Pawn-shop bets

Analysis and Design Document

Student: Iacob Gabriel

**Group: 30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 23/APR/19 | 1.0 | First take at architecture | Iacob Gabriel |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

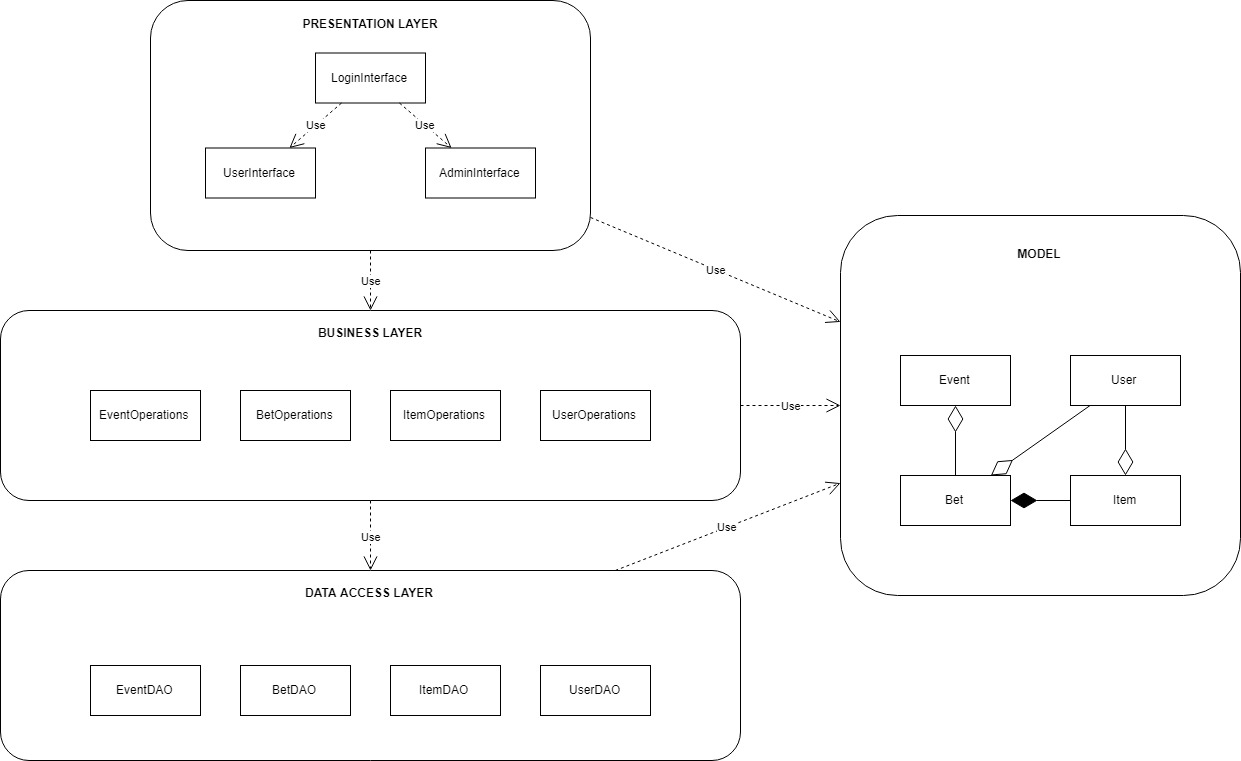
VI. Bibliography 5

# Project Specification

The Pawn-shop bets is an application which allows placing physical objects as bets. The admin will be physically present at the pawn-shop, will receive the item and evaluate its value, insert it into the database and add it to the user’s account. The user can bet one or more items on different events with different stakes. The system will find the closest value item(s) from the users who lost bets and assign them to the winners.

