SSN COLLEGE OF ENGINEERING

KALAVAKKAM-603110

Department of computer science and Engineering

END SEM REVIEW

FPSD-PROJECT REPORT

P2-MOVIE SCHEDULING IN A MULTIPLEX MALL

CSE-A BATCH 1

TEAM-**TechUp**

1.Apurva Narayan : 3122 21 5001 011

2.A Bhavana : 3122 21 5001 019

3.G Deepti : 3122 21 5001 023

4.Harini Mohan : 3122 21 5001 029

ABSTRACT

The project is to develop a model that supports a system that generates weekly movie schedules in a multiplex movie theater. A movie schedule specifies, for each day of the week, on which screen(s) different movies will be played, and at which time(slots). The model integrates elements from marketing (the generation of demand figures) with approaches from operations research (the optimization procedure). Therefore, it consists of two parts: (I) conditional forecasts of the number of tickets booked previously and (II) a scheduling procedure that quickly finds a near-optimal schedule (which can be demonstrated to be close to the optimal schedule by taking into consideration various scores as parameters). To generate this schedule, the problem is formulating the “movie scheduling problem” as a generalized set partitioning problem. The latter is solved with an amended schedule optimization procedure in a multiplex by using basic concepts of Linear programming and implementing column generation method.. Various techniques and methods were studied, analyzing the pros and cons of each algorithmic technique. A list of constraints pertaining to the problem were analyzed, discussed, and were designed to produce the most efficient system to solve the problem. The main objective of this software is to reduce the workload on the multiplex management and to maximize the profit.

INTRODUCTION:

The project is to develop a model that supports a system that generates weekly movie schedules in a multiplex movie theater. The problem was analyzed, and the solution was formulated using the concepts of linear programming and column generation algorithm.

Various constraints were specifically taken into account, considering the ultimate aim of the project that is to schedule an efficient schedule for the multiplex by maximizing the profit.

The demand score of a particular movie, the regional language of the multiplex locality and the language of the movie were taken to calculate the popularity score of the movie. The number of tickets booked for a particular movie in the past week are also considered. The number of slots per screen per day will not be exceeding five and thus the number of slots is fixed to five.

EXISTING WORK:

The existing system used complex mathematical logics and algorithms to meet the demand of the multiplexes.

TOOLS USED IN SCHEDULING:

IBM CPLEX OPTIMIZER

CPLEX Optimizer provides flexible, high-performance mathematical programming solvers for linear programming, mixed integer programming, quadratic programming and quadratically constrained programming problems. This helps in making better decisions by finding the best solution for planning, resource allocation problems using CPLEX Optimizer.

Optimizer implementation is especially useful for fine-grained scheduling. Scheduling problems also require the management of minimal or maximal capacity constraints for resources over time and of alternative modes to perform a task.

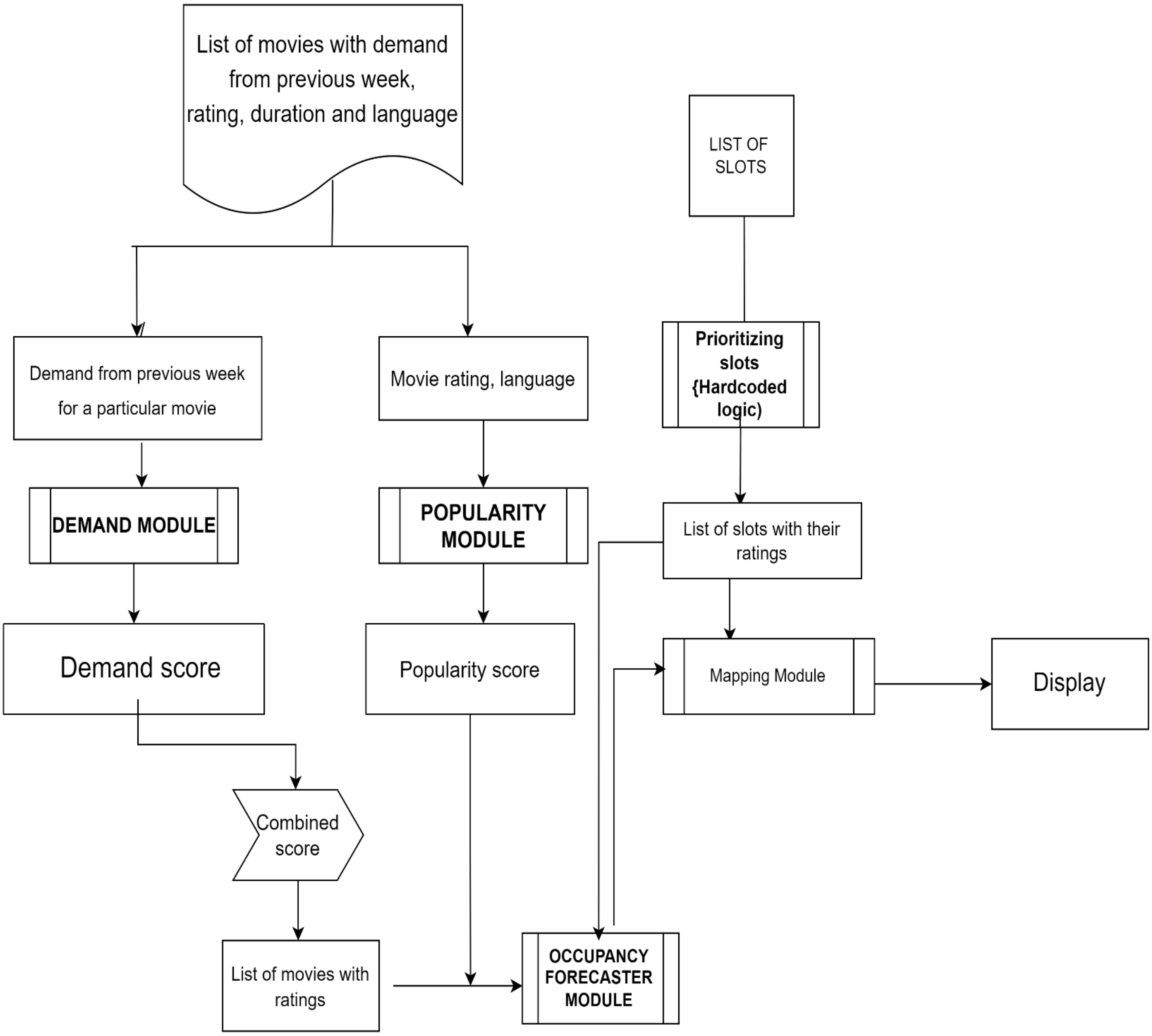
RosettaBridge Theatre Management System

A software that integrates with the point of sale for scheduling, creating Show PlayList from templates to build shows, allowing pre-show integration with your advertising or trailer provider while managing both hardware and automation of multiplexes.

