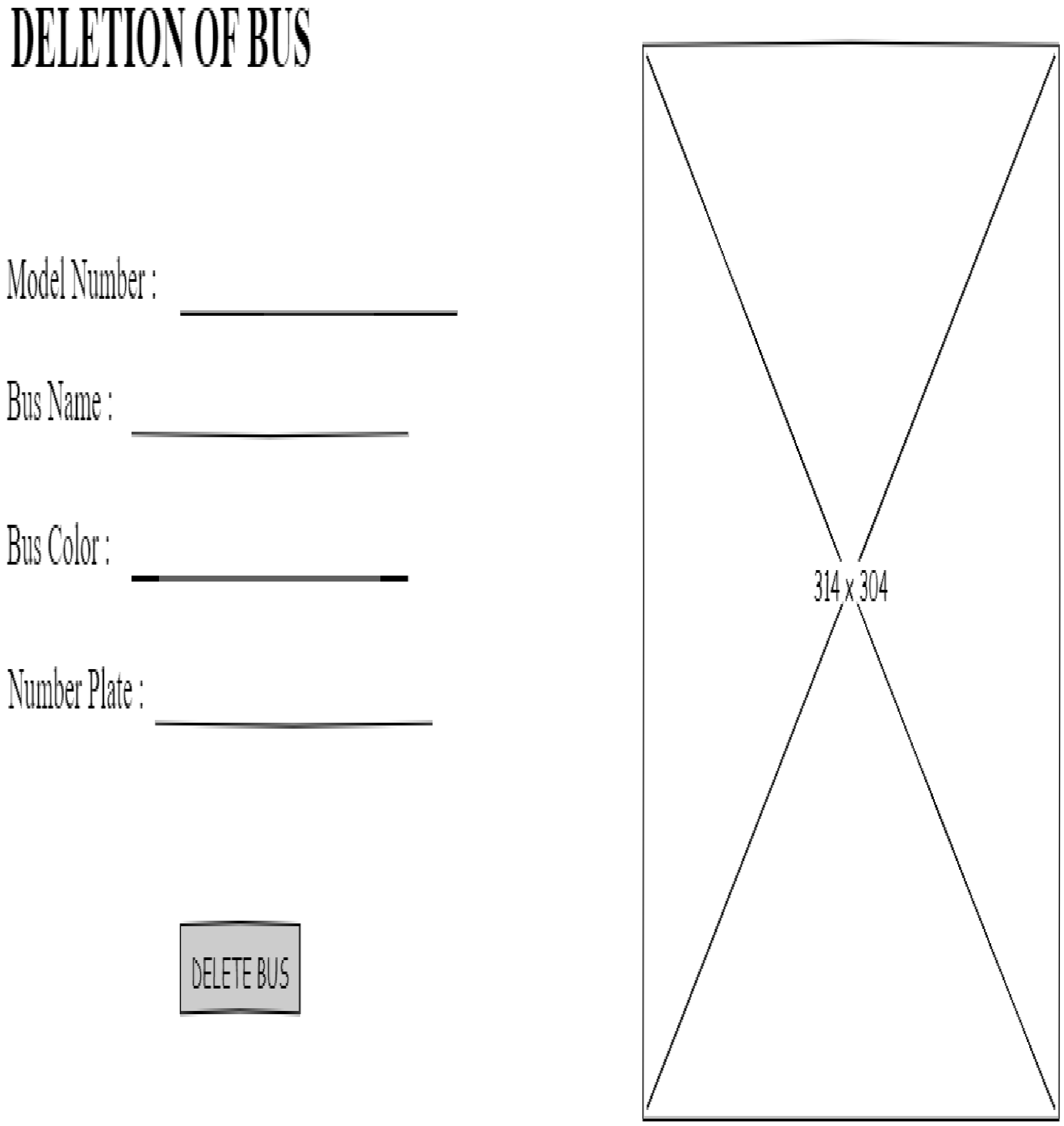
Update Bus Window:

This window also contains the search panel at the top of the page. The admin will search the bus which he wants to update. As the bus is selected, then rest of the information is automatically filled. Then, the admin will change the information which he wants to change. As the **Update Bus** button is pressed, then as a result, the object of the bus is dequeued.



