Final Project Report

For

MeetMe

Group Members

|  |  |
| --- | --- |
| **Name** | **Role** |
| Dr. Kevin Gary | Project Manager |
| Praful Gaikwad |  |
| Syed Zafar Shah |  |
| Dhawal Soni |  |
| Aditya Kanakamedala |  |
| Viplav Kadam |  |

**1.**    **Final Project Summary**

**1.1.**    **Key Decisions**

**Scope**

The key decision for choosing this system is create a better scheduling meeting planner which is easier and efficient to use than other existing meeting planners. It allows users to log in into the applications either as a scheduler to schedule meetings and send invites to participants or as a participant to accept invites and participate in meetings. Moreover the user can also view the events that has been scheduled in his calendar dashboard. The application also provides a feature to user to log in using google credentials and allow them to provide their google calendar to scheduler in order to decide preferred time slots for the meeting without violating user privacy.

**1.3.**    **Learning Outcomes Summary**

**1.4.Project Plan**

All members of the team have responded, contributed, and interacted well with each other throughout the project. We have arrange meetings on regular basis to discuss about the status of the project and tasks to be done in the next sprint. We have focused on the project deliverables of one sprint at a time and discussed them in detail. After that, we have distributed the work among the team. The tasks have completed the assigned work in a timely manner to finish the sprint on the time. We have updated the sprint status on the tiaga and updated source code on the github on the regular basis.

**1.6.**    **Implementation**

We have used spring MVC to design this system. We have used bootstrap and javascript for the front end design and java for the backend language. We have used MYSQL database to store the project data. We have also developed an efficient algorithm to help scheduler to select a preferred time among the participants to schedule meetings.

The steps of the algorithm is as following:

1.       Create counters same as preferred time slots generated by the meeting scheduler.

2.       Send emails to the participants with preferred time slots.

3.       Insert preferred time slots of every user into database.

4.       Fetch time slots for the event and store into a list.

5.       Run a loop till the end of list.

6.       Compare list elements with the time slots provided by meeting scheduler.

7.       If matched, increase respective counters of the time slot.

8.       Sort the time slots with respect to counter.

9.       Select time slot with highest counter value as the preferred time slot.

10.   Rank remaining time slots from highest counter value to lowest counter value.

11.   Send the preferred time slot and remaining time slots with rank value to the participants.

For google User

1.       Fetch google calendar of every user.

2.       Create an array in which every element represent a time slot of the window preferred by the scheduler.

3.       Run a loop for each user.

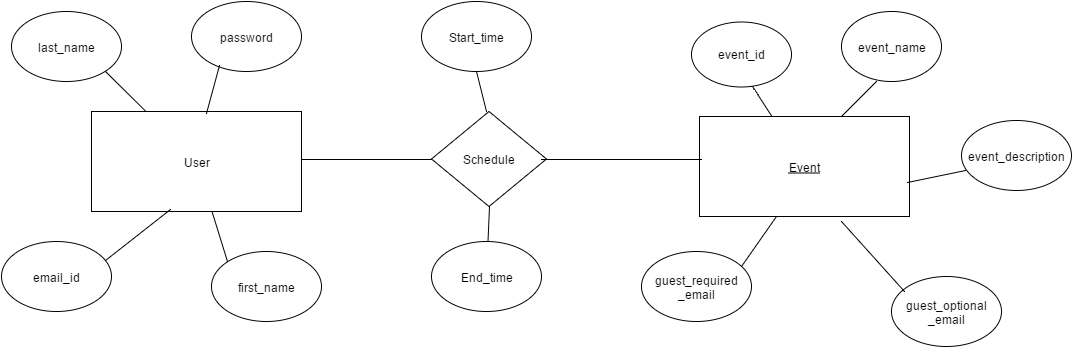
4.       Fetch time slots of events (busy time slots) during the given window and mark respective array element.

5.       After that, find unmarked array elements and its respective time slots.

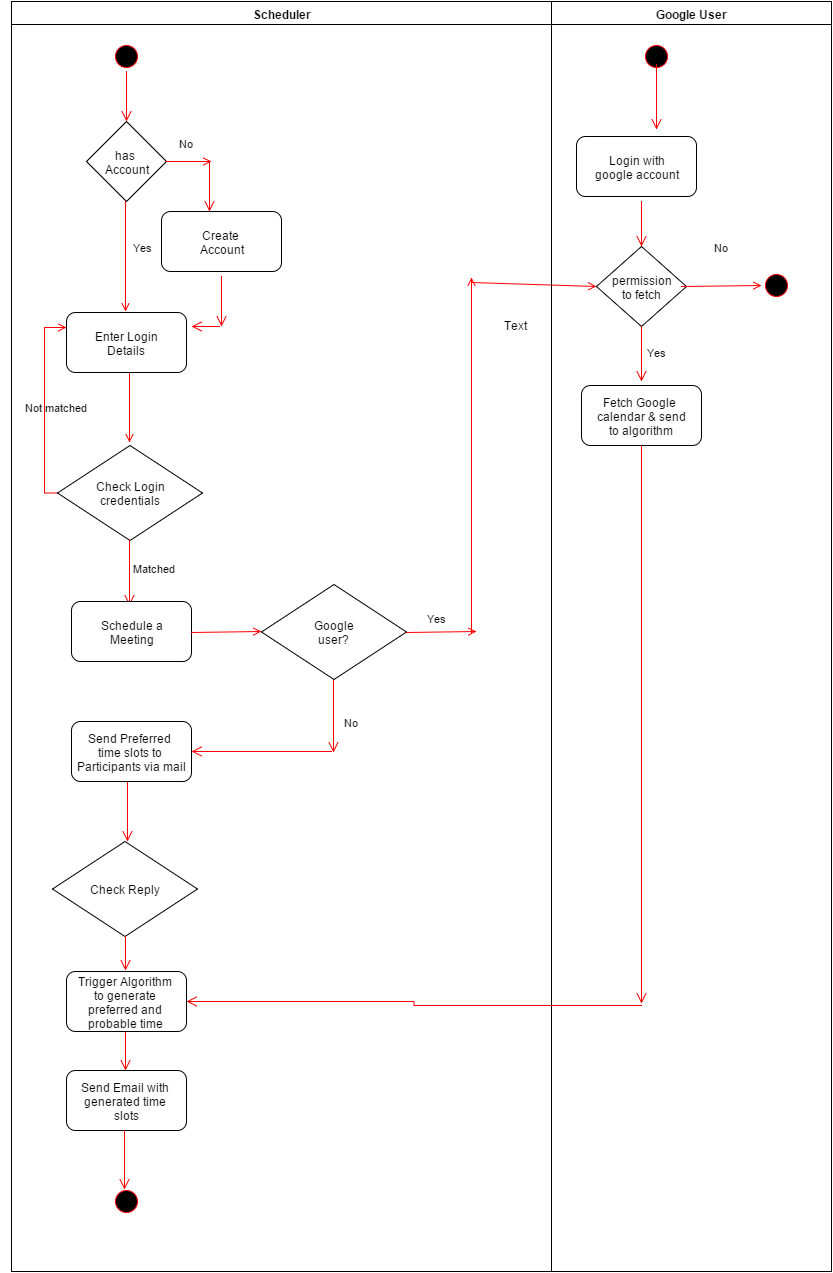
6.       Send the time slots to the participants.

