# (Team Name)

**(*Deep Purple* Art club)**

# Software Design Document

Name (s):

- Flori Heidi

- Popescu Georgiana

- Paun Gabriel

Lab Section: Workstation:

Date: (03/26/2021)

**TABLE OF CONTENTS**

1. [INTRODUCTION 2](#_bookmark0)
   1. [Purpose 2](#_bookmark1)
   2. [Scope 2](#_bookmark2)
   3. [Overview 2](#_bookmark3)
   4. [Reference Material 2](#_bookmark4)
   5. [Definitions and Acronyms 2](#_bookmark5)
2. [SYSTEM OVERVIEW 2](#_bookmark6)
3. [SYSTEM ARCHITECTURE 2](#_bookmark7)
   1. [Architectural Design 2](#_bookmark8)
   2. [Decomposition Description 3](#_bookmark9)
   3. [Design Rationale 3](#_bookmark10)
4. [DATA DESIGN 3](#_bookmark11)
   1. [Data Description 3](#_bookmark12)
   2. [Data Dictionary 3](#_bookmark13)
5. [COMPONENT DESIGN 3](#_bookmark14)
6. [HUMAN INTERFACE DESIGN 4](#_bookmark15)
   1. [Overview of User Interface 4](#_bookmark16)
   2. [Screen Images 4](#_bookmark17)
   3. [Screen Objects and Actions 4](#_bookmark18)
7. [REQUIREMENTS MATRIX 4](#_bookmark19)
8. [APPENDICES 4](#_bookmark20)

### INTRODUCTION

## Purpose

Whether the reader is an experimented developer or simply a customer with big dreams, this SRS will serve as a complete guide to aid them in understanding the functionalities of the application.

## Scope

The web application for the art club “Deep Purple” allows its members to access various events through a simple account creating method. Once the user becomes one of us, they can create events using any available resource for a fair amount of money.

## Overview

*Introduction*: Contains a short description of the website’s purpose along with some explained terms which the user might face when reading the SDD.

*System Overview*: Contains the Use-Case diagram along with a presentation of the website’s functionalities.

*System Architecture*: Contains the modular program structure and a brief description of it.

*Data Design*: Contains the relational model of the database.

*Component Design*: Contains details about the data structure.

*Human Interface Design*: Contains a list of the website’s functionalities and the wireframe diagram.

