FINAL SEMESTER ASSESSMENT (FSA)

B.Tech. CSE
Semester VI

UE18CS355 – Object Oriented Analysis and Design with Software Engineering Laboratory

Project Report on

CALENDAR MANAGEMENT

Submitted by

Sree Pranavi G	**-**	**PES1201800368**
Sanriya S	**-**	**PES1201801233**
Sivanjali J	**-**	**PES1201801926**

January - May 2021

Department of Computer Science & Engineering
RR Campus.

Bengaluru - 560100, Karnataka, India.

TABLE OF CONTENTS

I. Abstract

This calendar management project is a web application which mimics the google calendar for personal or workplace use, to manage and schedule events or to assist users with tasks and reminders. The application is intended for any users needing the convenience of an online calendar that can be shared, accessed, and modified. In general, this calendar application is meant to help individuals with busy schedules as it’s quite exhausting to keep track of various events. So the goal is to develop an application to make it convenient and efficient for the users to easily manage and also add reminders to ensure all tasks and events are successfully carried out without letting them slip.

The application comprises the hereinafter discussed primary features. Authentication or user sign in is the first step in the application, it is essential to ensure that the right user is modifying the calendar contents for all security reasons. The next feature for user convenience is the availability to change a calendar view from 3 different available views - month, week and day of the calendar using a simple drop down menu. The key trait is adding events, tasks and reminders along with modifying and deleting them in any of the 3 views and displaying them in a color coded fashion for user comfort. And finally, the application also helps the user in sending invites to all the mail IDs in the guest list of a created event with the date and time details.

II. Software Requirement Specification - SRS

1. Introduction

1.1 Purpose

Google Calendar is an easy-to-use, free online calendar for personal or workplace use, such as listing classes or library events, coordinating meetings and events with colleagues, or scheduling work shifts for residents at a hospital or librarians at the reference desk.
The intent of Google Calendar is to make project management convenient and the ability to use a personalised calendar. This open source application is offered by Google which manages all events and works like a reminder which alerts the users.

1.2 Intended Audience

Google Calendar is for anyone needing the convenience of an online calendar that can be shared, accessed, and modified by a single user, invited users in a group, or the public.

1.3 Product Scope

The product scope is to make sure users can view their Google Calendars via a mobile phone's browser or a desktop application. While setting up Google Calendar, users can choose from twenty-four languages, calendar events can be created by using the Create an Event or Quick Add links or by clicking directly on a time and day on the calendar. Google user credentials are used to login in to google calendar. Calendar Settings offers many options, such as language, country, time zone, time format, date format, weather icons based on geographic location, and much more.
Individual events in shared calendars can be set as “Private,” which provides no details and only allows others to see the marked-off time as busy. The calendar can also be shared with non-Google Calendar users by creating a uniform resource locator (URL) they can click on or subscribe to via a feed reader.

1.4 References

google calendar API

https://www.sprintzeal.com/blog/use-google-calendar-as-a-project-management-tool

https://support.google.com/a/answer/2905486?hl=en

2. Overall Description

2.1 Product Perspective

In general, this calendar application is meant to help individuals with busy schedules. Those who must balance their time between working on multiple simultaneous projects would categorize key users since projects are present in every industry, it can be difficult to keep track of them all without some tools to help keep track of them for us. So the goal is to develop an application to make it convenient and efficient.

2.2 Product Functions

- Schedule meetings with groups using "Find A Time"
- Add a google hangout to your event
- Email event guests
- Arrange appointment slots
- Share calendars with others (making it private or public)
- Users can create master or resource or personal calendars
- Customising calendars
  → manage event notifications
  → choose calendar's view
  → change calendar's look
- Schedule events
  → create event
  → add event description and attachments
  → set up notification

2.3 User Classes and Characteristics

- Students and Teachers: The majority of students & teachers shall use this application during the start of a new semester. Teachers use it keeping track of due dates and upcoming events and planning them accordingly.
- IT Staff: ITS staff will be the primary maintainers of the application and may have to perform some administrative functions relating to the academic calendar and holidays.
- The technical experience of these users should not matter as the system will be straightforward and easy to use. Basically, it can be used by anyone who has minimum technical knowledge and a device to view it.

- The end user (no classifications), who can view the calendar, add events, invite people to events via email, only after authentication.

2.4 Operating Environment

- The application will be aimed as a web application, and should be able to run on any device connected to the internet.
- Google Calendar is fully compatible with most Apple and PC computer browsers: Microsoft Internet Explorer versions 6–7, Mozilla Firefox version 2 and later, and Apple Safari 3.1.
- Users must enable cookies and JavaScript.
- Google Calendar can sync with the 2003 and 2007 Microsoft Outlook Calendar versions, as long as one has Windows Vista or Windows XP, but not the Windows XP 64-bit edition.
- Hardware requirements : Pentium IV 1GHz speed processor , hard disk of 40 GB and RAM 4 GB.

2.5 Design and Implementation Constraints

- Creating more than 100,000 events in a calendar during a short period, users might lose calendar edit ability for a few hours.
- Creating more than 60 new calendars in a short period, the calendar might go into read-only mode for several hours.
- To prevent spamming, Google Calendar limits the number of invitations a user sends to external guests. If a user sends 10,000 invites in a short period, the user’s calendar might go into read-only mode.
- Implementation constraints would be synchronising the Calendar of the same user across multiple devices.

2.6 Assumptions and Dependencies

Users already familiar with calendar and project management applications should be able to readily understand how this application functions.

1.Use of a third-party service will improve the service and reduce coding time and effort.

2. Users will be able to use the application on both mobile and desktop devices equally well.

3. Projects will not require an official administrator or moderator.

4.User documentation will be minimal as the application will be structured
much like other calendar applications.

3. External Interface Requirements

3.1 User Interfaces

The user interface will be accessible via any standard web browser and shall mimic typical calendar scheduling software appearances to encourage ease of use for users.The Google calendar can be accessed through a secure user interface requiring the use of a predetermined login name and password of their Google account.

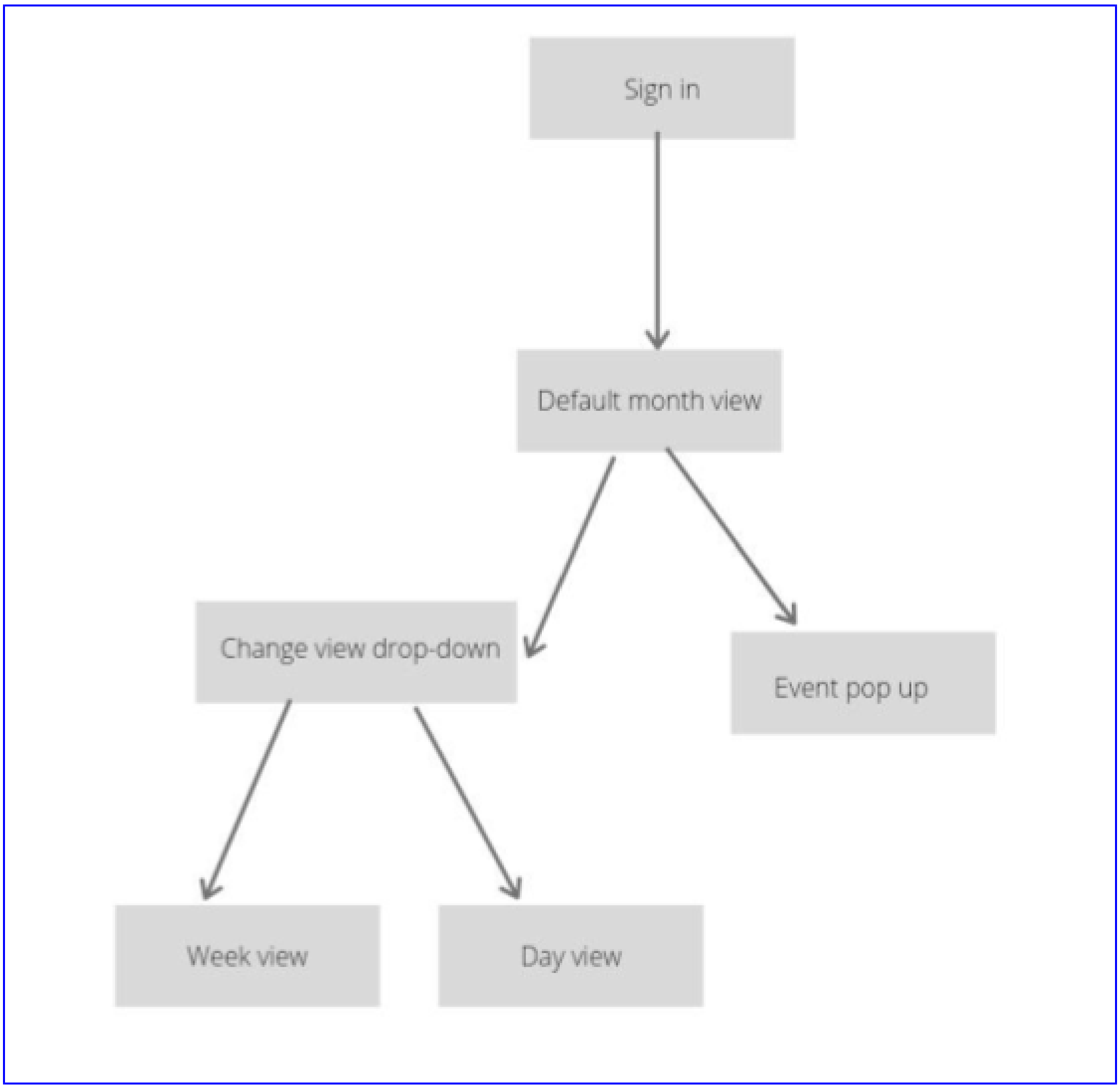


Fig 3.1.a

3.2 Software Interfaces

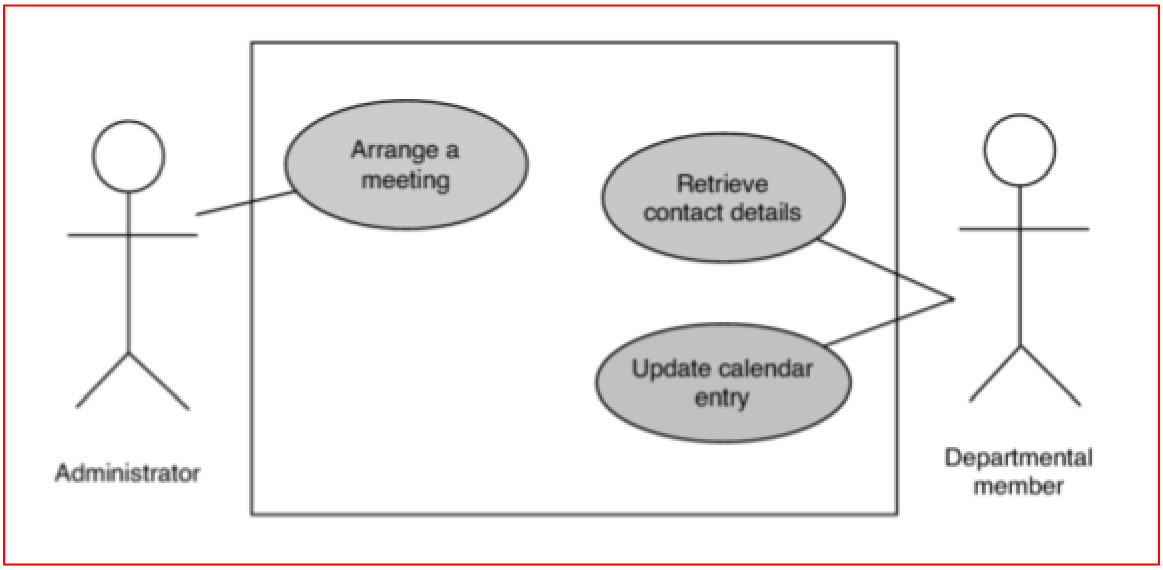
1. Database-MySql.
2. Server-Apache.
3. Development tools.
4. Network-internet.
5. OS-Windows/linux

3.3 Communications Interfaces

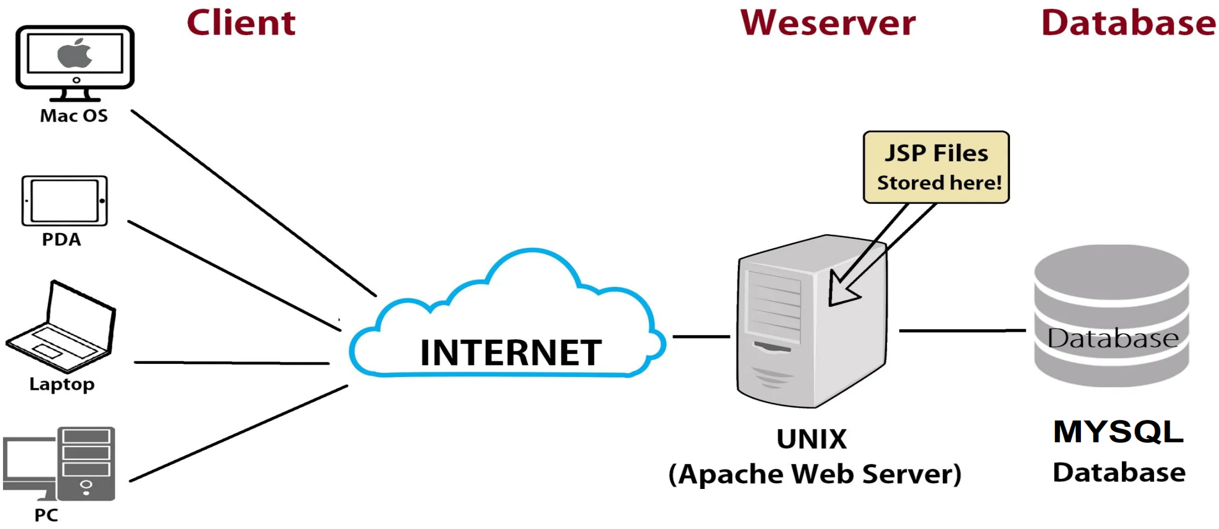
- SMTP for reminders and alerts of any event.
- HTTP protocol as a way of communication between server and client.

4. Analysis Models

4.1 Scheduling a meeting :

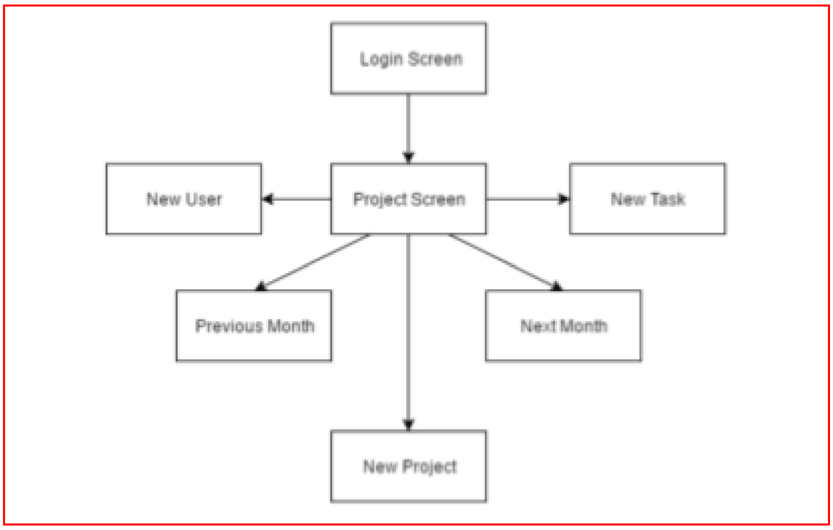


4.2 Communication between Apache web server and client browser :

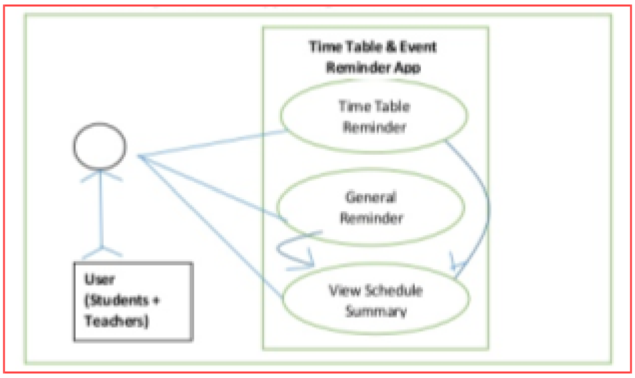


4.3 Overall :

Upon first use of the application, the user is directed to the login screen, where they will be prompted to login via their Google account. After creating an account, the user is sent to a screen where they view their personal project in the current month.



