MEC

McMaster Engineering Competition 2019

Programming Challenge

November 2nd 2019 – November 3rd 2019

Table of Contents

Schedule

Day 1

Time	Activity	Location
10:45AM	Registration	BSB Lobby
11:30AM	Opening Ceremony	TBA
12:00PM	Competition Debrief	BSB 120
12:30PM	Competitions Begin	BSB 120
1:30PM	Lunch/Dinner	TBA
6:30PM	**DELIVERABLES DUE**	BSB 120
7:00PM	Competitions End	BSB 120

Day 2

Time	Activity	Location
9:00AM	Breakfast / Sign-In	BSB Lobby
9:30AM	Finalists Presentations	BSB 105
11:30AM	Presentations End	BSB 105
12:30PM	Lunch	Celebration Hall
1:30PM	Networking	Celebration Hall
2:15PM	Closing Ceremony	Celebration Hall

Problem Statement

McMaster Hospitality Services is one of the biggest operations at McMaster, yet it lacks in technological advancements and renewed innovation

Currently, students and campus staff can encounter long wait times, irregular costs, mediocre food quality and unsustainable food maintenance. It has now become a hassle to purchase food on campus, but it does not have to be this way.

In your team of 4, you are to come up with a digital program that is a solution to one of the problems the McMaster Hospitality Services face daily. Divided into four main categories, here are some current outlined issues:

- Checking Out
    - What can be done to combat long wait times?
    - What can be done to attract more consumers?
    - How can the check-out process be optimized?
- Keeping Track of Inventory
    - What can be done to ensure the freshest food is always given to the consumer?
    - What can be done to keep track of inventory at multiple stations and/or locations?
    - How will food be transported throughout the store?
- Sustainability
    - What can be done to reduce food waste?
    - What can be done to promote sustainable food practices to workers and customers?
- Accessibility
    - How can the services be more accessible to everyone?
    - If a change is made, will it be more accessible or less accessible?

Since there are multiple solutions for these problems, it is your task to come up and deploy a feasible solution that can help vamp up McMaster Hospitality Services. It is also possible that your solution falls under two or more of the categories above.

The implementation and use of order sales and statistics is also encouraged and may help you develop your solution.

For simplicity you can assume your solution will be used to improve the La Piazza, the central McMaster Hospitality Service in the McMaster University Student Centre.

Deliverables

All deliverables must be emailed to **vanvlm1@mcmaster.ca** and provided via USB by the end of the competition. More details on the submission process can be found under "Submission"

The solution must contain the following deliverables

Program

Your team must design and develop a program that is a solution to the given problem statement. The program should at the very least contain a graphical user interface with an appropriate input and output system.

Please submit the program via a GitHub repository link (recommended) or Zip file.

Demo

The program must be demoed at the end of the competition period. The demo can take place on your personal device or on one of the judge's devices.

The judges are free to ask for and do specific commands to your program during this demo period to ensure all test cases are covered during your team's testing.

The demo is also an appropriate time to communicate with the judges' aspects of the program's design you think they should know. For example, telling the judges how certain aspects of your program could be changed or improved if given more time. You should not assume this question will be asked during testing and it is up to your team to communicate your solution to the best of your abilities.

A concrete laid out presentation accompanying the demo is not necessary however it is highly recommended. If you are to qualify onto the next round, a proper laid out presentation can be the difference between first and second place.

Any presentations should be made in Google Slides or Microsoft PowerPoint and submitted with the rest of the deliverables.

Documentation

Your program must be submitted with software documentation. This can be done multiple ways and does not have to follow industry protocol. It is up to your discretion how you go about documenting the process it took to build your solution but at the very least the code should be properly commented with meaningful and insightful comments.

Your documentation must be in PDF, Microsoft Word or Google Document format.

