

Sched-Bot




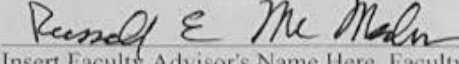
by

Andrew Barker, Adam Brooks, and Ahmad Nassar

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 2017 Andrew Barker, Adam Brooks, and Ahmad Nassar

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

 _____ Andrew Barker	<u>4/17/17</u> Date
 _____ Adam Brooks	<u>4/17/17</u> Date
 _____ Ahmad Nassar	<u>4/17/17</u> Date
 _____ Insert Faculty Advisor's Name Here, Faculty Advisor	<u>4/17/2017</u> Date

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

April 2017



Sched-Bot

Andrew Barker, Adam Brooks, and Ahmad Nassar

Students of University of Cincinnati

College of Education, Criminal Justice, and Human Services

School of Information Technology

Cech.uc.edu/it

Contents

Abstract	1
Problem Statement	2
Introduction	2
Project Description	2
Problem.....	2
User Profile.....	3
Use Case Diagram	6
Project Management	7
Objectives/Deliverables	7
Project Schedule	7
Budget.....	8
Technical Elements.....	8
Software Languages	8
Server.....	9
Hosting	9
Application.....	10
Security.....	10
User Interface	10
Home Interface:	10
Management Interfaces:.....	10
Employee Interfaces:	11
testing.....	11
Conclusion	15
Fall Semester 2016	15
Spring Semester 2017	16
APPENDIX A: REFERENCES.....	17
APPENDIX B: ADDITIONAL INFORMATION	18
APPENDIX C: THREAT MODEL	19

APPENDIX D: TECH EXPO 2017 POSTER 20

Appendix E: Application Wireframes 21

Appendix F: Open Source Software and Use Components 24

Figures

Figure 1: User Profile Form.....	5
Figure 2: Use Case Diagram.....	6
Figure 3:Project Gantt Chart.....	7
Figure 4: Database Schema	9
Figure 5: Threat Model Diagram.....	19
Figure 6: Tech Expo Poster	20
Figure 7: Sched-Bot dashboard (calendar view) on a desktop	21
Figure 8: Sched-Bot dashboard (calendar view) on a mobile device.....	21
Figure 9: Employee details and availability page on a desktop	22
Figure 10: Employee details and availability page on a mobile device	22
Figure 11: Table view of a schedule on a desktop	23
Figure 12: Table view of a schedule on a mobile device	23

Tables

Table 1: Restaurant Managers' Time and Money Spent on Employee Scheduling	3
Table 2: Project Objectives/Deliverables Due Dates	7
Table 3: Project Budget	8
Table 4: Application Development Testing Timeline.....	14
Table 5: Round Table Testing Timeline	15

ABSTRACT

To some, scheduling employees is a monotonous and often times difficult task to correctly manage. You have all your employees asking for their perfect schedules, resulting in a complex, and time-consuming mess in which countless errors can occur. With Sched-bot, manually creating weekly shifts is a thing of the past! By developing a Web platform that is also accommodated by Android and iOS mobile applications, managers can finally devote their time to more meaningful, or even neglected tasks. With access to Sched-bot, managers acquire the opportunity to manually, or automatically allow generation of their employee schedules, while also providing employees the ability to have live views of the current and forecasted schedules. The convenience of the mobile applications presents each employee with the opportunity to effortlessly complete off day, or shift trade requests; Sched- bot also requests approval from the manager and sends alerts whenever a schedule or request has been finalized. With Sched-bot, say goodbye to the headache of scheduling all of your employees!

PROBLEM STATEMENT

Introduction

Businesses and other organizations which require hourly shifts, must create schedules to fill the shifts. Management teams and/or managers try to make schedules which accommodate all parties involved, to prevent any possible conflicts. The managers are aware of shift needs and desired time periods. Scheduling is a crucial aspect of hourly shift systems and the continuous operation of these organizations rely on their employees, and these employees rely on an accurate schedule. Without this, the business and employee partnership is tarnished.

Project Description

An automated scheduling system will be created to address the needs of both businesses and employees. The system will be completely Web based, accessible from computers, and will be mobile friendly. Schedules will be created automatically using a custom algorithm. Employees will be able to request days off at their own discretion through the application and the system will build schedules around available days.

Problem

The problem organizations face is that manually creating schedules is time consuming, making it difficult to accommodate individual availability. Restaurant managers using a spreadsheet for scheduling have been shown to spend an average of 3.14 hours per week on employee scheduling.¹ These managers are spending 7.86% of their work week on this mundane task.¹

Scheduling is an important key to running these businesses; Eliminating this task would allow this time to be better spent elsewhere. Furthermore, schedules are only available on the site of operations. This means there is no way to see what a future schedule would look like. No one has an opportunity to plan ahead.

Weekly	Monthly	Yearly
3.14 hours	12.56 hours	150.72 hours
\$62.80	\$251.20	\$3,014.40

Table 1: Restaurant Managers' Time and Money Spent on Employee Scheduling

Table 1 above demonstrates the average time and money wasted on scheduling by restaurant managers at an assumed rate of \$20/hour. This amount of time and money is only amplified by the number of employees and size of the businesses.

User Profile

Management:

Management consists of the people managing the scheduling for hourly shifts. They will have general knowledge of the business and its scheduling needs. Management/Managers will benefit from the Automated Scheduling Tool by eliminating the time spent creating schedules. They will be relieved of the duty of taking care of shift coverage when workers need days off in advance. However, management will still be in charge of keeping the necessary staff in the case of an emergency.

Hourly/Shift Workers:

Workers will benefit from the use of the application by being able to access work schedules wherever they are. Employees will be able to simply input their days of availability and be automatically notified of the days they are working. Furthermore, all workers will have the capability to request days off, request a cover of shift, or pick up a shift. Instant notifications allow both managers and workers to benefit from this intuitive and innovative system. Figure 1 depicts the User Profiles for the product.