4.4 Task management example when a meeting is scheduled :



4.5 Alerts/ Reminders of events :

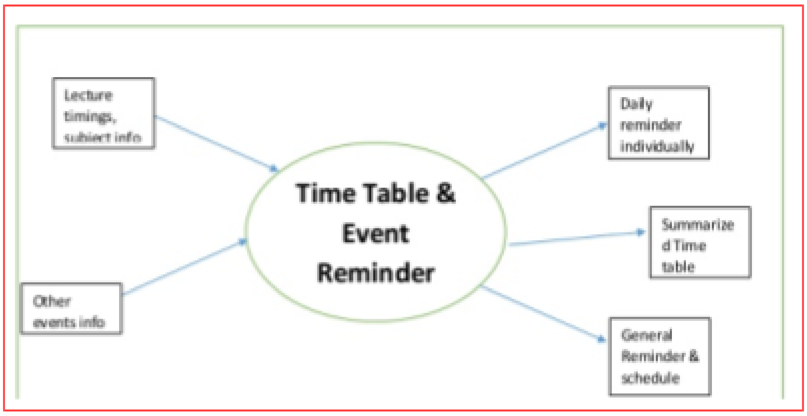

Lecture timings, subject info
Other events info
Time Table & Event Reminder
Daily reminder individually
Summarize d Time table
General Reminder & schedule

Fig 4.1.a is the use case diagram.

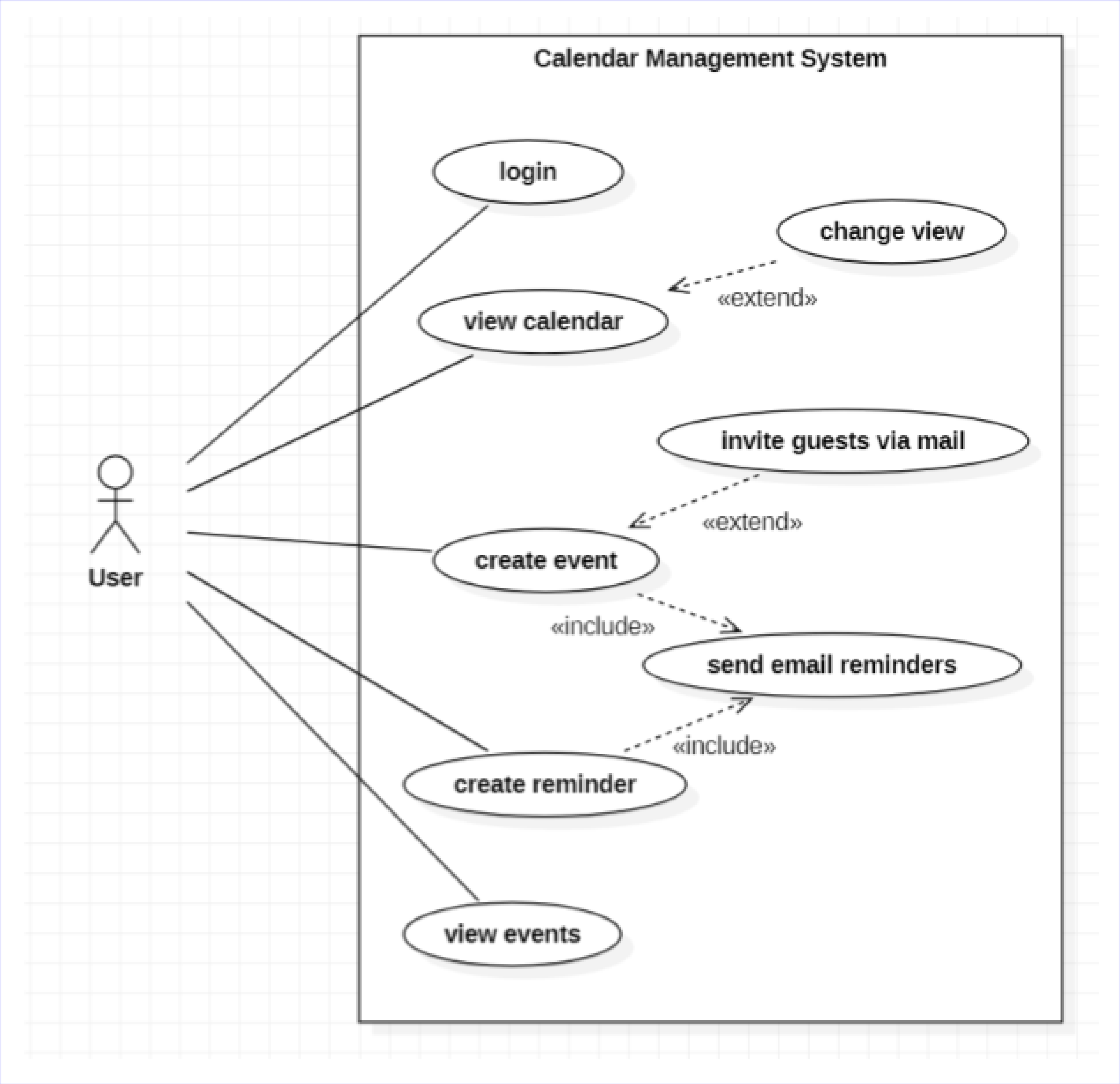


Fig 4.1.a

5. System Features

5.1 Login

5.1.1 Description and Priority

The calendar application can be accessed only by the users who own a mail ID. The user can either enter the mail ID along with the password and login or login using their email account. The priority of this system feature is high.

5.1.2 Stimulus / Response Sequences

The user will have to enter the mail ID and the password and login. And if details are correct, then the user will be logged into the calendar. Else, an error message shall be displayed.

5.1.3 Functional Requirements

REQ-101	Users will have a UI to login.
REQ-102	The users will enter the mailID and the password and login.
REQ-103	Valid credentials will take the user into the calendar interface.
REQ-104	Invalid credentials will throw an error on the login page.
REQ-105	The user credentials are backed up in the database.

5.2 Different views on Calendar

5.2.1 Description and Priority

Once a user logs in, the user can toggle between the week, month and the day view. The default view would be the month view and the priority associated is high.

5.2.2 Stimulus/ Response Sequence

On successful login, the user is on the default month view. The user can change views using a dropdown.

5.2.3 Functional Requirements

REQ-201	On successful login, the user enters the month view.
REQ-202	A change view dropdown box to change between the views.
REQ-203	The user must be able to change between the different views. On clicking the name of the view, the user must be able to enter that view.

5.3 Event Management

5.3.1 Description and Priority

Projects are an essential classification for the system and its functionality is of the highest priority. In the instance of this planner application, a project is primarily for organizing tasks. When a project is shared, all users have the ability to view and create tasks related to that project.

A user can set events and send out invites to their guests. The user must enter details such as title of the event, date and time of the event and the guest list. On saving the event, the invitations are sent to the guests. The priority associated is high.

5.3.2 Stimulus / Response Sequence

The user should be able to fill the details for the event. If the details entered passes all the validation checks then the email invitations are sent out for the guests in the list.

5.3.3 Functional Requirements

REQ-301	Users shall be able to create a new event.
REQ-302	When creating a event, users shall be able to provide an event name
REQ-303	Once an event is created, a user shall be able to view a calendar that displays the event and the invitations are sent out.
REQ-304	The invitations are sent out.
REQ-305	A user shall be able to add guests to a pre existing event.
REQ-306	A user shall be able to cancel an event.

5.4 Task Management / Reminder Management

5.4.1 Description and Priority

The user must be able set reminders with task name and time for personal reference. The priority of this feature is high.

5.4.2 Stimulus / Response Sequence

The user should be able to fill the details for the task/reminder. If the details entered passes all the validation checks then the task is entered into the database and it also shows up on the user’s calendar.

5.4.3 Functional Requirements

REQ-401	Users shall be able to add tasks to the calendar.
REQ-402	Users shall be able provide a task title to the task.
REQ-403	Users shall be able to add a start and end date to the task.
REQ-404	Users shall be able to add a description to the task.
REQ-405	Users shall be able to cancel creation of the task

6. Other Nonfunctional Requirements

6.1 Performance Requirements

1. Projects must take no longer than one minute to update revisions.
2. Project revisions must take no longer than one minute to be processed by the server.
3. Projects revisions must take no longer than one minute to be received from the server.
4. The planner service must be online/functional 99% of the time.

1. The logging in of users must happen within 3 seconds.
2. The mail reminders must be sent to all the mailID's mentioned in the guest list.
3. The calendar service must be functional 99% of the time.
4. The event or task edits made must reflect on the database and the UI.

6.2 Safety Requirements

1. Users are responsible for managing the content they view.
2. The application will not censor, modify, or restrict user language.

6.3 Security Requirements

The server must support user authentication.

6.4 Software Quality Attributes

No prior knowledge is required to use the software but the user is required to have a basic computer knowledge. It is required that the user has a Google account in prior.

Appendix A: Glossary

Project : A group of tasks of a similar or same topic
Task : Represents an user's work

Event: It is a meeting or an activity that involves more than one user. Hence, invitations are sent out for events.

Task: It is a personal to-do action. This does not involve other users and hence, no invitations are sent out.

Appendix B: Field Layouts

Field	Length	Data Type	Description	Is Mandatory
Start Date	8	Date	Date of start of event	Y
End Date	8	Date	Date of end of event	Y
Start Time	4	Time	Start of time of event	Y
End time	4	Time	End of time of event	Y
Name of event	100	Alphanumeric	Name of event	Y
Recurring	8	Int	Date of Mandate	Y
Location	100	Alphanumeric	Location of Event	Y

Appendix C: Requirement Traceability Matrix

Sl. No	Requirement ID	Brief Description of Requirement	Architecture Reference	Design Reference	Code File Reference	Test Case ID	System Test Case ID
1	REQ-101	UI to login.	1.a	2.a, 3.a, 4.a, 5.a	login.php/ login.css	UC-101	SC-001
2	REQ-102	Enter the mailID and the password to login.	1.a	2.a, 3.a, 4.a, 5.a	login.php/ login.css	UC-101	SC-001
3	REQ-103	Valid credentials will take the user into the calendar interface.	1.a	2.a, 3.a, 4.a, 5.a	login.php/ login.css	IC-001	SC-001
4	REQ-104	Invalid credentials will throw an error on the login page.	1.a	2.a, 3.a, 4.a, 5.a	login.php/ login.css	UC-101	SC-001
5	REQ-105	The user credentials are backed up in the database.	1.a	2.a, 3.a, 4.a, 5.a	login.php/ login.css	IC-007	SC-001
6	REQ-201	On successful login, the user enters the month view.	1.a	2.a, 3.a, 4.a, 5.a	login.php/ login.css	IC-001	SC-001
7	REQ-202	A change view dropdown box to change between the views.	1.a	2.a, 3.a, 4.a, 5.a	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-103	SC-001
8	REQ-203	The user must be able to change between the different views. On clicking the name of the view, the user must be able to enter that view.	1.a	2.a, 3.a, 4.a, 5.a	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-103	SC-001

9	REQ-301	Users shall be able to create a new event.	1.a	2.a, 3.a 4.a, 4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	IC-002	SC-001
10	REQ-302	When creating a event, users shall be able to provide an event name	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-203	SC-001
11	REQ-303	Once an event is created, a user shall be able to view a calendar that displays the event.	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	IC-003	SC-001
12	REQ-304	The invitations are sent out.	1.a	3.a, 4.c, 5.c	month.php / day.php / week.php	UC-301 IC-006	SC-001
13	REQ-305	A user shall be able to add guests to a pre existing event.	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-206	SC-001
14	REQ-306	A user shall be able to cancel an event.	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-207	SC-001
15	REQ-401	Users shall be able to add tasks to the calendar.	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-202 IC-002	SC-001
16	REQ-402	Users shall be able provide a task title to the task.	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-203	SC-001
17	REQ-403	Users shall be able to add a	1.a	4.b, 5.b	month.php / day.php /	UC-203 UC - 204	SC-001

		start and end date to the task.			week.php/ month.css/ day.css/ week.css/commonjs.js	UC-205	
18	REQ-404	Users shall be able to add a description to the task.	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-203	SC-001
19	REQ-405	Users shall be able to delete task	1.a	4.b, 5.b	month.php / day.php / week.php/ month.css/ day.css/ week.css/commonjs.js	UC-207	SC-001

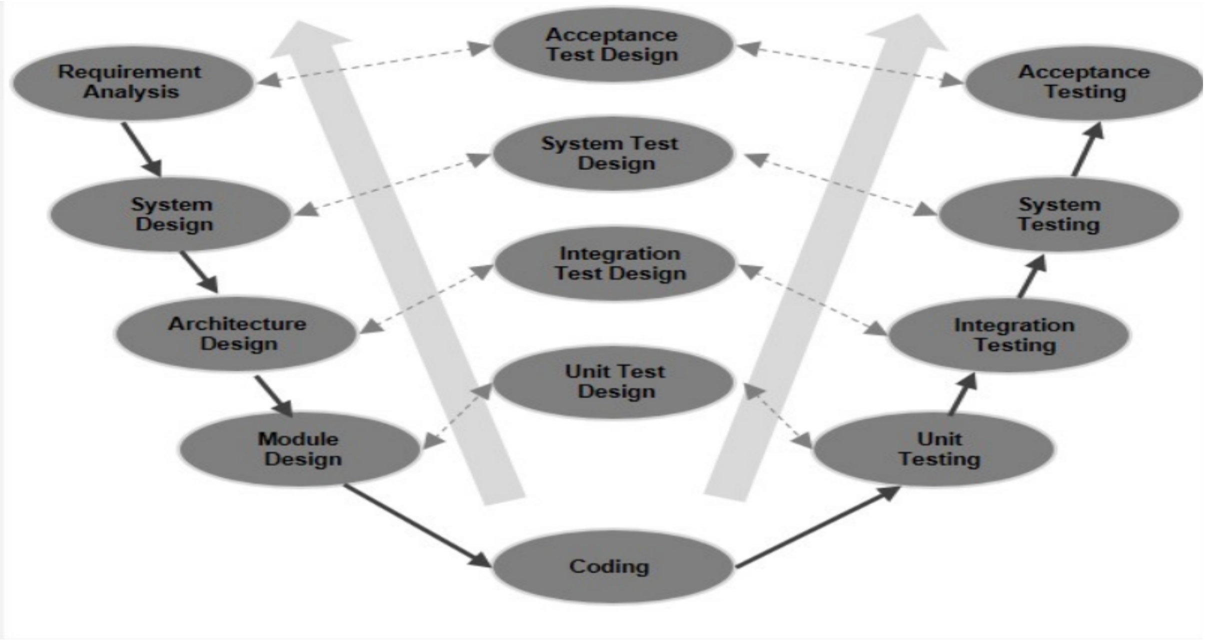
III. Project Plan

1: Identify the lifecycle to be followed for the execution of your project and justify why you have chosen the model.

We have chosen the V model because before releasing the complete application we plan to build the application in small stages before complete release of the implementation. This is an extension of the waterfall model, but in the V model the testing is done parallely at every stage with development in a sequential way.

The advantage of the V-Model method is that it is very easy to understand and apply. The simplicity of this model also makes it easier to manage and control.

The disadvantage is that the model is not flexible to changes and just in case there is a requirement change, which is very common as it becomes very expensive to make the change. The v model depicts the different phases of the SDLC :



The different phases of this model involves :

Business Requirement analysis : The first phase in the development cycle where the product requirements are understood from the customer's perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirements.

System Design : Once there are clear and detailed product requirements, it is time to design the complete system. The system design will have the understanding and detailing the complete hardware and communication setup for the product under development. Doing this at an earlier stage leaves more time for the actual test execution later.

Architectural Design : Architectural specifications are designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken.

Module design : The unit tests are an essential part of any development process and help eliminate the maximum faults and errors at a very early stage which is done here.

