INTEGRATED SOLUTION DOCUMENT

TABLE OF CONTENTS

Objective 2
Solution Summary 2
Conceptual solution overview 2
Design Assumptions 3
Assumptions, Dependencies and Constraints 3
Assumptions 3
Dependencies........ 3
Constraints........ 4
UML Views........ 4
High Level Class Diagram........ 4
High Level Sequence Diagram........**Error! Bookmark not defined.**
Infrastructure........ 5
Implementation Support........ 5
References........ 7

OBJECTIVE

The purpose of this document is to specify the solution architecture, design and implementation for Conference Management System, an assignment from ThoughtWorks. The scope of this document is to define the architecture, design and implementation details for the requirements to plan a big conference, where the stakeholders are finding trouble in fitting the slots into the time constraints of the day.

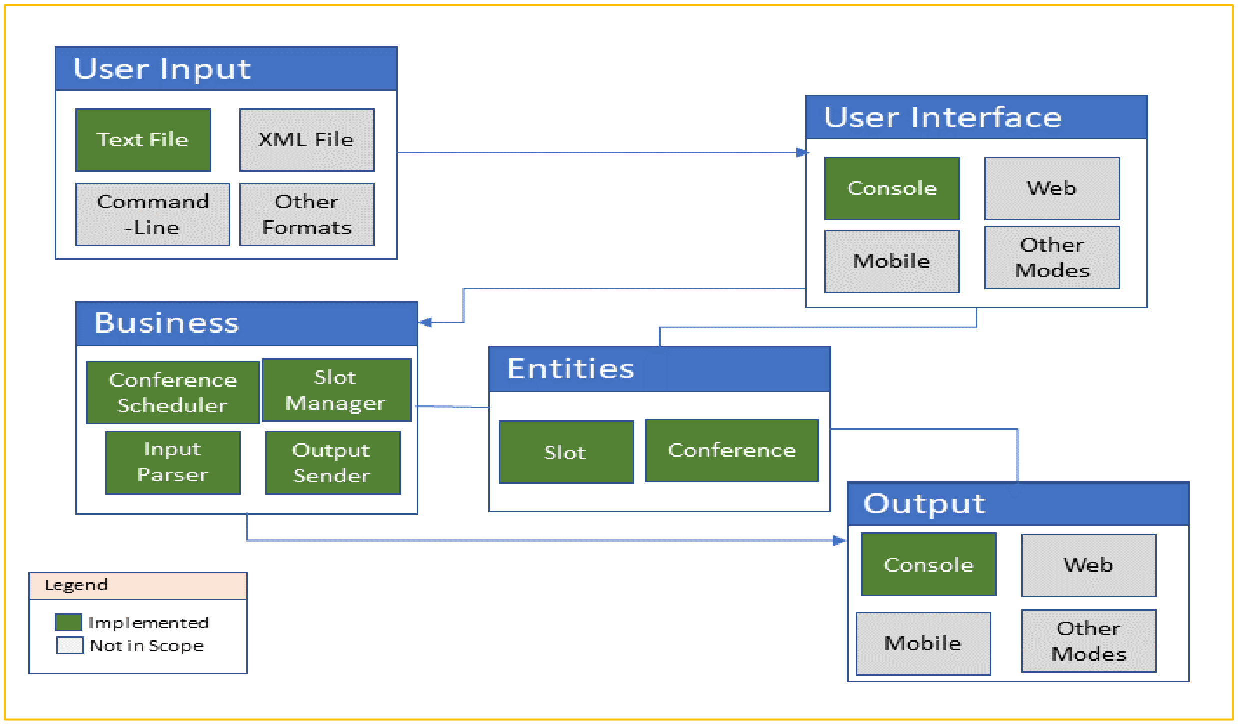
This document will provide a solution overview and required infrastructure details and required design and implementation details of the requirements.

SOLUTION SUMMARY

The solution for Conference Management System will use Microsoft.NET platform as a technology solution. The current solution will take input from a flat file, where the file location can be provided through command line arguments and place the file in the project folder having name as "input.txt".

The solution will also include a logger component for logging purpose across different stages of the solution. Every status will be logged into a file "LogFile.txt" that is placed under root folder of the project. The solution can be extended to capture input from multiple devices and display/store output on multiple modes.

CONCEPTUAL SOLUTION OVERVIEW



The solution for Conference Management System covers the solution for conferences fitting the slots based on duration into the timing constraint of different sessions of the day. The solution takes input through a flat file, where the path can be specified either through command-line arguments or update the file "input.txt" located under project folder.