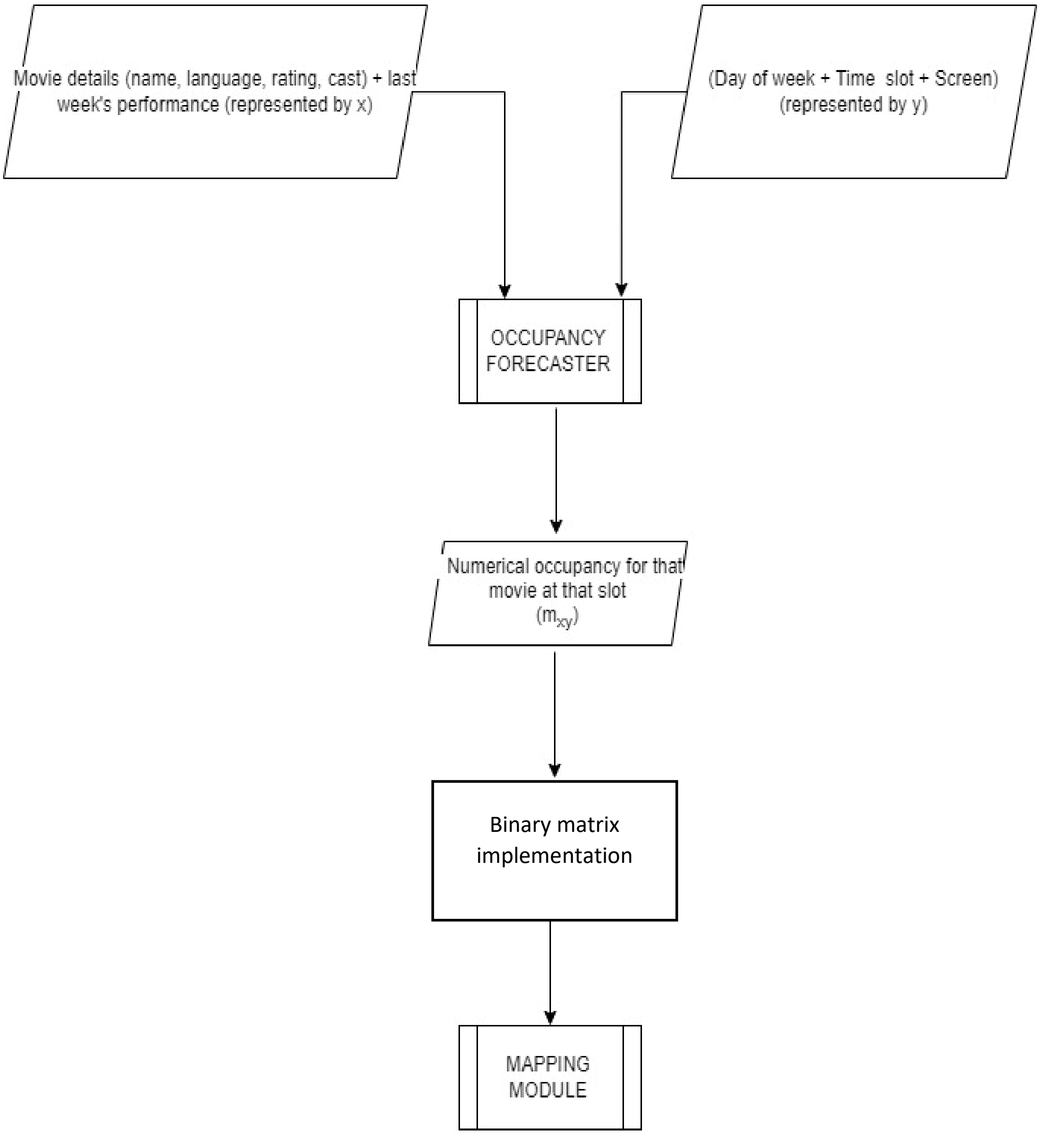
ARCHITECTURAL DESIGN

For the first version of this application, the parameters associated with the movie ratings, cast weightage, language etc are taken as direct input with the specifications of a given multiplex as predefined input. The application is to be developed in C language. The data required for scheduling is stored in C data structures itself in the first version. A mathematical solver such as IBM Cplex is used to compute the column generation algorithm. The input is read using basic interactive interface with the output displayed as a table with the required schedule as well as stored in a file for display.

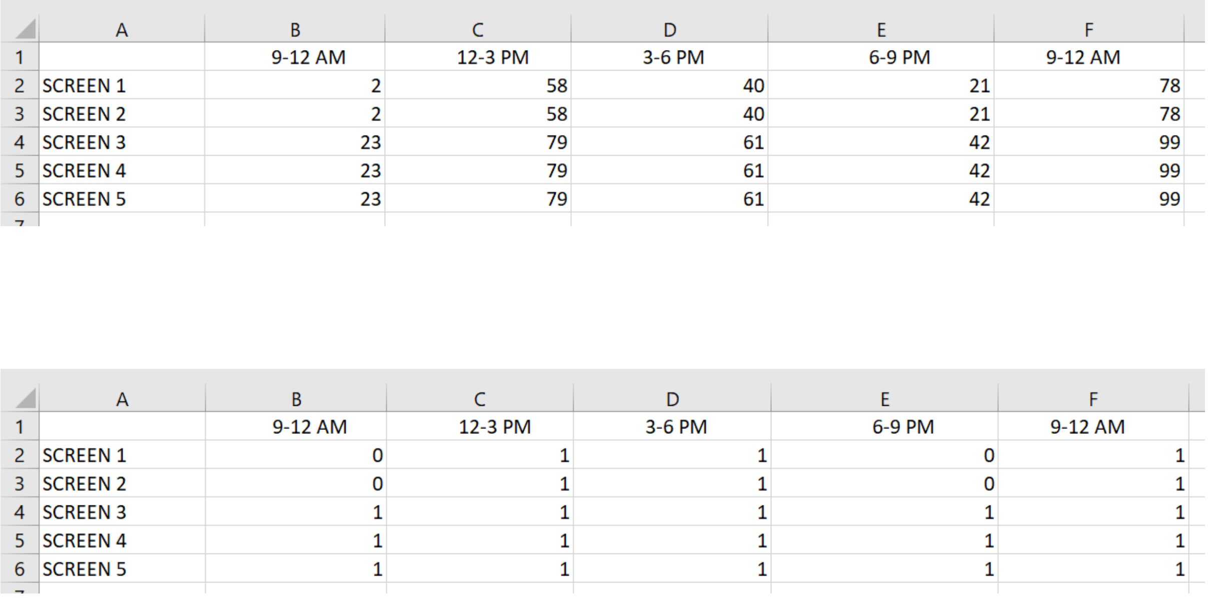
ARCHITECTURE DIAGRAM


List of movies with demand from previous week, rating, duration and language
LIST OF SLOTS
Demand from previous week for a particular movie
Movie rating, language
Prioritizing slots {Hardcoded logic)
DEMAND MODULE
POPULARITY MODULE
List of slots with their ratings
Demand score
Popularity score
Mapping Module
Display
Combined score
List of movies with ratings
OCCUPANCY FORECASTER MODULE