## Reference Material

*This section is optional.*

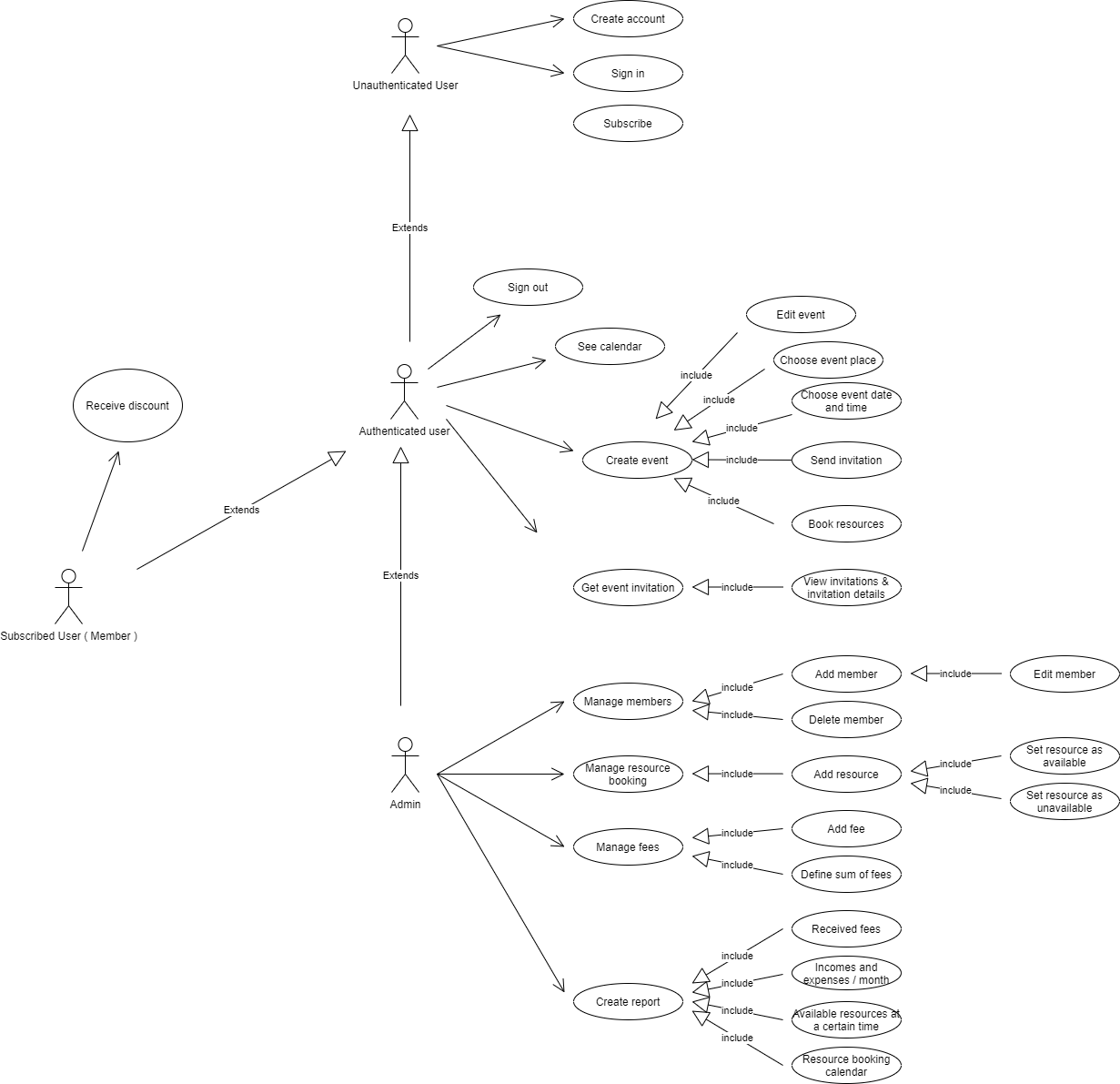
List any documents, if any, which were used as sources of information for the test plan.

## Definitions and Acronyms

|  |  |
| --- | --- |
| User | Any person that accesses the website and owns an account |
| Member | A user that subscribed |
| Administrator | A special member that has the ability to manage events, invitations and see whether the expenses are lower or higher than the earnings and thus decides if there will be any event reservation for the month |
| Resources | Places that users and members can rent for an event |

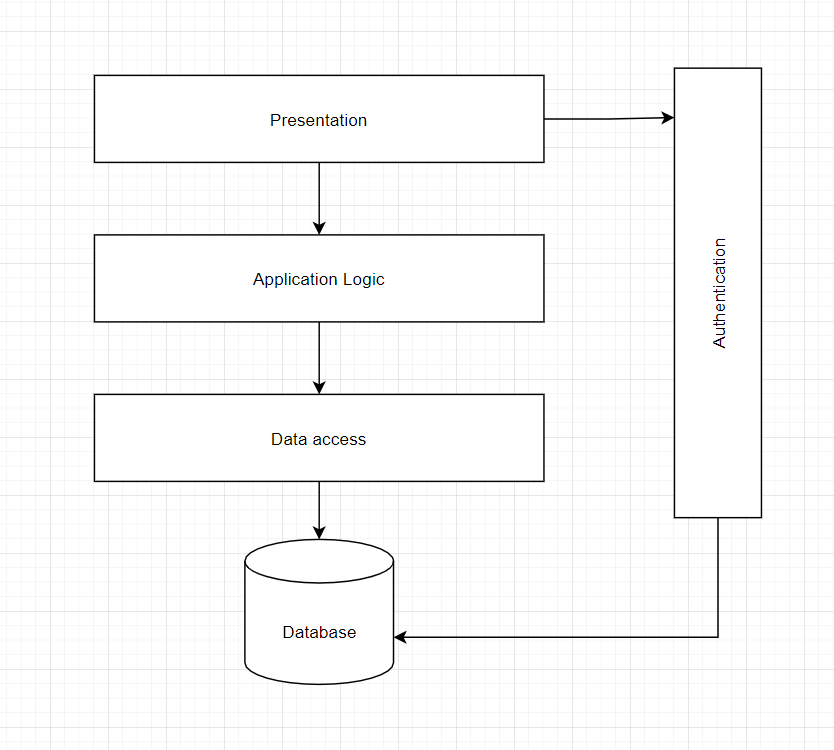
### SYSTEM OVERVIEW

The purpose of the website is to help the art club book events, resources and post articles for members and users. The app also helps the admins maintain and keep track of the financial situation of the art club and make donating and becoming a member easier for day-to-day user.



