

Tech Wizard

by

Charles Holt, Matthew Miracle, and Erica Partin

Submitted to
the Faculty of the School of Information Technology
in Partial Fulfillment of the Requirements for
the Degree of Bachelor of Science
in Information Technology

© Copyright 2019 Charles Holt, Matthew Miracle, and Erica Partin

The author grants to the School of Information Technology permission
to reproduce and distribute copies of this document in whole or in part.

Erica Partin	04-26-2020
Erica Partin – Project Manager	Date
Matthew Miracle	04-26-2020
Matthew Miracle – Software Developer	Date
Charles Holt	04-26-2020
Charles Holt – Software and Database Manager	Date
Yahya Gilany	04-26-2020
Yahya Gilany – Technical Advisor	Date

University of Cincinnati
College of
Education, Criminal Justice, and Human Services
November 2019

TABLE OF CONTENTS

ABSTRACT	1
1. PROJECT STATEMENT	2
1.1 Introduction	2
1.2 Project Description	2
1.3 Problem	2
1.4 Solution	3
1.5 User Profile	4
1.5.1 Project Title	5
1.6 Use Case Diagram	6
2. PROJECT MANAGEMENT	7
2.1 Budget	7
2.2 Project Deliverables/Objectives	9
2.3 Project Schedule	10
2.4 Work Breakdown Structure	11
3. TECHNICAL ELEMENTS	12
3.1 Network	12
3.2 Database	13
3.3 Application	13
3.4 Security	13
4. APPLICATION	14
4.1 Application Architecture	14
4.1.1 Application Software	14
4.2 Screenshots and Examples	15
4.2.1 Homepage	15
4.2.2 User Homepage	16
4.2.3 Signup Page	17
4.2.4 User App Homepage	17
4.2.5 GPS Page	18
5. TEST PLAN	19
5.1 Overview	19
5.2 Objective	19

5.3 Test Cases	19
5.3.1 Testing Procedures	20
5.4 Testing Scenarios	21
5.4.1 End User View	21
5.4.2 Wizard's View	23
5.5 Pass/Fail Conditions	26
5.5.1 Start-Up and Launch Testing	27
5.5.2 Stability Testing	27
5.5.3 User Interface Testing	27
5.5.4 Testing Timeline	28
5.6 Lessons Learned	28
6. CONCLUSION	30
6.1 Fall Semester 2019	30
6.2 Spring Semester 2020	30
APPENDIX A: ADDITIONAL INFORMATION	32
Poster	32
Abbreviations	32
APPENDIX B: REFERENCES	33

List of Illustrations

Figures

Figure 1. User Profile	4
Figure 2. Use Case Diagram	6
Figure 3. Project Deliverables	9
Figure 4. Project Schedule Gantt Chart	10
Figure 5. Gantt Chart	11
Figure 6. Work Breakdown Structure	12
Figure 7. Architecture Diagram	14
Figure 8. Home Page	15
Figure 9. Log-In Page	16
Figure 10. Sign Up Page	17
Figure 11. User App Homepage	17
Figure 12. GPS Page	18
Figure 13. Home Screen	21
Figure 14. Sign-Up Page	22
Figure 15. Sign-Up Page Example	22
Figure 16. Login Screen	23
Figure 17. Tech Wizard Homepage	24
Figure 18. View Jobs Page	25
Figure 19. Job Details Page	25
Figure 20. Client Info	26
Figure 21. Phone Dialer Window	26
Figure 22. Chat Window	26

Tables

Table 1. Cost Breakdown of Elements	8
Table 2. Cost Breakdown of Human Resources	8
Table 3. Software	9
Table 4. Start-Up and Launch Testing Conditions	27
Table 5. Stability Testing	27
Table 6. User Interface and Features Testing	27
Table 7. Testing Timeline	28

Graphs

Graph 1. Original Budget Chart	7
Graph 2. Budget Chart	8

ABSTRACT

Technology is exponentially evolving, sometimes faster than we expect it to. Because of this, it's difficult for those who are just trying to keep up with the newest technologies. Of course, there are many outlets that people can look to for help. However, trying to troubleshoot over the phone can be frustrating for others on both ends and some problems just simply go unsolved. With Tech Wizard, we are providing an extra hands-on approach to users who are facing technical difficulties at home. We created an app that allows users to sign up and search for an available Tech Wizard in the area that can offer support via chat, phone support, or in person. We expect the result of this is to increase the availability of tech support to people of all ages and all levels of technical knowledge.

1. PROJECT STATEMENT

1.1 Introduction

Tech support can be a double-edged sword. There's always a possibility that your issue can't be simply solved over the phone or chat, or that you won't have to spend hours on the phone talking to different automated prompts that may or may not solve your issue or connect you to someone who does. More and more companies are finding it difficult to keep up with the demands of their customers when it comes to understanding their products and issues that can occur with them. Tech Wizard can help ease the pain and frustration that comes with this expanding technical world. With our easy to use app, you can sign up and find a Tech Wizard in your area that knows just how to solve your issue.

1.2 Project Description

Tech Wizard allows users to sign up for an account and enter their technical issues they are having. The app will then ask for how they would like to receive support, via chat, phone, or in person. It will then find a Tech Wizard whose expertise match the technical issue and is available for immediate support. If the user chooses to find someone in their area to solve an issue, it will search their nearby area for an available Wizard to assist. Tech Wizards are able to create their profiles based on their personal expertise and experience.

1.3 Problem

Technology is ever increasing, but sometimes it's difficult to keep up with learning and understanding new technologies that we have to live with. Help desks, online chat help,

and independent troubleshooting are great ways to help with that understanding. But, sometimes waiting an hour on the phone to reach a live person can be frustrating. Sometimes the fix for an issue can be simple, something that phone support and online chat help took too long to solve or failed to provide a solution. ResearchNow conducted a study and found that the average person will spend around 43 days or more of their life on hold and only 1 in 3 callers will even try to call back¹. This not only has an impact on the customer, but also the company itself and its performance.

1.4 Solution

What we are introducing is an app that allows tech savvy folks to join and offer their expertise to others in their area. Users who find themselves in need of a tech wizard can join the app and search for an available tech in the area who can help via chat, phone, or in person. The app will allow you to track the tech wizard while they are on their way to you. This can help users who don't want to wait on the phone for hours while a robot directs their answers and instead talk to a real person who can help solve their issues quickly and efficiently.

