

# TECHNICAL MANUAL

The process model used while developing the project is “Waterfall Model”

## User Interface Design

The user interface is designed with the rules of usability in mind. Examples of these include simple instructions, allowing for ease of navigations, and grouping of similar instructions/buttons together.

## Types of Users

Users are divided on basis of their position in the event management system. There are five different positions in event management system .Admin is the one who is above all the other managers. Event manger concerns himself with managing events . Resource manager scales the resources and venues on basis of availability. Financial Manager manages the financial resources. Volunteer just gets the instructions from one or more of the above mentioned people .

## Functions Implemented and forms used

On the launch of this software a login page would be displayed, one has to give valid credentials (username and password) to login. New users of this software has to sign up first. In case one forgets his/her password they can regain them by clicking on the “Forgot Password?” button. There is also a search option which searches by event.

Once valid username and password are entered, it will be redirected to a new form based on the position of user. There also few common forms for all type of users.

Event Manager: the form corresponding to event manager is “Form\_Event\_Manager”

One can see buttons for adding event, viewing events, sending message, viewing inbox, logout. On the click of “Add Event” button it is redirected to Add\_Event form where the event manager has to add the details of an event .

Details requested in this form are as follows:

### Title of the event:

A text box is provided for this column where only char variables are allowed to enter the title of the event. Date and Time are two different columns which are individually implemented using a date and time picker. Duration has to be mentioned in terms of minutes in the corresponding text box. Venues, the user has to make venue requests( in his preference order ) for the event. For this we have used combo boxes one for each which takes input from database table venue\_list. Input to the combo box has been implemented with function ListFill(). We have taken that once a venue is selected as one of the preferences it is not available for another selection. Description, is a text box which is to be not more than 500 characters.

Form2: This form is opened when a new user wants to register himself for the system. This form contains five text fields asking for the details of the user. The user gets registered on the click of button "Register". Details don't get saved if form is not completely filled or the domain in the email field list is not proper.

### Admin

- **Form\_admin Form** : This form at the top contains a sliding images in a picture box. This has a logout button which directs to the login page. There three buttons namely View Finalized Events, View Resources, and View Financial Database which on click displays the details of finalized events, available resources, and financial database in the datagrid present in this form. A button named "View Event Request" is present which on click is directed to the form "Event\_Request\_admin". Another button named " View Volunteer Data" which redirects to "Form\_Assign\_Volunteer".
- **Event\_Request\_admin**: As soon as this form is loaded both the combo box and the list view in the form are filled with data. This list box is connected with the database table financerequests showing the complete detail of the event. The combobox is also connected to the same database table listing the event names. There are two buttons namely Approve and Reject which on click makes a change in the remaining budget.
- **Assign Volunteer**: in this form there is a listbox in which usernames of all the volunteers would be displayed from which one can be selected. On making a selection the username
- **INBOX form**: This form contains a listView showing the messages received by them. This is done by connecting to database.
- **Message form**: This is the form used to send messages to others users but with a restriction of maximum 3 recipients .This form contains few labels along with textboxes. It contains two listviews displaying the username and positions of different users. By looking at this, one can choose the users to whom they want to send the message. The selected usernames has to be entered in the textboxes.
- **Change password**: This form is used in case any user wants to change their password. This requires the following details to be entered username, email id, new password. On click of the button "Reset" the software checks for a match in username and email id, and then it updates the password with the text entered in textbox corresponding to new password.
- **Search**: In this form the user can search among evens on basis of any substring from the table containing all the details of the event.

### Event Manager

- **Add Event**: Fills in all the details in the corresponding textboxes which are saved into the database1.table2.
- **View event**: As soon as this form is loaded the datagrid present in this form is filled with details of all the events from the database.events.