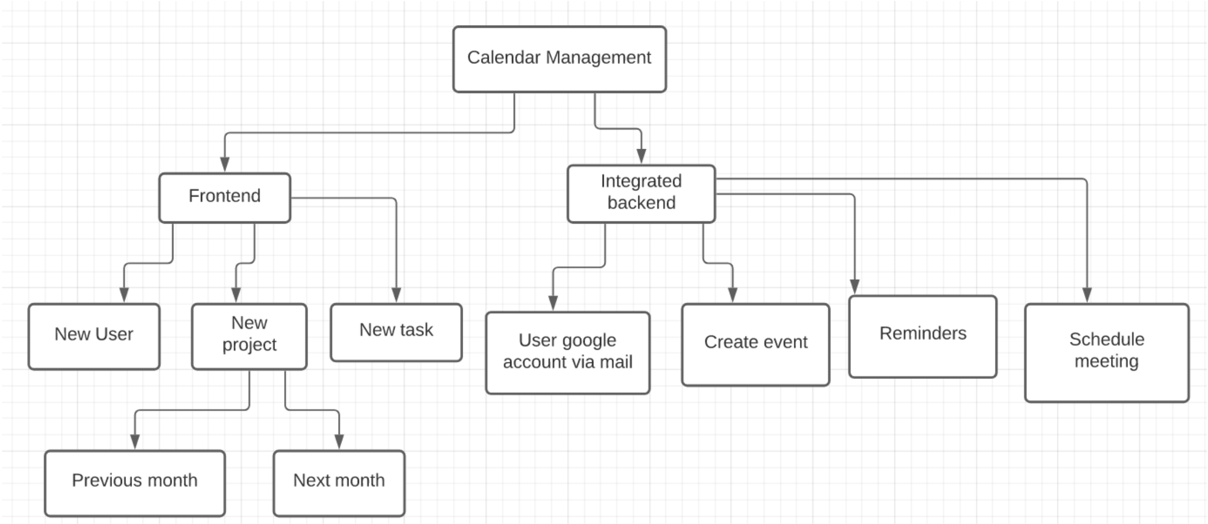
Coding Phase : The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements.

2: Identify the tools which you want to use at different phases of SDLC like planning tool,design tool, version control, development tool, bug tracking and testing tool.

- Planning tool- Microsoft Project.
- Design tool- Bootstrap and CSS.
- Version control- Git.
- Development tool- web development framework(HTML,PHP, MySql)
- Bug tracking- no bug tracking tool used but a beta version of the website   before complete release
- Testing tool- a beta version of website before actual release for testing and bug tracking
- Bug tracking and testing tool - Selenium

3: Create a Work Breakdown Structure for the entire functionalities of your project in detail.

WBS -
1. Frontend development
2. Backend development
3. Integration of both
4. Adding the required functionalities and additional functionalities



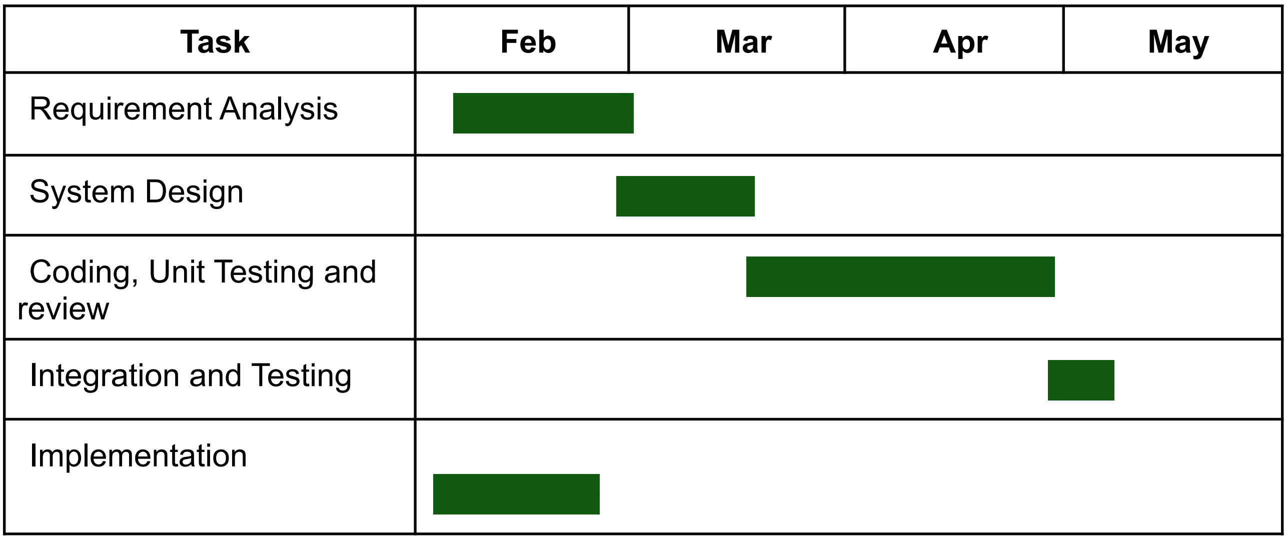
4. Determine all the deliverables and categorize them as reuse/build components and justify the same.

The deliverables would be a website with a working event managing mechanism to notify a user of the occurrence of the event (as alerts or notifications) entered by the user. This is a small and simpler mini version of the Google Calendar. All the functionalities have to be built (coded) from scratch but when **built**, they can be **reused** and more functionality can be added. The main web frameworks remain the same (only the design or other small changes in the frontend) so hence this can be **reused**. The storage system used to implement the backend has to be **reused** to maintain consistency of data. The application servers (multiple servers so as to handle fault tolerance as in the case of one application server) have to be **built** so that they run 24 x 7, so that customers receive their event alerts at any time of the day without any trouble.

5. Do a rough estimate of effort required to accomplish each task in terms of person months.

As a team, it would take us approximately 4-5 months to build and make this project completely functional. The frontend would take no more than a month but the backend and integrating the two with all other functionalities is where most of the time may go.

6: Create the Gantt chart for scheduling the defined tasks.



IV. Design Diagrams

1. System architecture block diagram

The architecture diagram is a birds eye view of the entire calendar management application. The system consists of a UI through which the users can avail the various calendar application functionalities.
The UI is connected to the backend server that performs all the business logic (validation checks, database updation, trigger to send out mails etc). The server is connected to a database that is a storehouse of all the necessary details of the users and their actions.
The server is also connected to a third party service to send out emails to the guests of an event.

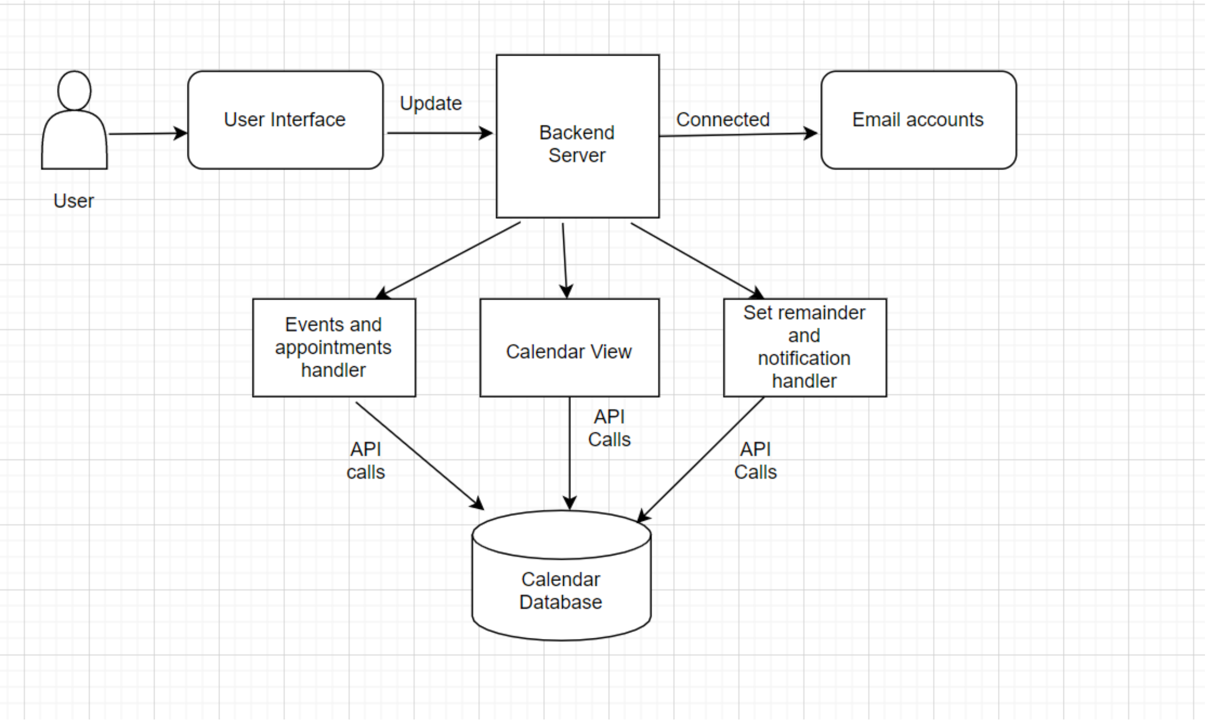


Fig 1.a

2. Class diagram

The classes are user, email service, event, day view, week view and month view generalized into calendar. The user class has email id & status as attributes and signin and sing out operations. It is related to the calendar class which has authentication, change view and add events as operations. The day view has hours and events per hour as attributes, week view has week dates, hours and events per hours and month view has month name, month dates and events per day as attributes. The events class has a lot of attributes and operations as shown below, it is related to email service (3rd party) with 0..* and 1 cardinality. It is related to the calendar in 0..* and 1..* cardinality. The email service is also related to users in 1 and 1..* cardinality.

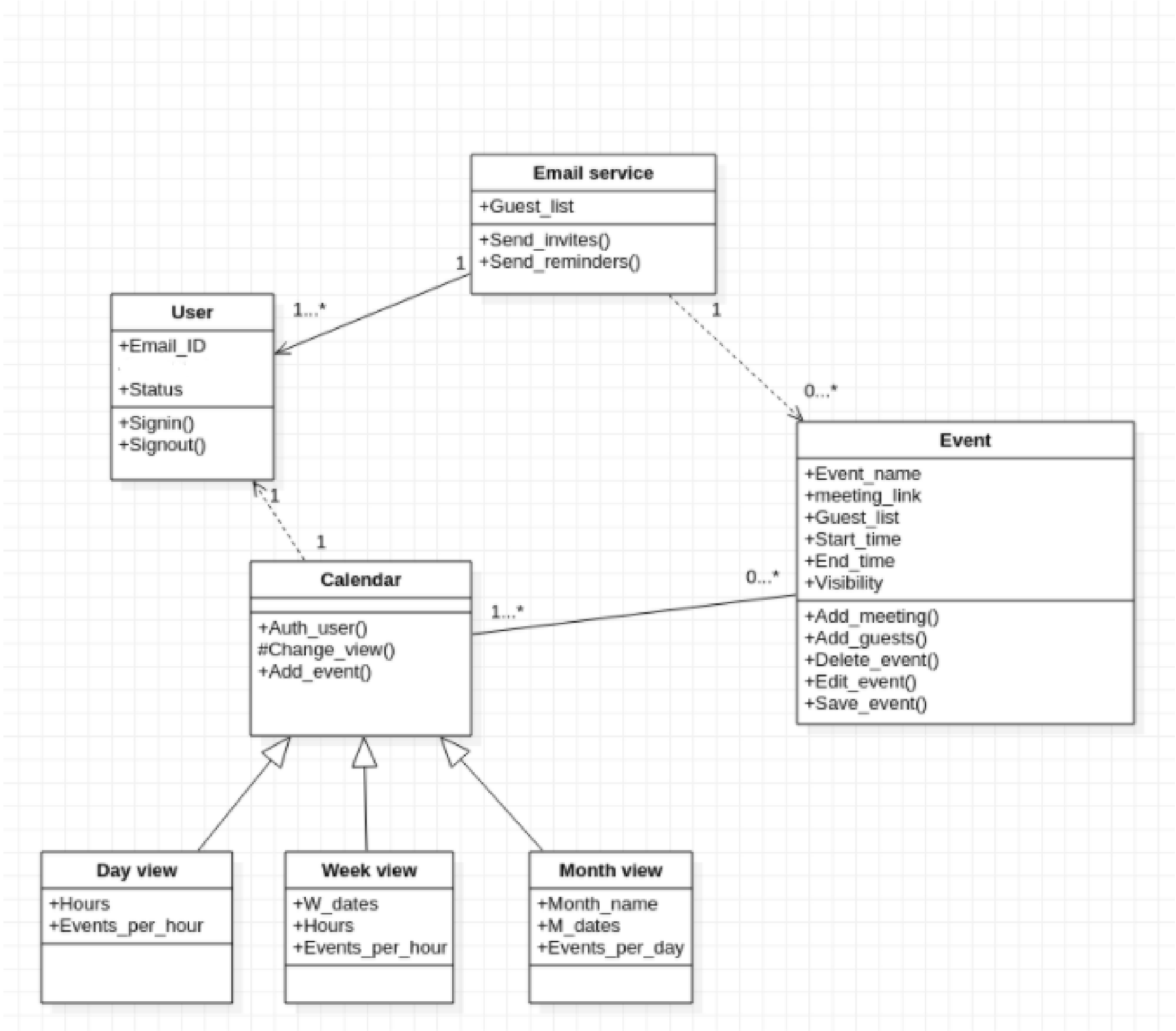


Fig 2.a

3. State diagram

The following diagram depicts the various states a user will go through in the calendar application. The initial state is the login state and on entering valid credentials, the user proceeds as a logged in user. Else, upon failure, the user can reattempt to login. On logging in, the user is at the default month view from where he could change the views or create an event. On choosing to change the view, the user can toggle between month, week and day view. The user still can create an event or task. On creating an event, the action is successful if all valid entries are made, else it's a failure. If the event creation is successful, it is seen that mail invites are also sent out. The user can set reminders/tasks. On filling the valid details, the event and task becomes a part of the user's calendar.