Many current tech support companies such as Geek Squad offer similar services, but they can require calling to schedule an appointment or having you come up to the store to deliver your equipment to them. With Tech Wizard you can make life much simpler by entering your information in and then having one of our technicians that specialize in the area you are having trouble with contact you directly by phone, chat or in person. This

¹ "Arise Customer Service Frustration Series: Phone Hold Times."
<https://www.arise.com/resources/blog/arise-customer-service-frustration-series-phone-hold-times>.
Accessed 14 Oct. 2019.

saves you the time and frustration of waiting on hold or having to schedule an appointment later for a problem that you would like fixed in a timely manner.

1.5 User Profile

Figure 1: This illustrates the user interaction with the Tech Wizard App and It gives examples of similar applications that user may have experience with that are similar to the behavior of the Tech Wizard app. This is important for us to look to as a guideline to make sure our users have a good experience while using our app.

User Profile Form
PROJECT: Tech Wizard
POTENTIAL USERS: - Anyone who needs technical assistance. - Technicians can use the app to create their profile, answer via phone, in-person and text chat questions that match their skills and knowledge, as well as setting up an in-person appointment and location.
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: This application is geared towards users that need technical help so the app will be easy to comprehend and follow with an easy to use interface. Users should have a basic knowledge of how to operate a smartphone and application before using the Tech Wizard App.
EXPERIENCE WITH SIMILAR APPLICATIONS: - Uber - Lyft - Facebook - ZipCar

- Maven Any app that requires you to create an account and allows you to communicate with someone via the app and gives the option to track the technician in a ride-share app style. However, if no experience in any of the above apps or similar, the app will be easy to understand.
TASK EXPERIENCE: - Create an Account - Select your issue - Select method of assistance (Phone, Chat, In-Person) - Payment method - Chat with support representative - Create Job Listing - View list of support representatives with skillset.
FREQUENCY OF USE: Whenever a user needs technical assistance, they can look for an available support representative. This can happen at any point in time and as frequently as needed.
KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS: - Easy to use app interface - Quick and efficient problem-solving - Ability to choose method of assistance - Timely response time when scheduling in-person assistance

Figure 1: User Profile

1.5.1 Project Title

Tech Wizard

1.6 USE CASE DIAGRAM

Figure 2: This displays our Use Case Diagram and explains how both the Tech Wizards and Client Users use and interact with the app. A user can be both a customer of Tech Wizard or an employer and how they both can use the app is illustrated below.

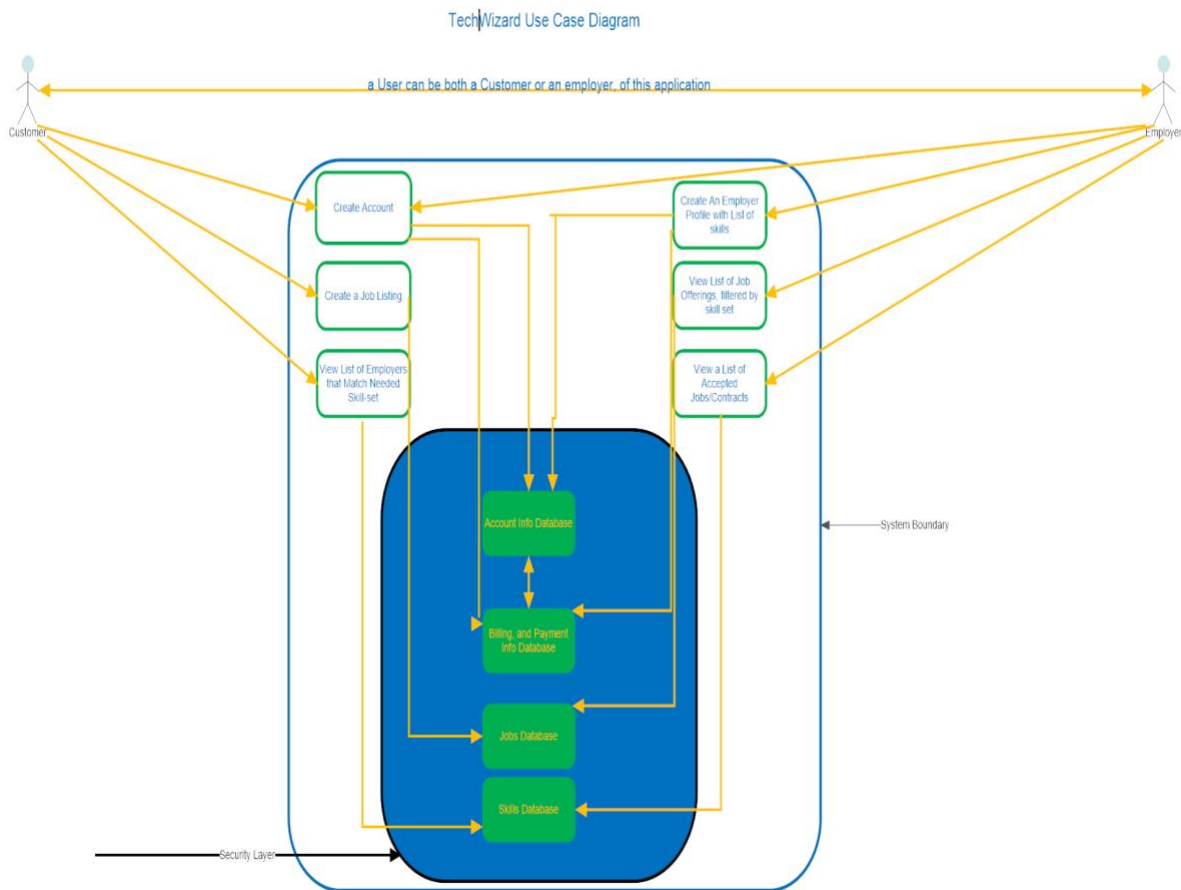


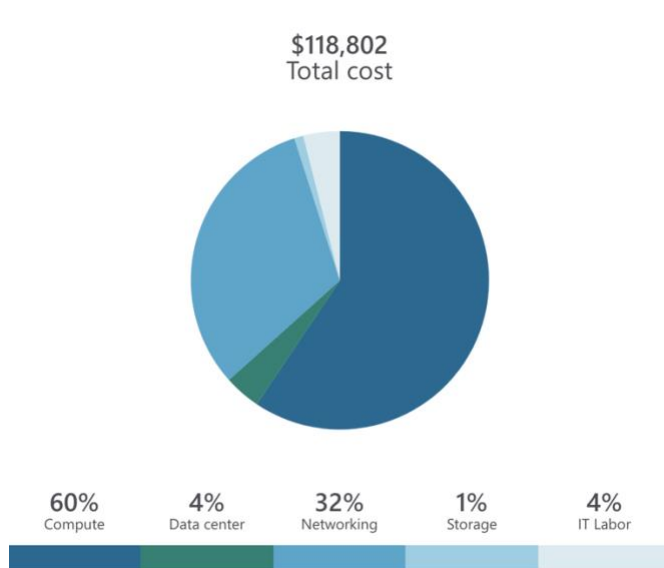
Figure 2: Use Case Diagram

2. PROJECT MANAGEMENT

2.1 Budget

Our budget total comes to \$334,722.66 which includes the salaries of the staff needed. Since this is a senior project, we have all been working for free so some estimates may be off, but the software and infrastructure needed is included. Our original budget can be found in **Graph 1:**