PROJECT: Sched-Bot
POTENTIAL USERS: <ul style="list-style-type: none">- Hourly/Shift Workers- Management- Small/Medium/Large Businesses
SOFTWARE, INTERFACE, AND RELATED EXPERIENCE: <ul style="list-style-type: none">- Operating Systems- Computer Web Applications- Mobile Web Applications- Web Browsers
EXPERIENCE WITH SIMILAR APPLICATIONS: <ul style="list-style-type: none">- Google Chrome- Safari- Twitter Bootstrap Web Applications

<p>TASK EXPERIENCE:</p> <ul style="list-style-type: none">- Creating Schedules- Using Calendars- Spreadsheets
<p>FREQUENCY OF USE:</p> <ul style="list-style-type: none">- Daily Monitoring- Weekly Tasks- Monthly Tasks- Reporting
<p>KEY PROJECT DESIGN REQUIREMENTS THAT THE PROFILE SUGGESTS:</p> <ul style="list-style-type: none">- Industry Designs Standard- Easy to use Web Interface- Easy to use Mobile Interface- Useful Notifications

Figure 1: User Profile Form

Use Case Diagram

Figure 2 depicts the use case diagram for our application

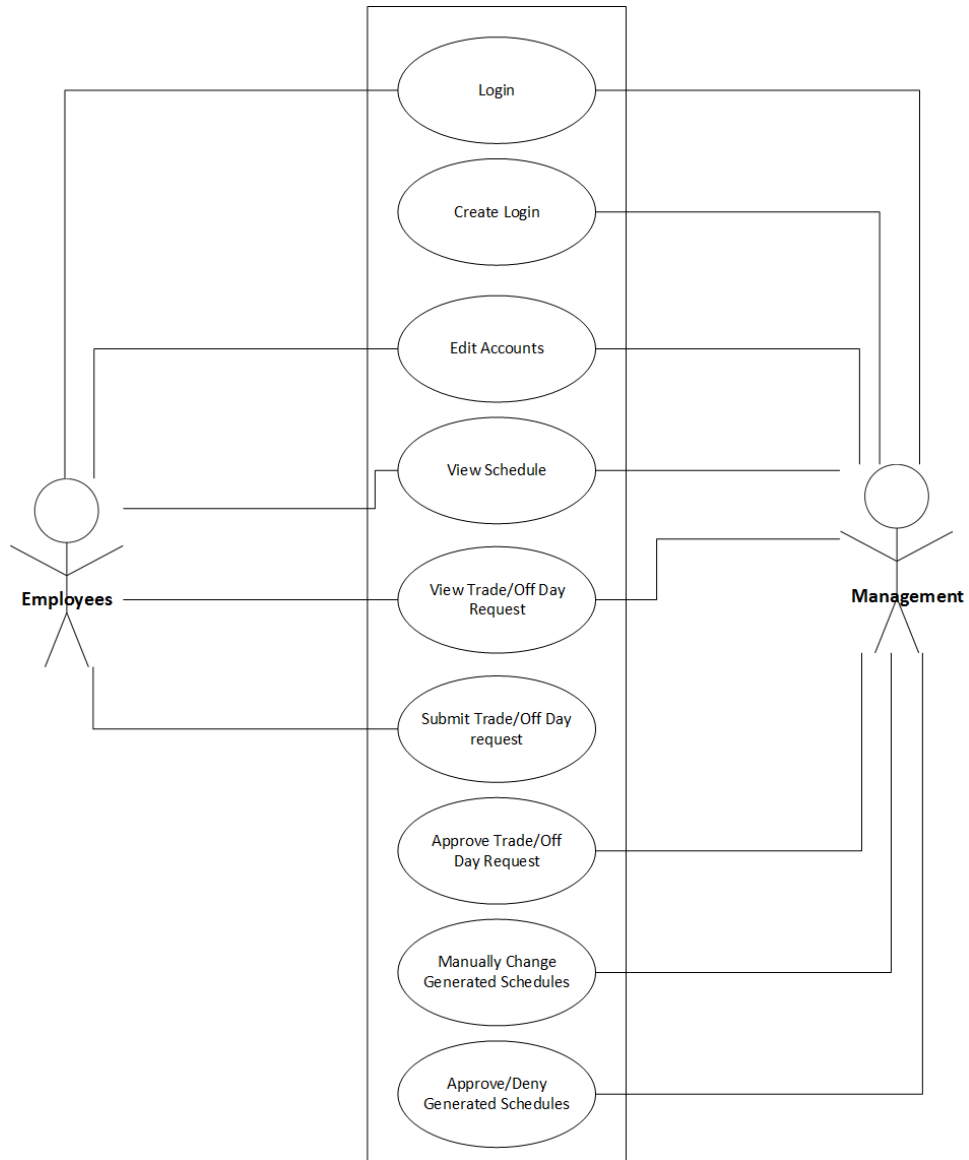


Figure 2: Use Case Diagram

PROJECT MANAGEMENT

Objectives/Deliverables

Table 2 presents the project deliverables and deadlines

MAJOR PROJECT MILESTONES (DELIVERABLES)				
Pre-Project Planning	10/24/16		User Interface Milestone	11/7/16
Wireframes Milestone	10/24/16		Software Demo Milestone	11/23/16
Pre-Network Setup Milestone	10/31/16		Fall Presentation Milestone	11/28/16

Table 2: Project Objectives/Deliverables Due Dates

Project Schedule

Figure 3 illustrates our project timeline from the beginning of the fall semester through completion of Sched-Bot.