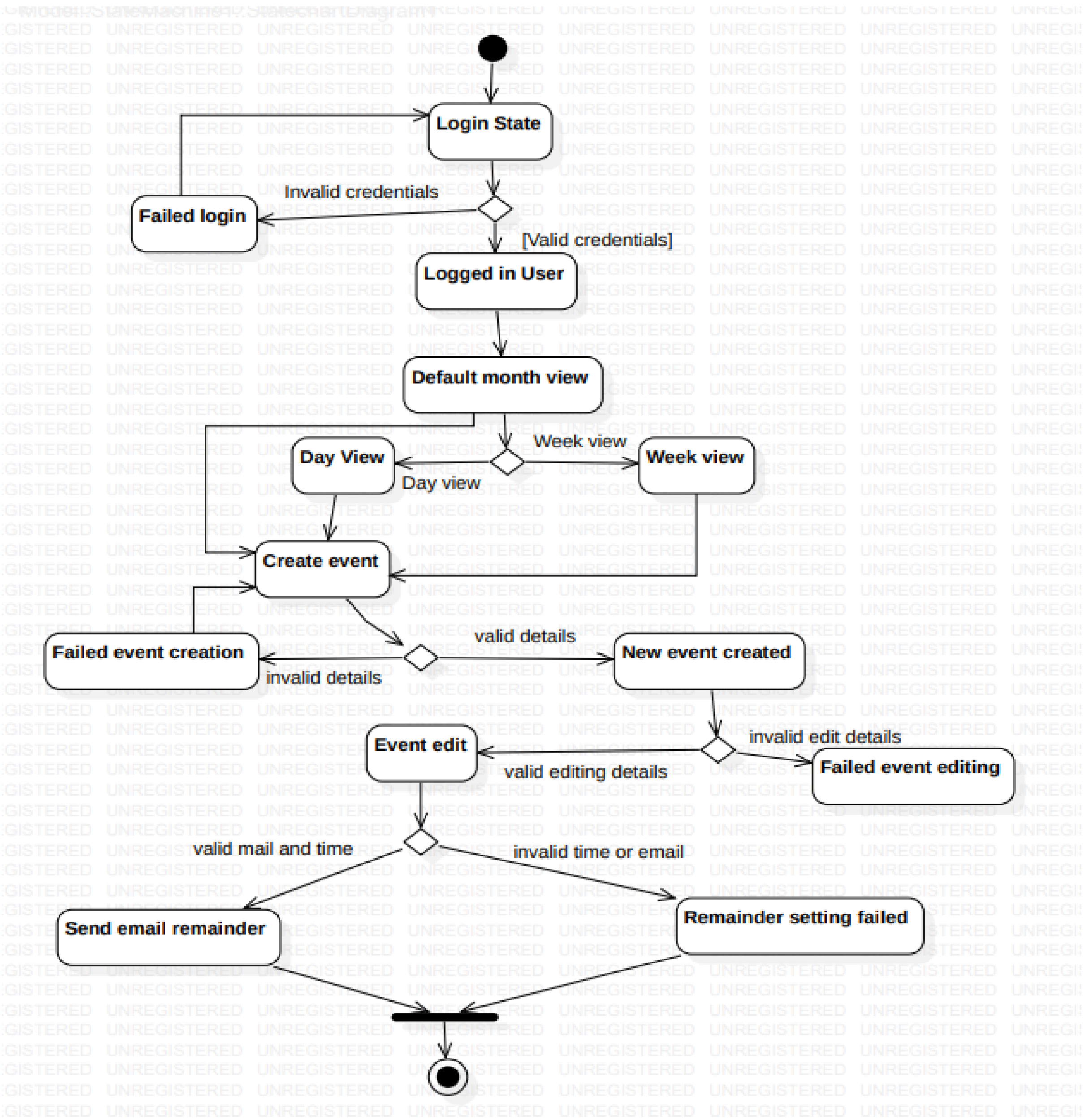


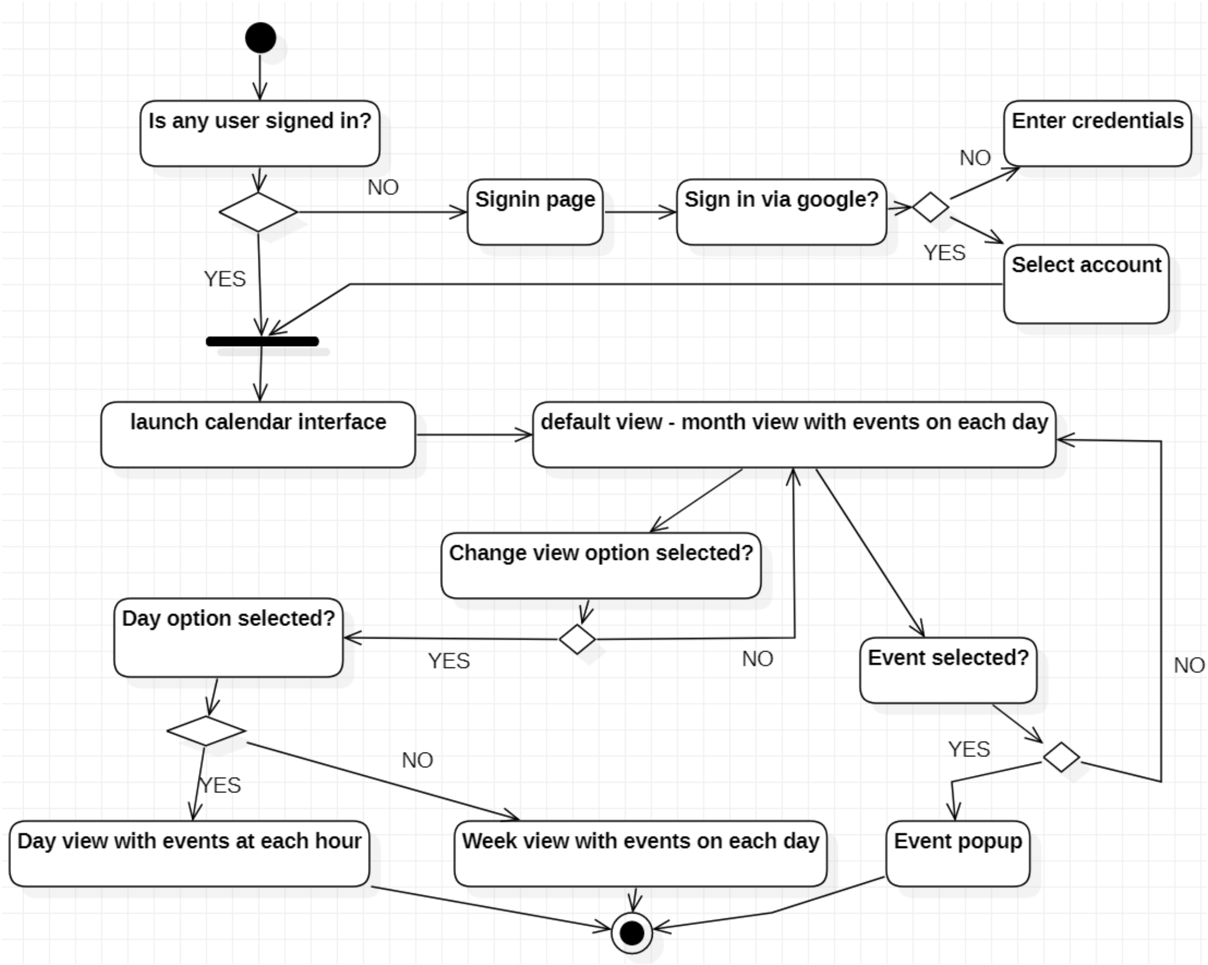
Fig 3.a

4. Activity diagrams

4.1 Show calendar based on day/week/month

The system checks if any user is signed in, if yes, launches calendar if not, goes to sign in page, checks if the sign in via google option selected, if yes displays logged in accounts else, takes the credentials entered, if authenticated successfully, launches calendar.

Default view is the month view. If change view option is selected, dropdown displayed, if day option selected, goes to day view, else if week option is selected goes to week view, else stays on same view. If date box/ event selected, event pop up displayed.



Is any user signed in?
NO
Signin page
Sign in via google?
Enter credentials
NO
YES
Select account
YES
launch calendar interface
default view - month view with events on each day
Change view option selected?
Day option selected?
YES
NO
Event selected?
NO
YES
NO
YES
Day view with events at each hour
Week view with events on each day
Event popup

Fig 4.a

4.2 Calendar entry management (reminder, event - invite people)

Once the user is logged in they will be able to access the default calendar view i.e the Month view first. They can choose to either create a new event or select an existing one. A pop up is displayed on choosing to create an event. The user can opt to add a reminder, task or an event. If event is chosen, they can add guests to whom invites would be sent. If task is chosen, the user can add a description of the task. All three types of events take date and time. On saving the event, the calendar is updated.
If the user chooses to delete an event, the calendar gets updated and the guests, if any, should be notified. If the user chooses to edit, the current details should be displayed and saved onces edited.

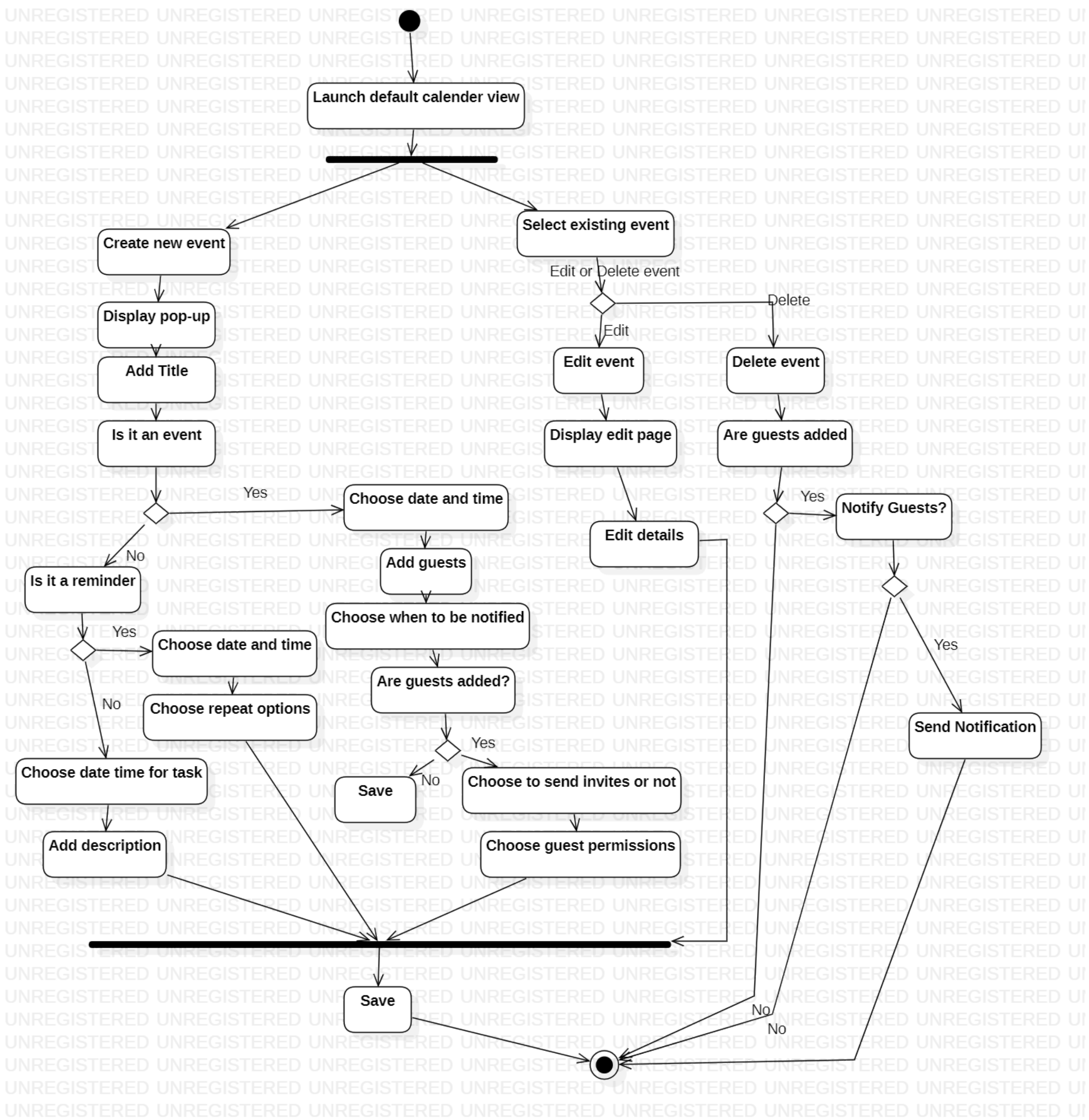


Fig 4.b

4.3 Send email reminders

When a user is logged in and is on any of the view pages of the calendar, the user can add a new event or edit an existing event. A click on a date will enable the event section for the user. The user can enter details such as title of the event, date and time of the commencement of the event and the guests to be invited.

If the details entered passes all the validation checks, then upon clicking the save button, an email invite is sent out to all the guests on the list. And in case any of the validation checks are being failed, an alert is sent. The user can modify the details and then save the event details.

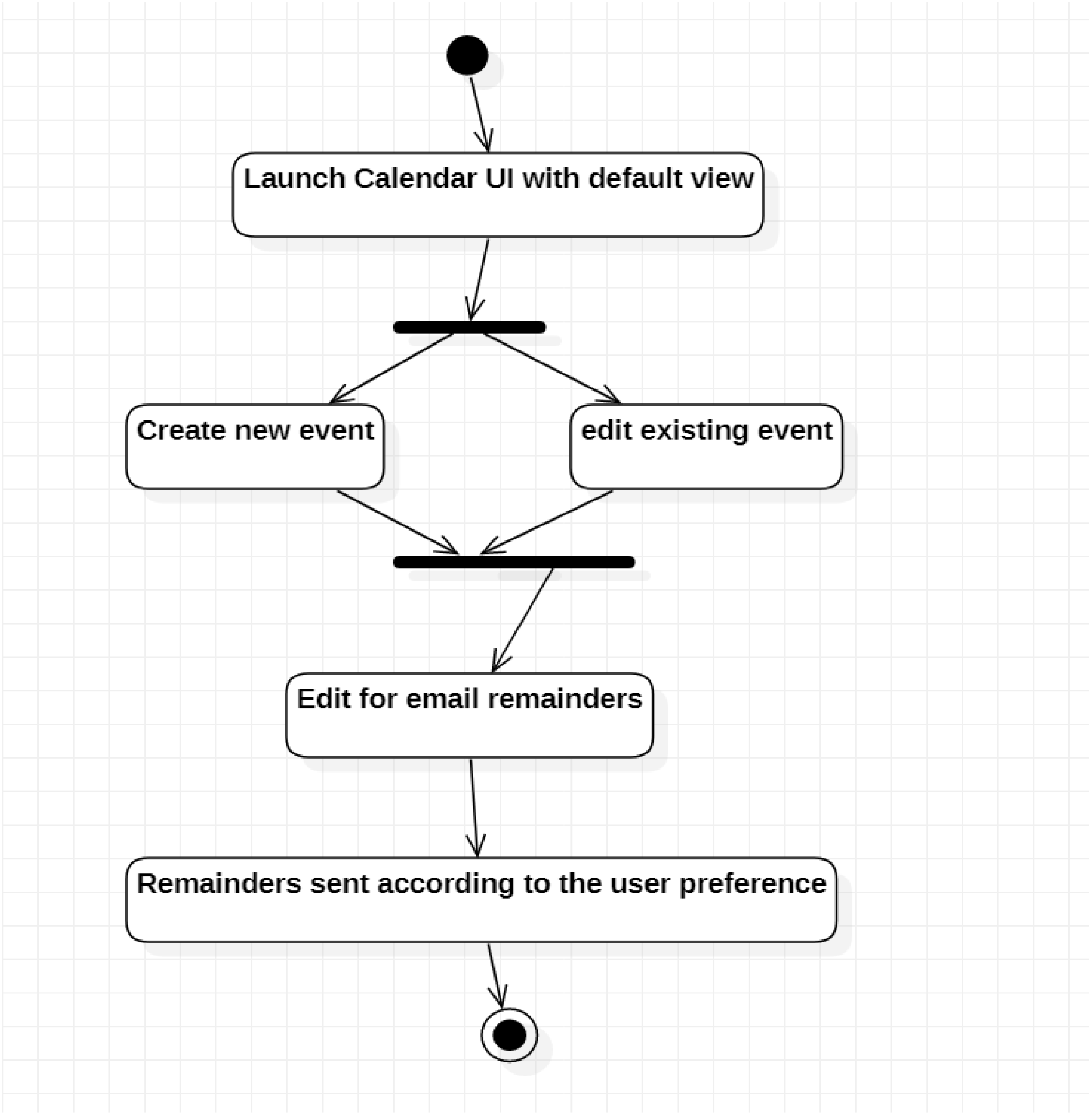

Launch Calendar UI with default view
Create new event
edit existing event
Edit for email remainders
Remainders sent according to the user preference

Fig 4.c

5. Sequence diagrams

5.1 Show calendar based on day/week/month

UI launched, reads sign in credentials, sends to server and gets it authenticated, if inavlid, displays error message and asks to sign in again, if valid, fetches month view page.

If change view option is selected, dropdown displayed, if day option selected, fetches day view page, else if week option is selected fetches week view page, else stays on same view. If date box/ event selected, event pop up displayed.