### SYSTEM ARCHITECTURE

## Architectural Design

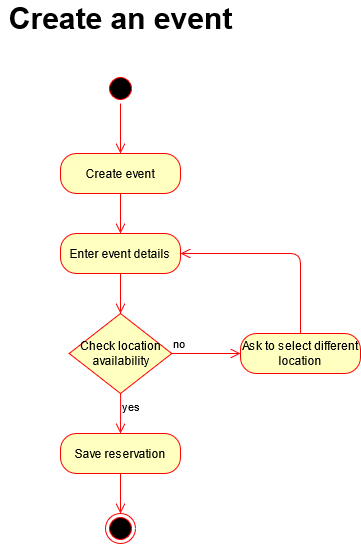


*< The visual part of the website is the front-end part, the presentation. Through the design of the page, the user will have the option to either log in or create an account, so the presentation is connected to the authentication, which gets the data or enters the data in the database.*

*Returning to the presentation, it is bounded to the application’s logic, its functionalities which, some of them, need access to the database to function properly.>??*

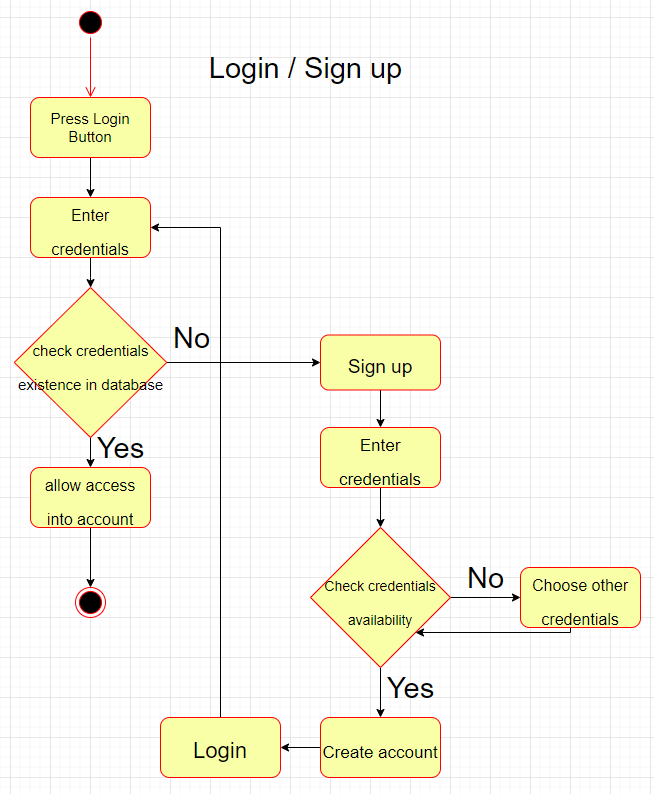
## Decomposition Description

## *< + diagrame clase>*



This diagram represents the description of the “create an event” functionality.

At first, the member will create the event. The details will be entered in the event description and then will choose a resource. The respective resource will have its availability checked then, in case it’s free, the reservation will be made successfully; else, they will be required to choose a different resource (location).

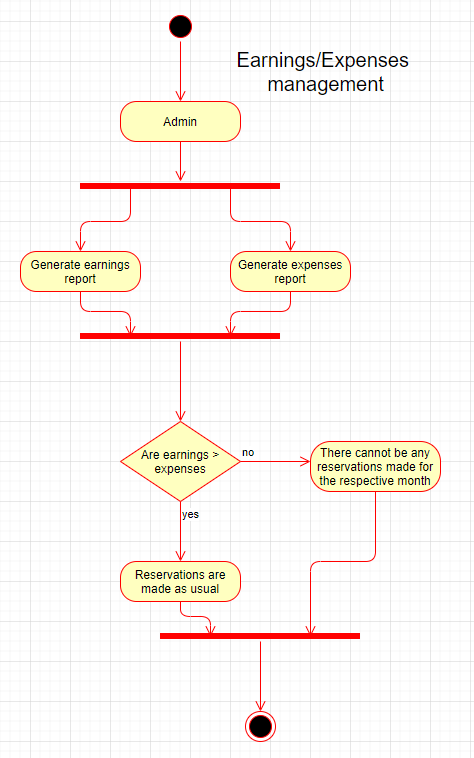


This diagram represents the description of the “Login/Sign up” functionality.