Original Budget Chart below:

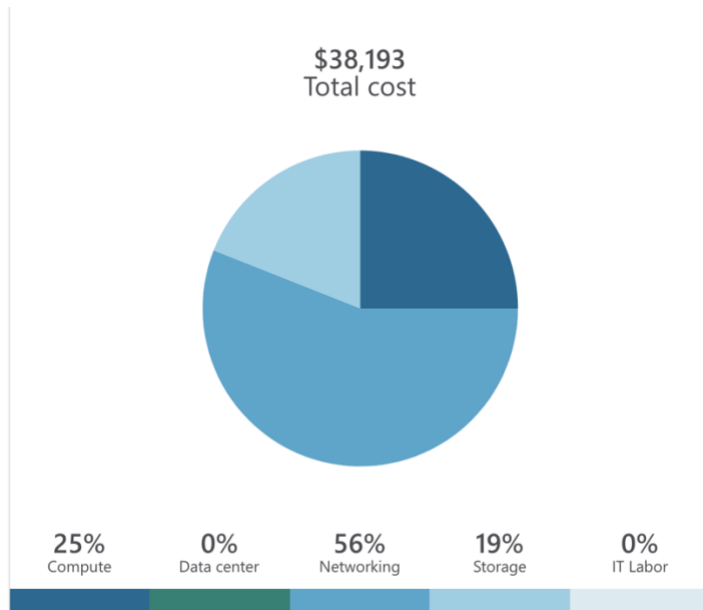


Graph 1: Original Budget Chart

Included in the **Graph 1: Original Budget Chart** is how much we would need if we had a physical on-premise environment. We decided to use Microsoft's Azure services to eliminate the need for an on-premise site to save us a lot of money. Note that percentages and total is the cost for 5 years.

Below you will find **Graph 2: Budget Chart** which shows our overall budget for the app over the course for 5 years. If this app were to go live, we would utilize Microsoft Azure for most of our app. It cuts down the cost to have a full IT staff and an on-premise environment which is mostly

driven by Compute and Data Center costs. You can see the percentages below of what each function would cost over the course of 5 years.



Graph 2: Budget Chart

In **Table 1: Cost Breakdown of Elements** you can see each function broken down into what each element costs over the course of 1 year.

Name:	Units:	Cost:	Total:
Computing	1 Core, 1.75GB RAM - SQL Database	\$1938.48	\$7638.66 1 year
Network	4TB outgoing bandwidth	\$4271.04	
Storage	- Premium Azure Disk Match 2048GB in Backup and Archive Volume	\$1429.14	

Table 1: Cost Breakdown of Elements

In **Table 2: Cost Breakdown of Human Resources** you can see the cost of staff needed.

Name:	Units/Hours:	Cost:	Total:
Project Manager	1/1200	\$80,000/year	\$320,000 1 year

Software Developers	2/1200	\$160,000/year	-
Database Manager	1/1200	\$80,000/year	-

Table 2: Cost Breakdown of Human Resources

In **Table 3: Software** shows what software is needed and the cost pertaining to the software.

Together, our entire budget including all 3 tables is \$334,722.66.

Software:	License:	Cost:
Visual Studio Enterprise	1	\$5999/1 st year
Xamarin	1	\$999/year
ASP.NET	1	\$86.00/year
Microsoft Azure w/MySQL	1	\$7638.66/year

Table 3: Software

2.2 Project Deliverables/Objectives

Below is **Figure 3: Project Deliverables** which describes the goals for our project. The table is broken into two sections. The top section is dedicated to the fall semester while the bottom section is for the spring semester. The left most columns are for assignments due for the class while the right most columns are the goals we worked towards for our actual project.

Fall Semester			
Course Work		Project Goals	
Team Contract	09/23	Research/Determine Software	10/01
Project Abstract	10/14	Create Welcome/Log-In	10/30
Contract Resubmission	10/14	Create User Database	11/07
Speech	10/21	Create Registration Page	11/14
Use Case Diagram	10/21	Create Workorder Page	11/21
User Profile	10/21	Create Workorder Database	12/01
Draft Report	11/10	Determine Further	12/15
Final Report	12/02	Requirements/Network/Security	
Presentation	12/??		
Spring Semester			
Course Work		Project Goals	