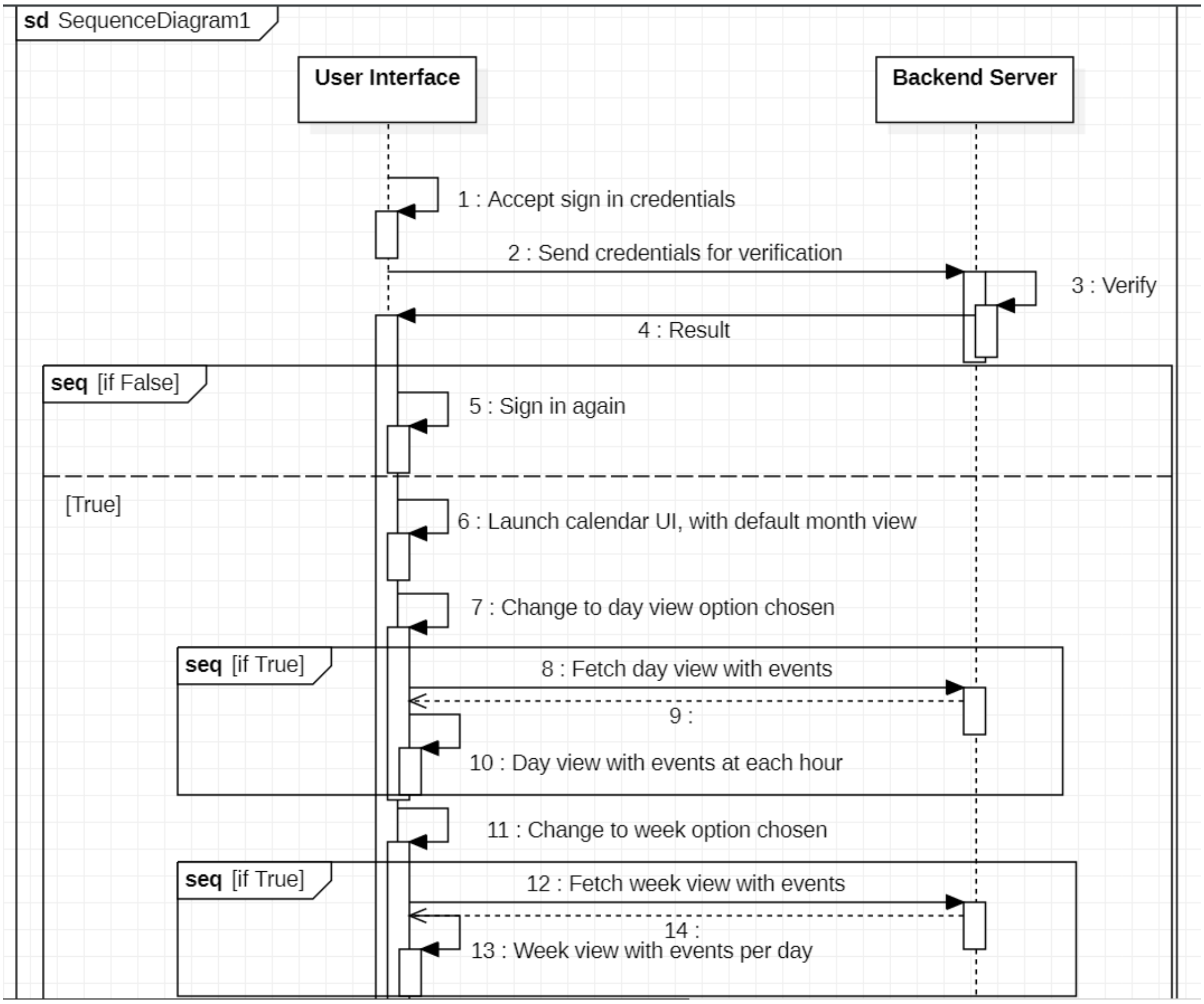

sd SequenceDiagram1
User Interface
Backend Server
1 : Accept sign in credentials
2 : Send credentials for verification
3 : Verify
4 : Result
seq [if False]
5 : Sign in again
[True]
6 : Launch calendar UI, with default month view
7 : Change to day view option chosen
seq [if True]
8 : Fetch day view with events
9 :
10 : Day view with events at each hour
11 : Change to week option chosen
seq [if True]
12 : Fetch week view with events
14 :
13 : Week view with events per day

Fig 5.a

5.2 Calendar entry management (reminder, event - invite people)

UI displays default calendar view. If the user chooses to add an event, add event details to the database. If the user chooses to edit details, fetch details from the database to display and update details in the database. If the user chooses to delete an event, remove the event rom the database

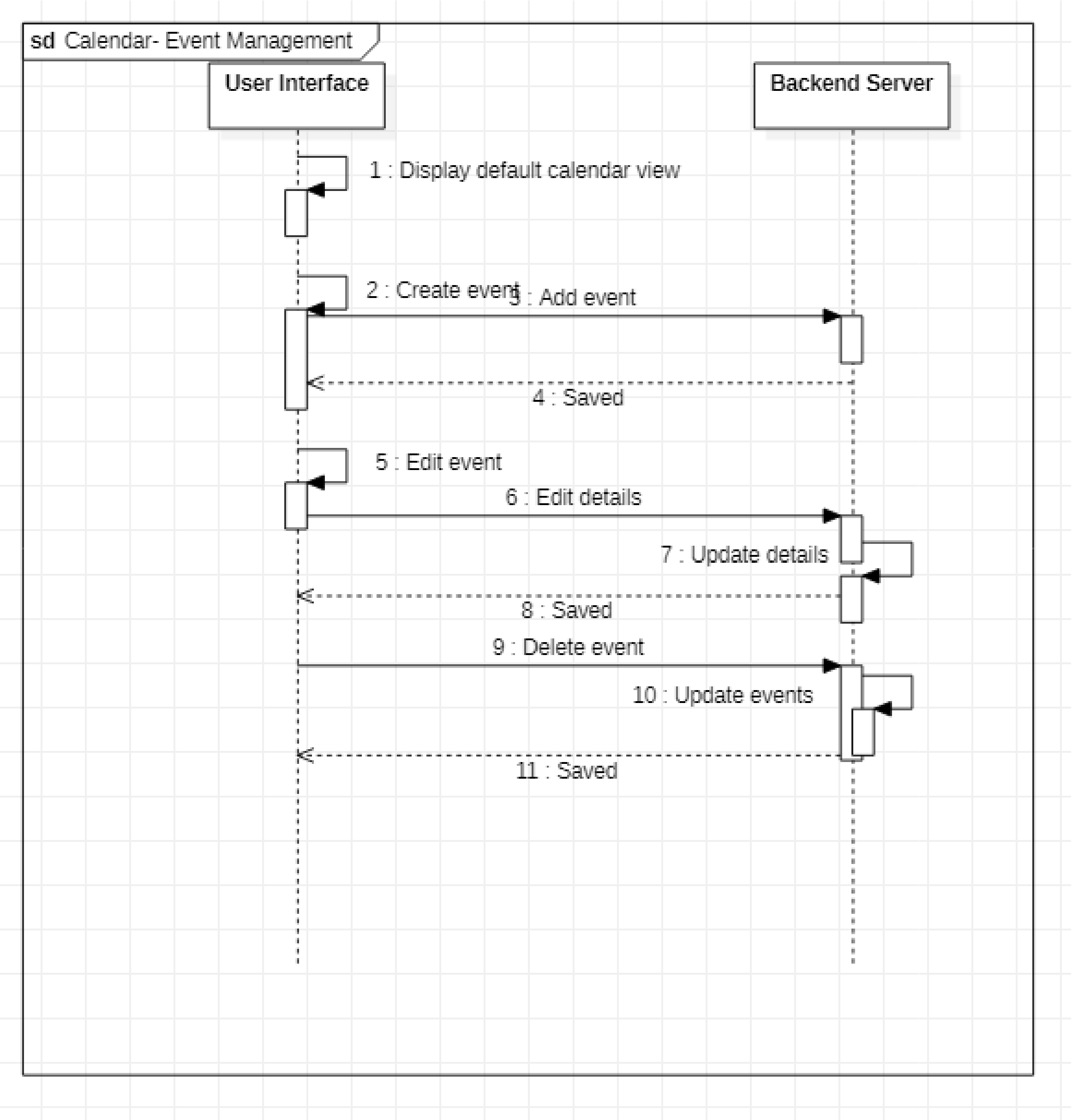


Fig 5.b

5.3 Send email reminders

The user can be on any of the view pages. Once the user clicks on any date, the options to create an event pops up. The user can enter the necessary details. The system will not throw an error if all the validation checks are passed. If not the system throws errors , and the user will have to modify the field accordingly.
Once the details are entered, the user can click on the Save button. This triggers the third party library to send out email invites to the guests mentioned on the list. After successful completion of the task, a successful message is displayed on the current screen.

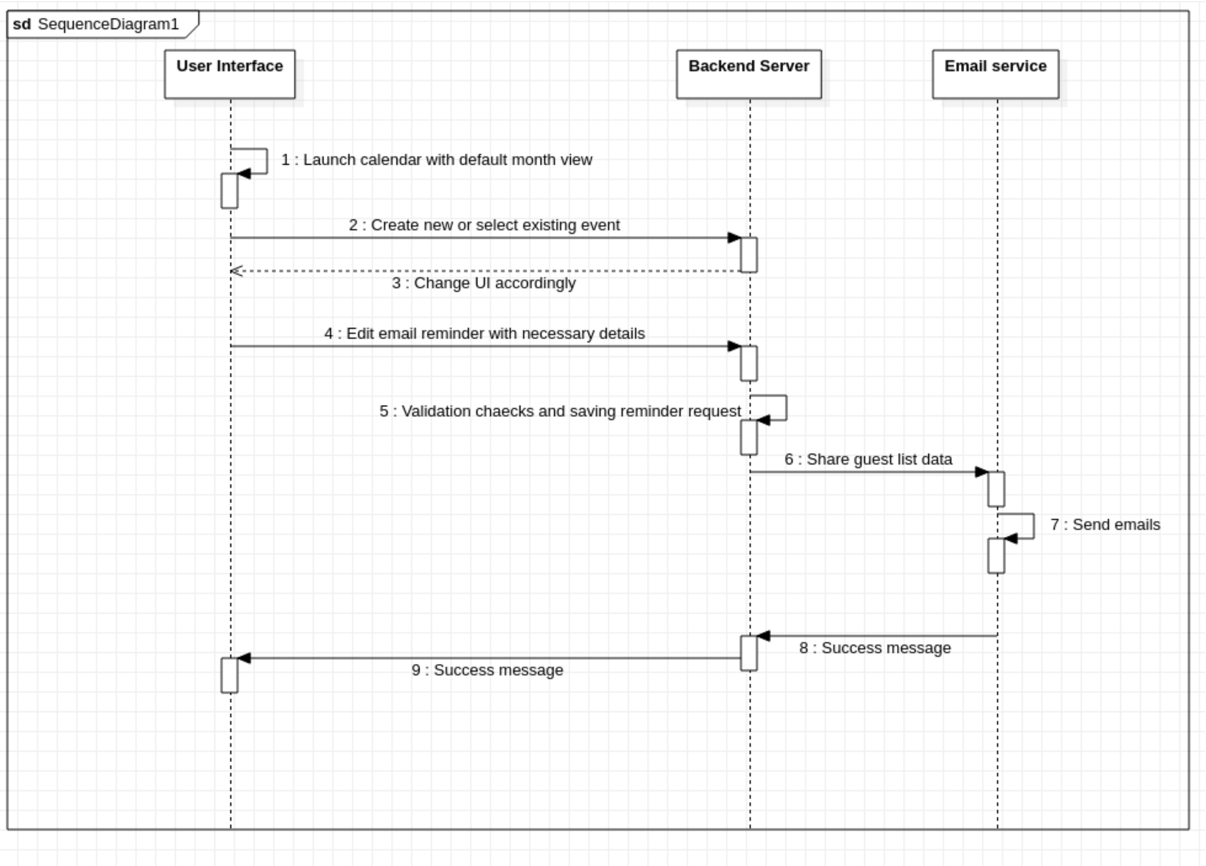

sd SequenceDiagram1
User Interface
Backend Server
Email service
1 : Launch calendar with default month view
2 : Create new or select existing event
3 : Change UI accordingly
4 : Edit email reminder with necessary details
5 : Validation chaecks and saving reminder request
6 : Share guest list data
7 : Send emails
8 : Success message
9 : Success message

Fig 5.c

V. Module Description

1. Authentication

The user is required to enter credentials - emailID and password, these are verified with the default credentials stored in the users table of the database using PHP. On successful authentication, navigates to default month view.

2.Show Calendar

A logged in user should be able to view the calendar in three different views, drop down menu available to change the views:
Month View - Displays the dates of the current/selected month. This is also the default view.
Week View - Displays the days of the current/selected week.
Day View - Displays 24hrs of the current/selected week.
Each of the views are implemented tables in HTML with CSS and JS and PHP which handles the event details display, addition to the database.

2.Calendar Entry Management

A user, once logged in should be able to add events, tasks or reminders to their calendar. They should also be able to delete or edit an existing event, task or reminder. They can also invite guests while adding an event. The database is updated every time a user adds, deletes or edits an event, reminder or task. The respective changes should also reflect on the view.
Implementation: Used PHP, HTML, JS to create three modal pop ups. One for each type of event . It is also used to store and fetch details of the events/reminders/tasks in the database.

3. Send Email Invites

A user, once on any of the view pages should be able to create events. Once the details are entered and if the details pass all the validation checks, then an email invite is sent out to all the guests mentioned by the user. This is implemented using a third part library PHPMaile which uses SMTP protocol and is linked using the PHP files.

VI. Test Cases

Unit Test Cases : Use Case 01

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UC-101	User sign in on the module.	To test the sign in functionality.	Access to chrome browser.	1. Go to the login.php page. 2. Enter the gmail id. 3. Enter the password. 4. Click on the sign in button.	Gmail id=jsivanjali@gmail.com password =abc123@345 Gmail id=fake@gmail.com password =abc_123	Successful sign in with the current month view of the calendar. Unsuccessful sign in and the user is still on the login.html. The error message given is “Invalid username or password”.	Successful sign in with the current month view of the calendar. Unsuccessful sign in and the user is still on the login.php. The error message given is “Invalid username or password”.	PASS PASS
UC-102	Sign in via google account.	To test if a user can sign in via a connected google account.	Access to chrome browser.	1. Go to the login.html page. 2. Click on the connect to sign in button. 3. Choose an account from the displayed pop up.	Choosing any connected google account.	Successful sign in with the current month view of the calendar.	Unsuccessful sign in and the user is still on the login page.	FAIL
UC-103	Changing view	Testing the change view functionality.	Should be on one of the view pages.	1. Click on the change view button. 2. Select an option from the dropdown.	Click on the ‘Week’ option.	Must open week view page.	Opens week view page. Opens day view page.	PASS PASS

					Click on the ‘Day’ option. Click on the ‘Month’ option.	Must open day view page. Must open month view page.	Opens month view page.	PASS
UC-104	Order of dates	Testing if the correct order of dates and corresponding days are displayed.	Should be on one of the view pages.	Stay on the view page.	Stay on the view page.	The correct order of dates and corresponding days are displayed.	The correct order of dates and corresponding days are displayed in all views.	PASS
UC-105	Leap year dates	Testing if the dates are modified correctly for a leap year, especially february.	Should be on one of the view pages.	Stay on the view page.	Stay on the view page.	Must display the correct order of dates in the case of leap year. February must have 29 days in whole for that year.	Doesn’t display the correct order of dates in the case of leap year. February does not have 29 days in whole for that year.	FAIL
UC-106	Next / prev navigation	Testing the navigation feature between months/ days/ weeks.	Should be on one of the view pages.	1. Click on the previous button. 1. Click on the next button.	Click on the previous button. Click on the next button.	Must display previous month/ day/ week. Must display next month/ day/ week.	Doesn’t display previous month/ day/ week. Doesn’t display next month/ day/ week.	FAIL FAIL

UC-107	Correct week/ month/ date on the top bar.	Testing if the correct week number, date and month name are displayed on week view, day view and month view page.	Should be on one of the view pages.	Stay on the view page.	Stay on the view page.	The correct week number, date and month name must be displayed on the week view, day view and month view page in the top bar.	The correct week number, date and month name displayed on the week view, day view and month view page in the top bar.	PASS

Unit Test Cases : Use Case 02

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UC-201	Task and Reminder button on event pop up	Testing if the Task and reminder pop ups are displayed.	Event pop up must be displayed	1. Click ‘Task’ button 2. Click ‘Reminder’ button	Click ‘Task’ button Click ‘Reminder’ button	Task pop up displayed on click Reminder pop up displayed on click	Task pop up displayed on click Reminder pop up displayed on click	PASS
UC-202	Event, Task, Reminder pop up Navigation	Testing the navigation between Event, Tasks and Reminders.	Any of the three pop ups must be displayed.	1. Click the ‘Event’ button from any of the other two pop ups. 2. Click the ‘Task’ button from any of the other two pop ups. 3. Click the ‘Reminder’ button from any of the other two pop ups.	Event button Task button Reminder button	The pop ups should be displayed according to the button clicked irrespective of the pop up its originally on.	The pop ups are displayed correctly according to the button that is clicked.	PASS