OCCUPANCY FORECASTER MODULE


Movie details (name, language, rating, cast) + last week's performance (represented by x)
(Day of week + Time slot + Screen) (represented by y)
OCCUPANCY FORECASTER
Numerical occupancy for that movie at that slot (m_{xy})
Binary matrix implementation
MAPPING MODULE

LOGIC BEHIND OPTIMIZATION



	A	B	C	D	E	F
1		9-12 AM	12-3 PM	3-6 PM	6-9 PM	9-12 AM
2	SCREEN 1	2	58	40	21	78
3	SCREEN 2	2	58	40	21	78
4	SCREEN 3	23	79	61	42	99
5	SCREEN 4	23	79	61	42	99
6	SCREEN 5	23	79	61	42	99

	A	B	C	D	E	F
1		9-12 AM	12-3 PM	3-6 PM	6-9 PM	9-12 AM
2	SCREEN 1	0	1	1	0	1
3	SCREEN 2	0	1	1	0	1
4	SCREEN 3	1	1	1	1	1
5	SCREEN 4	1	1	1	1	1
6	SCREEN 5	1	1	1	1	1

Fig.1: Occupancy Matrix

The movies are sorted based on the demand score, popularity score, rating, language as well as slot and day of the week weightage. Screens are allotted per movie based on the combined demand and popularity score. (Fig.1)

Calculations are made to generate an occupancy matrix with the percentage of audience predicted to fill the capacity in that particular slot in a given screen and a binary matrix is generated to indicate whether the movie is being assigned to that slot or not.

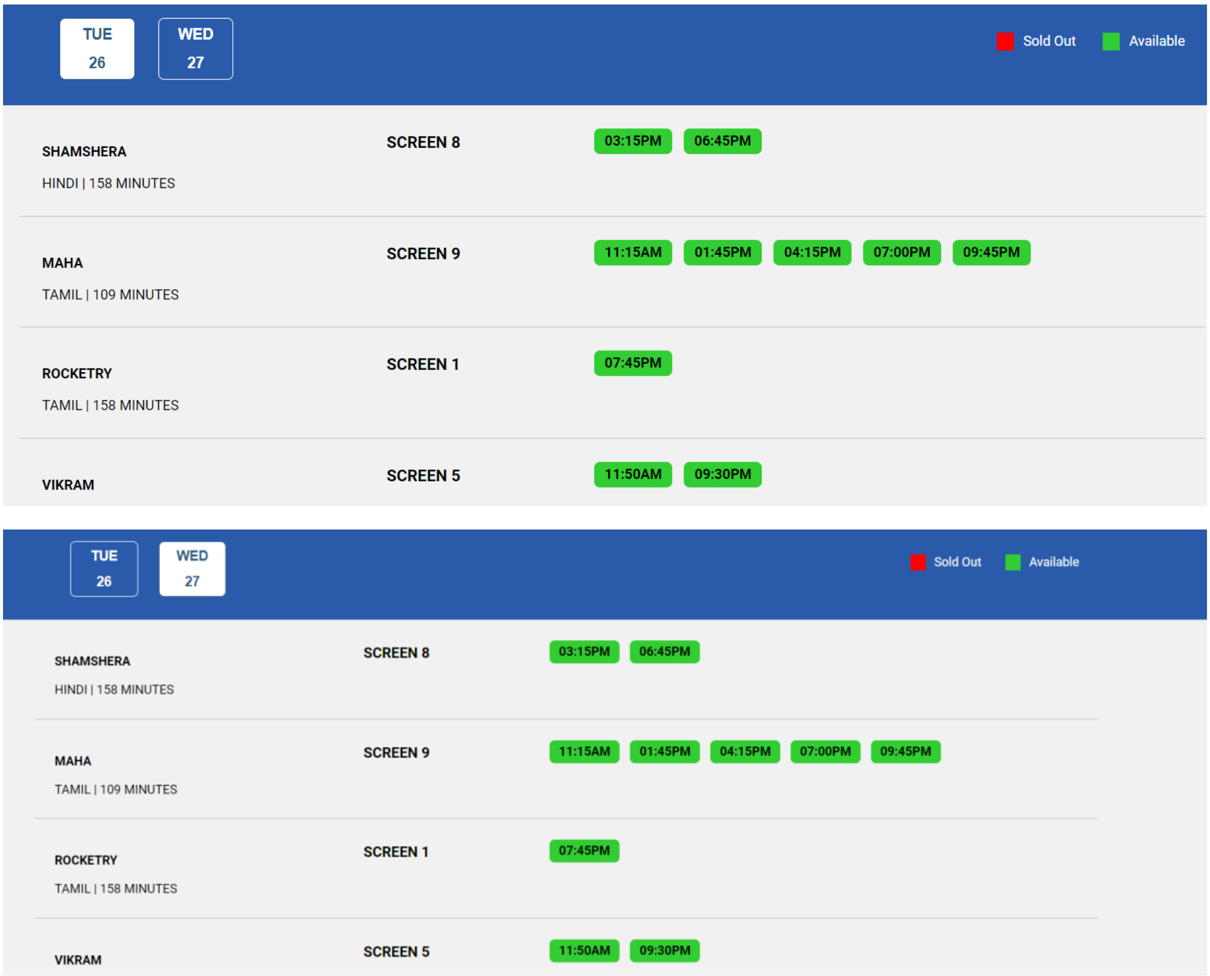
The schedule for a week is generated and crowd management is incorporated by adding appropriate time gaps for each consecutive slot and screen.

DATA ANALYSIS AND VERIFICATION

The Movie Multiplex Scheduling problem has been solved using the logic of the column generation technique simplified into simple linear programming equations for optimum time table generation.

