Wedding Planner

Analysis and Design Document

Student:Suciu Delia

**Group:30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# Proiectarea si implementarea unei aplicatii client-server folosita pentru a putea planifica o nunta online. Aplicatia are 2 tipuri de utilizatori: client si administrator.

Clientul va putea: -inchiria o sala

-alege un meniu si o trupa

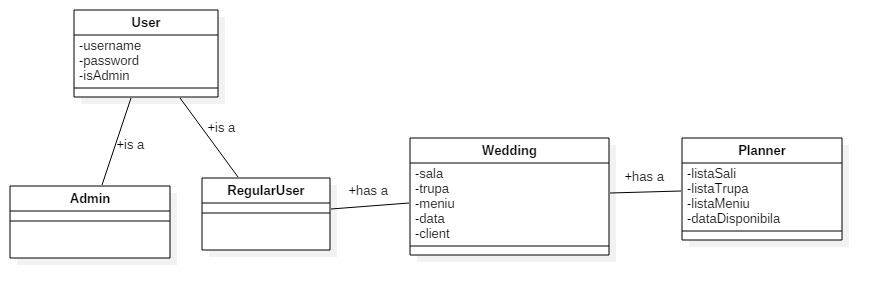
-vizualiza data si serviciile disponibile

Administratorul va putea: - face operatii CRUD

* Accepta rezervari

# Elaboration – Iteration 1.1

# Domain Model

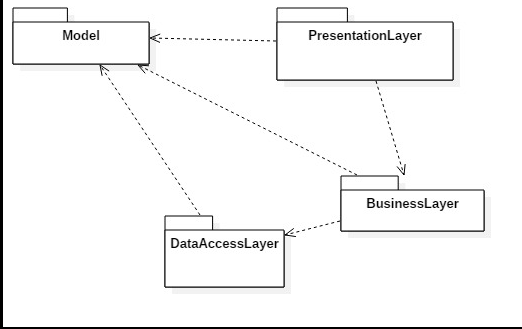


# Architectural Design

## Conceptual Architecture

Am ales sa implementez utilizand design pattern-ul arhitectural Model View Controller pentru a putea separa logica de design de cea de back-end. In plus, folosind acest design patter se poate refolosi partea de interfata si la alte proiecte.

## Package Design



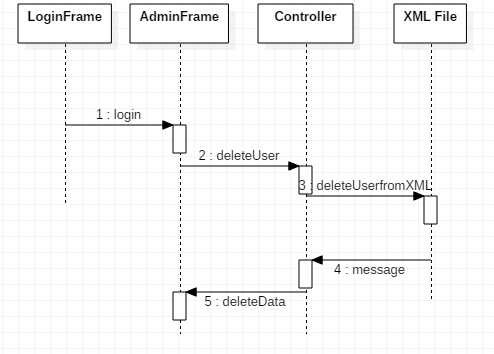
## Component and Deployment Diagrams

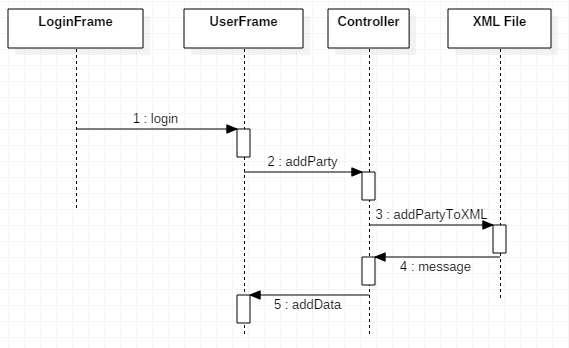
# 

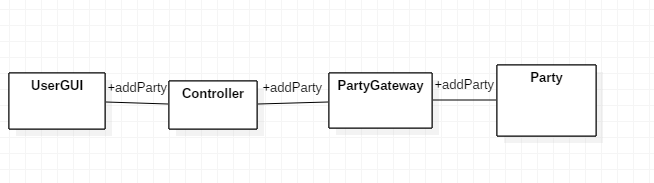
# Elaboration – Iteration 1.2

# Design Model

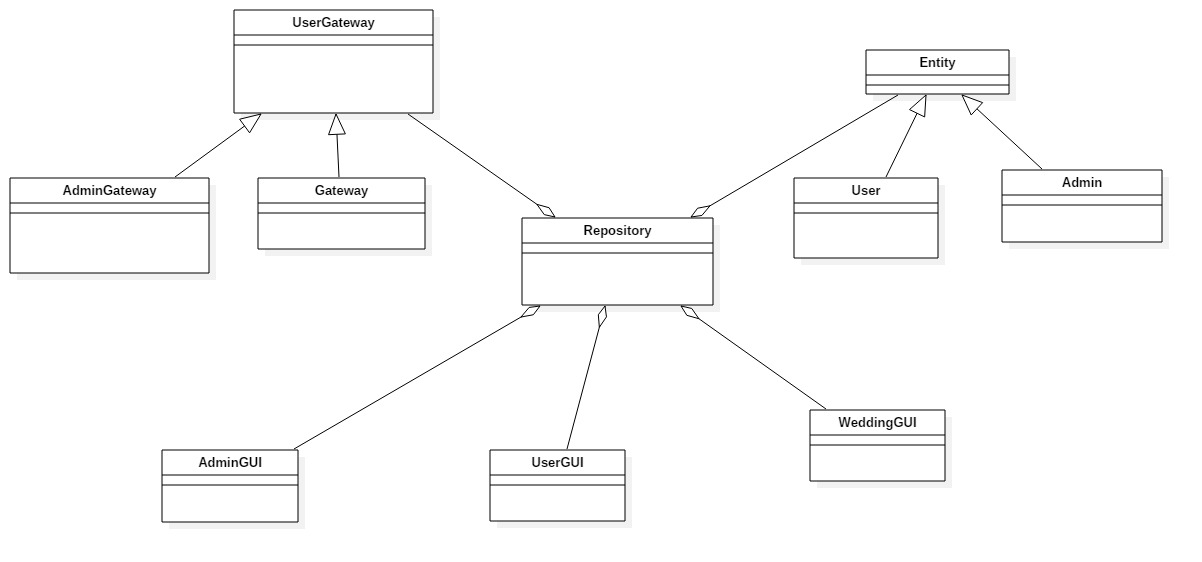
## Dynamic Behavior







## Class Design



# Data Model

Modelul de date folosit in acest proiect este baza de date SQL.

Acest model contine 3 tabele: user, band si room. Tabelul user contine detalii despre administratorii sau clientii care folosesc aplicatia, band contine: numele, pretul si disponibilitatea, iar room contine numele, nr de locuri, pretul si disponibilitatea.

# Unit Testing

# Elaboration – Iteration 2

# Architectural Design Refinement

Pentru implementara acestui program am folosit design pattern-ul arhitectural Model View Controller. Am folosit acest design pattern pentru a putea separa partea de design de cea de back-end.

# Design Model Refinement

In aceasta aplicatie am folosit chei primare in cadrul bazei de date.

# Construction and Transition

# Future improvements

Imbunatatirile ulterioare ar putea fi:

* Adaugarea unui meniu specific fiecarei Sali
* Incorporarea acestui proiect intr-un client-server

# Bibliography