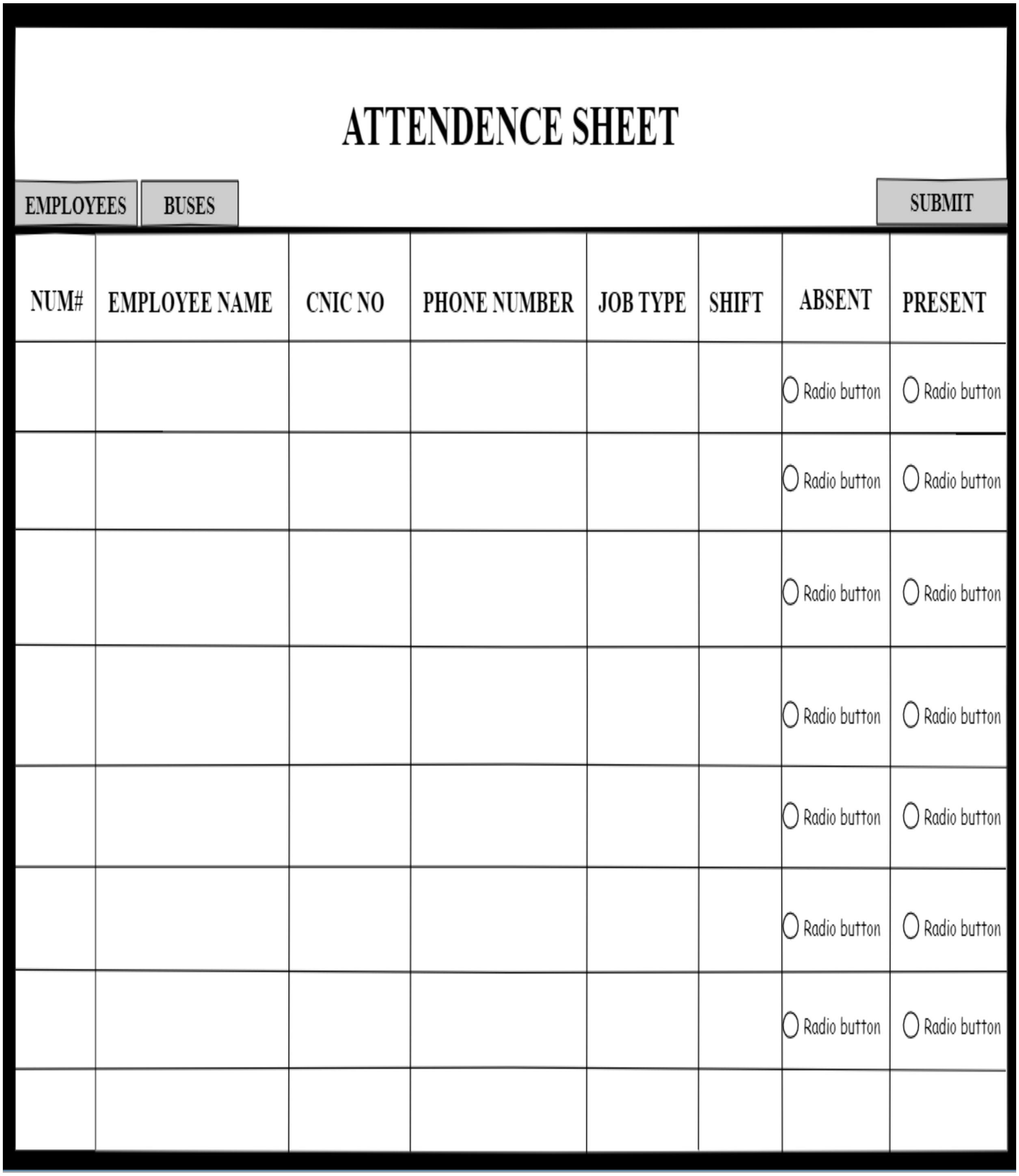
Delete Bus Window:

This window also contains a search panel. The admin of all the buses searches the bus which he wants to delete. When he will select the bus, rest of all the information regarding the bus will be autofilled. The buses will be deleted from the queue as the **Delete bus** is pressed.