Real Time Data

Variation in movie-scheduling in Mayajaal on weekdays



TUE 26 | WED 27 | 🟥 Sold Out | 🟩 Available

SHAMSHERA HINDI | 158 MINUTES	SCREEN 8	03:15PM 06:45PM
MAHA TAMIL | 109 MINUTES	SCREEN 9	11:15AM 01:45PM 04:15PM 07:00PM 09:45PM
ROCKETRY TAMIL | 158 MINUTES	SCREEN 1	07:45PM
VIKRAM	SCREEN 5	11:50AM 09:30PM

TUE 26 | WED 27 | 🟥 Sold Out | 🟩 Available

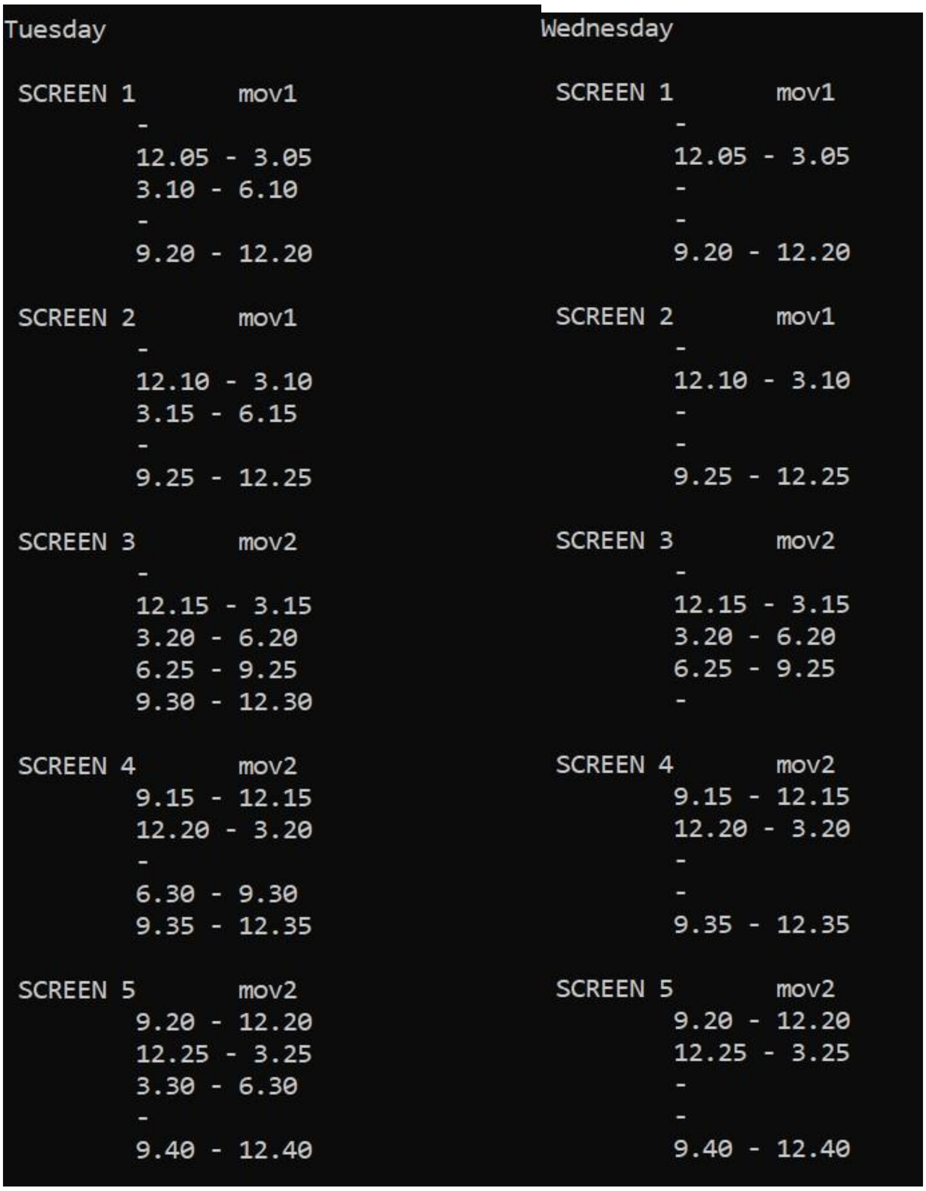
SHAMSHERA HINDI | 158 MINUTES	SCREEN 8	03:15PM 06:45PM
MAHA TAMIL | 109 MINUTES	SCREEN 9	11:15AM 01:45PM 04:15PM 07:00PM 09:45PM
ROCKETRY TAMIL | 158 MINUTES	SCREEN 1	07:45PM
VIKRAM	SCREEN 5	11:50AM 09:30PM

Fig.2: Real Time Data

Observation from above data: (Dated: 26/07/2022)

We observe that there is little variation in the schedule on days Tuesday and Wednesday. All the movies run on the same screen and the same slots.(Fig.2)

TechUp Schedule Solution: (Dated:26/07/2022)

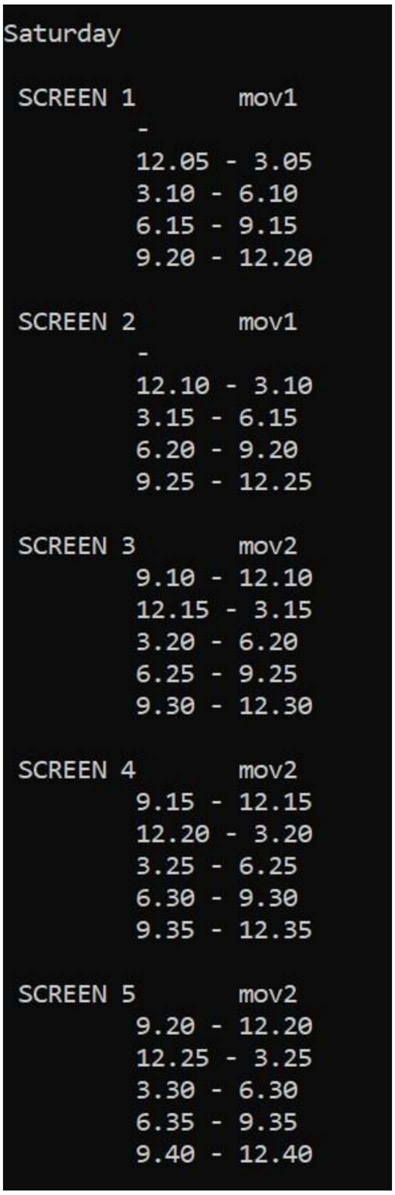


```
Tuesday

 SCREEN 1       mov1
        -
        12.05 - 3.05
        3.10 - 6.10
        -
        9.20 - 12.20

 SCREEN 2       mov1
        -
        12.10 - 3.10
        3.15 - 6.15
        -
        9.25 - 12.25

 SCREEN 3       mov2
        -
        12.15 - 3.15
        3.20 - 6.20
        6.25 - 9.25
        9.30 - 12.30

 SCREEN 4       mov2
        9.15 - 12.15
        12.20 - 3.20
        -
        6.30 - 9.30
        9.35 - 12.35

 SCREEN 5       mov2
        9.20 - 12.20
        12.25 - 3.25
        3.30 - 6.30
        -
        9.40 - 12.40
```

```
Wednesday

 SCREEN 1       mov1
        -
        12.05 - 3.05
        -
        -
        9.20 - 12.20

 SCREEN 2       mov1
        -
        12.10 - 3.10
        -
        -
        9.25 - 12.25

 SCREEN 3       mov2
        -
        12.15 - 3.15
        3.20 - 6.20
        6.25 - 9.25
        -

 SCREEN 4       mov2
        9.15 - 12.15
        12.20 - 3.20
        -
        -
        9.35 - 12.35

 SCREEN 5       mov2
        9.20 - 12.20
        12.25 - 3.25
        -
        -
        9.40 - 12.40
```