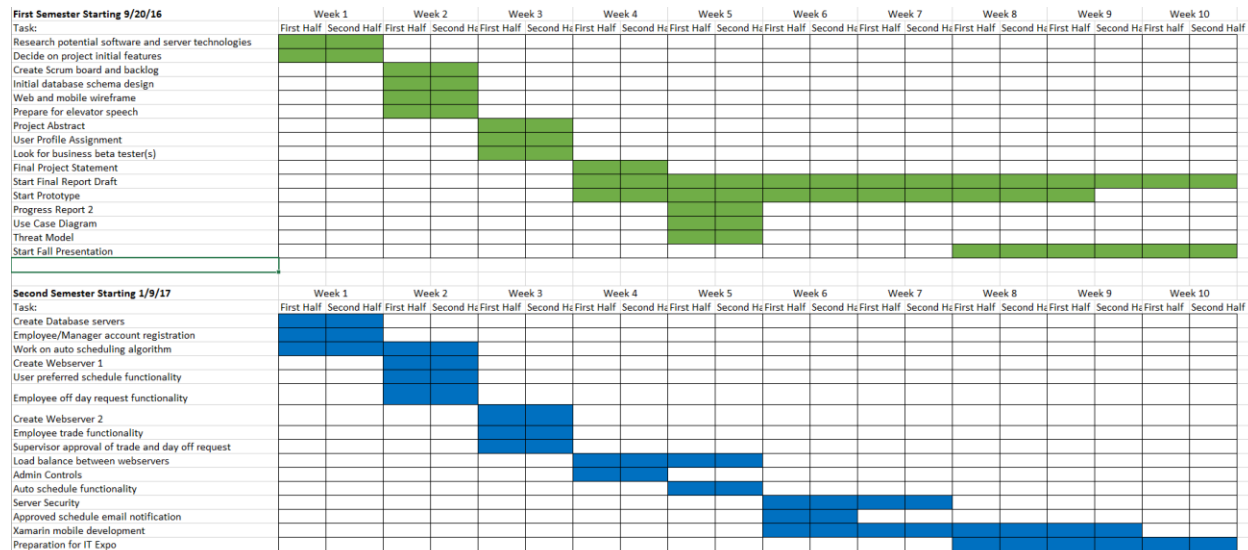


Figure 3: Project Gantt Chart

Budget

Table 3 presents the budget for Sched-Bot. This budget replicates the cost of our solution in the business world; however, as we are developing an open source solution our total comes is \$0.

No.	Item	Unit, Each	Unit, Price	Line Item Total
Networking				
1	Labor	100	\$75	\$7,500
2	Resources	1	\$510	\$510
Software				
3	Developer 1 Labor	150	\$75	\$11,250
4	Developer 2 Labor	150	\$75	\$11,250
	Subtotal			\$30,510
	Total to public			\$0

Table 3: Project Budget

TECHNICAL ELEMENTS

Software Languages

Sched-Bot will be developed within Visual Studio, utilizing C# ASP.NET MVC. Early phases will run locally on Visual Studio. It will use Entity Framework for sending and retrieval of data from databases. We chose asp.net and entity framework, as it will allow us to produce and publish the Sched-Bot Web site in a time-sensitive manner. Also, ASP.NET is one of the most widely used platforms for web development, which gives us access to many libraries and documents should we run into any blockers. The Sched-Bot team will utilize Model View Controller to keep code organized, automatically generate forms, and to implement the principle

of least privilege throughout the website easily. JavaScript and Html will be used on the UI. We will use cshtml along with Razer syntax for model binding. Mobile applications may be developed, as well as Java for Android and Swift for iOS. GitHub will host our code repository.

Server

Sched-bot will run on a windows Web server. SQL Server will be the database utilized on the server. Figure 4 illustrates the initial database schema we have produced for Sched-Bot -backups will be made. When ready to deploy, another server will be started as a production server. The old server will become the development server, allowing us to test changes before they are pushed public.

Hosting

The server will be hosted by the Sched-Bot team. This allows us to handle networking security such as a firewall. Also, we can provide as many backups as needed. This will also allow for easy scalability, should the need arise. Figure 4 depicts the Database Schema for Sched-Bot.

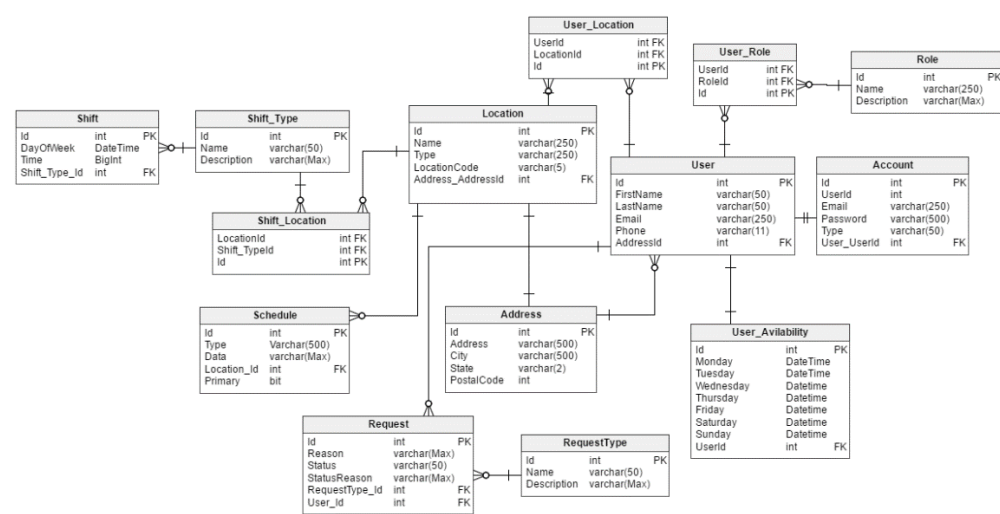


Figure 4: Database Schema

APPLICATION

Security

One major component of our solution is security of private data. The application will be developed to defend against any SQL threats, as storing information about when employees will be away from their homes, and the location of homes can be a real security issue. The application will ensure that users can only perform actions limited to their roles; therefore, normal employees will not be able to view or perform any of the actions strictly related to only management users.