Before starting implementation, we have developed diagrams to build the system such as ER diagram, Activity diagram and Use case diagrams.

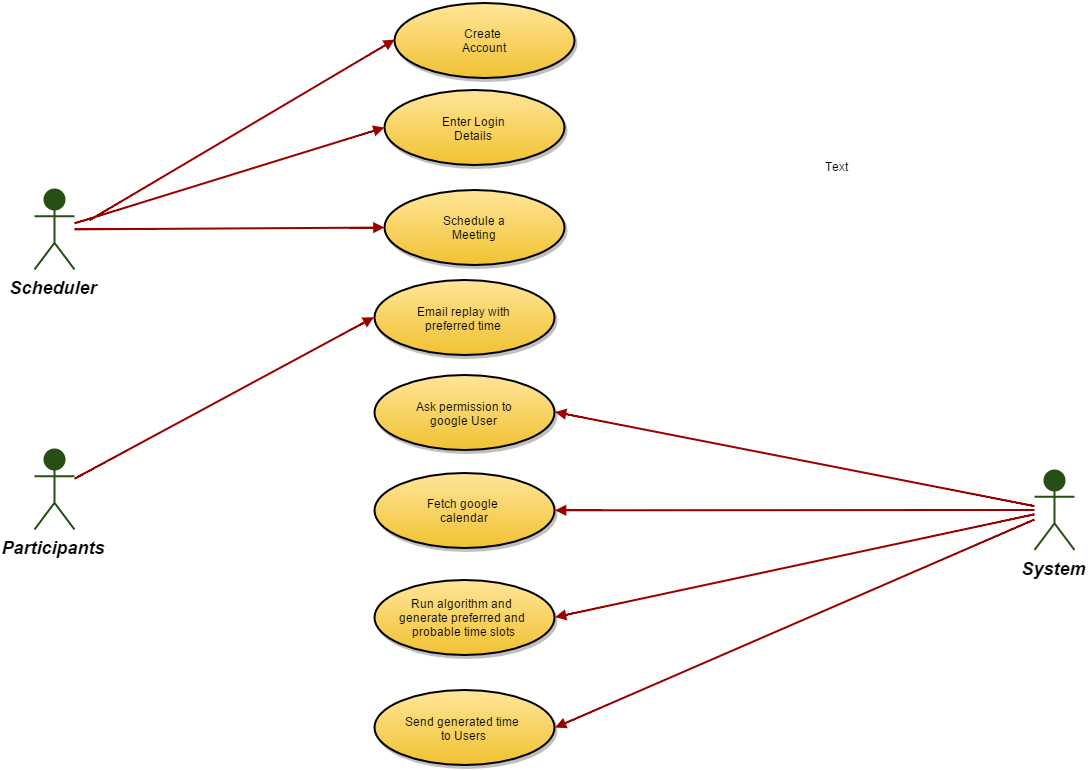
**ER Diagram:-**



**Activity diagram**



**Use case Diagram:-**



**1.7.**    **Validation and Verification**

We have provided validation on the login credentials so that only authorised user will have access to the system. We have also taken into account user’s privacy. We have asked user to give permission before fetching google calendar. For testing, We have performed Unit testing, Integration testing and functional testing. In unit testing, we have tested every small testable part of the application. After that, we have done the integration testing at time of combining modules. In the end, we have taken into account validation testing and checked weather system meets its intended purpose. In this testing, we have tried to improve quality of the system.

**1.8.**    **Outcome & Lesson learned**

During this project, we have learned about different aspects of the project development. We have learned the importance of the requirement elicitation phase. We have learned how to gather requirements and how to create user stories corresponding to the requirements. We learned scrum agile methodology and how to use this methodology during this project. We also got opportunity to identify the advantages of this methodology as compare to other existing methodology. We also learned how the diagrams such that ER diagram, Activity diagram and Use case diagram helps to develop system easily. We also learn spring MVC architecture and how to implement this architecture to design the system. We also learned about front end languages such that bootstrap, javascript, back end languages such that java and MySQL database during this project.