Software Design Document (SDD) Template

Software design is a process by which the software requirements are translated into a representation of software components, interfaces, and data necessary for the implementation phase. The SDD shows how the software system will be structured to satisfy the requirements. It is the primary reference for code development and, therefore, it must contain all the information required by a programmer to write code. The SDD is performed in two stages. The first is a preliminary design in which the overall system architecture and data architecture is defined. In the second stage, i.e. the detailed design stage, more detailed data structures are defined and algorithms are developed for the defined architecture.

This template is an annotated outline for a software design document adapted from the IEEE Recommended Practice for Software Design Descriptions. The IEEE Recommended Practice for Software Design Descriptions have been reduced in order to simplify this assignment while still retaining the main components and providing a general idea of a project definition report. For your own information, please refer to [IEEE Std 1016­1998](http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee-SDD.pdf)1 for the full IEEE Recommended Practice for Software Design Descriptions.

1 <http://www.cs.concordia.ca/~ormandj/comp354/2003/Project/ieee>­SDD.pdf

# (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”, still at its first version, 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

Provide an overview of this document and its organization.

## Reference Material

*This section is optional.*

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

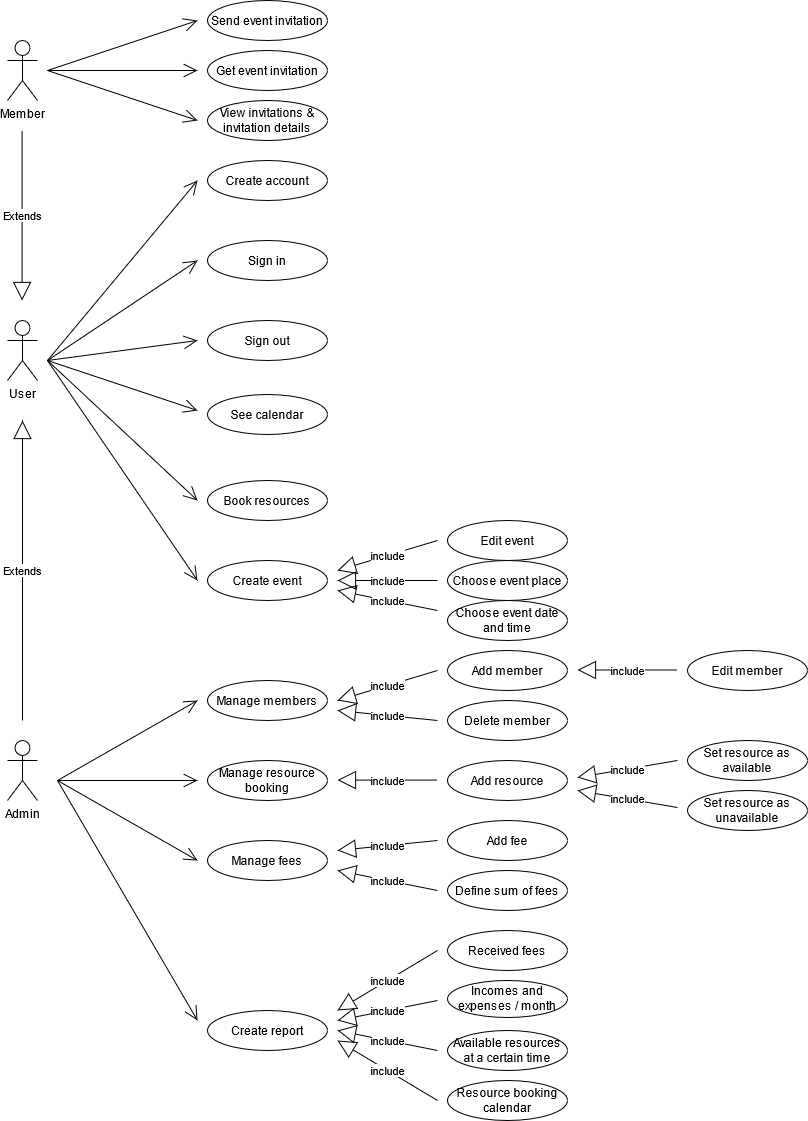
## Definitions and Acronyms

*This section is optional.*

Provide definitions of all terms, acronyms, and abbreviations that might exist to properly interpret the SDD. These definitions should be items used in the SDD that are most likely not known to the audience.