User Interface

Home Interface:

The home interface is the first page of the application. When a user enters the Web site, this is the very first page the user will see. The page will contain descriptions of our scheduling solution, benefits, and provide the user with the ability to sign into their accounts if he/she has one.

Management Interfaces:

The management interface will grant management users access to the application and view trade/off day requests made by employees, as well as granting them the ability to approve or deny the requests. Management users will also be able to view or edit their employee's roles. Lastly, and most importantly, management will be able to approve generated schedules and manually make changes as they see fit.

Employee Interfaces:

The employee interfaces will grant employees access to view current and forecasted schedules based on everyone's availability. Furthermore, employees will have the ability to send trade requests to other employees and request particular days off.

TESTING

Overview

The testing methodology for the Sched-Bot web application will be detailed in this section. This should be used as a guide for the testing plan. The following individuals should use this section:

- Developers
- Project Managers
- Team Members
- Outside Testers

Scope

The scope of testing will include all implemented features, ensuring they work as expected. Tests will also be based on core requirements to verify a useable application. These tests will take place on the web application Sched-Bot, along with companion applications on mobile.

Objective

The objective of testing is to confirm the usability and functionality of the application. These tests will examine all of the different aspects of the application. Testing will be conducted by developers as well as outside, third-party testers.

Entry and Exit Criteria

Entry Criteria:

- High-Level Design Specifications
- Software Development Project Plan Complete
- Business Requirement Document complete
- Compiled Code
- Configured Test Environment
- The user interface and the feature set are frozen

Exit Criteria:

- Unit Tests pass
- All priority a bug have been fixed and closed
- Internal documentation has been updated to reflect the current state of the product.

Logging Test and Reporting

While executing the test cases, if an individual finds a defect, the individual will document the defect thoroughly on our scrum board. Once the defect has been created on the scrum board, the individual will then assign the ticket to the project manager to assess the defect and distribute the work. Once the defect has been examined and assigned to be fixed, one or both of the developers will work on resolving the bug.

System Testing

The Sched-Bot web application will be tested thoroughly as a complete application. Any bugs or other flaws will be revealed better when the complete web application is tested as one entire unit. It is intended to verify Sched-Bot's functionality as a whole.

Testing Procedures

Testing will require the ensuing steps:

- Construct all scenarios and test cases
- Create a document containing steps taken to administer the test and the expected outcome
- Identify any bugs in the report

Below are descriptions of the classifications of tests we will perform:

- User Interface Testing: This set of test cases will ensure the interface operates as expected. The UI will be tested amongst different screen sizes and resolutions.

- Feature Testing: This series of tests check the functionality of the applications. Ensuring that features actually function and create results as expected.
- Performance Testing: This series of tests verifies that our servers are running quickly and efficiently under load. This will be done with multiple testers testing at once.

Pass/Fail Conditions

Testing will be considered successful if all major functionality of the application works as intended. If a core feature is not working it will not pass the testing phase. Otherwise minor bugs will be patched.

Schedule of Team Member Testing

Table 4 is a timeline and frequency in which our team tested our application.

Tester	Timeline	Frequency
Backend Developer	9/19/2016 - 4/10/2017	Weekly
Frontend Developer	9/19/2016 - 4/10/2017	Weekly
Server Administrator	9/19/2016 - 4/10/2017	Each Build Release

Table 4: Application Development Testing Timeline

Schedule of Round Table Testing

Table 5 can be reference to view the schedule of our round table testing.

Round Table	Timeline	Frequency
Management	1/1/2017 - 4/10/2017	Monthly
Employees	1/1/2017 - 4/10/2017	Monthly
UX Expert	1/1/2017 - 4/10/2017	Monthly

Table 5: Round Table Testing Timeline

CONCLUSION

Fall Semester 2016

In the 2016 Fall Semester, the team devised a plan for designing and creating Sched-Bot. Hosting options were researched and a plan for using the UC Sandbox as a testing environment was formed. A database was designed and created using Vertabelo. A prototype was created using C# ASP.NET in Visual Studio. Wireframes for both the Web application and mobile applications were also completed. One of the largest roadblocks that we have run into so far, is the time required for developing the equation for automatically generating the schedule. Once the algorithm is developed, it will also be a challenge to correctly code through the application, and test against it to ensure it is correct.

Spring Semester 2017

In the 2017 Spring Semester, the team completed the Sched-Bot project. The final development of the application was completed. The application was optimized for desktop and mobile access. The algorithm for automating the randomization of schedules was created and finished. The database was built and finalized. Two web servers and a database server were created. The web servers were load balanced to handle large loads of traffic. The project was presented at the 2017 UC IT Expo.

What We Learned

We learned how to setup multiple servers and link them together to host a web application. We learned how to load balance our web servers to handle large amounts of traffic using Application Request Routing and creating a server farm. We learned how to build and package an application that was ready to be deployed through the publishing steps within Visual Studio as well as the package deployment process on the both IIS web servers to correctly display the application online. We learned how to build and utilize a database server. We also learned how to connect an application to a remote database and the steps needed to open the correct ports within the SQL client as well as the firewall.

Next Steps

We will move our application to servers outside of UC. This will allow us to use the application from wherever in the world without being connected to UC's VPN. We can show off our project or continue to expand and build it if we choose to.