Fig.3: Intermediate Schedule on 26/07/2022

The above observations(Fig.3,4) have been made for days Tuesday and Wednesday, scheduling 2 movies within a range of 5 screens and a maximum of 5 slots. Here as well, it is noticed that there is little variation in data compared to the schedule of Saturday, where almost all showtimes are scheduled due to population preference as shown below.



```
Saturday

 SCREEN 1       mov1
        -
        12.05 - 3.05
        3.10 - 6.10
        6.15 - 9.15
        9.20 - 12.20

 SCREEN 2       mov1
        -
        12.10 - 3.10
        3.15 - 6.15
        6.20 - 9.20
        9.25 - 12.25

 SCREEN 3       mov2
        9.10 - 12.10
        12.15 - 3.15
        3.20 - 6.20
        6.25 - 9.25
        9.30 - 12.30

 SCREEN 4       mov2
        9.15 - 12.15
        12.20 - 3.20
        3.25 - 6.25
        6.30 - 9.30
        9.35 - 12.35

 SCREEN 5       mov2
        9.20 - 12.20
        12.25 - 3.25
        3.30 - 6.30
        6.35 - 9.35
        9.40 - 12.40
```

Fig.4: Saturday schedule

Verification and Validation

From the above comparisons between the data, it is seen that the given solution on scheduling is efficient and accurate method to develop a time table. As the weeks proceed, we predict a downward slope of the number of tickets booked. The solution aims at optimization of seats occupied hence tickets booked.

MODULE DESIGN

- Release_date,valid_date,date_diff: checks if release date is valid
- Avg_calc,demand_calc : calculates demand score based on initial demand
- Screen_calc: allots screens based on demand score
- Screen_show: show times for each screen allotted
- popularity: calculates popularity score based on movie rating, language
- Slot_generation: generation of slot timings taking into account crowd management
- Occupancy_calc : forecasts occupancy of screening by optimization of all parameters reduced to numerical value-namely popularity and demand score
- Bin_scheduler: Reduced matrices to binary logic to allot movie to screen and schedule
- Display: dispalys multiplex schedule

MODULES USED: ALGORITHMS

```
1.Algorithm: Demand
  Input: movie_release_date, total_seats_booked, capacity
  Output: demand_score
  Variable: array of daily demand scores(demand_arr) , number of screens (screen)
1.Read movie_release_date, total_seats_booked, capacity
2.if (movie_relese_date=current date)
   then Initial_demand <- 100
   else
         demand_score <- (total_seat_booked / capacity) * 100
         demand_arr <- demand_arr+[demand_score]
         Initial_demand <- average(demand_arr)
  endif
3.if (Init_demand>80 and Init_demand<=100)
         screen <- 7
  else if (Init_demand>60 and Init_demand<=80)
         screen <- 5
  else if (Init_demand>33 and Init_demand<=60)
         screen <-2
  else screen <- 1
  endif
4.return demand_score, screen
```

2. Algorithm: Popularity

```
Input: movie_rating , language

Output: popularity_score

Variables: popularity_1, popularity_2

1. Read movie_rating
2. popularity_1 <- movie rating
3. popularity_2 <-0
4. Read language
5. If (language=Tamil)
   then popularity_2 <- 10
   else if(language=English)
   then popularity_2 <- 7
   else popularity_2 <-5
   endif
6. popularity_score <- popularity_1+popularity_2
7. return popularity_score
```

3. Algorithm: Prioritize_slot

```
Input: day

Output: array of slots per screen for the ith screen (screen_slot)

 //as a two-dimensional array with slot and priority//

Variables: integer i

1.i <- 1

2.for (i<=10)

        if (day= “Monday”,” Tuesday”,” Wednesday”, “Thursday”)

        then screen_slot <- [[9+(i-1*5)-12+(i-1*5),5],[12+(i-1*5)-3+(i-1*5),4],[3+(i-1*5)-6+(i-1*5),3]..]

        else

        screen_slot <- [[9+(i-1*5)-12+(i-1*5),5],[12+(i-1*5)-3+(i-1*5),4],[3+(i-1*5)-6+(i-1*5),1]..]

        endif

  return screen_slot

  endfor
```

//for crowd management the slot times for each screen is delayed by a 5-minute buffer time//

```
4.Algorithm: Occupancy_Forecaster
    Input: demand_score, popularity_score, Day, Time_slot, screen
    Output: estimated numerical occupancy for a movie for a given slot (Mxy)
    Variables: integer x, integer y
1.Read demand_score, popularity_score
2. x <- demand_score + popularity_score
3. y <- numerical value taking into account Day, Time_slot, screen
4. Mxy <- numerical occupancy calculated as a percentage of x and y
5. return Mxy
```

