

**Deliverable J**  
***User Manual***

**Accessible Wi-Fi**  
**Group 15**

Jason Smith 300043461  
Kawakon Quansah 300018273  
Ruchira Perera 8714992  
Samuel Oyediran 300016349  
Xiongxin Xiao 7939074

## **Abstract**

This report will outline the problem at hand and how it was solved. It will discuss the Autoconnect application which was made specifically for the St. Vincent Hospital (SVH) to help them with their Wi-Fi network connection issues. The report will demonstrate some of the core components of the software application and how it is supposed to be used. The report will also reflect both the prototype(s) made during the development of this application and the final product. Furthermore, maintaining the app is important as an admin for the application. Therefore this user manual will help discuss ways of maintaining the application and ensure it functions properly for its users. In addition, this manual will go over any troubleshooting needed to fix any issues people may experience.

## **Table of Contents**

<b>Abstract</b>	<b>2</b>
<b>Table of Contents</b>	<b>2</b>
<b>Table of Figures</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
<b>Main Body</b>	<b>4</b>
Important features	4
Instruction for Server	6
Troubleshooting	6
<b>Recommendations for Future Work</b>	<b>7</b>
<b>Conclusion</b>	<b>8</b>
<b>Bibliography</b>	<b>8</b>

## **Table of Figures**

<b>Figure Number</b>	<b>Page</b>
Figure 1	4
Figure 2	5

## Introduction

The problem of internet connectivity is relevant for many reasons at this hospital. During password changes or other network disconnections, staff must be requested to come help patients reconnect to the internet. This process is very time-consuming but is also avoidable with the proper technology. The solution we had to design must keep up-to-date passwords on the user device and bypass the Terms & Conditions page when reconnecting. We decided to create a software based solution because of its accessibility on android and ios systems. Additionally, our method of updating and distributing information using a server minimizes the staff assistance when passwords are changed. The accessibility, convenience and simplicity of our product makes it a dynamic solution.

The main goal of our design concept was to resolve the frequent disconnections and monthly password change in St Vincent hospital. With our application *AutoConnect*, a software based solution was provided that automatically accept the terms and conditions page, monitors disconnection and automatically reconnects users to Wi-Fi when disconnected as requested by our client, Bocar. The following questions will be answered: so what? With *AutoConnect*, it is easier for patients to connect to the Wi-Fi and be more independent. With users being independent, less time is spent on attending to different users who have disconnection issue in the hospital[1]. The application setup is simple, the Admin, Boucar, is responsible for connecting all the devices in the hospital to the Wi-Fi. Our application meets all our client needs and a working project was achieved within proper time and budget[2].

**Who cares?** In order to use our application, *AutoConnect*, users are simply required to download the application in their play store/Apple store. *AutoConnect* works on both Android and iOS. Our application is important for patients of the hospital especially because they have difficulty with multiple reconnection to the Wi-Fi. So to simply resolve that, the Admin of the application, Boucar, will be responsible for typing in the Wi-Fi name and password on the router page. The router page allows the admin to input the password for the next two months[1]. The Wi-Fi name, Wi-Fi password, and the next two months passwords are sent to the server. After the router page has been set up, to connect to the Wi-Fi, simply click on the “switch\_power\_on icon”. After the icon has been clicked on, the server sends the Wi-Fi password to the device. The terms and conditions page is automatically accepted and instant access to the Wi-Fi is granted [1].