UC-203	Event, Task, Reminder pop up Form fields validation	Testing the functioning of the Event, Task, Reminder forms	Any of the three pop ups must be displayed	1. Enter Title 2. Select Date 3. Select Time 4. Add Guests (in Event popup) 5. Add Description (in Task popup)	TestTitle 12-04-2021 9:00:00 Guest@gmail.com TestTitle123chars678901923 12-04-2021 9:00:00 Guest@gmail.com	The fields should be able to take the Test Data inputs. Should restrict title field to 15 character limit	The form fields takes the Test Data inputs Doesn't let the user input more than 15 characters.	PASS PASS
UC-204	Setting an event in the past date.	Testing if an event can be set in the past date.	The user must be logged in and be on one of the view pages.	1. Click on the event. 2. Set the date to a past date. 3. Enter other details. 4. Click on the save button.	Click on save button Date: 22/02/2021	The UI takes in the date without any validation, mail also sent like google calendar.	The UI takes in the date without any validation, mail also sent.	PASS
UC-205	Appropriate time settings.	Testing for the display of appropriate time. Eg: the hours must be between 00 to 23 and the minutes must be between 0 and 59.	The user must be logged in and be on one of the view pages.	1. Click on a date. 2. Click on the event. 3. Click on the time field.	Click on the time field. Hours: 05 Minutes: 35 Hours: 28 Minutes: 90	The UI must show appropriate time as entered. Must automatically change hours to 23 and minutes to 59.	The UI shows appropriate time. Automatically changes hour to 23 and minutes to 59.	PASS PASS
UC-206	Add Guests	Testing if the 'Add' button on the Event popup	Event popup must be displayed	1. Click the 'Add' button under the 'Add Guests'	Click the 'Add' button.	Should add a new empty text field to enter	Adds new empty text field to enter Guest Email ID.	PASS

		allows users to Add more Guests.		label in the Event form.		Guest Email ID.		
UC-207	Delete Event/Task/Reminder	Testing if the Event/Task/Reminder details gets deleted in the database as well as the on the Month,Week and Day views.	An Event/Task/Reminder must already exist in the database and on the UI	1. Click on an Event/Task/Reminder in Month/Week/Day view. 2. Click on 'Delete' Option	Click on an Event/Task/Reminder in Month/Week/Day view. Click on 'Delete' Option	Details of the selected event should be displayed along with the option to 'Edit' or 'Delete'. Details of selected events should be deleted from the database. Event/Task/Reminder should be deleted on Month, Week and Day views.	Event details not displayed on view.	FAIL
UC-208	Edit Event/Task/Reminder	Testing if the Event/Task/Reminder details gets updated in the database as well as the on the Month,Week and Day views upon 'Edit'	An Event/Task/Reminder must already exist in the database and on the UI	1. Click on an Event/Task/Reminder in Month/Week/Day view. 2. Click on 'Edit' Option 3. Edit details of the event 3. Click 'Save' button	Click on an Event/Task/Reminder in Month/Week/Day view. Click on 'Edit' Option Change Title,Date, TIme Click 'Save' button	Details of the selected event should be displayed along with the option to 'Edit' or 'Delete'. Details of the event should be updated in the database. Events should be updated in Month, Week and Day views.	Event details not displayed on view.	PASS

UC-209	Event, Task and Reminder popup exit condition - click anywhere	Testing when the event, Task and Reminder popup should be closed.	The pop ups must already be displayed.	1. Click anywhere on the UI but not on the popup.	Click anywhere else on the UI other than the popup.	The popup currently displayed should close and return to the view it is currently on.	Popup is closed and returns to view	PASS
UC-210	Event, Task and Reminder popup exit condition - click 'save'.	Testing when the event, Task and Reminder popup should be closed.	The pop ups must already be displayed..	1. Click the 'Save' button.	Click 'Save' button with fields filled. Click the 'Save' button with empty fields.	Adds event to database and closes popup. Returns to view. Closes popup. Returns to view.	Adds event to database and closes popup. Returns to view. Closes popup. Returns to view.	PASS PASS

Unit Test Cases : Use Case 03

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
UC-301	Send a mail invite	Testing if the mail gets sent provided all the details are entered.	The user must be logged in and be on one of the view pages and pop up must be opened..	1. Click on a date. 2. Select the event button. 3. Fill in the necessary details. 4. Click on the save button.	TestTitle 12-04-2021 9:00:00 svanjali128@gmail.com TestTitle 12-04-2021 9:00:00 fake@gmail.com	Successful mail sending if all the details entered are correct. Unsuccessful mail sending if one or more details are wrong/empty.	Successful mail sending if all the details entered are correct. Unsuccessful mail sending if one or more details are wrong/empty.	PASS PASS
UC-302	Accurate Details in the email	Correct event details must be specified	The user must be logged in and be on one of	1. Click on a date. 2. Select the event button.	TestTitle 22-04-2021 3:30	Subject : TestTitle	Mail received. Subject :	PASS

		in the mail.	the view pages and pop up must be opened.	3. Fill in the necessary details. 4. Click on the save button.	svanjali128@gmail.com	Body : This is to notify that you have a meeting on 2021-04-22 at 3:30 regarding TestTitle.	TestTitle Body : This is to notify that you have a meeting on 2021-04-22 at 3:30 regarding TestTitle.	
UC-303	Sending mail without any guests	Testing for errors when no guest is added to the event pop-up.	The user must be logged in and be on one of the view pages.	1. Click on a date. 2. Select the event button. 3. Fill in the details, except the add guest field. 4. Click on the save button	Click on the save button without any mailID.	The mail is not sent as no guest mailID is entered. Gives an “Unsuccessful” message on the view page.	The mail is not sent as no guest mailID is entered. Gives an “Unsuccessful” message on the view page.	PASS
UC-304	Multiple recipients	Testing if emails are sent to each recipient in case of more than one emails in the guest list.	The user must be logged in and be on one of the view pages. The event pop up must be open.	1. Click on a date. 2. Select event button. 3. Fill in the details. And add guests. 4. Click ‘Save’	svanjali128@gmail.com ooad.project.mail@gmail.com sreepranaviganugapati@gmail.com	Emails must be sent to each recipient in the guest list.	Emails sent to each recipient in the guest list.	PASS
UC-305	Sending mail without using PHPMailer (3rd party code library).	Testing if the mail gets sent without using PHPMailer.	The PHPMailer code must be eliminated from the server.	1. Click on the event. 2. Enter all the details including email id. 3. Click on the save button.	Click on the save button.	The UI must show an error for the PHPMailer files not found.	The UI shows an error for the PHPMailer files not found.	PASS

Integration Test Cases

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
IC-001	Navigate to default view on successful authentication.	Testing if month view appears on successful sign in.	Should be successfully signed in.	1. Click on the sign in button.	1. Click on the sign in button.	Must open the current month view.	Opens the current month view.	PASS
IC-002	Event pop up.	Testing the event pop up feature.	Should be on one of the view pages.	1. Stay on the view page. 2. Click on one of the numbered boxes/ buttons i.e date.	Click on one of the numbered boxes/ buttons i.e date.	New event pop up must be displayed with empty fields.	New event pop up displayed with empty fields.	PASS
IC-003	Event/ task/ reminder bars.	Testing if the correct event/ task/ reminder bars are displayed on each box.	Should be on one of the view pages.	Stay on the view page.	Stay on the view page.	The correct event/ task/ reminder bars are displayed on each box with correct color codes.	Not displayed.	PASS
IC-004	Update Event/Task/Reminder in the database	Testing if the Event/Task/Reminder details gets updated in the database upon clicking the 'Save' button.	Details must be filled in the form of respective pop up.	1. Fill details of the form such as Title, Date, Time in the pop ups. 2. Click the 'Save' button.	Trial event 13-04-2021 12:00:00	Event details should be reflected in the database.	Event details are added to the database.	PASS

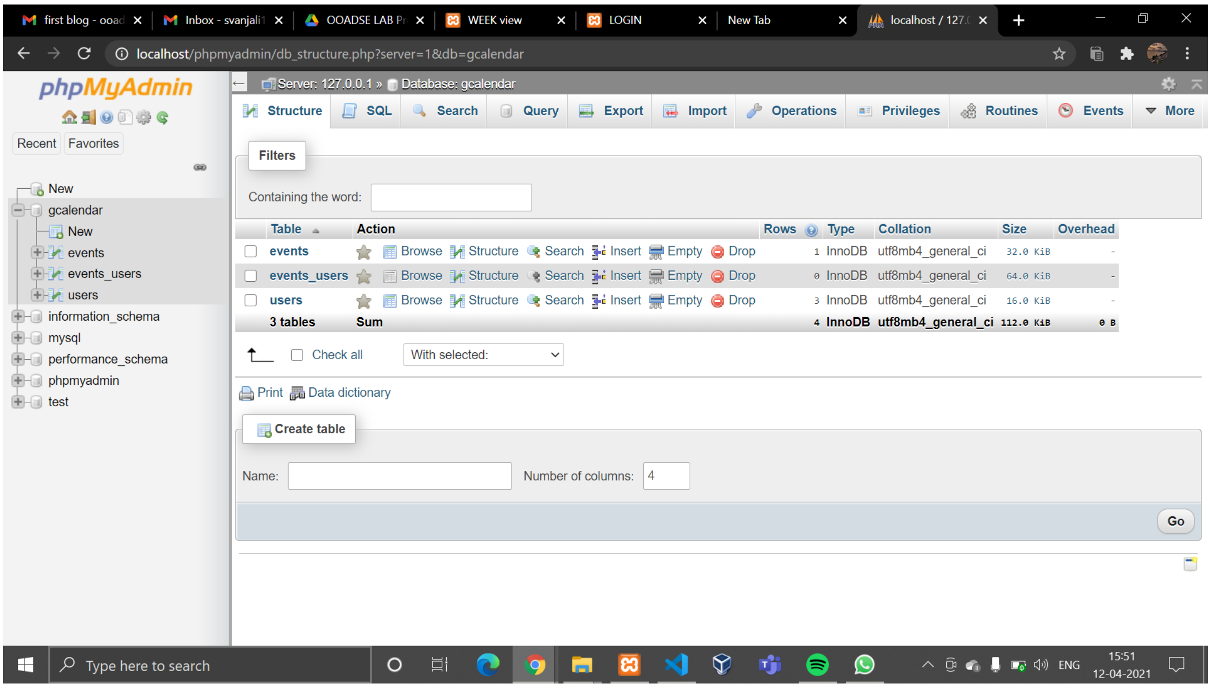
IC-005	Auto increment ID	Auto increment ID in the database every time an event is added.	Details must be filled in the form of respective pop up.	1. Fill details of the form such as Title, Date, Time in the pop ups. 2. Click the ‘Save’ button.	Trial event 13-04-2021 12:00:00	Auto incremented ID in the database.	Auto incremented ID in the database.	PASS
IC-006	Mail confirmation message.	Display mail confirmation message on view pages.	1. The user must be logged in. 2. Must be on one of the view pages. 3. A pop up must be open.	1. Fill all the details in the pop up. 2. Enter mailID. 3. Click on the save button.	TestTitle 12-04-2021 9:00:00 svanjali128@gmail.com TestTitle 12-04-2021 9:00:00 fake@gmail.com	“Mail sent successfully” message appears on the view pages. “Unsuccessful” message appears on the view pages.	“Mail sent successfully” message appears on the view pages. “Unsuccessful” message appears on the view pages.	PASS PASS
IC-007	Add user details to database	Add user details to database	Add details in the database	Add details in the database	svanjali128@gmail.com abc123@345	Success message	Success message	PASS
IC-008	Email validation	Test to check if the incorrect email ids are allowed,	1. The user must be logged in. 2. Must be on one of the view pages. 3. A pop up must be open.	1. Fill all the details in the pop up. 2. Enter mailID. 3. Click on the save button.	GmailID: ooad.project.mail@gmail.com GmailID:Dgkahjfjalkjfkaj	The UI accepts any mailID that follows the validation check. The UI shows a warning text “Please include @ , the mailID has a missing @”	The UI accepts any mailID that follows the validation check. The UI shows a warning text “Please include @ , the mailID has a missing @”	PASS PASS

System Test Cases

Test Case ID	Name of Module	Test case description	Pre-conditions	Test Steps	Test data	Expected Results	Actual Result	Test Result
SC-001	Consistency throughout views.	Testing if upon adding an Event, Task, Reminder, all three views gets updated.	An Event/Task/Reminder must exist in the database.	1. View Month View (Month = Month the event in added in) 2. Change to Week View (Week = Week the event is added in) 3. Change to Day View (Day = the date the event is added in) 4.View the Time slot the event is added in.	April View 3rd week View 13th View at 12	In all 3 views(Month ,Day,Week) the event details should be displayed.	Event details are not displayed.	FAIL

VII. Screenshots of Output

The structure of the database used for Gcalendar application:



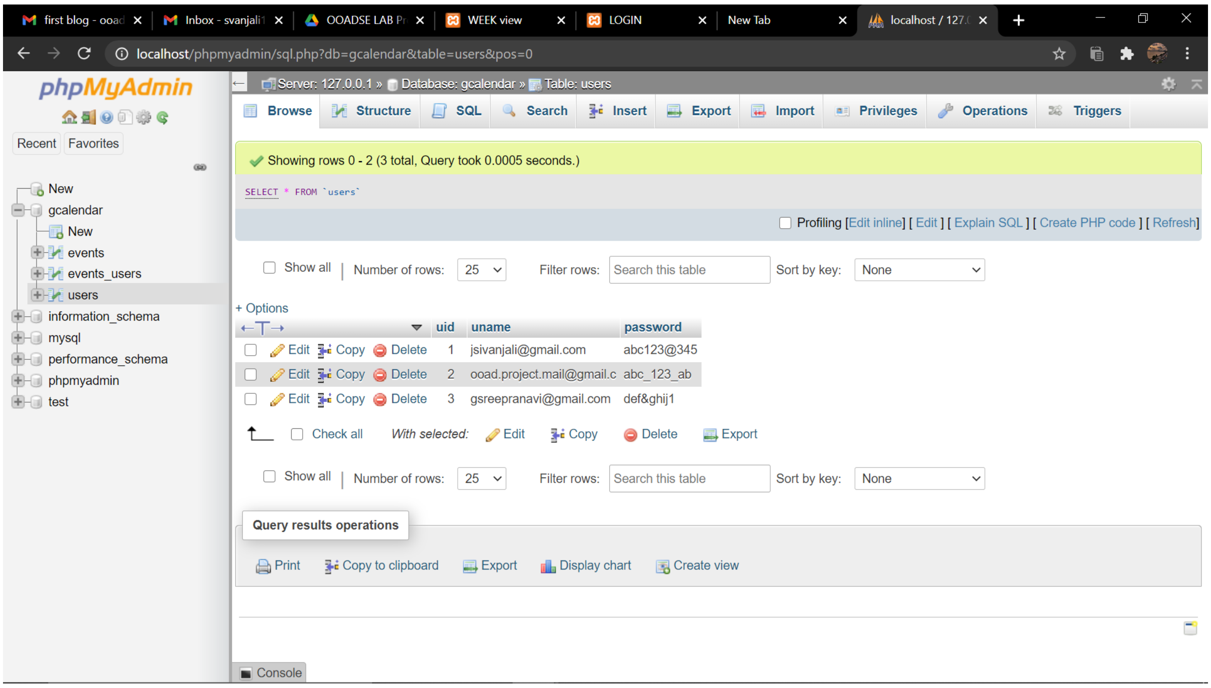
localhost/phpmyadmin/db_structure.php?server=1&db=gcalendar

Server: 127.0.0.1 » Database: gcalendar

	Table	Action	Rows	Type	Collation	Size	Overhead
□	events	Browse Structure Search Insert Empty Drop	1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
□	events_users	Browse Structure Search Insert Empty Drop	0	InnoDB	utf8mb4_general_ci	64.0 KiB	-
□	users	Browse Structure Search Insert Empty Drop	3	InnoDB	utf8mb4_general_ci	16.0 KiB	-
	3 tables	Sum	4	InnoDB	utf8mb4_general_ci	112.0 KiB	0 B

Name: Number of columns: 4

The user’s table in the database:



localhost/phpmyadmin/sql.php?db=gcalendar&table=users&pos=0

Server: 127.0.0.1 » Database: gcalendar » Table: users

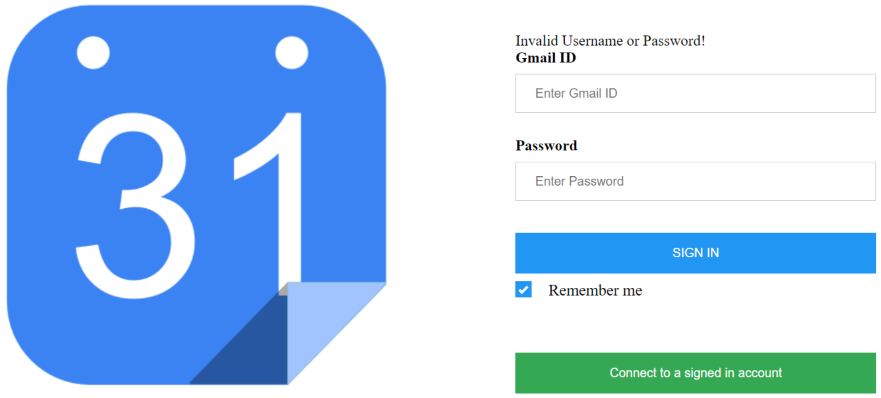
✓ Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

```
SELECT * FROM `users`
```

	uid	uname	password
□ Edit Copy Delete	1	jsivanjali@gmail.com	abc123@345
□ Edit Copy Delete	2	ooad.project.mail@gmail.c	abc_123_ab
□ Edit Copy Delete	3	gsreepranavi@gmail.com	def&ghij1

Number of rows: 25 Filter rows: Search this table Sort by key: None

The login page on entering invalid credentials:



Invalid Username or Password!