Attendence Sheet:

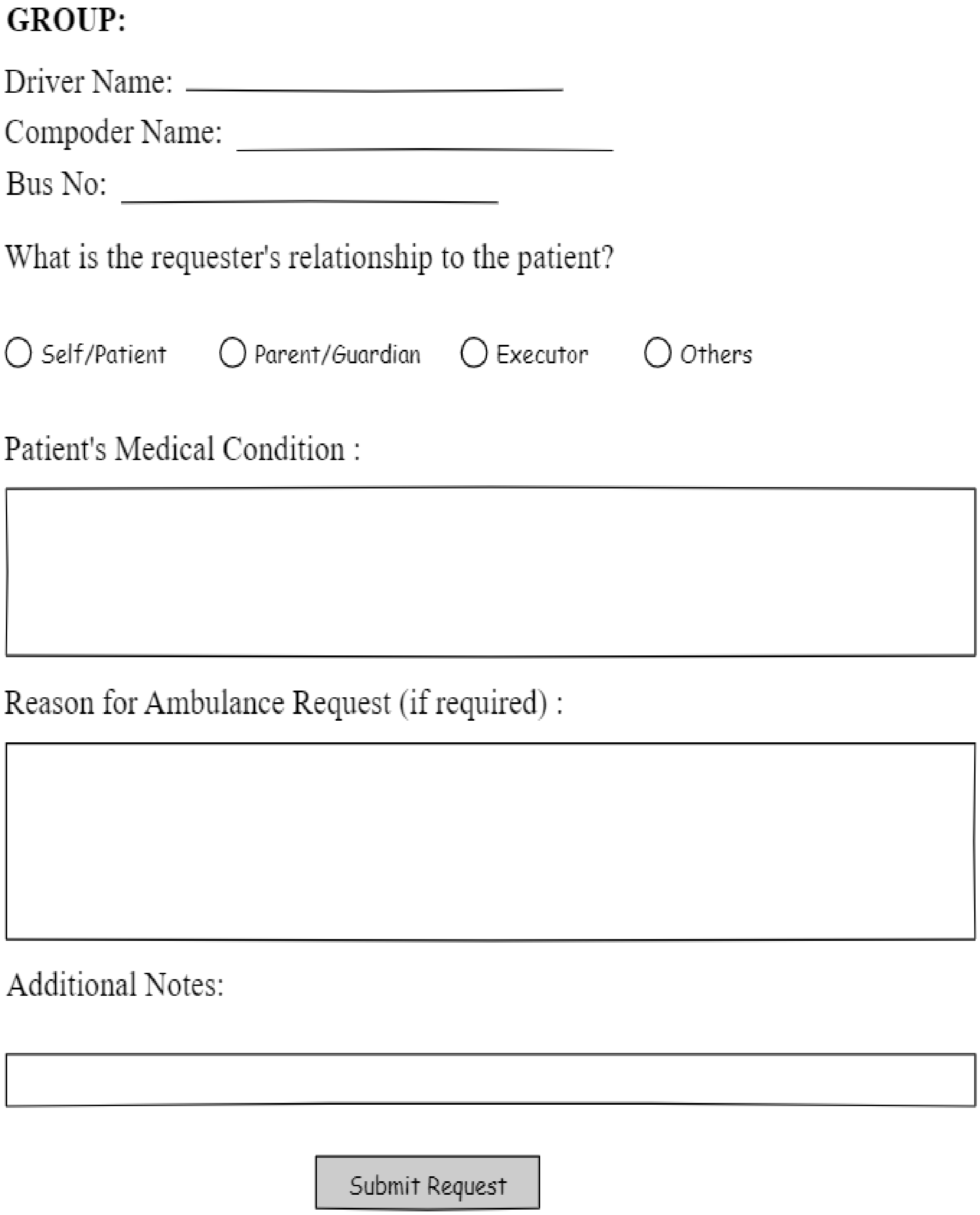
This window contains all the information related to employees and buses. This window also contains two radio buttons at the end of each row. These radio buttons are used for marking absent or present of the employees or buses. In this way, attendance of the buses and employees can be taken.