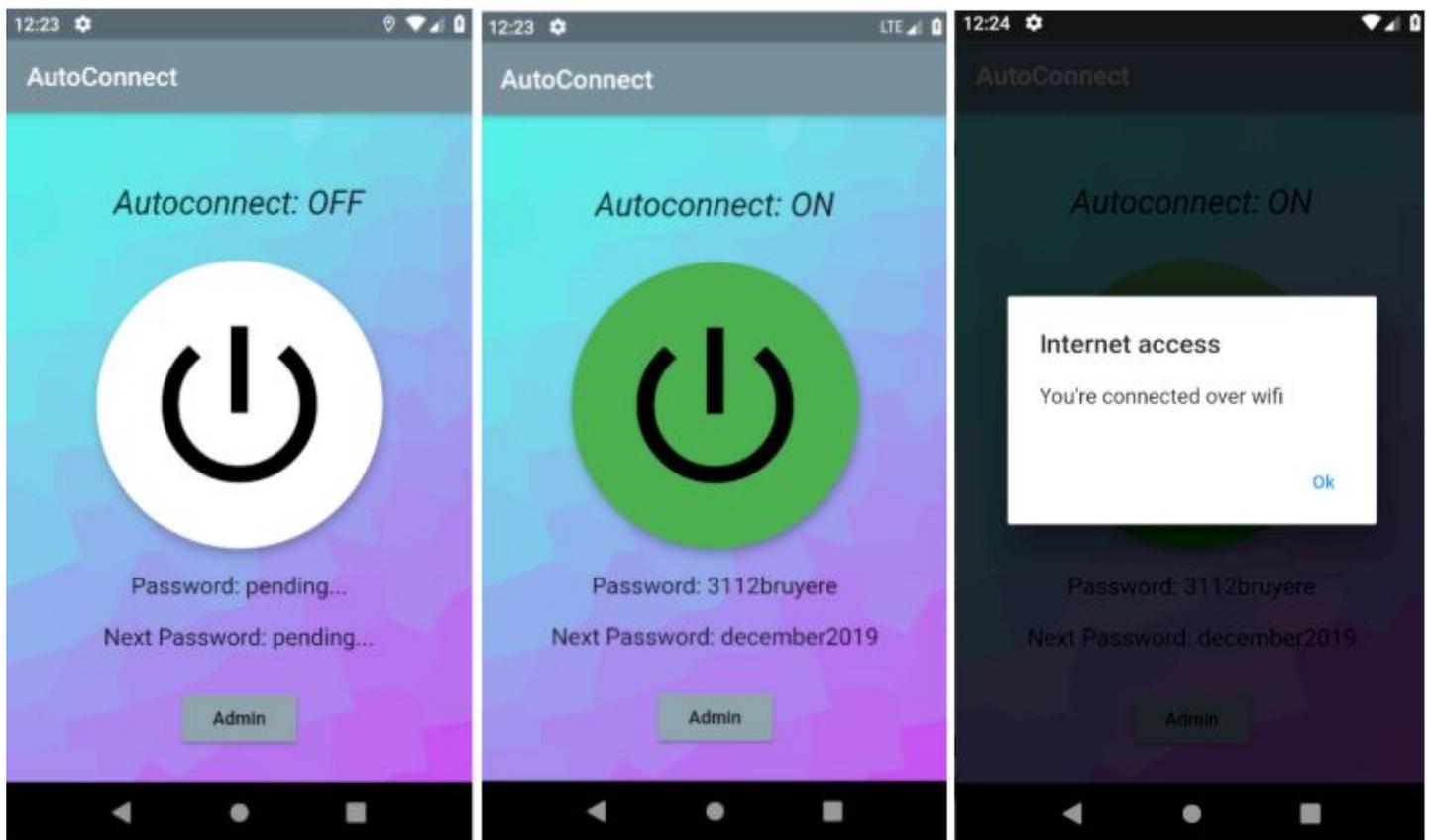
**Why us?** The Application, *AutoConnect*, is designed such that patients and guests at St. Vincent Hospital have access to the Wi-Fi and accept the hospital terms and condition for internet upon one button click. The admin of the application is to provide the Wi-Fi name and password to connect devices to the router page. The passwords for the next two months of the associated Wi-Fi name can also be entered by the admin on the router page [1]. The stored passwords are saved in order, as the month and year of the passwords are prioritized on the listview saves the password. Finally, when users open the application, clicking on the “power\_switch\_on icon” the network name and password are sent to the user's device. The next months passwords automatically update monthly after the old one expires [1].

Our final design prototype is a working version of our design solution. Our client is satisfied with our solution, and gave positive feedback on our design concept. The design budget and deadline for completing the application was met just as initially planned.