- **Compose:** This button redirects to the form message\_form. This is the form used to send messages to others users but with a restriction of maximum 3 recipients .This form contains few labels along with textboxes. It contains two listviews displaying the username and positions of different users. By looking at this, one can choose the users to whom they want to send the message. The selected usernames has to be entered in the textboxes.
- **Inbox:** This form contains a listView showing the messages received by them. This is done by connecting to database.
- **Change Password:** This form is used in case any user wants to change their password. This requires the following details to be entered username, email id, new password. On click of the button “Reset” the software checks for a match in username and email id, and then it updates the password with the text entered in textbox corresponding to new password.
- **Upload file:** On selecting “Upload File” this form is redirected to another form named Upload\_File. This recently opened form contains a text box to take the name of the file you want to upload. To select the file you want to upload click on the button “select” and browse for the file you want to want to upload.

## Resource Manager

- **Search:** In this form the user can search among evens on basis of any substring from the table containing all the details of the event
- **View profile:** The user can view his profile regarding the tasks assigned to the volunteers and the approved events.
- **Change password:** This form is used in case any user wants to change their password. This requires the following details to be entered username, email id, new password. On click of the button “Reset” the software checks for a match in username and email id, and then it updates the password with the text entered in textbox corresponding to new password.
- **Upload file:** On selecting “Upload File” this form is redirected to another form named Upload\_File. This recently opened form contains a text box to take the name of the file you want to upload. To select the file you want to upload click on the button “select” and browse for the file you want to want to upload.
- **Contacts:** In this form the position of every user is displayed in a list box.
- **Tasks:** Menu strip item “Tasks” has two sub-menu items namely Assign and view tasks. You can assign tasks to volunteers individually or in groups. If you go for assigning tasks individually you will be directed to the form “r\_tasks\_individual” which contains a dropdown box which contains list of volunteer names .This dropdown menu is connected to the database project.volunteers. Name of the Item and no. of pieces should be added into corresponding text box. Deadline should be selected from a date time picker. On click of the button “Add” all this information is added to the database table project.r\_tasks. Similarly, if you go for assigning tasks in groups you have to select “to a group” which will be redirected to the form “r\_task\_individual1” which is similar to the form “r\_task\_individual” except in the place of

selecting an individual username in the previously discussed form here we select group name .the database to this information gets passed is project.r\_tasks\_grp.

- To the left top corner of the Form\_Resource\_Manager there is a button named “Track Performance” which on click is directed to the form r\_performance. This form displays the progress of the volunteers individually as well as in groups. Display is made using a chart with usernames (group number) on x-axis and a performance value on a scale of 100 on the y-axis. This performance value is calculated as  $[(\text{tasks completed}) * 50] / [\text{tasks given}]$ .
- There is a button named “Add Resources”, which is used to add a particular resource for an event. On click of this button a new form named “Resources” would be displayed .
- Resources form: This form contains a groupbox which is intended to include name, number of pieces and price for each piece of the resource and their corresponding text boxes. And a button named “add” which on click adds that resources to database table named “items” only if that resource doesn’t exist.

## Financial Manager

The following forms concern themselves with the financial manager.

- 1) **FinanceManager.h**: This at the top has sliding image picture box. Below that there is strip of buttons in the following order:
  - **Home**: On click of this button the tab page below this strip is filled. Among them there is a list box towards right of the tab page. Along with filling of this tab page the list box is filled with the event requests for the financial manager. On selecting one of the request corresponding of the selected request are filled in the text boxes which are present towards the left of the tab page.
  - **Event**: There is data grid view which is filled the details of events which are taken from the database table database.events. Looking at this data grid we can fill the event name and new cost and update the event cost .
  - **Event Calendar**: There is a button along with a text box which display the event calendar for all the events .This text is filled by connecting to the database database.calendar.
  - **Manage Finance**: There is a list box for task approval requests and a corresponding button to clear the list. Then there is a group box for entering cheque details containing following fields – cheque id, bank name, task name, amount, date of submission, and isgroup. There are 2 buttons to cancel or approve the cheque.
  - **Contacts**: This page redirects to the contact information about the people related to the ‘event management system’ .People include the users of this software viz. admin,financeManager,ResourceManager,EventManager and volunteers of the three sections.
  - **View Profile**: Profile of the person who is currently the finance Manager contains photo of the person which he can upload manually, personal contact details, and interests.