Pressing the Login button, the user will be required to enter their credentials. Then, the credentials will be checked in the database so that, if they coincide with the ones entered by the user, the member is granted access into the account. Else, if the credentials don’t exist in the database, the user will be required to make an account through the Sign up form.

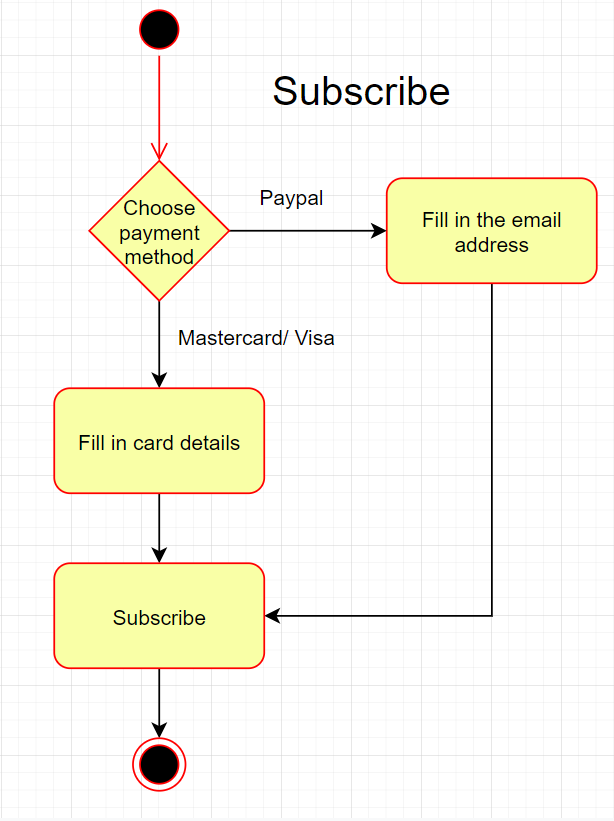
Then, the user will enter the credentials he/she wishes to use. Their availability will be checked and, in case they are free, the account will be created and the user will be redirected to the Login page once again.

However, if the credentials are already used, the user will have to choose other credentials that will be checked again, and so on until an available combination of credentials will be found.



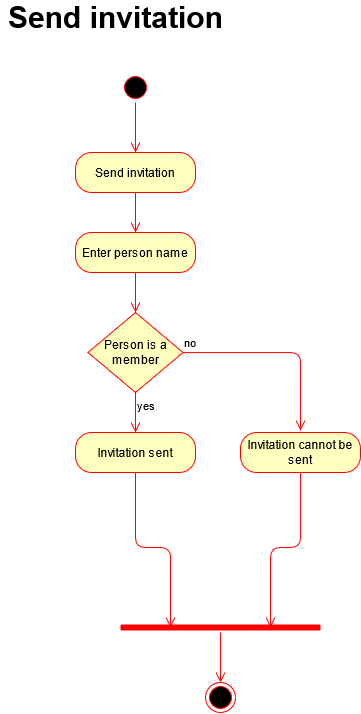
This diagram represents the description of the “Earnings/expenses management” functionality.

Each month, an admin will generate both an earnings report and an expenses report. Then, it will be checked if the expenses amount is lower than the earnings. If it’s true, the reservation of resources will continue normally that month; else, there cannot be any reservations made for that specific month.