5.Algorithm: Mapping

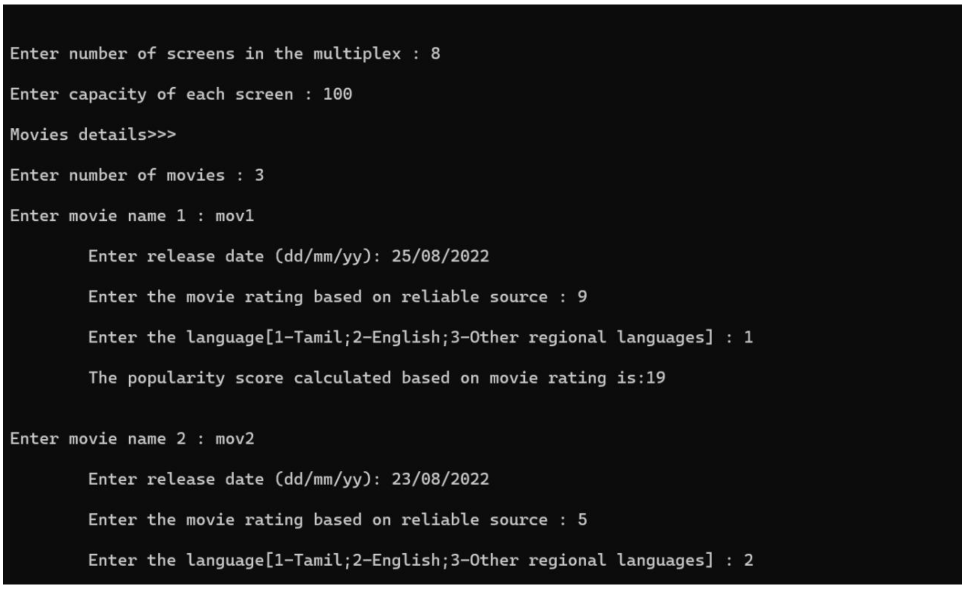
Input: Movie_rank (movies are ranked based on their combined score of popularity and demand), screen_slot, movie_specifications (2D/3D, Dolby, U rated/A rated)

Output: Mapping of movies to a screen based on above mentioned constraint in an array

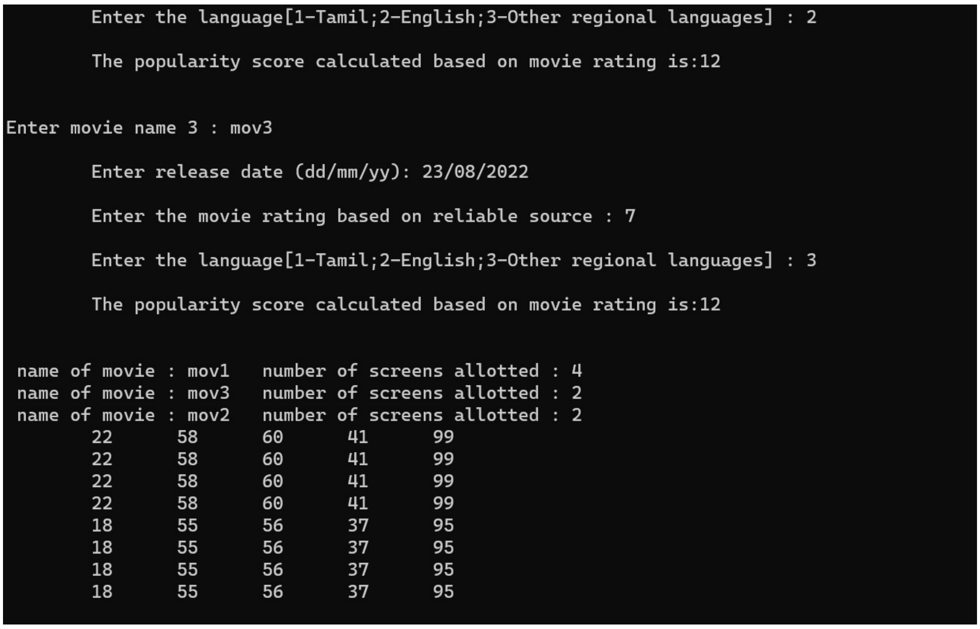
Variables: An indicator variable Cxy that takes value 0 or 1 to indicate if the movie is scheduled for that slot or not respectively.

Based on the given constraints the movies are allotted suitable screens for screening based on the predefined input of facilities available in those screens