The current input will be through a console window, and the solution is provided to extend the User Interface to multiple modes. The business process majorly distributed through Conference Scheduler and Slot manager functions through which the raw input from file can be parsed and tracked and fitted based on the duration into the time constraint of the day.

The solution details the output to be displayed on a console window but featured to extend this onto multiple interfaces.

The solution is also provided a logging mechanism, a cross cutting concern that helps to track the control flow to identify any errors and necessary information details. The rest of cross cutting concerns can be extended in the solution.

DESIGN ASSUMPTIONS

ID	Problem Statement	Design Decision
1	➢ The conference has multiple tracks each of which has a morning and afternoon session. ➢ Morning sessions begin at 9am and must finish by 12 noon, for lunch. ➢ Afternoon sessions begin at 1pm and must finish in time for the networking event. ➢ The networking event can start no earlier than 4:00 and no later than 5:00.	In case if any session overlapping across regular events like Lunch, Networking event, the slot will be moved after regular event. It will be after lunch in case of overlapping Lunch event, and it will be next day in case overlapping Networking Event.
2	You should provide sufficient evidence that your solution is complete by indicating that it works correctly against the supplied test data	Logger.txt file will provide necessary evidence, contains details of every method of execution

ASSUMPTIONS, DEPENDENCIES AND CONSTRAINTS

ASSUMPTIONS

1. The solution can be developed using Microsoft Visual Studio 2017 Developer Community and .NET Core, C#
2. The data can be passed through flat file where the path can be input through command-line arguments or update the content of "input.txt" under project folder
3. A logger file is provided that will provide evidence that the solution is working as expected
4. Morning Sessions always start at 9 am and lunch will be starting exactly at 12 pm.

DEPENDENCIES

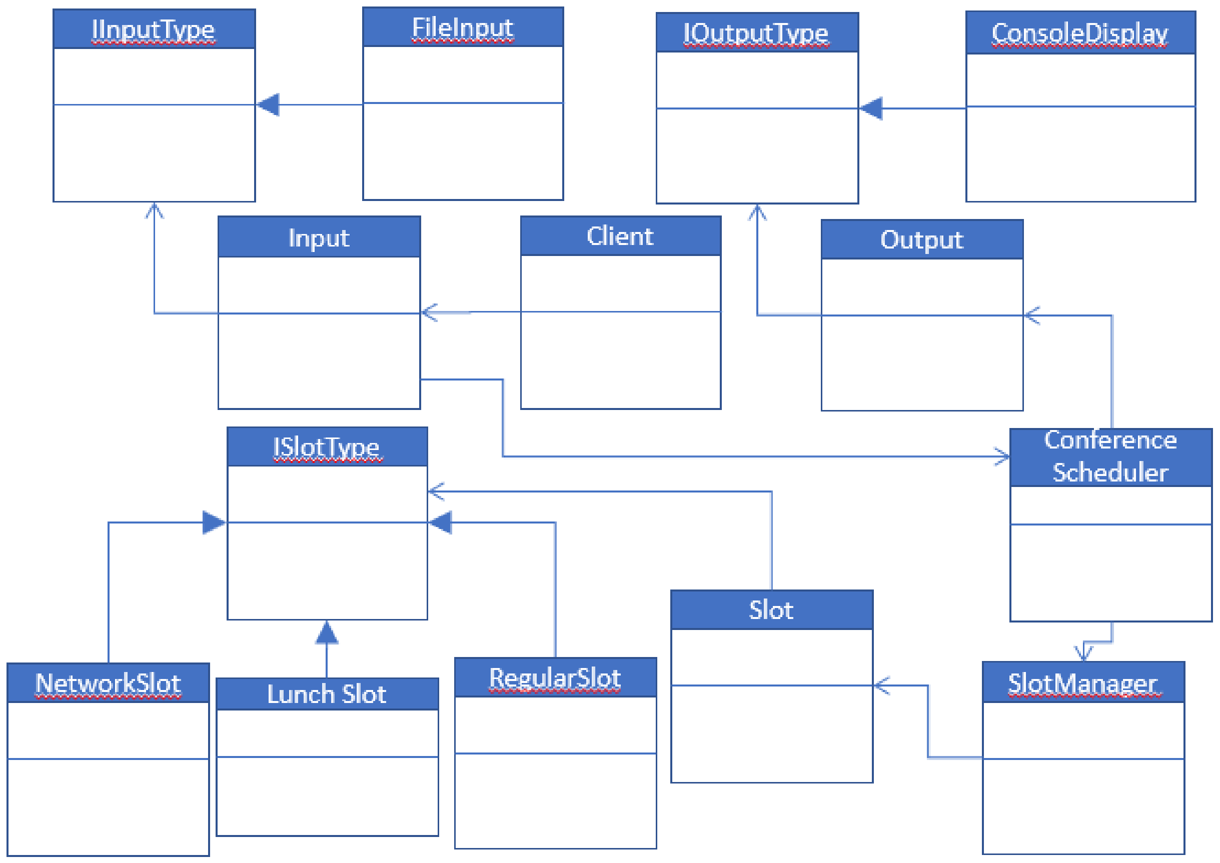
1. The solution might need Visual Studio 2017 and .NET Core 2.0 for evaluation purpose. But the code files can be manually scanned by opening them in a Notepad