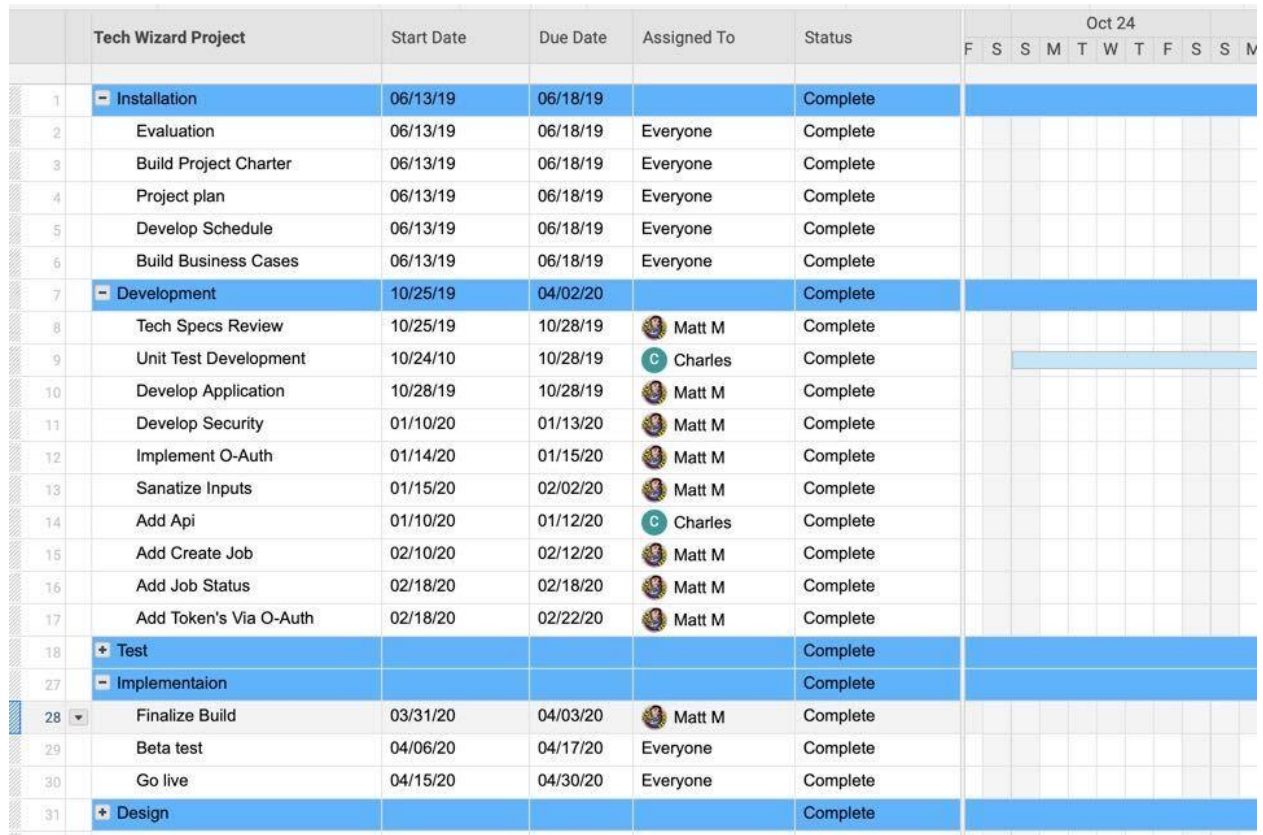


Figure 5: Gantt Chart

2.4 Work Breakdown Structure

Figure 6: Work Breakdown Structure shows the work that needs to be done in order for each objective and deliverable to be achieved.

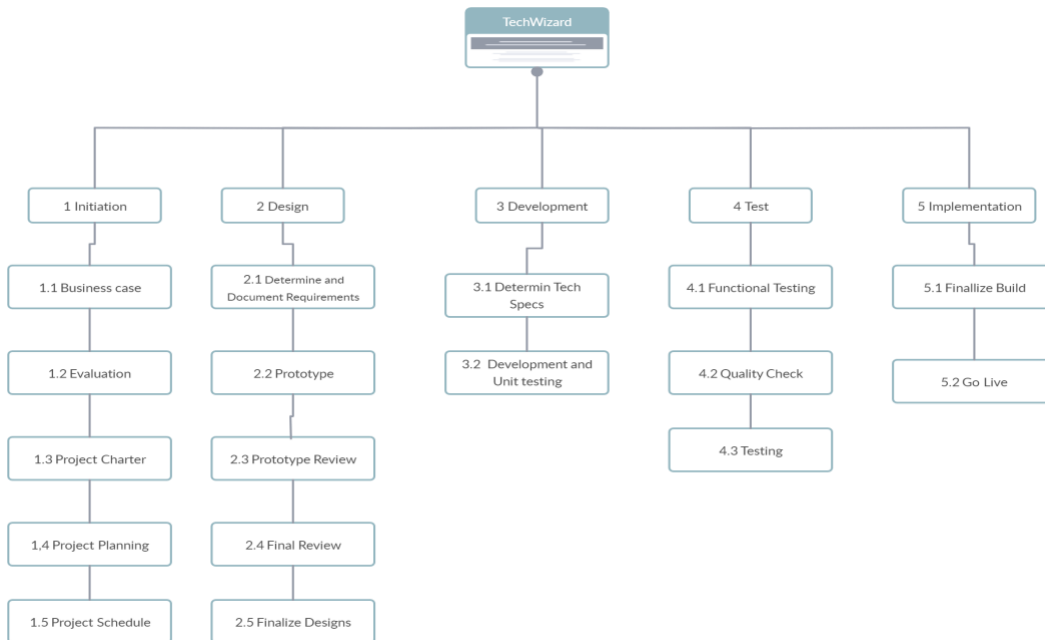


Figure 6: Work Breakdown Structure

3. TECHNICAL ELEMENTS

3.1 Network

For our web service we choose Microsoft Azure. Azure is a cloud computing platform development by Microsoft and as a Microsoft service it is easy to integrate with both Visual Studio and MySQL. Our Backend was developed using ASP.Net. ASP.Net is an open source framework that we used to develop our RESTful API. The API we created allows data to be transferred between our database and the application. When the application sends a request to GET, POST, PUT, or DELETE data the API will then open a connection with the database to take the appropriate action and return the information back to the application.

3.2 Database

Our project is used a MySQL database. MySQL is an open source relational database that supports a variety of programming languages. We'll be using MySQL to host all of our user, wizard technician, and work request data.

3.3 Application

Our project was designed in Xamarin forms. Xamarin Forms is a mobile framework program that allows users to create cross platform mobile applications using C#.

Xamarin uses a code sharing feature, which allows you to simultaneously create Apple, Android, and Windows applications using “native like” interface. The application itself has two different types of accounts. The standard user account and a “Wizard” account for our technicians. While both accounts function similarly the primary difference is that a Wizard account is able to accept created jobs within the system. The app also supports Xamarin Essentials which allows for accessing a mobile phone's text, email, phone dialer, and GPS features.

3.4 Security

We used the security features that come with using Xamarin to build the app. For Android apps, we will be using the AndroidClientHelper with TLS 1.2 support. For iOS support we will be using the Xamarin HttpClient Implementation Selector which offers the same as the AndroidClientHelper of TLS 1.2 support and supports Apples ATS (App Transport Security) which enforces that connections between internet resources and your app are secure (this also includes the backend of the app as well).