# Elaboration – Iteration 1.1

# Domain Model

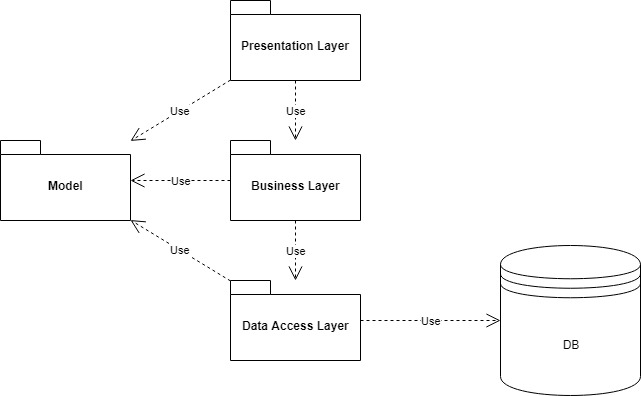
**

# Architectural Design

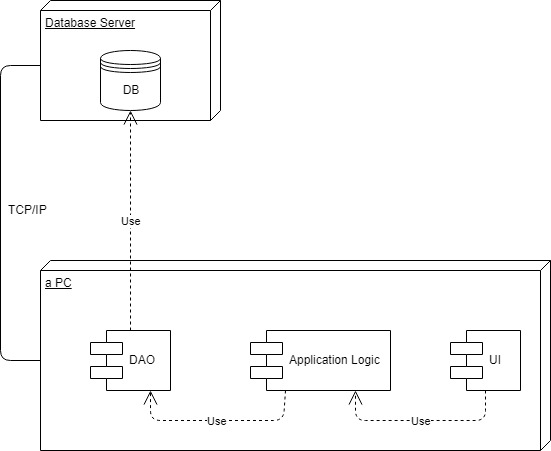
## Conceptual Architecture

The application will be structured around two design patterns: Layers and MVC. To have a more organized project, the user interface will be in the Presentation Layer or Controller, the Business Logic Layer or Service will have methods for the core application functionality, the Entity will have classes modelling real-life objects the application will interact with and finally the Data Layer or Repository will have database access methods for specific objects.

## Package Design

**

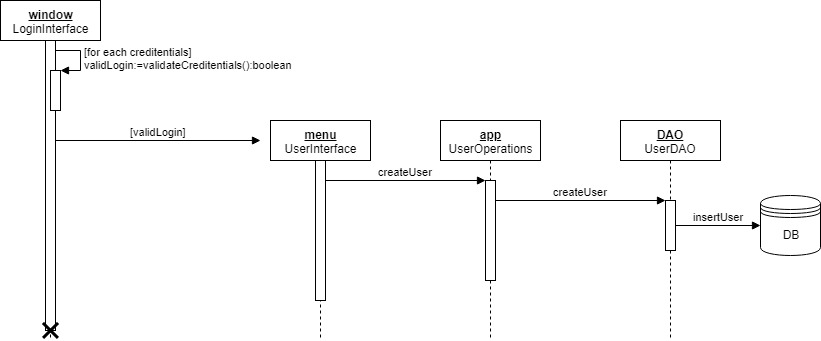
## Component and Deployment Diagrams



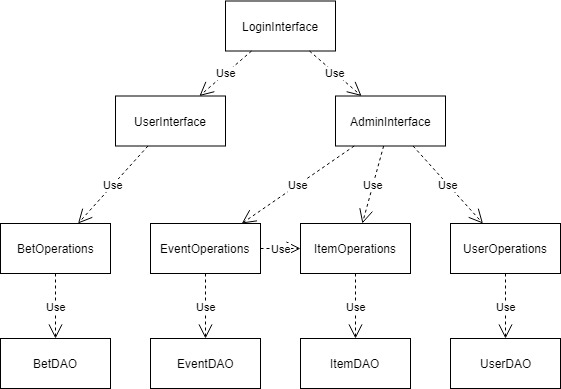
# Elaboration – Iteration 1.2

# Design Model

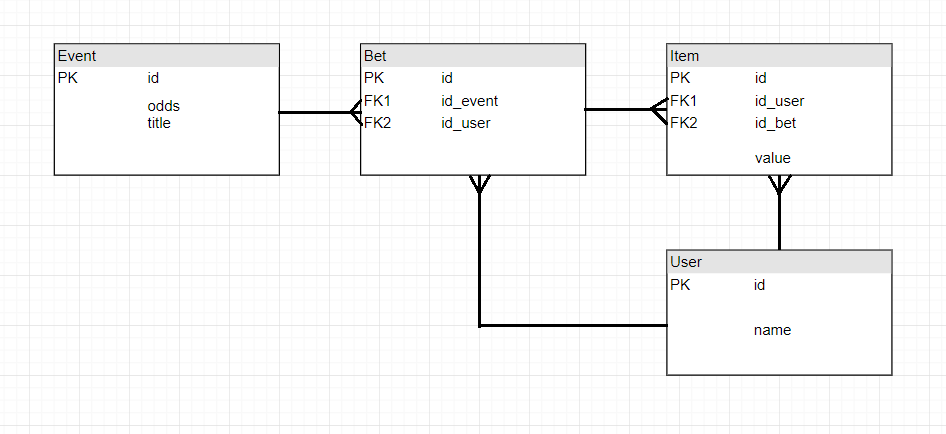
## Dynamic Behavior

**

## Class Design

**

# Data Model

**

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

*[Describe how you applied integration testing and present the associated test case scenarios.]*

# Future improvements

*[Present future improvements for the system]*

# Bibliography