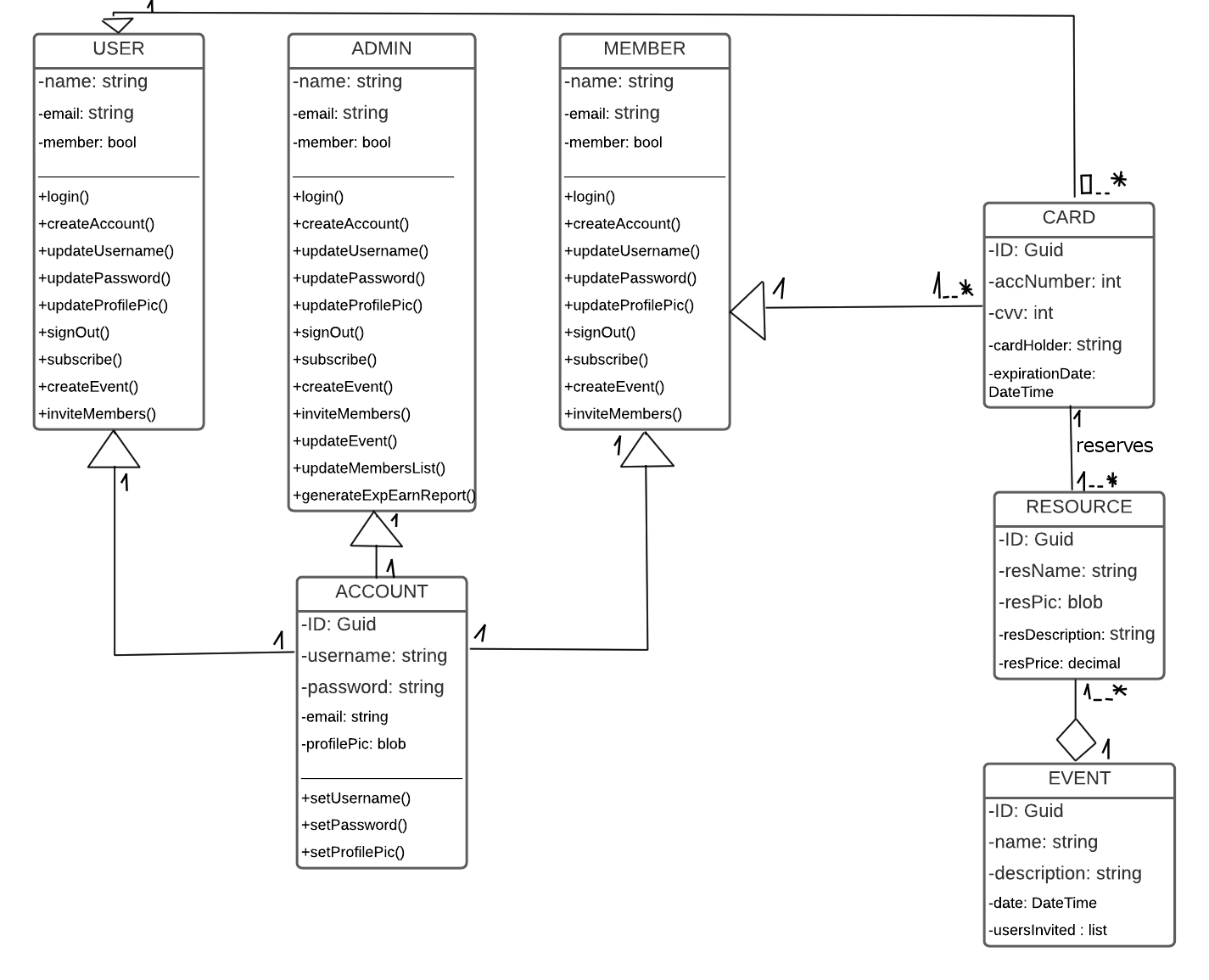
This diagram represents the description of the “Subscribe” functionality.

First, the user has to choose a method to pay through. If they choose the PayPal method, they will be required to enter their email address. If they choose to pay with a MasterCard or a Visa card, they will have to fill in the card details. Pressing the ‘Subscribe’ button, the process will end and the user will now have a subscription.



This diagram represents the description of the “Send invitation (to members)” functionality.

The user will have the option to send an invitation to a person. The user will enter a name and the application will check of the person is a registered member. If the person’s name is found in the database, the invitation will be sent, otherwise the person cannot receive an invitation.



## Design Rationale

The chosen structure is relatively simple, yet it perfectly portrays the functionality of the website.

First of all, what we first see when we enter the website is the visual part, its front-end, designed as a presentation meant to attract the client through the esthetic.

As the user dives deeper into the website, he/she will notice the most important functionality of it, the authentication, where they can either log in into the account or create one. The data provided by the user is checked (for login) or added (for sign up) into the data base, so there’s a strong correlation between authentication and database.

Now, returning to the presentation, it is bounded to the application’s logic through its functionalities which, some of them, may even require data from the database, and therefore it needs access to it.