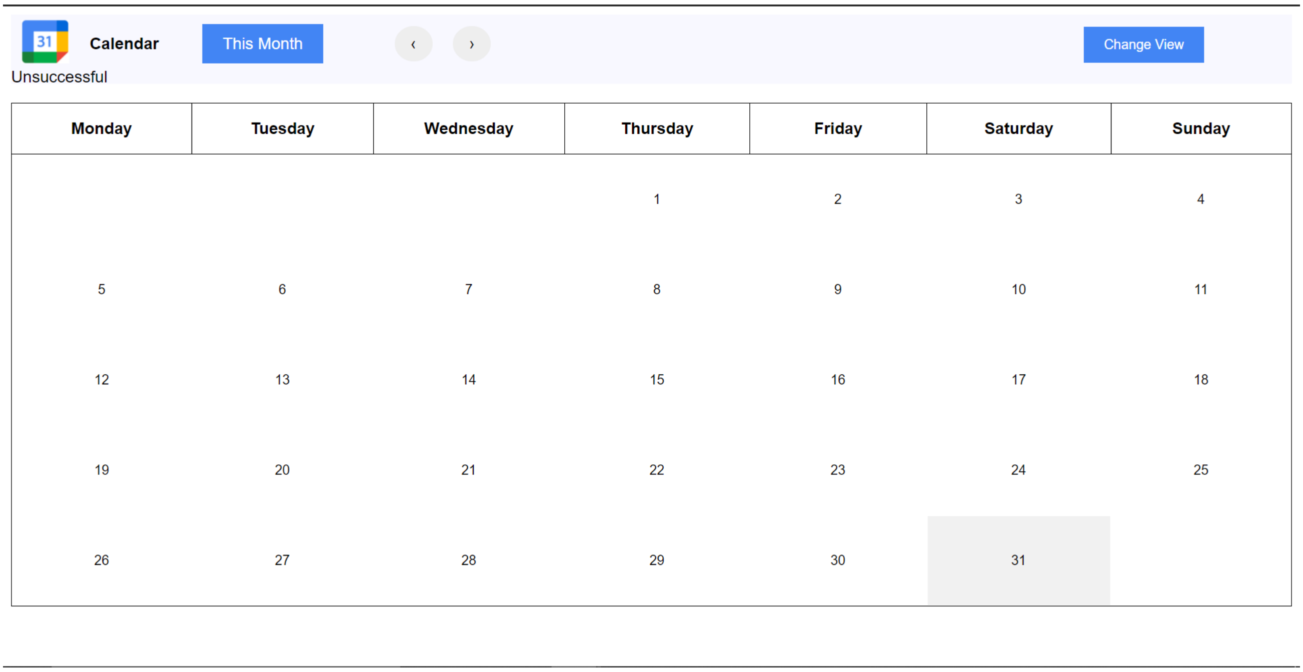
Gmail ID

Enter Gmail ID

Password

Enter Password

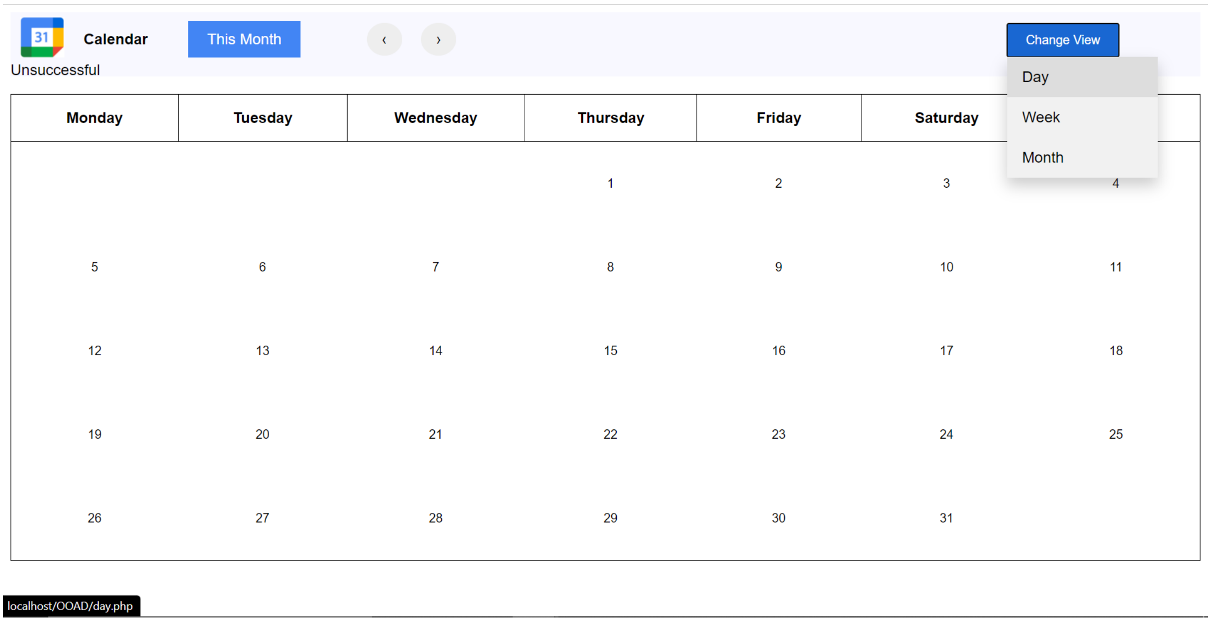
The default month view:



Unsuccessful

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

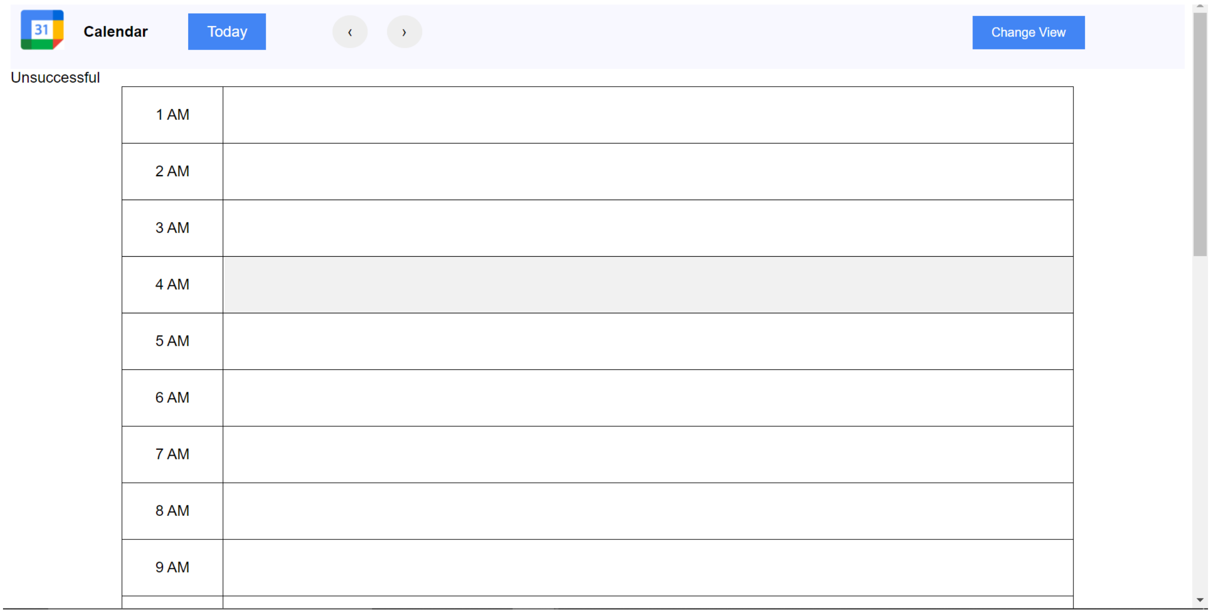
The dropdown to change the views:



Unsuccessful

Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30	31	

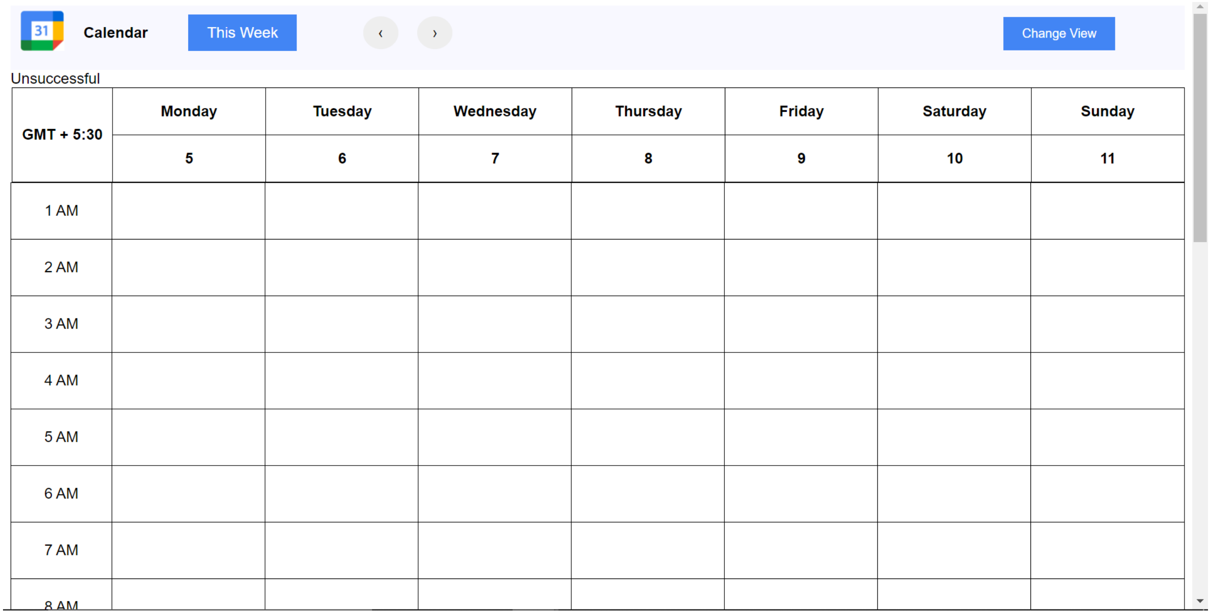
The day view of the Calendar application:



Unsuccessful

1 AM	
2 AM	
3 AM	
4 AM	
5 AM	
6 AM	
7 AM	
8 AM	
9 AM	

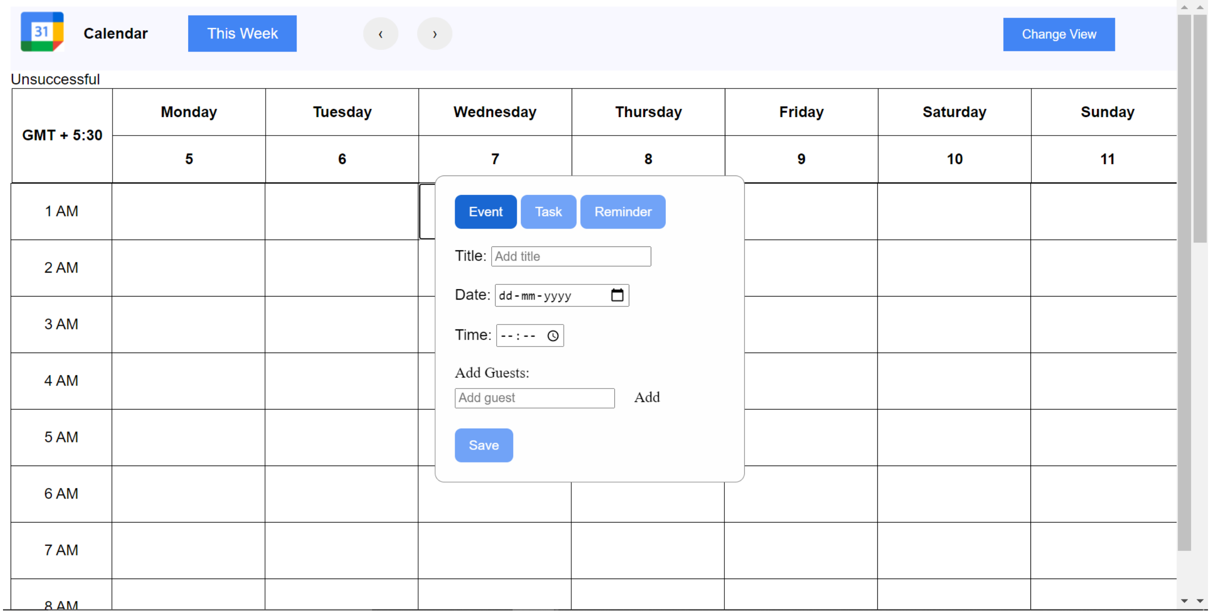
The week view of the Calendar application:



Unsuccessful

GMT + 5:30	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	5	6	7	8	9	10	11
1 AM							
2 AM							
3 AM							
4 AM							
5 AM							
6 AM							
7 AM							
8 AM							

The event pop-up:



Unsuccessful

GMT + 5:30	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	5	6	7	8	9	10	11
1 AM							
2 AM							
3 AM							
4 AM							
5 AM							
6 AM							
7 AM							
8 AM							