APPENDIX A: REFERENCES

1. Audrey Presley, “Restaurant Scheduling Statistics”, Timeforge, May 27, 2007, accessed October 30, 2016, <http://www.timeforge.com/site/scheduling-facts/restaurant-scheduling-statistics-2/>

APPENDIX B: ADDITIONAL INFORMATION

Josh Smith - Industry Advisor

Timothy Comer - Assisted in determining logic needed for scheduling algorithm

APPENDIX C: THREAT MODEL

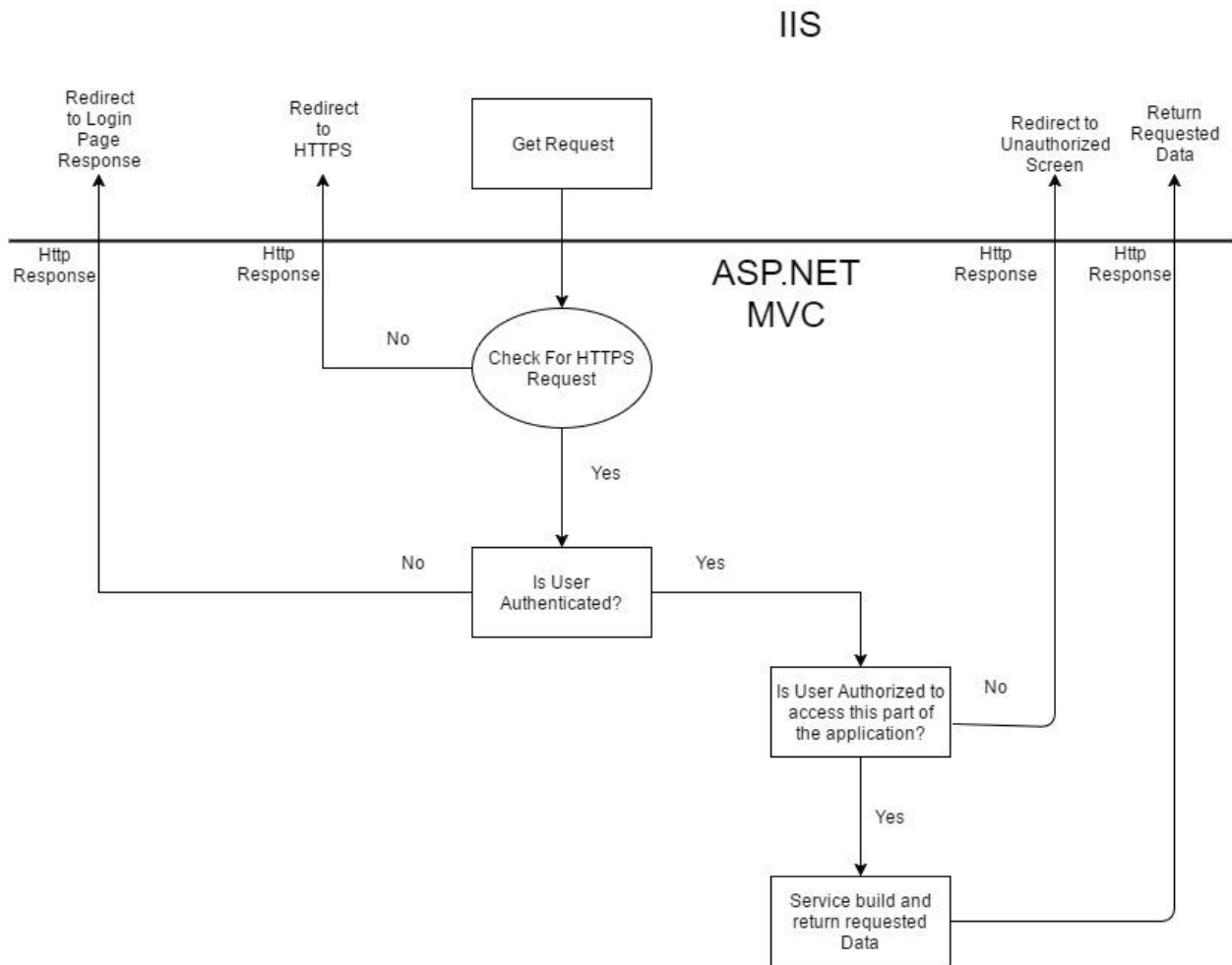
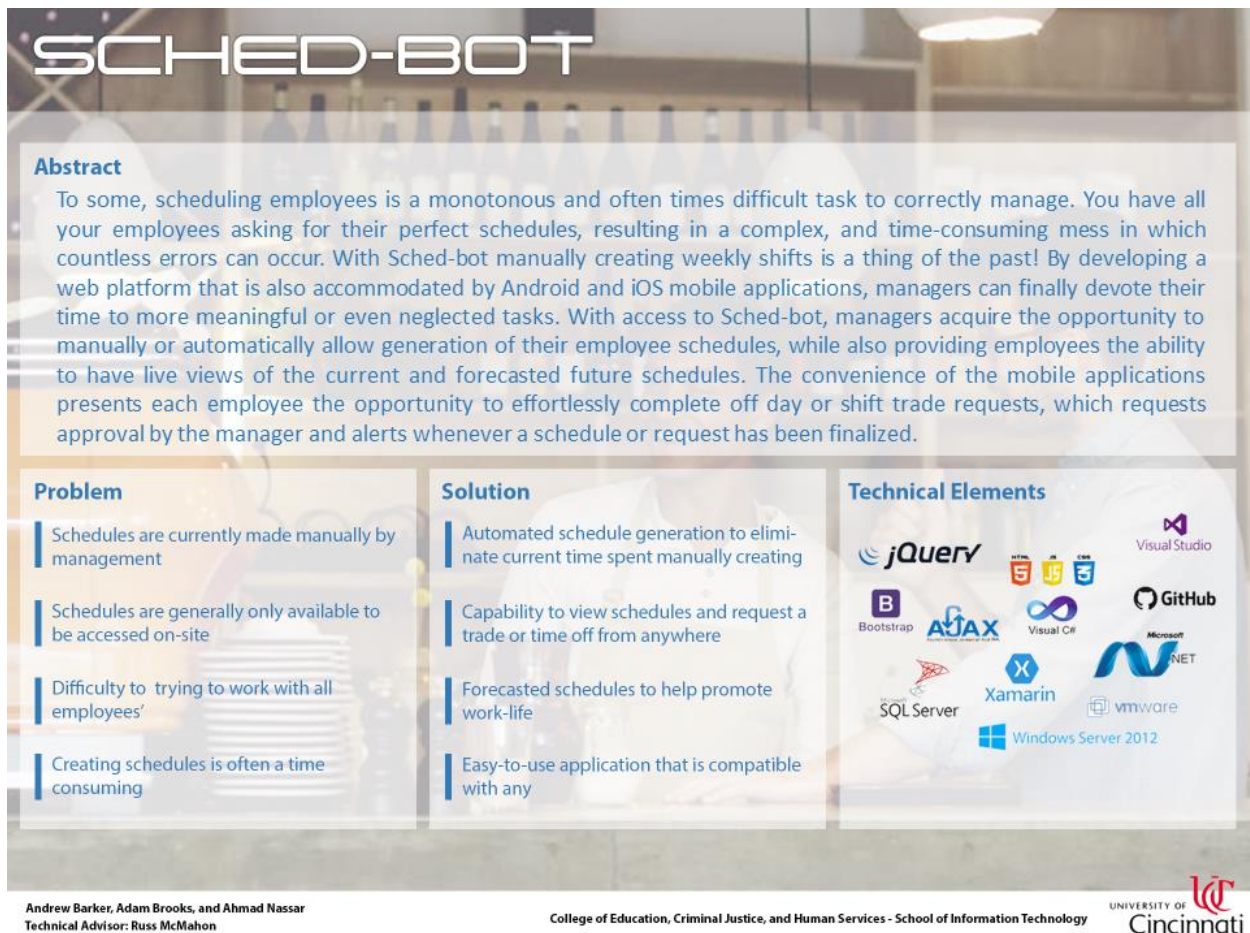