4. APPLICATION

4.1 Application Architecture

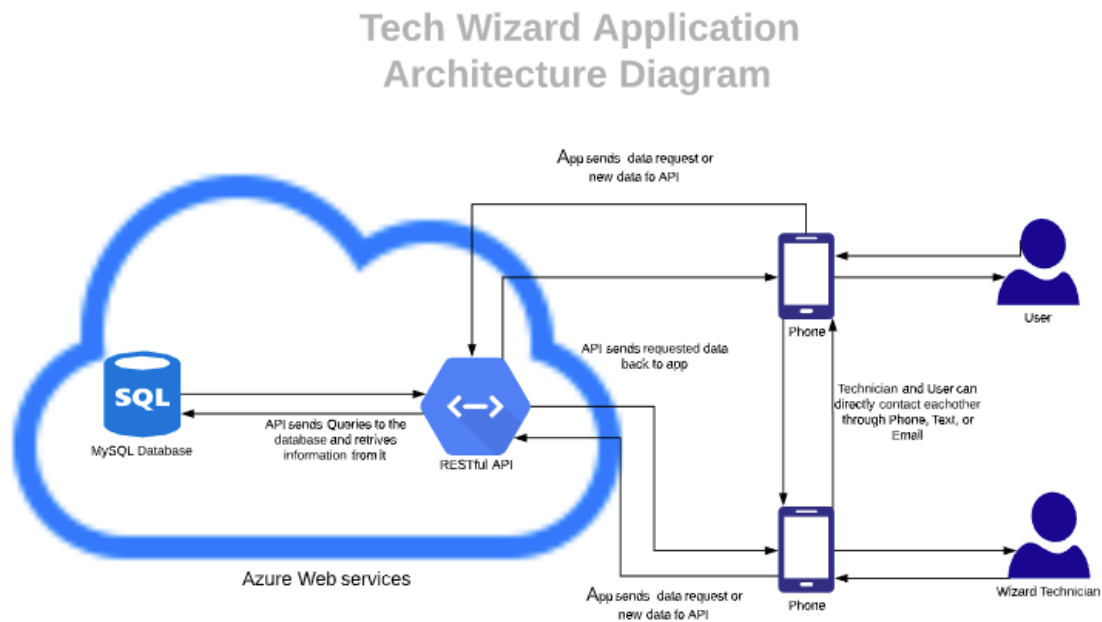


Figure 7: Architecture Diagram

Figure 7: Architecture Diagram: The application will use a Webservice to communicate with the database, and act as an intermediary for the application. This will allow a more seamless integration with the application as well as seamless integration with the database.

4.1.1 Application Software

Frontend: Xamarin Forms

Backend: ASP.NET

IDE: Visual Studio

Database: MySQL

Web Host: Azure

4.2 Screenshots and Examples

4.2.1 Homepage



Figure 8: Homepage This is the first page our users will see when opening the app. New members can choose to sign up and current users can login. The buttons are large enough for users to be able to easily distinguish between the two choices.

Figure 8: Homepage

4.2.2 User Homepage

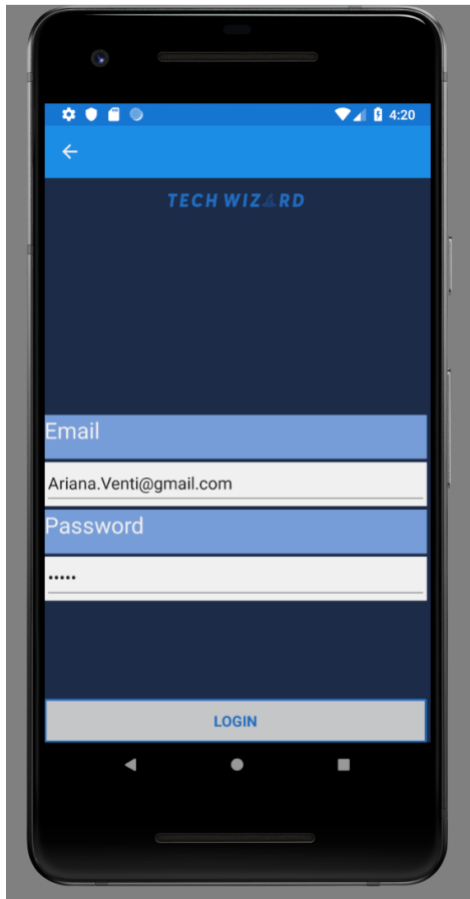


Figure 9: User Login Page

Figure 9: User Login Page

This is what is displayed when a current user selects the Login option from the homepage. This displays a text box to fill in the users email they used to sign up for a Tech Wizard account and a text area to fill in the password they set. The login button is conveniently at the bottom to avoid any mis-clicks.

4.2.3 Sign Up Page

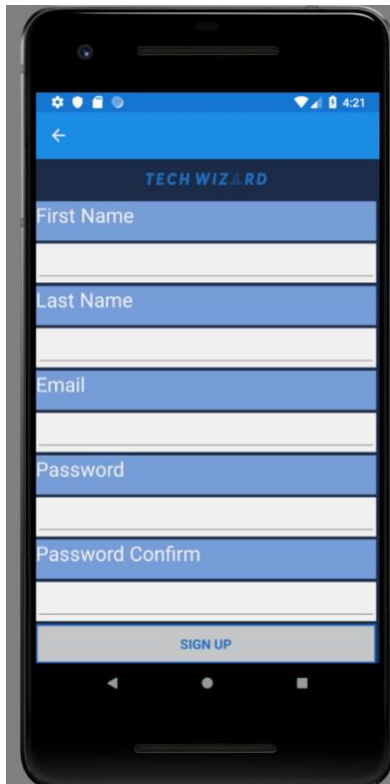


Figure 10: Sign Up Page The Sign-Up button will take the user to this page where they can fill out basic information about themselves such as Name, Email, and Password. This will create their account and is made to be a quick process so they can start using the app as soon as possible.

Figure 10: Sign Up Page

4.2.4 User App Homepage

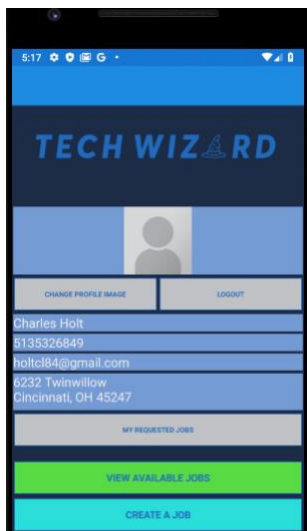


Figure 11: User App Homepage is the page current users will see. They will be able to create a job request from this page located at the bottom. They will also be able to view available jobs that they have open currently.