### SYSTEM OVERVIEW

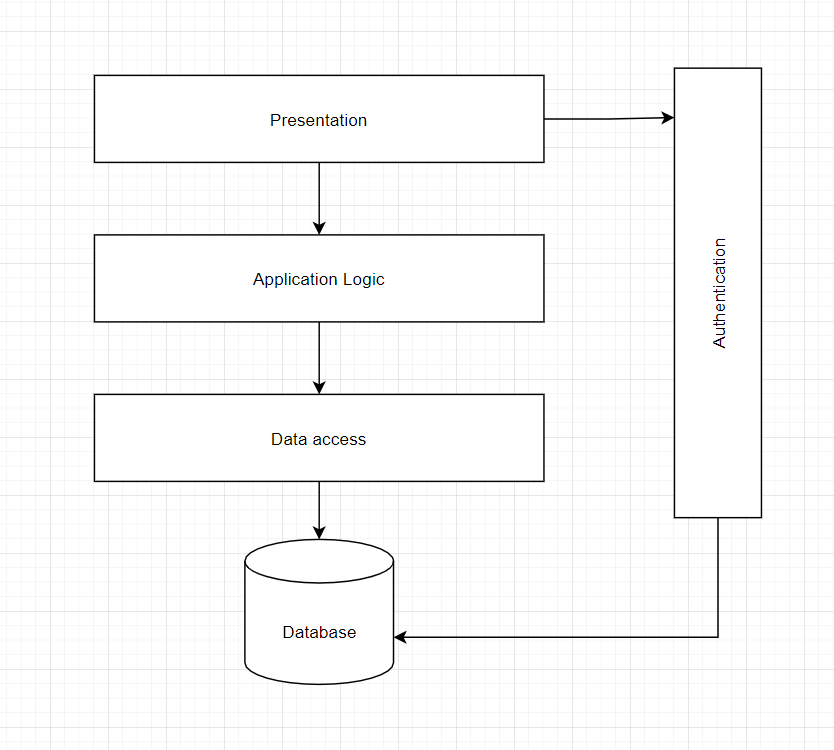
The purpose of the website is to help the art club book events, resources and post articles for members and users. The app also helps the admins maintain and keep track of the financial situation of the art club and make donating and becoming a member easier for day-to-day user.



### SYSTEM ARCHITECTURE

## Architectural Design

## <componentele implementate de app noastra + diagramele din primul ex?>

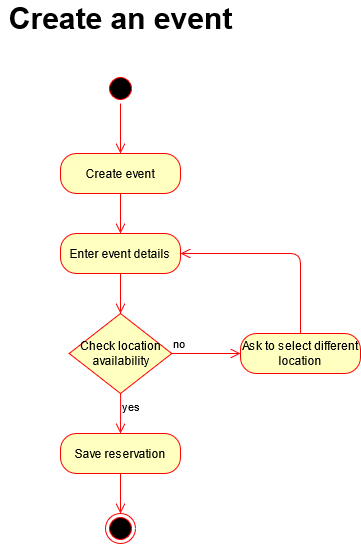


Develop a modular program structure and explain the relationships between the modules to achieve the complete functionality of the system. This is a high level overview of how

responsibilities of the system were partitioned and then assigned to subsystems. Identify each high level subsystem and the roles or responsibilities assigned to it. Describe how these subsystems collaborate with each other in order to achieve the desired functionality. Don’t go into too much detail about the individual subsystems. The main purpose is to gain a general understanding of how and why the system was decomposed, and how the individual parts work together. Provide a diagram showing the major subsystems and data repositories and their interconnections. Describe the diagram if required.