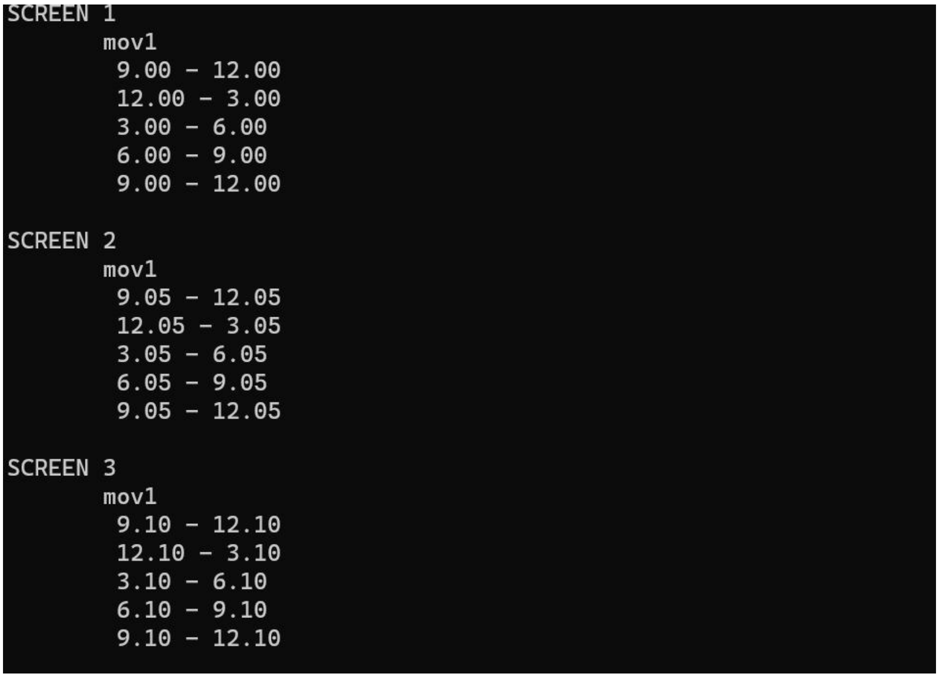
IMPLEMENTATION



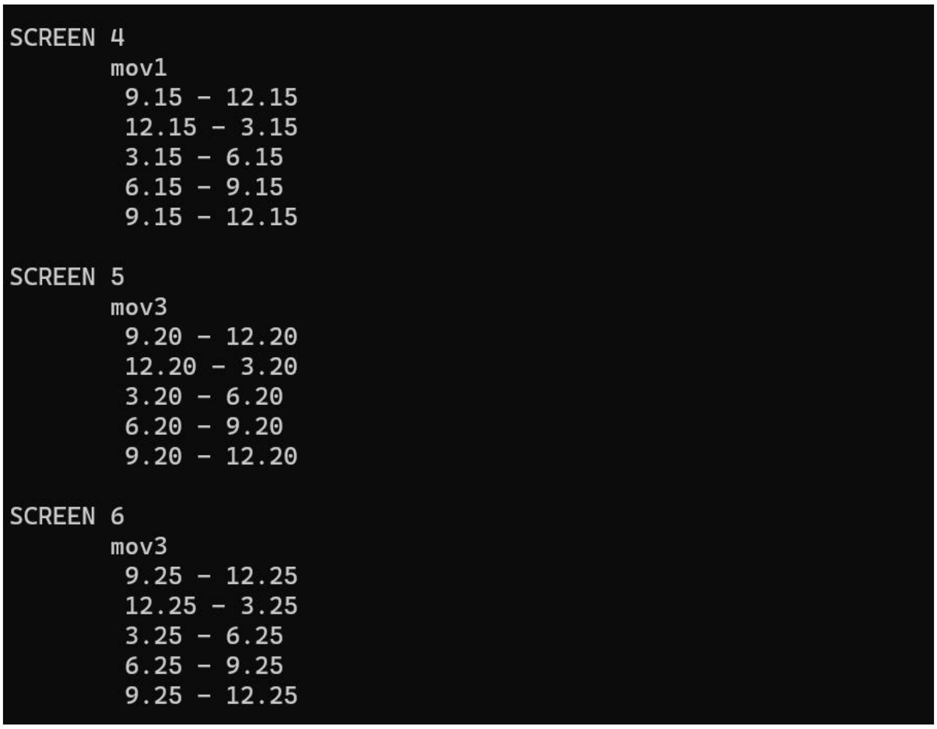
```
Enter number of screens in the multiplex : 8

Enter capacity of each screen : 100

Movies details>>>

Enter number of movies : 3

Enter movie name 1 : mov1

        Enter release date (dd/mm/yy): 25/08/2022

        Enter the movie rating based on reliable source : 9

        Enter the language[1-Tamil;2-English;3-Other regional languages] : 1

        The popularity score calculated based on movie rating is:19


Enter movie name 2 : mov2

        Enter release date (dd/mm/yy): 23/08/2022

        Enter the movie rating based on reliable source : 5

        Enter the language[1-Tamil;2-English;3-Other regional languages] : 2
```



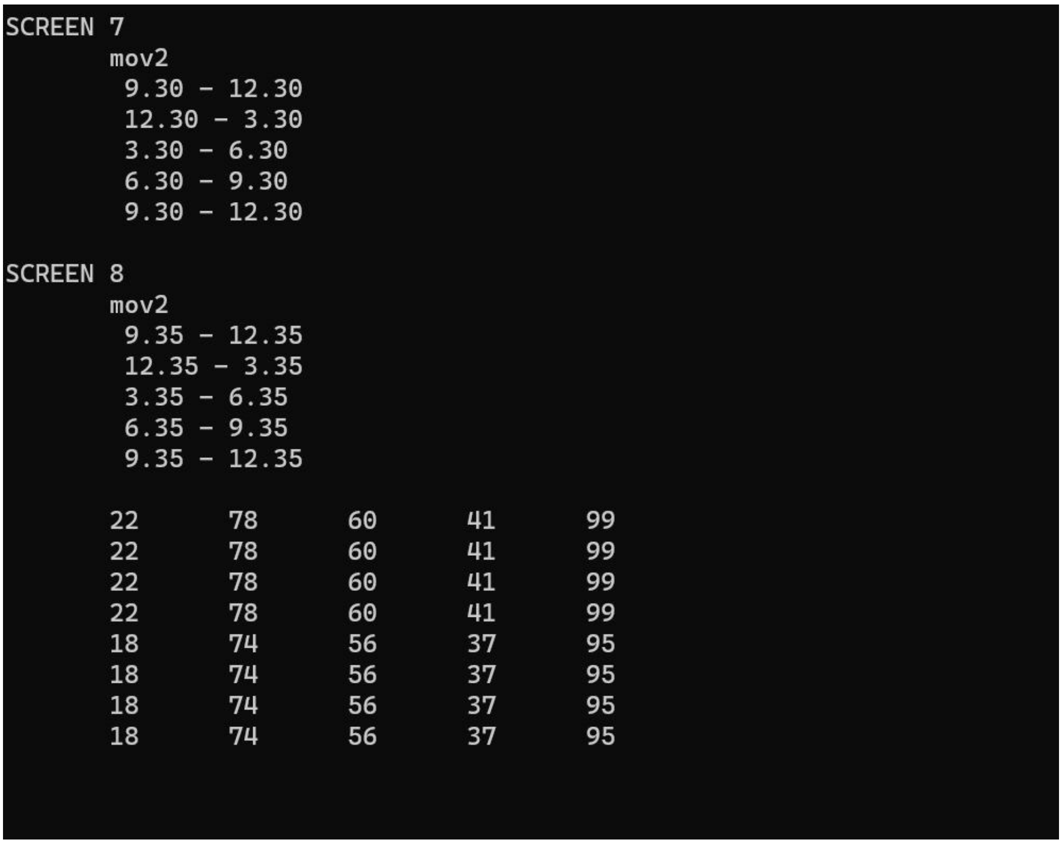
```
        Enter the language[1-Tamil;2-English;3-Other regional languages] : 2

        The popularity score calculated based on movie rating is:12


Enter movie name 3 : mov3

        Enter release date (dd/mm/yy): 23/08/2022

        Enter the movie rating based on reliable source : 7

        Enter the language[1-Tamil;2-English;3-Other regional languages] : 3

        The popularity score calculated based on movie rating is:12


 name of movie : mov1   number of screens allotted : 4
 name of movie : mov3   number of screens allotted : 2
 name of movie : mov2   number of screens allotted : 2
        22      58      60      41      99
        22      58      60      41      99
        22      58      60      41      99
        22      58      60      41      99
        18      55      56      37      95
        18      55      56      37      95
        18      55      56      37      95
        18      55      56      37      95
```



```
SCREEN 1
       mov1
        9.00 - 12.00
        12.00 - 3.00
        3.00 - 6.00
        6.00 - 9.00
        9.00 - 12.00

SCREEN 2
       mov1
        9.05 - 12.05
        12.05 - 3.05
        3.05 - 6.05
        6.05 - 9.05
        9.05 - 12.05

SCREEN 3
       mov1
        9.10 - 12.10
        12.10 - 3.10
        3.10 - 6.10
        6.10 - 9.10
        9.10 - 12.10
```



```
SCREEN 4
       mov1
        9.15 - 12.15
        12.15 - 3.15
        3.15 - 6.15
        6.15 - 9.15
        9.15 - 12.15

SCREEN 5
       mov3
        9.20 - 12.20
        12.20 - 3.20
        3.20 - 6.20
        6.20 - 9.20
        9.20 - 12.20

SCREEN 6
       mov3
        9.25 - 12.25
        12.25 - 3.25
        3.25 - 6.25
        6.25 - 9.25
        9.25 - 12.25
```