Figure 11: User App Homepage

4.2.5 GPS Page

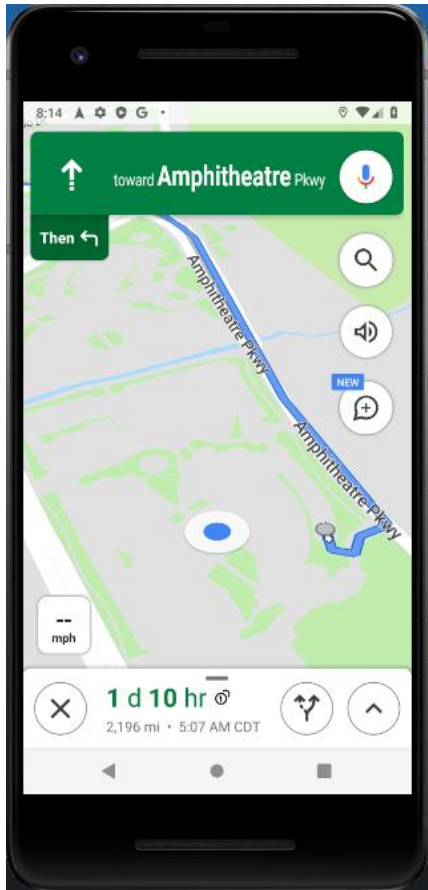


Figure 12: GPS Page

Figure 12: GPS Page shows the prompt that appears when a user requests an in-person Wizard. The Global Positioning System (GPS) routes directly to the address given to the technician.

5. TEST PLAN

5.1 Overview

This section references the tools and information used to conduct testing for our Tech Wizard app. The information found here consists of usability tests we've conducted to assure the functionality of the features we currently have for the app. The Tech Wizard app can run on Android software but will be integrated to iOS as well. This section will be helpful to Developers, Team Members, Beta Testers, and UI Designers.

5.2 Objective

The goal of testing the Tech Wizard app is to ensure the functionality and integrity of the features of the app work as intended without error. We want to be positive that our code can work on both Android and iOS devices without compromising certain features on each platform. We will test each part of the app individually to closely study the behavior and functionality.

5.3 Test Cases

The Tech Wizard app will be tested by individual features and as a full app itself in order to have full confidence in our app's performance. We want to tackle any performance or functionality issues before it will be put out for users to use. We want to test the full app from the viewpoint of the users' and the wizards' side of the application.

5.3.1 Testing Procedures

To start with the testing of the Tech Wizard application, we will set pass/fail conditions and document them accordingly including the duration of the test. By showing accurate measures of the testing procedures, team members and developers can see what important aspects of the app were tested and gives integrity of the app itself. We will be performing the following tests:

- Start-Up and Launch test – this test consists of launching the app itself to see if it crashes and at what point it crashes. This is important to the overall functionality of the app itself so users can use it without it crashing. We will test this using an android emulator and record the amount of attempts we conducted to launch the app and if it fails at any point.
- Duration and Stability – we will test the app itself to see if it runs as expected without any crashes or freezes. This is important to ensure the app will not crash on the user no matter the amount of time the app is open. Duration of how long the app runs will be recorded and the behavior.
- UI and UX – in this test we will go through the app as the user and run certain functionalities and options available to the user. We want to make sure our features work as intended without error and to ensure it is easy for the user to understand the options available to them. We will record this by running through a series of features we currently have and making sure it works as intended on the users end.

5.4 Testing Scenarios

5.4.1 End User View

First Scenario:

A user launches the app from their smartphone and should see the following screen in Figure 13 below. This should be the first screen the user sees when launching the app and has the choice of logging in or signing up for new users.



Figure 13: Home Screen

Second Scenario:

A new user launches the app and sees the home screen (referenced in Figure 14) and chooses to sign up and create an account. After selecting “Sign Up” the user should see the screen in Figure 14 and input their information. Figure 15 shows an example of information needed.

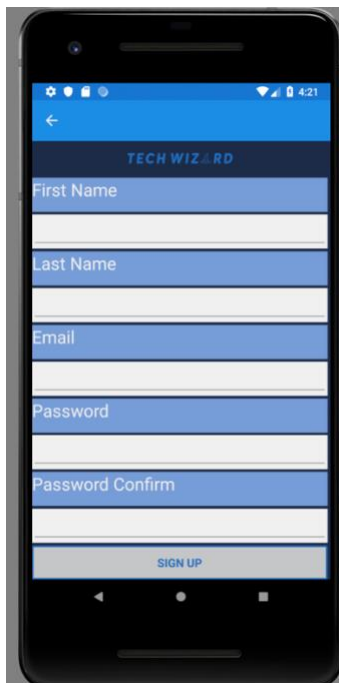
A screenshot of a mobile app's sign-up page. The page has a blue header with a back arrow and the text "TECH WIZARD". Below the header are five input fields: "First Name", "Last Name", "Email", "Password", and "Password Confirm". Each field has a blue label and a white input area. At the bottom of the form is a blue button labeled "SIGN UP". The status bar at the top shows the time as 4:21.

Figure 14: Sign Up Page

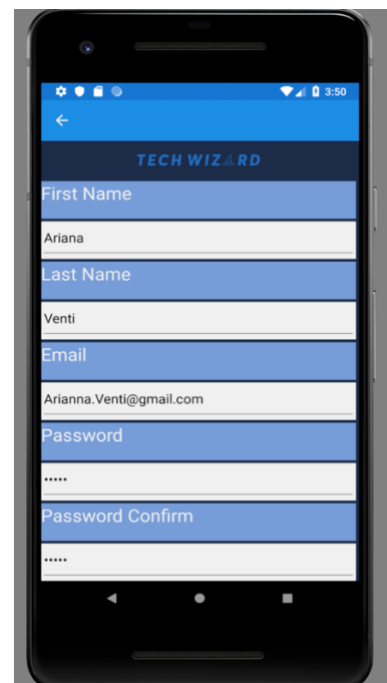
A screenshot of the same mobile app's sign-up page, but with example data entered. The "First Name" field contains "Ariana", the "Last Name" field contains "Venti", and the "Email" field contains "Arianna.Venti@gmail.com". The "Password" and "Password Confirm" fields contain six asterisks. The blue "SIGN UP" button is at the bottom. The status bar at the top shows the time as 3:50.

Figure 15: Sign Up Page Example

Third Scenario:

An existing user launches the app from their smartphone and selects the Login button.

The user should see the following screen in Figure 16 after logging in.



Figure 16: Login Homepage

5.4.2 Wizard's View

Fourth Scenario:

A Wizard logs into the app and should see the following screen. The Wizard can view open job requests from end users who have a technical issue. Figure 17 depicts the Wizard's page showing the option to view available jobs.