NUM#	EMPLOYEE NAME	CNIC NO	PHONE NUMBER	JOB TYPE	SHIFT	ABSENT	PRESENT
						○ Radio button	○ Radio button
						○ Radio button	○ Radio button
						○ Radio button	○ Radio button
						○ Radio button	○ Radio button
						○ Radio button	○ Radio button
						○ Radio button	○ Radio button
						○ Radio button	○ Radio button

Rescue Team Information Window:

This window contains all the information related to the rescue team. This form also contains the information of the patient to whom they have rescued. It also contains the information of the current condition of the patient. After pressing the **Submit Request** button, the information of this form is saves in their record.



GROUP:

Driver Name: ____________________

Compoder Name: ____________________

Bus No: ____________________

What is the requester's relationship to the patient?

○ Self/Patient ○ Parent/Guardian ○ Executor ○ Others

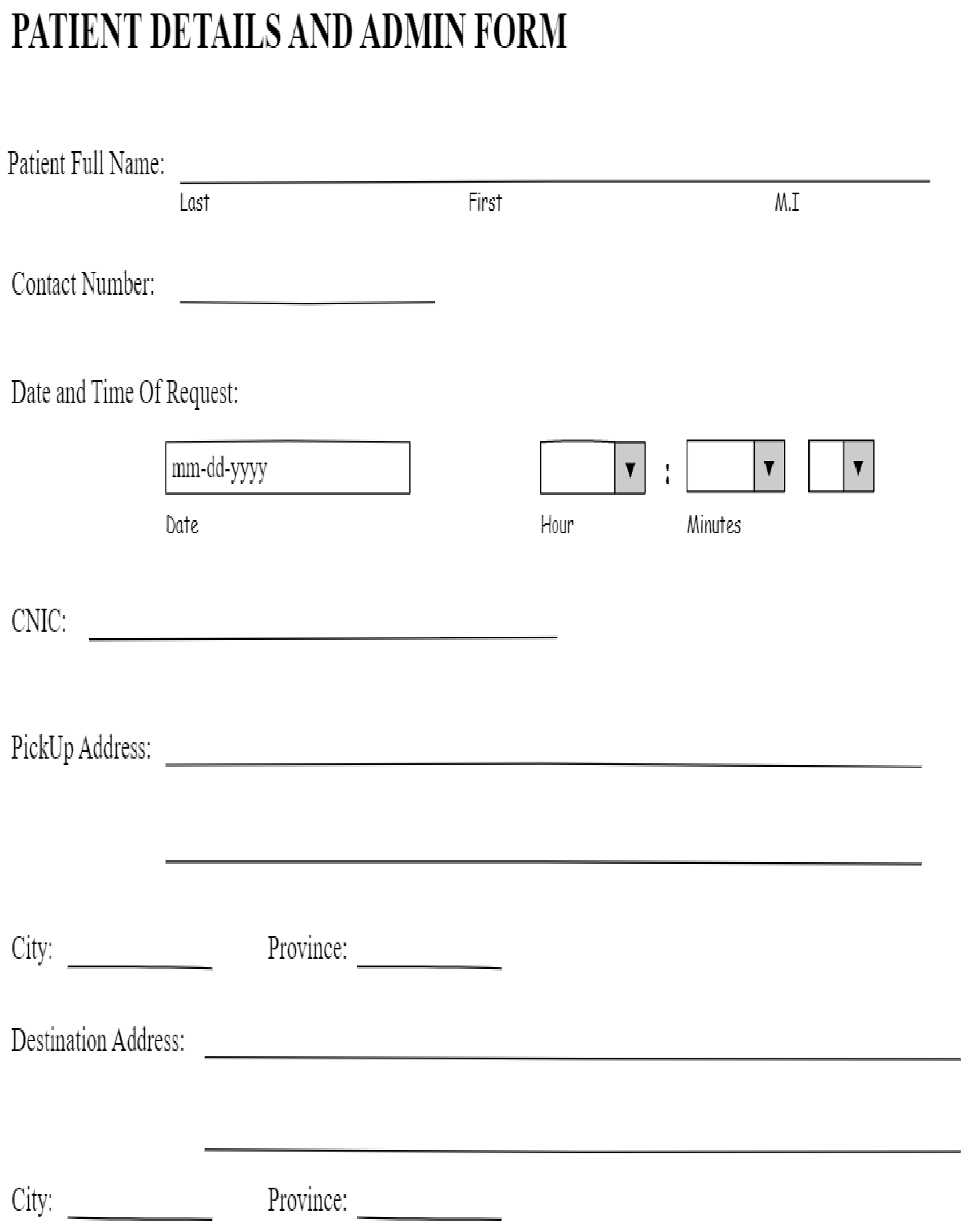
Patient's Medical Condition :