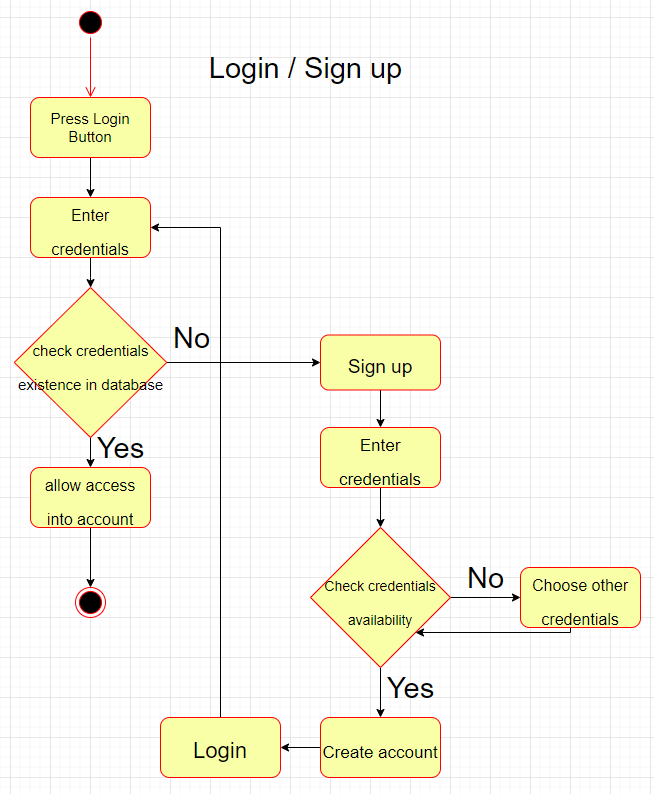
## Decomposition Description

## *< + 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).

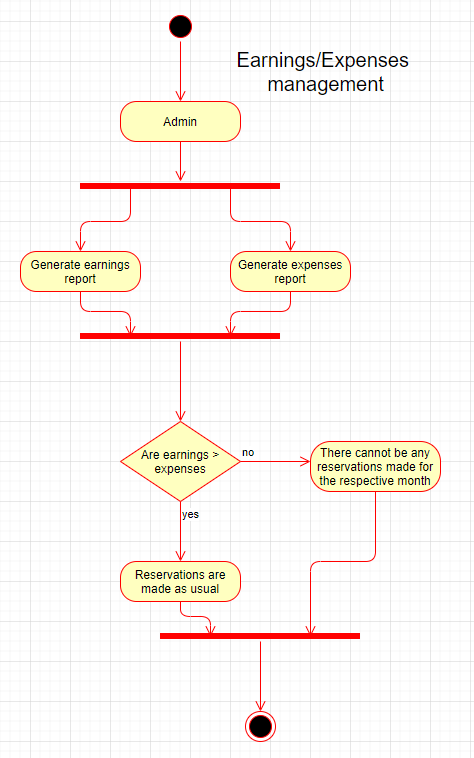


This diagram represents the description of the “Login/Sign up” functionality.

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.

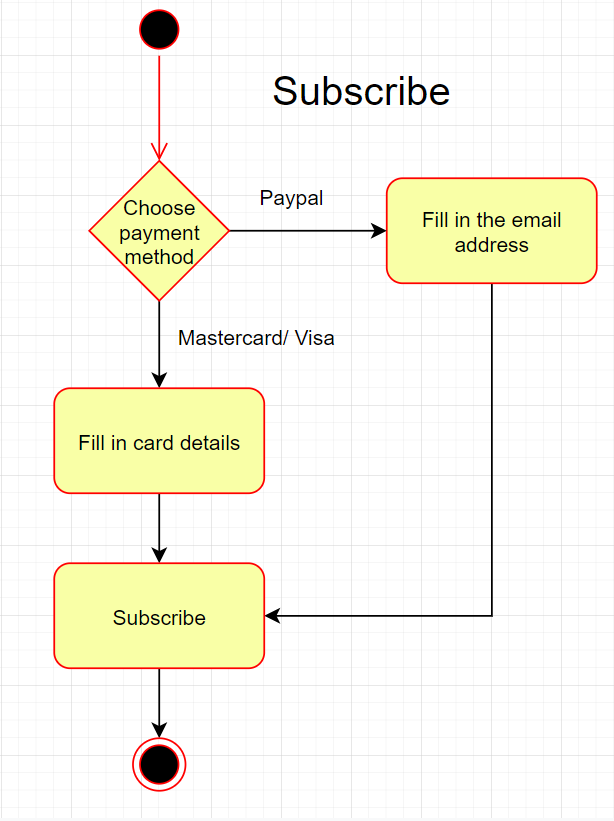
Then, the user will enter the credentials he/she wishes to use. Their availability will be checked and, in case they are free, the account will be created and the user will be redirected to the Login page once again.

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.



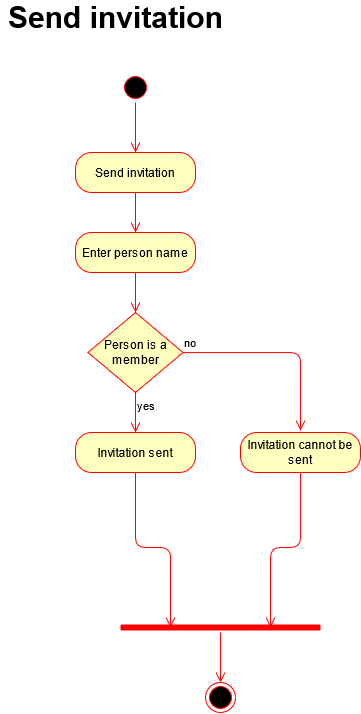
This diagram represents the description of the “Earnings/expenses management” functionality.

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

## <explicatie dc am folosit structura pe care am folosit-o in architecture design>

Discuss the rationale for selecting the architecture described in 3.1 including critical issues and trade/offs that were considered. You may discuss other architectures that were considered, provided that you explain why you didn’t choose them.

### DATA DESIGN

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

## Data Description

Explain how the information domain of your system is transformed into data structures. Describe how the major data or system entities are stored, processed and organized. List any databases or data storage items.

## Data Dictionary

Alphabetically list the system entities or major data along with their types and descriptions. If you provided a functional description in Section 3.2, list all the functions and function parameters. If you provided an OO description, list the objects and its attributes, methods and method parameters.

### 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>**

Describe the functionality of the system from the user’s perspective. Explain how the user will be able to use your system to complete all the expected features and the feedback information that will be displayed for the user.

## Screen Images

## <diagrame de wire frame/mock up>

Display screenshots showing the interface from the user’s perspective. These can be hand­ drawn or you can use an automated drawing tool. Just make them as accurate as possible. (Graph paper works well.)

## 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>**

Appendices may be included, either directly or by reference, to provide supporting details that could aid in the understanding of the Software Design Document.