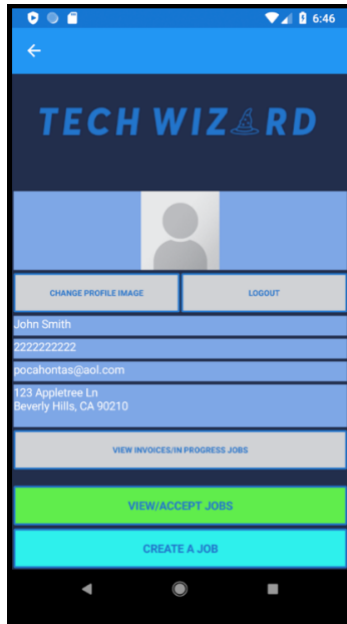


Figure 17: The Tech Wizard Homepage

Fifth Scenario:

A wizard logs in to view a list of jobs available to accept. The Wizard will log in and choose from a list of current jobs to accept and will be able to see more information of what each job entails. In **Figure 18: View Jobs Page**, you can see the list of available jobs awaiting a Wizard as well as Claimed Jobs which show jobs already in progress for that Wizard. **Figure 19: Jobs Details** shows the details of a particular job.

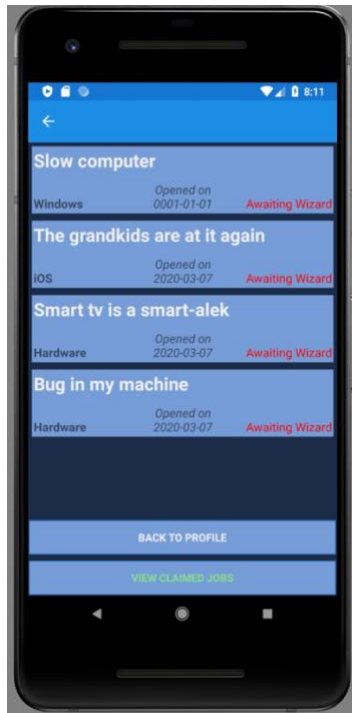


Figure 18: View Jobs Page

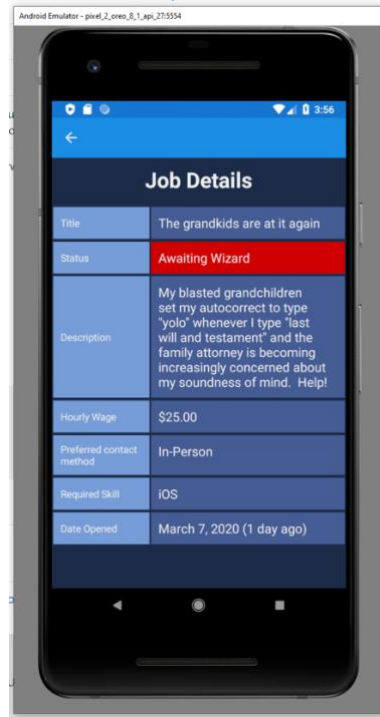


Figure 19: Job Details Page

Sixth Scenario:

A client has requested a text chat to solve their issue. A wizard will log in to view the invoice on the list of jobs and bring up a text chat window to solve the issue. **Figure 20: Client Info** shows the clients information, status of the job, description and method of assistance. **Figure 21: Phone Dialer Window** shows the prompt window that is brought up to assist the client with their phone number to text or directly call. **Figure 22: Chat Window** shows the text message sent from the Wizard to the client.

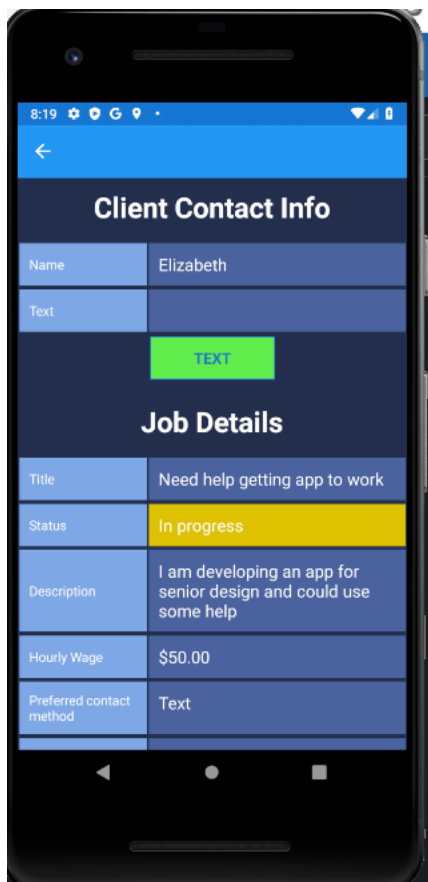


Figure 20: Client Info

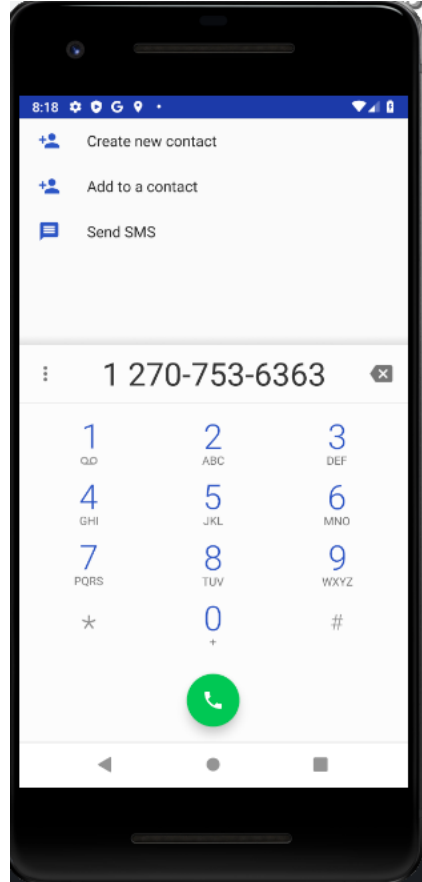


Figure 21: Phone Dialing Window

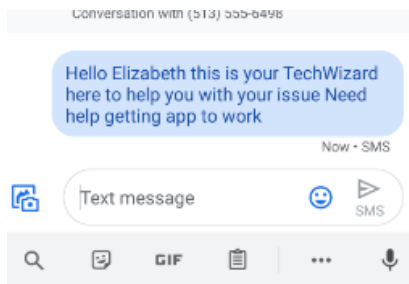


Figure 22: Chat Window

5.5 Pass/Fail Conditions

Each section of the following tables must pass to have a properly working and responsive app.

5.5.1 Start-Up and Launch Testing

Start-Up/Launch Testing	Description	Pass/Fail Condition
App Launch	App must load correctly and efficiently and not crash on the user when launching the app.	Pass ✓
App Response	App must open when the user clicks the app icon.	Pass ✓

Table 4: Start-Up and Launch Testing Conditions

5.5.2 Stability Testing

The following table shows the testing needed to look at the overall stability of the app once it's already been launched.