Figure 5: Threat Model Diagram

APPENDIX D: TECH EXPO 2017 POSTER



The poster features a background image of a bar with bottles and a lamp. The title 'SCHED-BOT' is prominently displayed at the top in a stylized, white, outlined font. Below the title, the poster is divided into several sections: an abstract, a problem-solution comparison, and a technical elements grid. The abstract describes the manual scheduling process and the benefits of the automated system. The problem-solution section lists four key issues and their corresponding automated solutions. The technical elements section displays logos for various technologies used in the project, including jQuery, Bootstrap, AJAX, Visual C#, Visual Studio, GitHub, Microsoft .NET, Xamarin, SQL Server, VMware, and Windows Server 2012. At the bottom, the authors' names and affiliations are listed.

SCHED-BOT

Abstract

To some, scheduling employees is a monotonous and often times difficult task to correctly manage. You have all your employees asking for their perfect schedules, resulting in a complex, and time-consuming mess in which countless errors can occur. With Sched-bot manually creating weekly shifts is a thing of the past! By developing a web platform that is also accommodated by Android and iOS mobile applications, managers can finally devote their time to more meaningful or even neglected tasks. With access to Sched-bot, managers acquire the opportunity to manually or automatically allow generation of their employee schedules, while also providing employees the ability to have live views of the current and forecasted future schedules. The convenience of the mobile applications presents each employee the opportunity to effortlessly complete off day or shift trade requests, which requests approval by the manager and alerts whenever a schedule or request has been finalized.

Problem	Solution
Schedules are currently made manually by management	Automated schedule generation to eliminate current time spent manually creating
Schedules are generally only available to be accessed on-site	Capability to view schedules and request a trade or time off from anywhere
Difficulty to trying to work with all employees'	Forecasted schedules to help promote work-life
Creating schedules is often a time consuming	Easy-to-use application that is compatible with any

Technical Elements

- jQuery
- Visual Studio
- Bootstrap
- AJAX
- Visual C#
- GitHub
- Microsoft .NET
- Xamarin
- SQL Server
- vmware
- Windows Server 2012

Andrew Barker, Adam Brooks, and Ahmad Nassar
Technical Advisor: Russ McMahon

College of Education, Criminal Justice, and Human Services - School of Information Technology