## Main Body

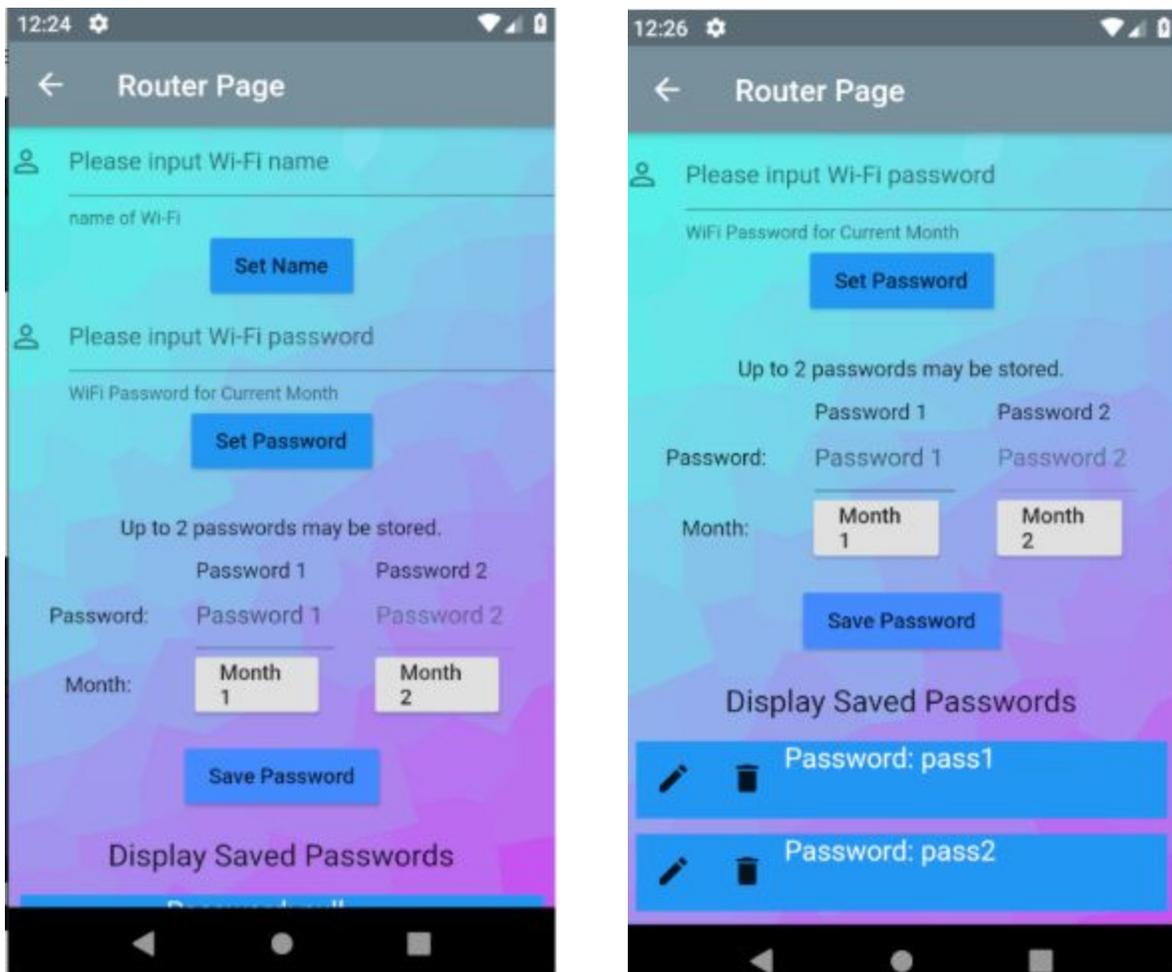
### Important features

In the Autoconnect application there are many important and key features we have provided our users in order to make the app usable by all. First and foremost is the large button that is seen on the landing page. This is known as the Autoconnect button. Simply tap it once or twice to successfully pull the data from the servers onto the device which is trying to connect to the Wi-Fi. This button is crucial and important as it is the purpose of this application. Turning the button on by tapping it will run back end scripts to connect the device to the Wi-Fi. It is important to note that once this is done the user should wait for the “You are now connected to the internet” dialogue box to pop up. Once this is done the user may click “Ok” and can then tap the home button to leave the app and use other apps on the phone.



**Figure 1** Landing page of the *AutoConnect* app showing the OFF and ON states of the application and showing the device to connect to Wi-Fi.

The second important feature our app has to offer is an administrative control over the Wi-Fi. An administrator can simply click the Admin button found on the landing page below the Autoconnect button. When pressed a dialogue box will appear prompting the user to enter a password. This password is fixed and cannot be changed to help simplify utility of the application. Once the proper password is entered the user may log in to what is called the router page. The router page is a place where a staff administrators may enter important information such as Wi-Fi name, current password, and the next two months passwords. Entering the proper data for each text field and then pressing the button to “Set” or “Save” data will send a post request to the server and the data is updated in real time.



**Figure 2** Shows the Router Page of the Autoconnect application showing the fields of where to input Wi-Fi Name and passwords. The buttons will send data to the servers to configure the Wi-Fi connection.