Stability Testing	Description	Pass/Fail Condition
Background Activity	The app must stay idle in the background when not manually crashed by the user. If the user minimizes the app, it should not completely exit the app itself.	Pass ✓
App Delay	When navigating through the app, all features must have decent response time and ensure that certain features do not impede or slow-down the performance of the app.	Pass ✓
Idling	The app should run smoothly even if left open and running for over an hour.	Pass ✓
Platform Testing	We will run the app on an Android phone to ensure stability on the Android platform.	Pass ✓

Table 5: Stability Testing

5.5.3 User Interface Testing

User Interface and Features	Description	Pass/Fail Condition
User Input	Test user text fields when signing up or logging into the app. All text fields must record what is inputted into it correctly.	Pass ✓
Button Placement and Nav	GUI buttons must be placed accordingly and not overlap with other button features. Accuracy of placement and navigation must be easy to understand and use.	Pass ✓
App Scrolling	App must be able to scroll as needed without error or crashing.	Pass ✓
Navigation Time	We want to make sure the user can smoothly run through the app without long loading times or delays in the feature.	Pass ✓

Page Links	Each page of the app must be linked correctly and obvious to the user of how to get to a certain feature.	Pass ✓
Features Testing (End User)	Features we will be testing to ensure functionality: • In-Person – GPS capability so end users can see an approximation of where their Tech Wizard is should they choose the In-Person support method. • Phone Support – If a user chooses phone support as their method of technical help, app must be able to handle call functions. • Text Chat – If a user chooses the text chat method of support, app must be able to handle a text feature so users can communicate promptly to the end Wizard.	Pass ✓
Features Testing (Wizard User)	• Wizard must be able to view and accept currently open job/tickets that the end users enter when they have a technical issue. • Wizard must be able to have the same functions as the end user (eg: access to text chat, phone, and in-person gps capabilities)	Pass ✓

Table 6: User Interface and Features Testing

5.5.4 Testing Timeline

We all tested it independently to ensure functionality.

Name	Role	Frequency
Erica Partin	Project Manager	Bi-Weekly
Matthew Miracle	Software Developer	Bi-Weekly
Charles Holt	Database and Software Dev	Bi-Weekly

Table 7: Testing Timeline

5.6 Lessons Learned

During testing we found a few errors when first trying to get the API to communicate with the app. That was solved by making sure the database password was correct. We often got the error “this project references NuGet packages that are missing on this computer” which was solved by redownloading NuGet packages. Another issue we had was “xamarin.android for visual studio requires Android SDK” which was solved by redownloading everything and making sure we all had the same SDK. Most of our

errors were easily solved by using google and each other for help. We found that sometimes downloading the code from GitHub wasn't always reliable as it wouldn't download every directory that was included. We found that trying to implement a video-chat feature using our own code instead of using a third-party service like Skype was much more time consuming because we did not figure out successfully to integrate a camera feature. Future features we would like to implement is the option to video-chat and also to enhance the GPS capability to be more in real time. There is a latency with the tracker as with most GPS-gearred applications.

6. CONCLUSION

6.1 Fall Semester 2019

We have been researching a lot of what the best avenues of application development exists for this kind of app and to begin creating the basic application usage. We want to integrate a GPS type of system for our users who need a physical technician which we have been testing out to see what would work best for our software we plan to use. We expect to have a working prototype by the end of the semester into the beginning of the Spring semester and move forward with more features to be added and fine tuning the code.

6.2 Spring Semester 2020

We are very proud of the working product we have, and we feel our skills have dramatically enhanced because of this project. We covered many areas in software development and database development all in one app. We made an app that can GPS track, text chat, email, and phone support function which all contained different elements to come together into one. We got to work with OAuth security functions and learned how to sanitize inputs within the code, which is pivotal to app security and performance. If we had more time, we would have implemented a video-chat function and enhanced the GPS function to make it more accurate. The video-chat idea had to be scrapped due to time and technology. We could have used a third-party service, but we wanted to host the video-chat feature ourselves and did not have a big enough timeline. If we could go back, I think we would have stuck to a few features at first and

gradually branched off of those and fine-tuned them instead of doing several different features like video-chat. Suggestions that was given to us was to have a better GPS function and to add a “forum” type of service where users of the Tech Wizard App can go and make how-to guides for simple technical issues like restarting a router.

APPENDIX A: ADDITIONAL INFORMATION

Poster:



TEAM 053

Erica Partin
Matthew Miracle
Charles Holt

UNIVERSITY OF CINCINNATI - COLLEGE OF EDUCATION,
CRIMINAL JUSTICE, HUMAN SERVICES - SCHOOL OF
INFORMATION TECHNOLOGY

TECHNICAL ADVISOR: YAHYA GILANY UNIVERSITY OF Cincinnati

THE AVERAGE PERSON WILL SPEND AROUND 43 DAYS OR MORE OF THEIR LIFE ON HOLD

PROBLEM

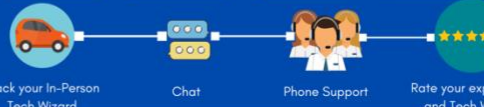
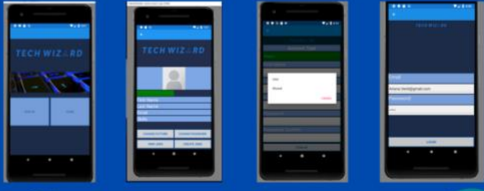
- **01**
Technology is difficult to keep up with. Extra help is needed.
- **02**
Long support desk wait times lead to unresolved issues.
- **03**
Help isn't always available when you need it.



OUR SOLUTION:

An app that connects users who need technical support to an expert who can help whenever and wherever you need it. Solving technical issues without the frustration of navigating automated menus and being put on hold. You choose how you want technical support - via chat, phone, or in-person when you need it.

DESIGN & TECHNOLOGY



Track your In-Person Tech Wizard Chat Phone Support Rate your experience and Tech Wizard

FRONT END	BACKEND
 VISUAL STUDIO	 XAMARIN
 DATABASE MySQL	 WEB SERVICE Microsoft Azure

Abbreviations: GPS (Global Positioning System)

APPENDIX B: REFERENCES

1. "Arise Customer Service Frustration Series: Phone Hold Times."
<https://www.arise.com/resources/blog/arise-customer-service-frustration-series-phone-hold-times>. Accessed 14 Oct. 2019.