Reason for Ambulance Request (if required) :

Additional Notes:

Patient Details Windows:

This form is filled by the admin .As the information related to the patient is received, the admin will fill this form. This form contains all the information related to the patient. It also contains the information from where the patient is received and at which place the patient will be delivered.



PATIENT DETAILS AND ADMIN FORM

Patient Full Name: ______________________________
Last First M.I

Contact Number: ______________

Date and Time Of Request:

mm-dd-yyyy ▼ : ▼ ▼
Date Hour Minutes

CNIC: ______________________

PickUp Address: ______________________________

____

City: ________ Province: ________

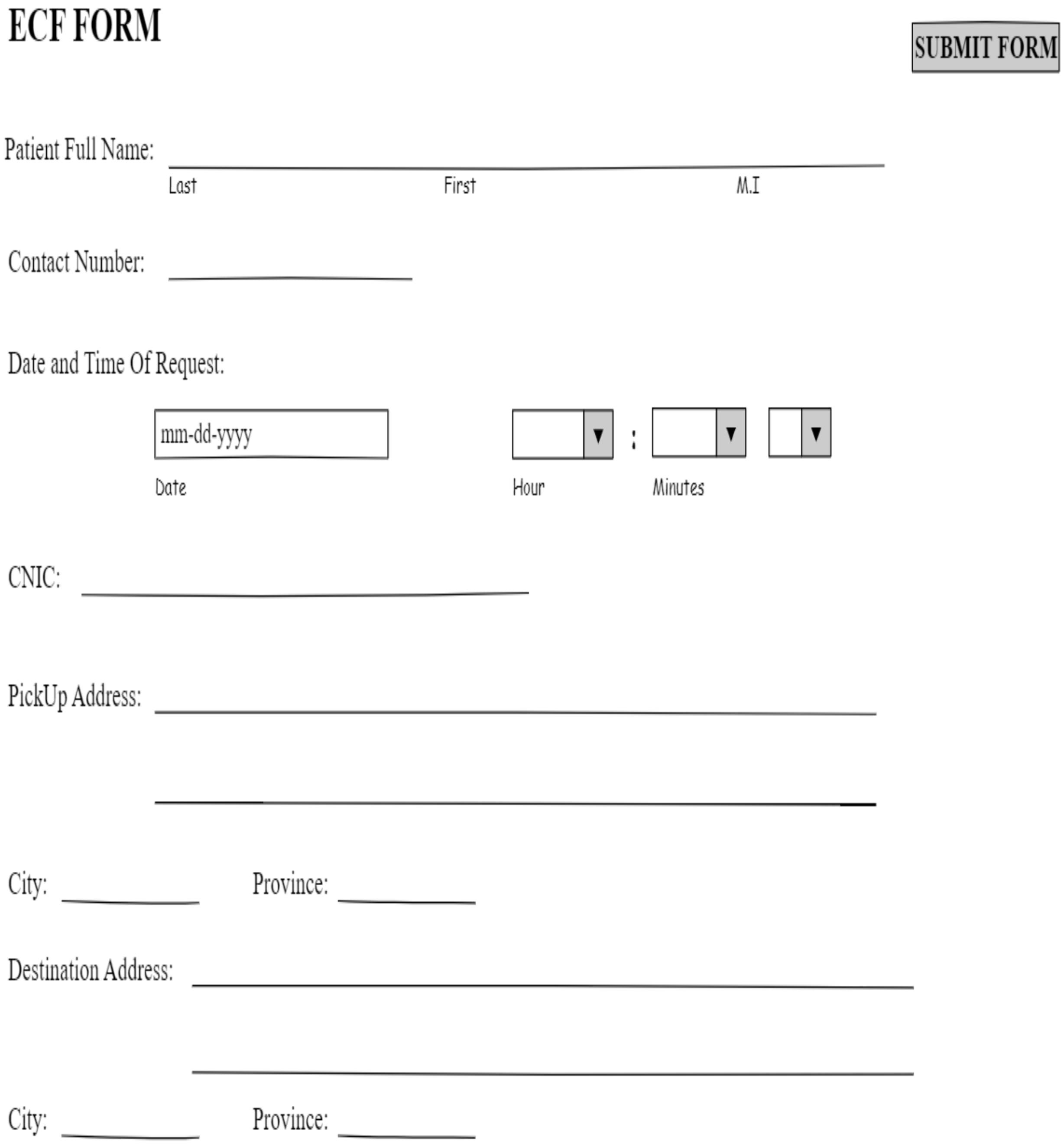
Destination Address: ______________________________