- **Tasks:** Finance Manager has to determine the availability of finance for the events, So after the analysis phases from event manager and resource manager the request for introducing new event is notified to the finance manager with all the details. Finance Manager can accept or reject the proposal depending on the budget and requirements status. If he accepts then request is sent to admin for granting permission. If he rejects then request is deleted from the database. The volunteers can be assigned the task in group or individual basis:
- 2) **assign\_task\_to\_individual.h:** This form is redirected from the tag page tasks. Finance manager uses this form to assign an individual task to a volunteer. In this he has to give task name, individual name and amount the task is supposed to collect, and deadline for the task. Following tables' attributes are updated after this action:
    - a. **f\_ind\_tasks:** Table for keeping the status of individual tasks assigned to the volunteers. Volunteers are notified fetching information from this table.
    - b. **f\_volunteers:** This table contains all the volunteers of finance manager section and the tasks assigned to them and the number of completed tasks. number of tasks assigned to the concern volunteer is updated after this action in this table.
  - 3) **f\_assign\_task\_to\_group.h:** This form is too redirected from the tagpage 'tasks' by a button named assign\_tasks\_to\_group. In this form he has to fill the following details of the task: taskname, groupname, amount, deadline. Following tables are updated when he assigns the task to a group:
    - **f\_group\_tasks:** This table contains the columns id, taskname, grpname, status, amount, deadline, status is by default 'not completed'. This gets changed to status completed when finance manager approves the task to be completed.
    - **f\_group:** This contains the id, groupname, tasks\_assigned, tasks\_completed. By this action tasks\_assigned is updated(incremented by 1).
    - **F\_volunteers:** This table contains the id ,volunteer names and their groups tasks\_given, Tasks\_completed and section.By this action tasks\_given column is updated.
  - 4) **f\_Search\_Cheque.h:** To provide the quick\_search about the details of the cheques submitted by individual or group volunteer filtered by task\_name,goup\_name,dateofsubmission,bank,chequeid,All.Substring search is also provided.User has to select the filter from the dropdown menu in the combobox and enter the text in the textbox provided.
  - 5) **Approval of tasks:** After completing the task volunteer will send report to the finance manager for the approval,After approval by the finance manager the amount is added in both the available budget and total budget in the budget table.Total budget is initially with a minimum amount.As volunteers fetch amount completing their tasks,these are added to it.Available budget is the amount available after spending on events.

## Volunteer

A volunteer is directed to one of the following forms according to which category he belongs to. In all the following forms there are two buttons for uploading and downloading files.

- 1) Volunteer\_Default.h: This is the homepage for any volunteer. If a volunteer is not assigned any of the following categories then this form will show.
- 2) Event\_V.h: If the volunteer belongs to event category, he is directed to this form. As the form opens a combobox is automatically filled with the tasks assigned to the volunteer. If the person selects a task, then the details of the task are displayed. Data source – in database *project* table *e\_ind\_tasks*.
- 3) Resource\_V.h: There are two comboboxes for task type and task name. According to the task type selected task names are accordingly loaded. Then there are 3 textboxes by which the volunteer can update the details of the task, which then need to be approved by resource manager. On clicking the update task button the task details are updated. Data source – in database *project* tables - *r\_tasks\_i*, *r\_tasks\_grp*, *volunteers*.
- 4) Finance\_V.h: It is the same as Resource\_V.h form except that the details of a task that are updated by a volunteer are different. . Data source – in database *database* tables – *f\_group\_tasks*, *f\_ind\_tasks*, *f\_task\_approval*.