Title: Add title

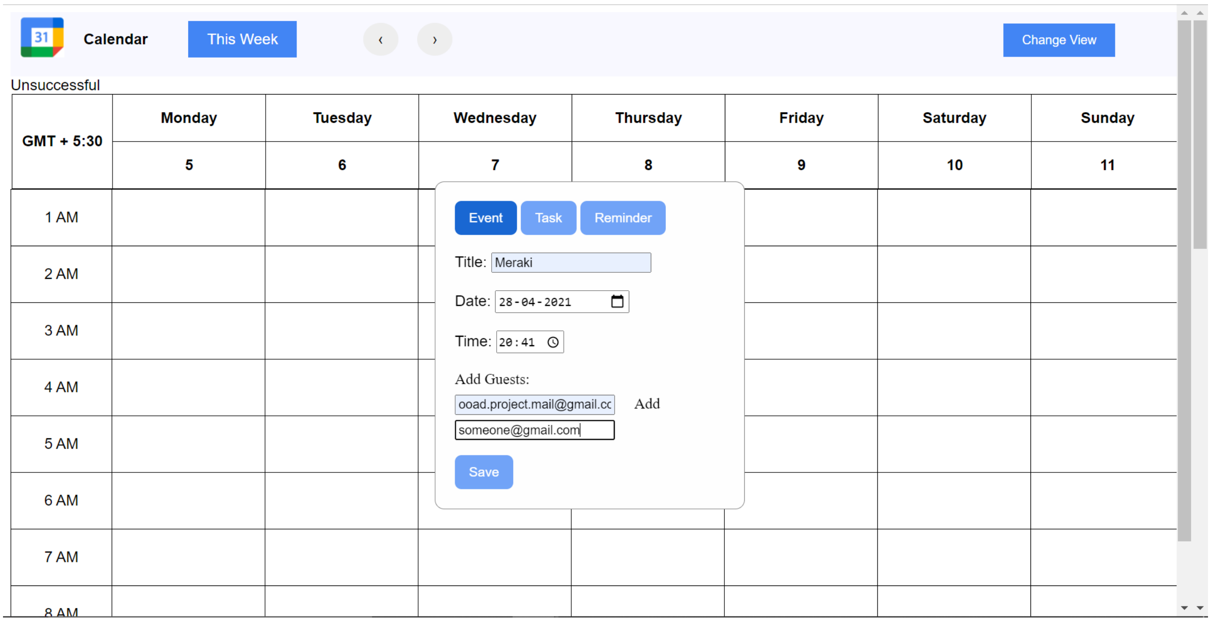
Date: dd-mm-yyyy

Time: --:--

Add Guests:

Add guest

The event pop-up with valid details for sending email invites:



Unsuccessful

GMT + 5:30	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
	5	6	7	8	9	10	11
1 AM							
2 AM							
3 AM							
4 AM							
5 AM							
6 AM							
7 AM							
8 AM							

Title: Meraki

Date: 28-04-2021

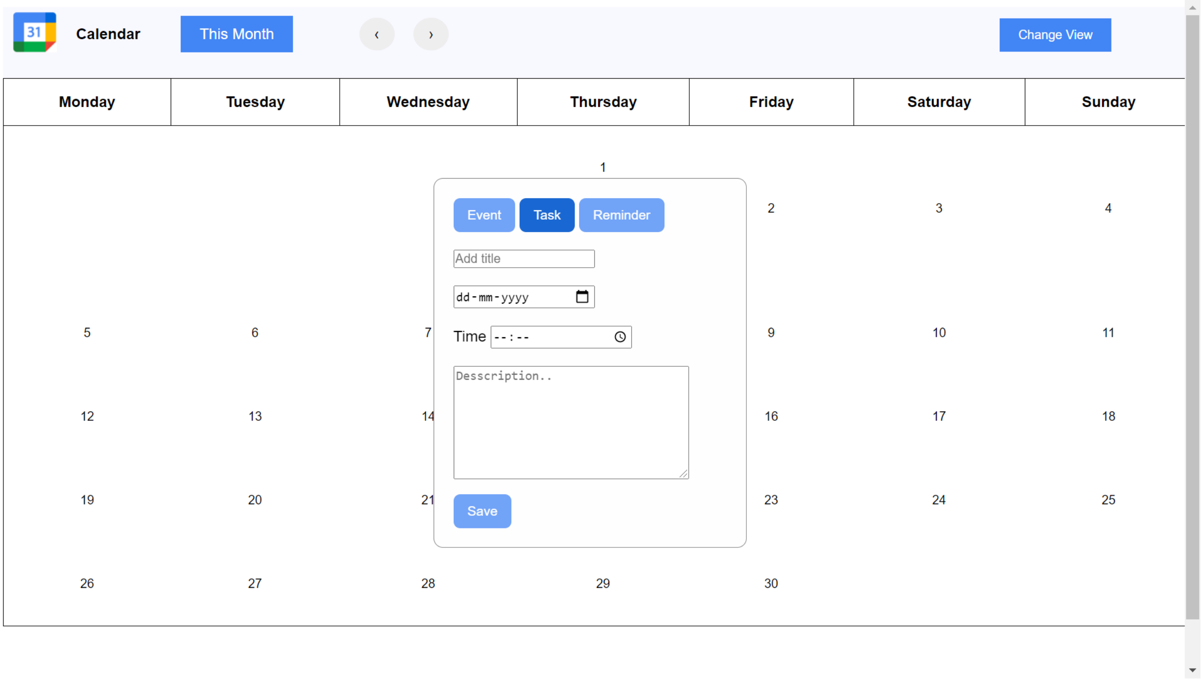
Time: 20:41

Add Guests:

ooad.project.mail@gmail.cc

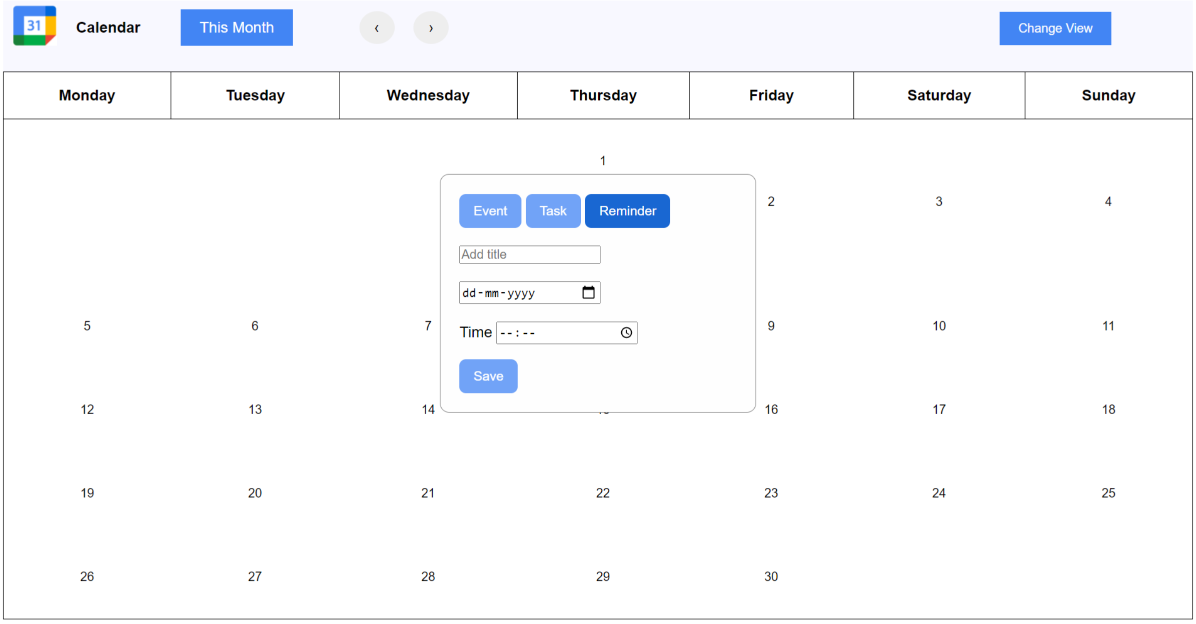
someone@gmail.com

The task pop up:



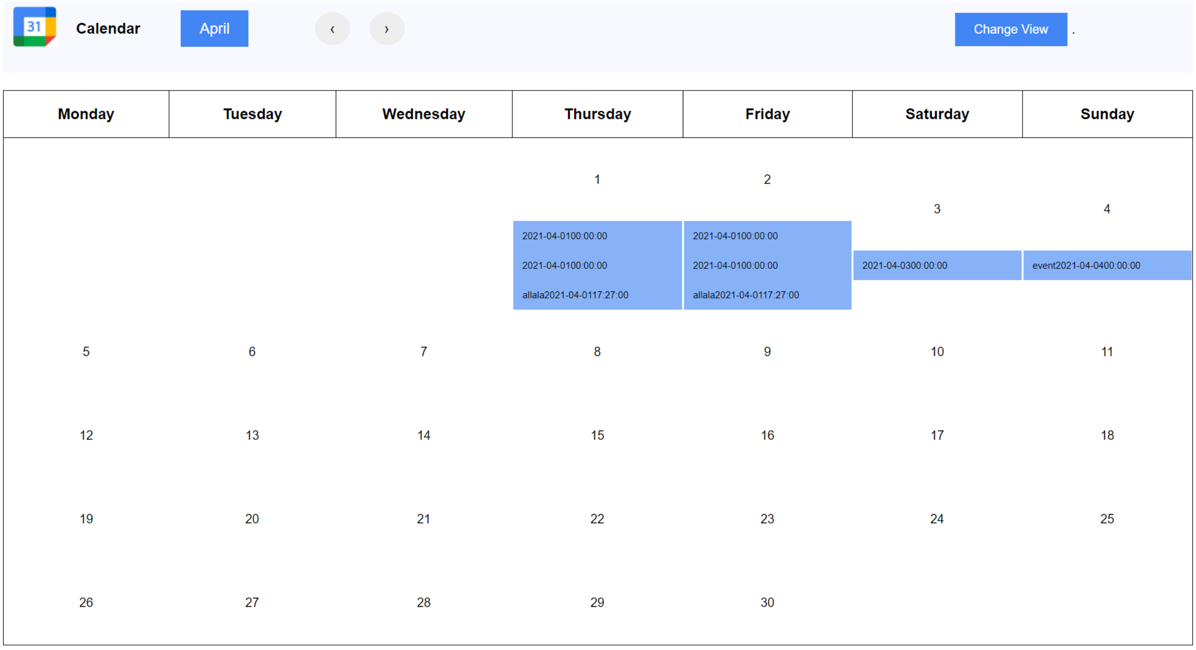
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
			1	2	3	4
5	6	7		9	10	11
12	13	14		16	17	18
19	20	21		23	24	25
26	27	28	29	30		

The reminder pop up:



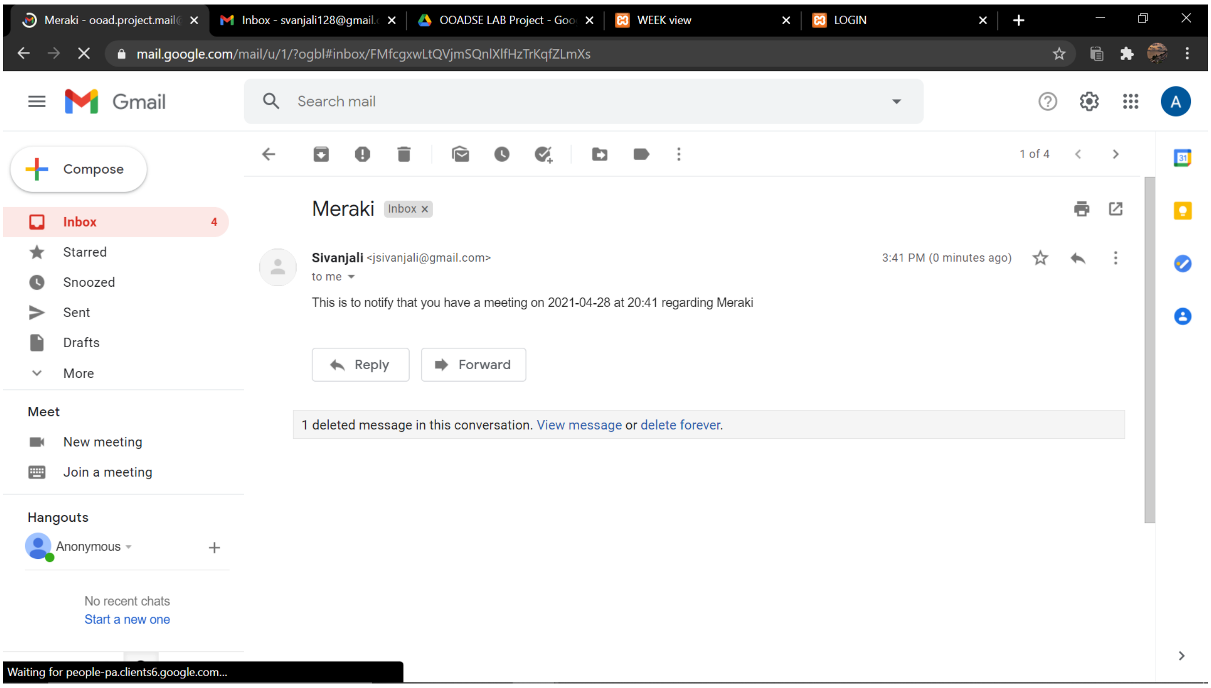
Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
			1	2	3	4
5	6	7		9	10	11
12	13	14		16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

Added events updated on view:



Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
			1 2021-04-0100:00:00 2021-04-0100:00:00 allala2021-04-0117:27:00	2 2021-04-0100:00:00 2021-04-0100:00:00 allala2021-04-0117:27:00	3 2021-04-0300:00:00	4 event2021-04-0400:00:00
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

The email invite on the inbox of the guest:



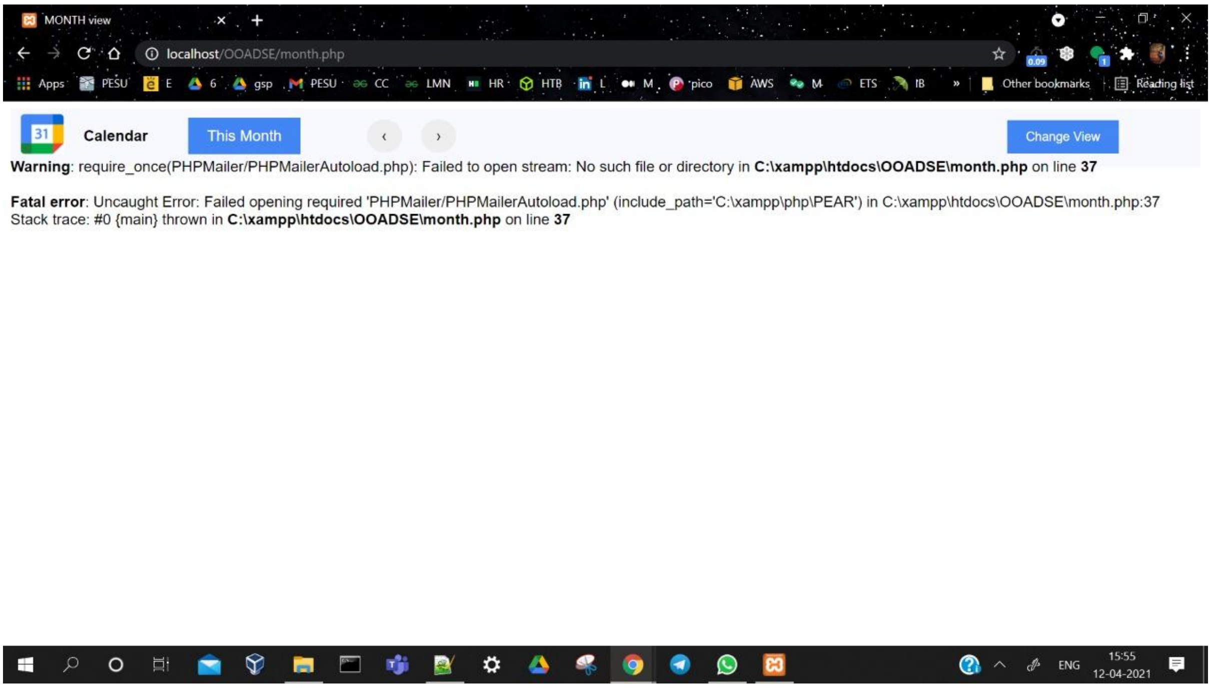
Meraki Inbox ×

Sivanjali <jsivanjali@gmail.com>
to me
3:41 PM (0 minutes ago)

This is to notify that you have a meeting on 2021-04-28 at 20:41 regarding Meraki

1 deleted message in this conversation. View message or delete forever.

The calendar application giving out error if PHPMailer (3rd party library) is not used:



```
Warning: require_once(PHPMailer/PHPMailerAutoload.php): Failed to open stream: No such file or directory in C:\xampp\htdocs\OOADSE\month.php on line 37

Fatal error: Uncaught Error: Failed opening required 'PHPMailer/PHPMailerAutoload.php' (include_path='C:\xampp\php\PEAR') in C:\xampp\htdocs\OOADSE\month.php:37
Stack trace: #0 {main} thrown in C:\xampp\htdocs\OOADSE\month.php on line 37
```

****