Rules and Guidelines

Please adhere to the following rules. Violation of any of the rules can lead to penalties, disqualifications or other deductions

1. You can access the internet during the competition period as well as online libraries, packages, APIs etc. Upon using any code that is not yours or was written by you outside of the competition period, you must reference it in your documentation and mention it during the demoing period. Plagiarizing entire codes will result in disqualification.
2. All deliverables must be made during the 6-hour competition period. Only material submitted during this period can be used during your presentation
3. You are not allowed to accept aid from anyone outside of your team. This includes having someone come in to help, substituting team members or asking for help on online forums.
4. You are not allowed to use any personal devices for online communication methods during the competition period. This includes going on and posting on social media during the competition period.
5. All questions regarding the problem are welcomed before the competition starts during the welcome and briefing session. During the competition, the coordinator and volunteers will be limited in terms of what questions they can answer.
6. Not submitting your solution and its associated files will result in penalty. Please consider submitting all deliverables 15 minutes prior to competition end.

Permitted Tools

Physical Tools

During the competition period you will be provided with a designated workspace in which you may only use

Computers (max one per person)

Peripherals (monitors, mice, headphones, keyboards etc.)

Reference Material (textbooks, course notes)

Software Tools

It is expected that teams participating in this competition have adequate knowledge in choosing what tools they wish to use to execute their solution. There are no restrictions in terms of what software tools are permitted however we do recommend using the following languages, IDEs, text editors and databases:

Languages:

- HTML/CSS/JavaScript
- jQuery
- Java
- C/C++
- Objective C
- PHP
- Ruby
- .NET (C#, VB, VBscript)
- Python
- Perl
- ActionScript
- Coffee script
- Scala
- Swift
- SQL
- MongoDB

IDE

- Eclipse
- NetBeans
- Microsoft Visual Studio
- JetBrains (IntelliJ, PyCharm, WebStorm etc.)
- XCode
- Android Studio

Text Editors

Sublime
Atom
Vim
Notepad++
Kwrite
Visual Studio Code
Bracket

Other Tools

- ReactJS
- MeteorJS
- NodeJS
- Bootstrap
- Flask
- MEAN Stack
- Digital Ocean
- Heroku
- Microsoft Azure
- Amazon Web Services
- Ionic
- Codava
- Ruby on Rails

Resources

For more information on McMaster Hospitality Services and to see what their current system is like please visit: https://hospitality.mcmaster.ca/

And more specifically view the page on La Piazza, the place you are designing a system for: https://hospitality.mcmaster.ca/locations/locations/la-piazza/#tab-content-ov

Rubric

Functionality

Solution is feasibly and could be used to improved McMaster Hospitality Services /20

- Solves one of the prompts outlined in the Problem Statement (rate out of 10)
- Is a solution that is better than the services currently McMaster Hospitality Services have in place (rate out of 10)

Solution can be easily run and implemented by the McMaster Hospitality Services /20

- Can be opened and run by anyone (rate of 10)
- Is robust and can be used large scale (rate of 5)
- Is accessibility friendly (rate of 5)

Solution is innovative and unique /20

- Solves the prompt the best compared to other teams (will be determined after all judging has commenced)
- Combines and solves more than one prompt with one solution (rate of 10)

Rare input cases, at the discretion of the judges, are covered /10

Code Structure and Documentation

Code is clean and well-structured /15

- Code appears easy to maintain and expand upon if needed (rate of 5)
- Code is not redundant or inefficiently written (rate of 5)
- Any given team member can explain how each aspect of the code works and why is was written that way (rate of 5)

Documentation covered major aspects of the code /10

- Documentation is clear and correctly covers each major aspect of code (out of 5)
- Documentation is efficient in explaining the major aspects of the code and can be read and followed without guidance from one of the developers (out of 5)

Code is commented /5

- Comments are meaningful and insightful (out of 2.5)
- Comments are not overbearing and redundant (out of 2.5)

Penalties

Code does not compile and/or run, deduct 20 points

Detrimental bug, deduct 10 points pe

Minor bug, deduct 3 points per

TOTAL SCORE /100

Submission

Please submit all files by 6:30pm by email to **vanvlm1@mcmaster.ca**

AND

Putting the files on the USB drive.

Please consider submitting your code via a GitHub repository link.