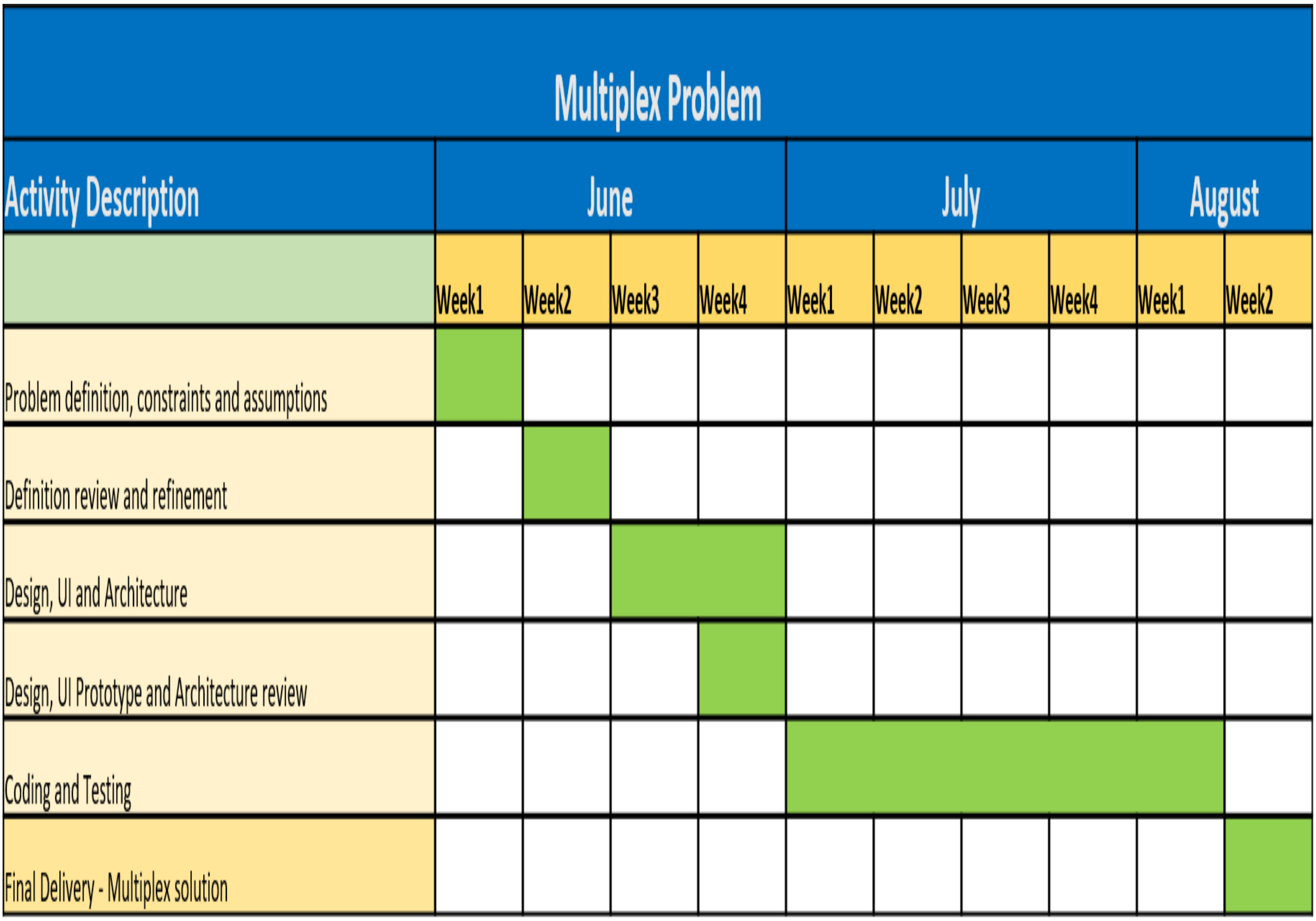
```
SCREEN 7
       mov2
        9.30 - 12.30
        12.30 - 3.30
        3.30 - 6.30
        6.30 - 9.30
        9.30 - 12.30

SCREEN 8
       mov2
        9.35 - 12.35
        12.35 - 3.35
        3.35 - 6.35
        6.35 - 9.35
        9.35 - 12.35

       22      78      60      41      99
       22      78      60      41      99
       22      78      60      41      99
       22      78      60      41      99
       18      74      56      37      95
       18      74      56      37      95
       18      74      56      37      95
       18      74      56      37      95
```

Fig.5: Implementation for a complete week’s movie schedule

- As seen in fig.5 we have computed the solution to scheduling the given movies using multiple matrices comprising various scores which were calculated based on factors such as demand score, popularity score, day score, slot score.
- The matrices were then reduced to binary sequence after forecasting occupancy and hence runtimes for each movie per day.
- The schedule takes into account factors such as crowd management as well.

The basic concepts of Linear programming were used to perform mathematical computation during mapping and scheduling

TIMELINE



Multiplex Problem

Activity Description	June				July				August	
	Week1	Week2	Week3	Week4	Week1	Week2	Week3	Week4	Week1	Week2
Problem definition, constraints and assumptions	🟩									
Definition review and refinement		🟩								
Design, UI and Architecture			🟩	🟩						
Design, UI Prototype and Architecture review				🟩						
Coding and Testing					🟩	🟩	🟩	🟩	🟩	
Final Delivery - Multiplex solution										🟩

CONCLUSION

The solution can be implemented by the multiplex malls to design an efficient schedule for their shows. The solution incorporates various constraints considering the demands and priority of the shows over other constraints.

It also takes into account the problems caused due to over-crowding at the mall area and hence serves as a tool for crowd management.

Enables the user to utilise the availabe resources in the most efficient way which in turn adds on to their profit.

Provides an optimal schedule which is user friendly as it takes into account the demands to the targets on different days of the week such as weekdays, festivals and holidays.

Sustainability of the solution:

- The model integrates elements from marketing (the generation of demand figures) with approaches from operations research (the optimization procedure)
- Various techniques and methods were studied, analyzing the pros and cons of each algorithmic technique. A list of constraints pertaining to the problem were analyzed, discussed, and were designed to produce the most efficient system to solve the problem.
- The solution can be utilized by the profit seeking multiplex management to schedule their slots and design a schedule to their shows in the most optimal form.
- The ultimate aim of our solution is to design a schedule that aims at maximum profit while keeping in mind the needs of the end users of the application. Hence, the system generates the most appropriate schedule.
- Considers crowd management while scheduling that improves the working efficiency of the multiplex by reducing the problems faced due to overcrowding.