____

City: ________ Province: ________

ECF Window:

This form is filled by the CTWO .As the information related to the patient is received, the CTWO will fill this form. This form contains all the information related to the patient. It also contains the information from where the patient is received and at which place the patient will be delivered. As the **Submit Form** button is pressed, the information of this form is stored in the record.



ECF FORM

Patient Full Name: ________________
Last First M.I

Contact Number: ________

Date and Time Of Request:

mm-dd-yyyy ▼ : ▼ ▼
Date Hour Minutes

CNIC: ________

PickUp Address: ________________

____

City: ______ Province: ______

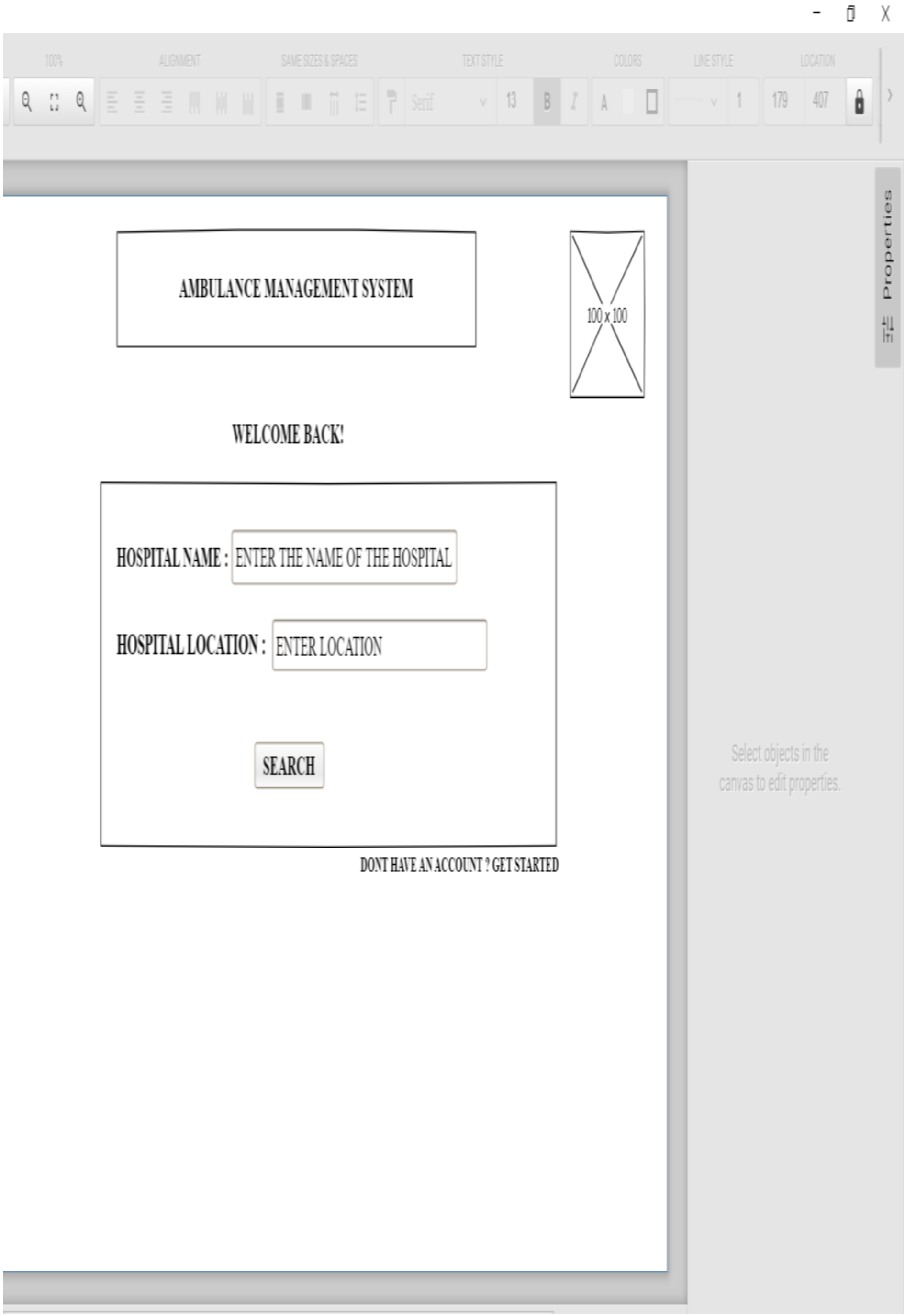
Destination Address: ________________

____

City: ______ Province: ______

Hospital Main Window:

This form is filled by the respective hospitals whenever the ambulance staff has to check for the information of these hospitals.This form will only take the name and location of the hospital .



AMBULANCE MANAGEMENT SYSTEM

WELCOME BACK!