## Instruction for Server

The server was developed by node.js, which can respond to get and post requests. To be specific, the server gets the password through Http post request from the administrator-side of the app and store the password as a variable, then the client-side app can get the password by the get() request, which is a typical application of the RESTful API. The content type of data been transported is the JSON object. I have made methods to transform JSON object to native object both in server and app. To be more specific, the js object is called “resData” and “resData2” in the server. The dart object called “PasswordClass” in the app. The transformation is very important, the flutter compiler would shoot error if any type-error has detected which occurred frequently in the development process. However, since a full set of request codes has been made, just follow the existing code to make some new instances of the request if needed. The request-id is included in the URL, which is followed after “?”, for example, “<http://ec2-35-182-74-15.ca-central-1.compute.amazonaws.com:9000/?id=1>” means this request is made aiming for the first password variable “password” in server. There are three ids that can be accepted by the server, “id=1” means the current password, “id=2” means password for next month, “id=3” is the name of the Wi-Fi, “no such id” would be the response for the incorrect id. For troubleshooting, most errors would be caught by the flutter compiler, which includes socket-error, type-error, handshake-error and any other internet problem. Most of them can be solved by asking google, which would guide you to a similar post on dio’s official debug centre, “<https://github.com/flutterchina/dio/issues>”. When you not sure about the cause of the problem is coming from server or app, we recommend using an HTTP request simulating tool named “Postman” to test server solely. If you got the right response in Postman, then the problem must come from the flutter app. For keeping the server running, the node.js file needs to be hosted on a cloud virtual machine. We have chosen the ubuntu EC2 instance from AWS with a 2.5 GHz vCPU and 1 GB RAM, which is free to use and sufficient for performance. Moreover, you also need a process manager running in the virtual machine to keep the process of server alive, or the process would be killed after you log out from the virtual machine. After research and comparison, the tool named “PM2” is chosen for this task, and it works perfectly well up to now, which keeps the server running in the virtual machine even after you close your computer and can restart the process of the server if an error happened automatically.

## Troubleshooting

### **Device is not connecting to Wi-Fi network**

Disconnect your device from any Wi-Fi network currently connected, if any. To do so, navigate to the Network Settings on your device and tap on the ‘Forget’ button of the network that the device is connected. Afterwards, proceed to *AutoConnect* start page and press the ‘ON’ button until it is displayed in a green colour. A pop-up dialog will appear and display the following: “Internet access - You’re connected over Wi-Fi.”

### **AutoConnect application will not start**

It is possible that your device is running out of memory to allocate to the *AutoConnect* application, as there are too many applications open. Please close any applications that you are not using. If the problem persists (*AutoConnect* application fails to start), please restart your device.

### **Password is not accepted to access Router page**

Ensure that the correct *Administrator* password is entered to access the *Router* Page. Please consult the default password given and ensure that lowercase and uppercase letters, spaces, and symbols are entered according to the default password.

### **Cannot edit passwords on Router page**

Refresh the *Router* page by pressing the “Back” button and returning to the *Router* page, re-entering the access password.

### **Term ‘null’ is seen after deleting a password. What does this mean?**

Upon deleting a password, the term ‘null’ is used to show *Administrators* that a password has recently been deleted. This value may be changed by editing a password.

## **Recommendations for Future Work**

In terms of future plans for the application, the *AutoConnect* team aims to provide this application to other businesses and organizations that use a terms and conditions page as an intermediary to Internet connection. Possible avenues for this use include libraries, hospitals, shopping centres, and small restaurants. This implementation of the *AutoConnect* application will be useful because it effectively solves the issue of reaccepting numerous terms and conditions in order to access the Internet. The problem is especially troublesome when a user has already navigate past the terms and conditions page and has been using the network for an extended period of time. This inhibits a user’s ability to access the Internet. Therefore, through the use of the *AutoConnect* applications, a user would only have to accept the terms and conditions upon accessing the Internet for the first time and will be able to use the Internet without re-encountering the page. In order to implement the *AutoConnect* application in different business, it will be necessary to provide a feature that allows *Administrators* to provide the URL for the terms and conditions page that must be viewed by users. Thus, upon pressing the “ON” button on the *AutoConnect* start page, the “I agree” option will be selected.

Furthermore, we would also like to extend our application to Linux and Windows devices. This added compatibility was included as one of our target specifications, but unfortunately, we were not able to implement it in our application due to time constraints. However, this extension will allow users to access the Internet from various devices, not limited to phones and tablets. This extension will be beneficial in businesses where different kinds of devices interact with a terms and conditions page before connecting to the Internet.

Finally, a password change feature will be required to allow *Administrators* to modify the default password to access the *Router* page.

## **Conclusion**

Over the course of the project we have learned several lessons which will help us in the future. The detailed planning and scheduling we created during the beginning of the semester helped us greatly to stay focused and on track. However, we realized that during the course of development that some tasks may take much longer than expected or some unexpected issues may appear and delay the schedule. We realized the value of communication during a group project. Meeting up for group discussions and staying active in group chats helped the team move as a unit and solve problems together instead of working alone. During times of disagreement we would have group meetings to discuss the benefits and disadvantages of each argument to come to a logical conclusion. Additionally, we noticed that communication with the client was crucial towards building software. The feedback and user requirements obtained from the client after each iteration was instrumental in designing the app to satisfy its future users. Through these strategies, we managed to create a working application which has the necessary functions to fulfill the user needs.

## **Bibliography**

[1] GNG2101-Deliverable I. *“Design Day Pitch and Final Project Evaluation”*. Accessed 3rd December, 2019.