### DATA DESIGN

**<img model relational baza de date>**

## Data Description

## 

## Data Dictionary

## <NU TREBUIE>

### COMPONENT DESIGN

**<detalii structura de date, dc am ales datele respective?>**

In this section, we take a closer look at what each component does in a more systematic way. If

you gave a functional description in section 3.2, provide a summary of your algorithm for each function listed in 3.2 in procedural description language (PDL) or pseudocode. If you gave an OO description, summarize each object member function for all the objects listed in 3.2 in PDL or pseudocode. Describe any local data when necessary.

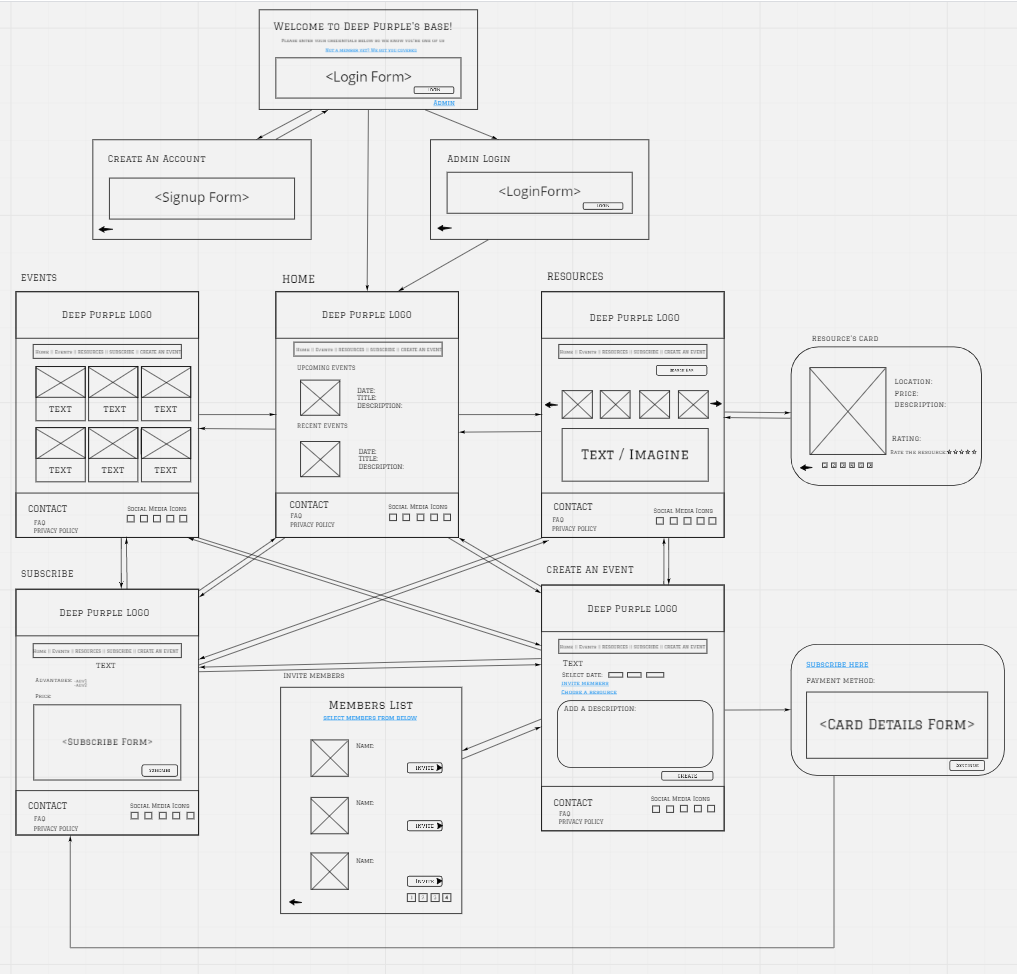
### HUMAN INTERFACE DESIGN

## Overview of User Interface

**<descriere + functionalitati interfata grafica>**

## Screen Images

## <diagrame de wire frame/mock up>



## Screen Objects and Actions

**<NU TREBUIE>**

A discussion of screen objects and actions associated with those objects.

### REQUIREMENTS MATRIX

**<NU TREBUIE>**

Provide a cross­reference that traces components and data structures to the requirements in your SRS document.

Use a tabular format to show which system components satisfy each of the functional requirements from the SRS. Refer to the functional requirements by the numbers/codes that you gave them in the SRS.

### APPENDICES

**<NU TREBUIE>**