HOSPITAL NAME : ENTER THE NAME OF THE HOSPITAL

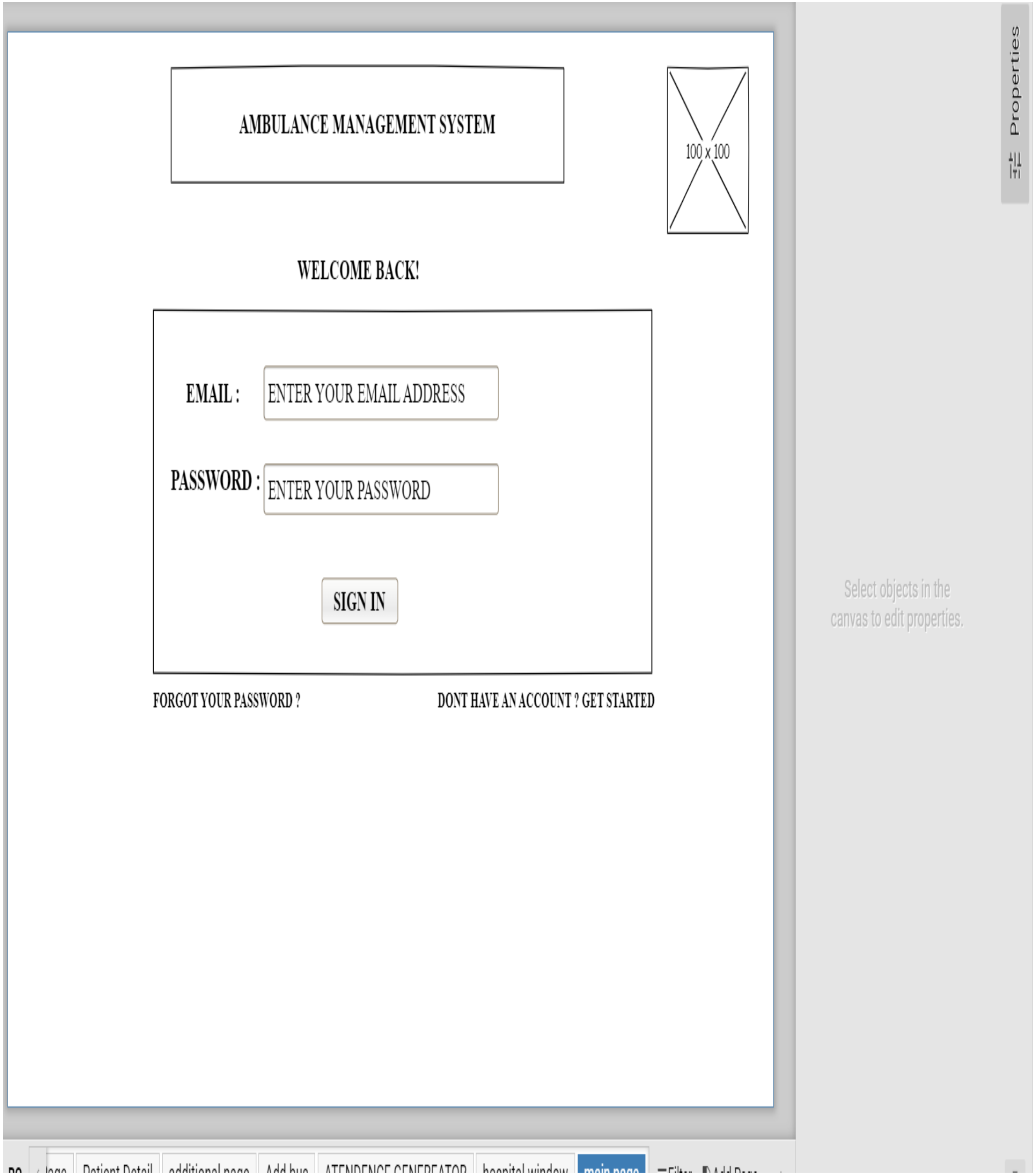
HOSPITAL LOCATION : ENTER LOCATION

SEARCH

DONT HAVE AN ACCOUNT ? GET STARTED

Employee Main Window:

This form is filled by the respective employees of the hospitals . This form will include the name and the password assigned to them by the administration of the hospital. Moreover in case he/she forgets the password , they will have the option to renew the password.



AMBULANCE MANAGEMENT SYSTEM

WELCOME BACK!

EMAIL : ENTER YOUR EMAIL ADDRESS

PASSWORD : ENTER YOUR PASSWORD

SIGN IN

FORGOT YOUR PASSWORD ?

DONT HAVE AN ACCOUNT ? GET STARTED

Plan of stimulating data:

Till now we have decided that we will generate random data in order to meet the requirements of the final project. We have searched a lot but we haven’t found even a single source from where we can get this huge amount of data. Our project requires different type of data. In other words, we can say that in our project there many portions where we need different type of data. For example:

- Data of hospitals
- Data of ambulances
- Data of Employees
- Data of patients

All of the above mentioned points are the different sections of our project. For each of these, we need a large amount of data. So, we have decided that for each section in our project, we will define a separate random function. In order to generate the data according to our own need, we will provide different ranges to these different random functions. So, it will generate the data according to our own need. So, in this way we will populate the data for our project.