UNIVERSITY OF CINCINNATI

Figure 6: Tech Expo Poster

APPENDIX E: APPLICATION WIREFRAMES

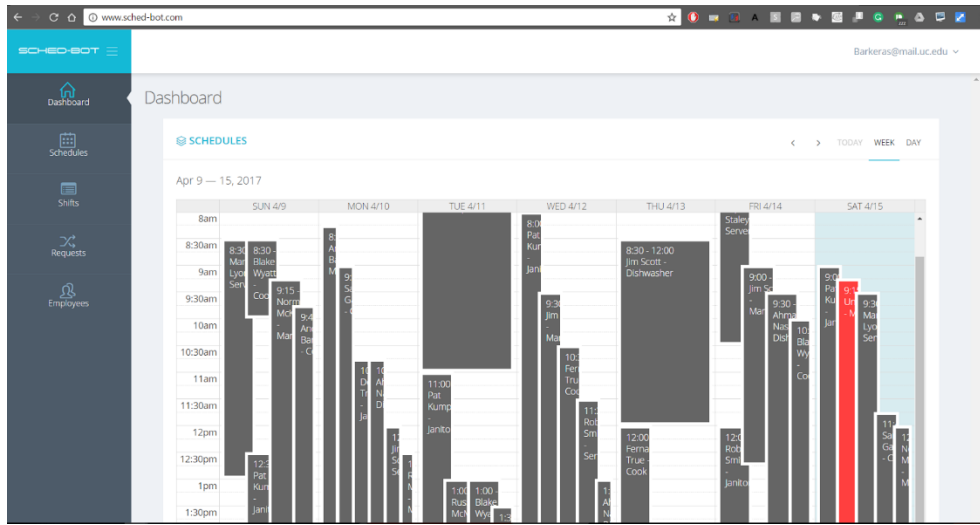


Figure 7: Sched-Bot dashboard (calendar view) on a desktop

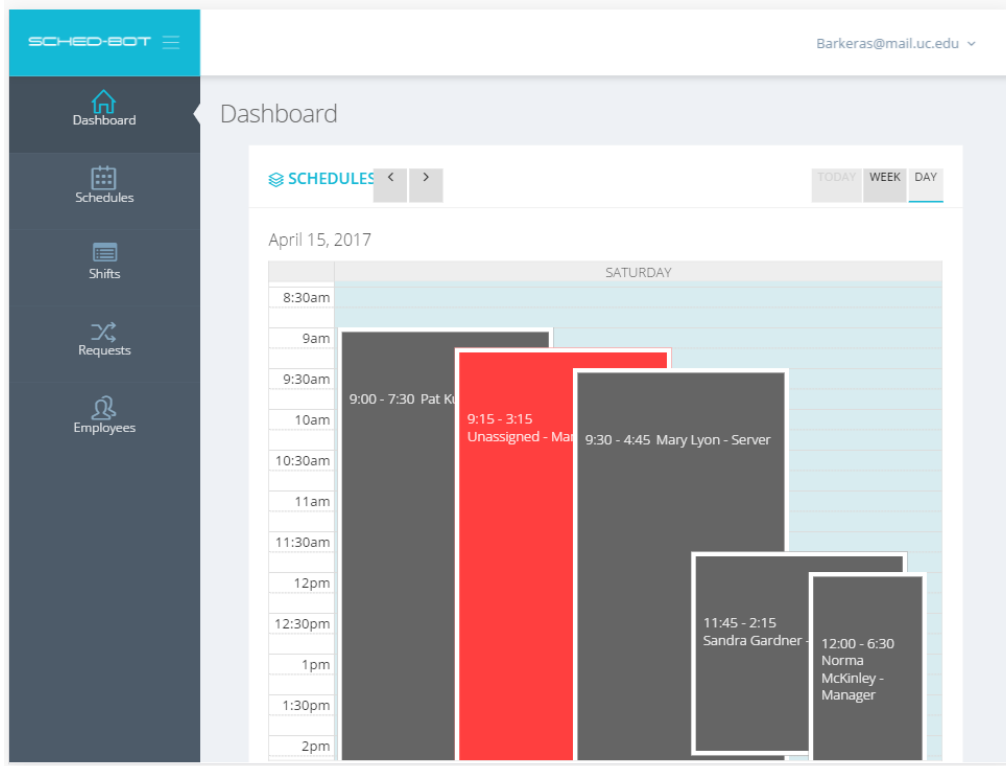


Figure 8: Sched-Bot dashboard (calendar view) on a mobile device

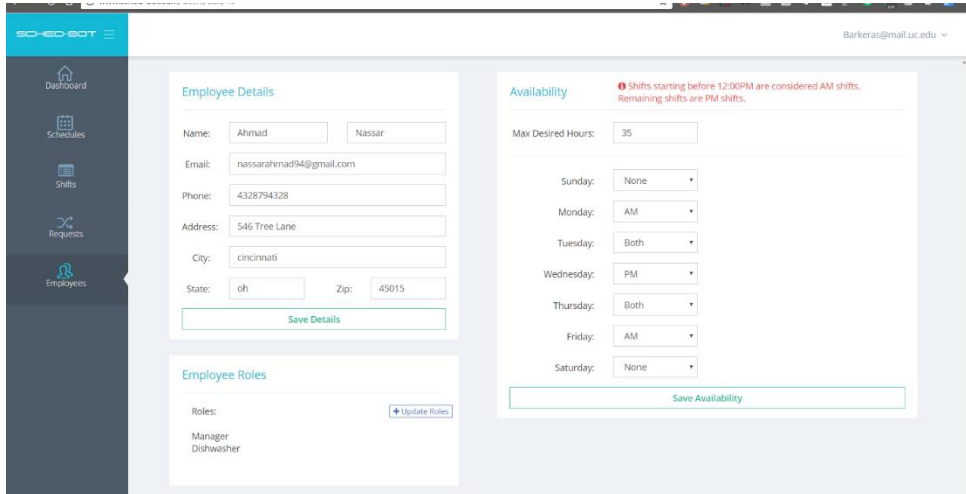


Figure 9: Employee details and availability page on a desktop

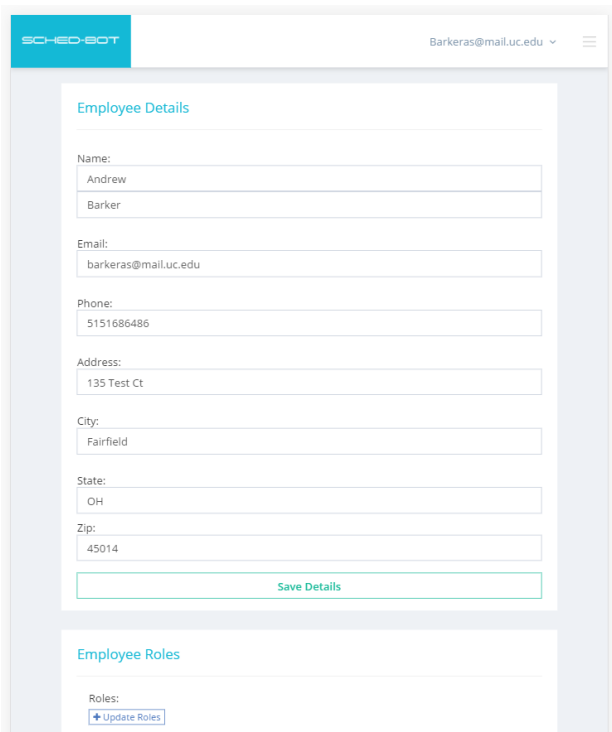


Figure 10: Employee details and availability page on a mobile device

The screenshot shows the SCHED-BOT desktop interface. On the left is a dark sidebar with navigation icons for Dashboard, Schedules, Shifts, Requests, and Employees. The main content area is titled 'Schedule' and features a search bar with 'Dates: 04/09/2017 - 04/15/2017' and 'Employee: All Users'. Below this is a table with columns for Employee, Day, Time Start, Time End, Shift Role, and Actions. The table lists 18 shifts for various employees from Sunday to Tuesday.

Employee	Day	Time Start	Time End	Shift Role	Actions
Blake Wyatt	Sunday	8:30 AM	10:00 AM	Cook	[edit] [delete]
Mary Lyon	Sunday	8:30 AM	1:00 PM	Server	[edit] [delete]
Andrew Barker	Sunday	9:45 AM	5:00 PM	Cook	[edit] [delete]
Pat Kumpf	Sunday	12:30 PM	9:00 PM	Janitor	[edit] [delete]
Norma McKinley	Sunday	9:15 AM	3:00 PM	Manager	[edit] [delete]
Donald Trump	Monday	10:45 AM	2:30 PM	Janitor	[edit] [delete]
Andrew Barker	Monday	8:15 AM	2:45 PM	Manager	[edit] [delete]
Russ McMahon	Monday	12:30 PM	4:45 PM	Manager	[edit] [delete]
Sandra Gardner	Monday	9:00 AM	5:30 PM	Cook	[edit] [delete]
Jim Scott	Monday	12:00 PM	2:30 PM	Server	[edit] [delete]
Ahmad Nassar	Monday	10:45 AM	2:15 PM	Dishwasher	[edit] [delete]
Pat Kumpf	Tuesday	11:00 AM	6:45 PM	Janitor	[edit] [delete]
Russ McMahon	Tuesday	1:00 PM	7:00 PM	Manager	[edit] [delete]
Blake Wyatt	Tuesday	1:00 PM	5:00 PM	Cook	[edit] [delete]

Figure 11: Table view of a schedule on a desktop