CONSTRAINTS

1. The solution will accept the input only through flat file. The absolute file path can be provided as command-line argument, or content can be placed in “Input.txt” placed under project folder.

UML VIEWS

HIGH LEVEL CLASS DIAGRAM



INFRASTRUCTURE

Please find below the details of development environment used to develop the solution

Software	
Operating System	Windows 10 64 bit
IDE	Microsoft Visual Studio 2017 Developer Community
Technology	..NET Core, Console App
Language	C#

Hardware	
Make	Lenovo G510
RAM	8GB
HDD	500GB
Processor	I5 4200 CPU @2.50GHz

This application can be executed on any windows 8 /10 operating system machines where .NET Core is available.

IMPLEMENTATION SUPPORT

The main components of the zip file contain below items

ID	FileName	Description
1	ConferenceManager	This is the Microsoft visual Studio Solution File
2	Input	This folder contains the implementation related to input processing
3	Input/FileInput.cs	This file contains the logic to parse the input file
4	Input/IInputType.cs	This is an interface that should be implemented by different class to process different kind of inputs
5	Input/Input.cs	This file uses IInputType to call the concrete input processing function.
6	Output	This folder contains the implementation related to Output processing

7	Output/ConsoleDisplay.cs	This file contains the logic to display content on console
8	Input/IOutputType.cs	This is an interface that should be implemented by different class to process different kind of Outputs
9	Input/Output.cs	This file uses IOutputType to call the concrete Output processing function.
10	Slot	This folder holds the files used to process slot management
11	Slot/Conference.cs	This is a composition of Slot.cs
12	Slot/ISlotType.cs	This is an interface should be implemented by different slots
13	Slot/LunchSlot	Having logic to apply Lunch Slot
14	Slot/Network Slot	Having logic to apply Network SLot
15	Slot/RegularSlot	Having logic to apply regular slots
16	Slot/Slot.cs	This class contains preliminary properties of Slot and function to initiate regular events
17	Conference.cs	This is a composition of Slot.cs
18	ConferenceManager.csproj	A visual studio project file
19	ConfrenceScheduler.cs	This file will be used to receive the input and call necessary business function, and call output function to display
20	SlotManager.cs	This file contains the main logic of the solution, to slot every track to fit them in the session of the day.
21	LogFile.txt	This is the file created by logger component and will write the details of every event/method triggered in the solution. This file can be used as evidence to showcase the completeness of solution that provides necessary indication against supplied data. The data in this file can be deleted before executing the program. The file will be appended with the data for every execution of the program
22	Input.txt	Contains input files
23	Program.cs	This file is main entry point of the control
24	CrossCutting	This folder contains logging cross cutting concerns implementation
25	CrossCutting/FileLog.cs	This file contains logic to write the logger content into a file, currently hardcoded at project folder
26	CrossCutting/Log.cs	This is an abstract class that defines message types and method types
27	CrossCutting/Logger.cs	A class having static methods used to log the content

Steps to Open, Build and Run

1. Please open the solution in the Visual Studio 2017 environment (as mentioned above in Software table)
2. Please make sure that a Logger.txt exists (with Read/Write Permissions) under project folder of the solution
3. The input can be provided in any of the below steps
    a. Absolute Input File Path through Command-Line Arguments
    b. Place the content in "Input.txt" located under project folder
4. The solution can be build/run successfully.

REFERENCES

ID	Description	
1.	Requirements Document	**Gmail - ThoughtWorks Codi**
2.	Solution File	ConferenceManager.zip