The screenshot shows the SCHED-BOT mobile interface. The top bar includes the SCHED-BOT logo and the user email 'Barkeras@mail.uc.edu'. The 'Schedule' section has a search bar with 'Dates: 04/09/2017 - 04/15/2017' and 'Employee: All Users', along with a 'View Schedule' button. The table below is a mobile-optimized version of the schedule data, with columns for Employee, Day, Time Start, Time End, Shift Role, and Actions.

Employee	Day	Time Start	Time End	Shift Role	Actions
Blake Wyatt	Sunday	8:30 AM	10:00 AM	Cook	[edit] [delete]
Mary Lyon	Sunday	8:30 AM	1:00 PM	Server	[edit] [delete]
Andrew Barker	Sunday	9:45 AM	5:00 PM	Cook	[edit] [delete]
Pat Kumpf	Sunday	12:30 PM	9:00 PM	Janitor	[edit] [delete]
Norma McKinley	Sunday	9:15 AM	3:00 PM	Manager	[edit] [delete]
Donald Trump	Monday	10:45 AM	2:30 PM	Janitor	[edit] [delete]
Andrew Barker	Monday	8:15 AM	2:45 PM	Manager	[edit] [delete]
Russ McMahon	Monday	12:30 PM	4:45 PM	Manager	[edit] [delete]
Sandra Gardner	Monday	9:00 AM	5:30 PM	Cook	[edit] [delete]
Jim Scott	Monday	12:00 PM	2:30 PM	Server	[edit] [delete]
Ahmad Nassar	Monday	10:45 AM	2:15 PM	Dishwasher	[edit] [delete]
Pat Kumpf	Tuesday	11:00 AM	6:45 PM	Janitor	[edit] [delete]
Russ McMahon	Tuesday	1:00 PM	7:00 PM	Manager	[edit] [delete]
Blake Wyatt	Tuesday	1:00 PM	5:00 PM	Cook	[edit] [delete]

Figure 12: Table view of a schedule on a mobile device

APPENDIX F: OPEN SOURCE SOFTWARE AND USE COMPONENTS

1. **UC Sandbox**
2. **Windows Server 2012**
3. **Microsoft SQL Server 2012**
4. [Jquery](#) - Javascript library used to manipulate the HTML DOM and make AJAX calls
5. [Metronic](#) - CSS and basic html layout used to jumpstart our front end development
6. [SweetAlerts2](#) - Javascript library that was used in conjunction with our AJAX calls to show success or error alerts to users
7. [FullCalendar](#) - Javascript library used to build the dashboard calendar view of shifts
8. [Bootstrap](#) - HTML, CSS, and Javascript framework used to develop our application to be responsive to work on any device, desktop or mobile.
9. [PickADate](#) - Javascript library linked to a input element to allow users to select a time of day
10. [Vertabelo](#) - Online site used to create our initial database diagrams