Android Blood Bank

Analysis and Design Document

Student: Danila Vlad-Mihai

**Group: 30432**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 4/4/2018 | 1.0 | Preliminary domain model, architectural design and deployment diagram | Danila Vlad-Mihai |
| 25/4/2018 | 1.1 | Design model, data model | Danila Vlad-Mihai |
|  |  |  |  |
|  |  |  |  |

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

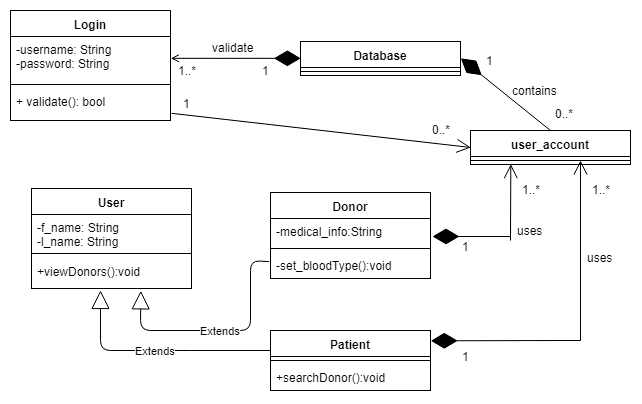
VI. Bibliography 5

# Project Specification

The Android Blood Bank project is a system whose intent is to create an android-based tool designed for people involved or willing to be involved in blood transfusion events.

# Elaboration – Iteration 1.1

# Domain Model



# Architectural Design

## Conceptual Architecture

For this system, the Layers Architectural Pattern is going to be used. We will group the logical functionalities of the application from the technical point of view as follows: Presentation Layer, Business Layer and Data Layer. This division is performed to increase maintainability, readability, reusability and to minimize the number of overlapping functionalities across the entire application.

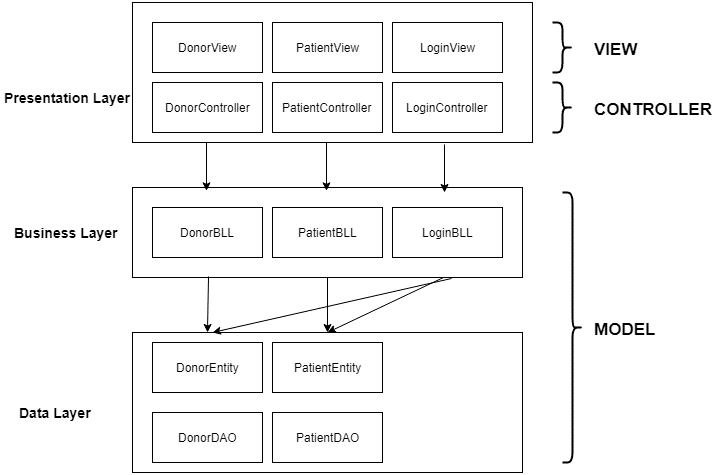
Presentation Layer: provides the application’s user interface.

Business Layer: implements the business functionality of the application.

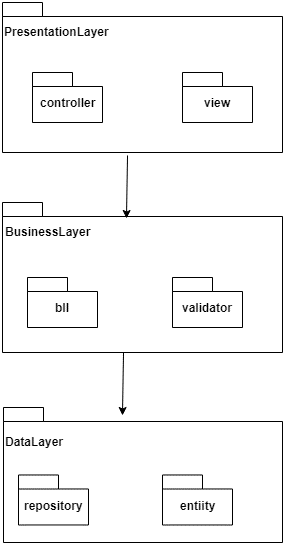
Data Layer: provides access to database.

MVC (Model-View-Controller) architectural pattern will also be use to divide the system intro three interconnected parts:

* Model represents the data and business logic of the application.
* View module is responsible to display data i.e. it represents the presentation.
* Controller module acts as an interface between view and model. It intercepts all the requests i.e. receives input and commands to Model / View to change accordingly.



## Package Design



## Deployment Diagram

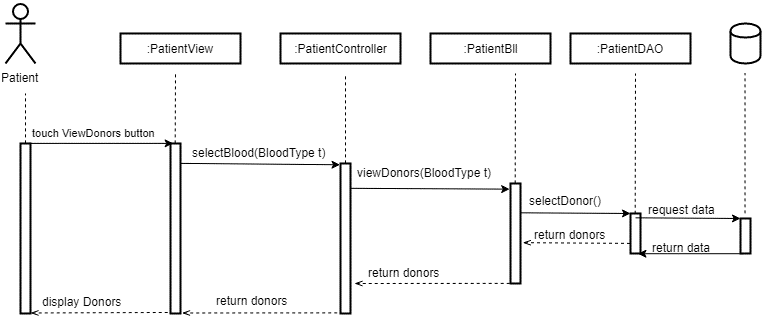
# 

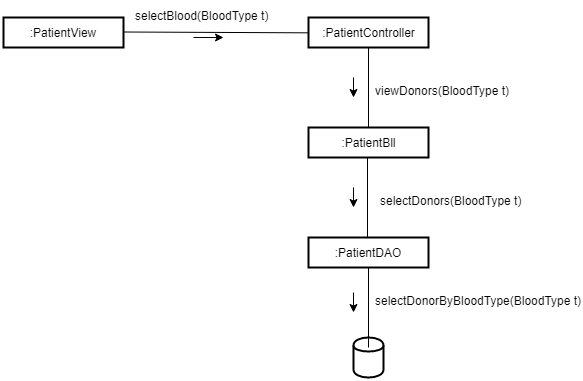
# Elaboration – Iteration 1.2

# Design Model

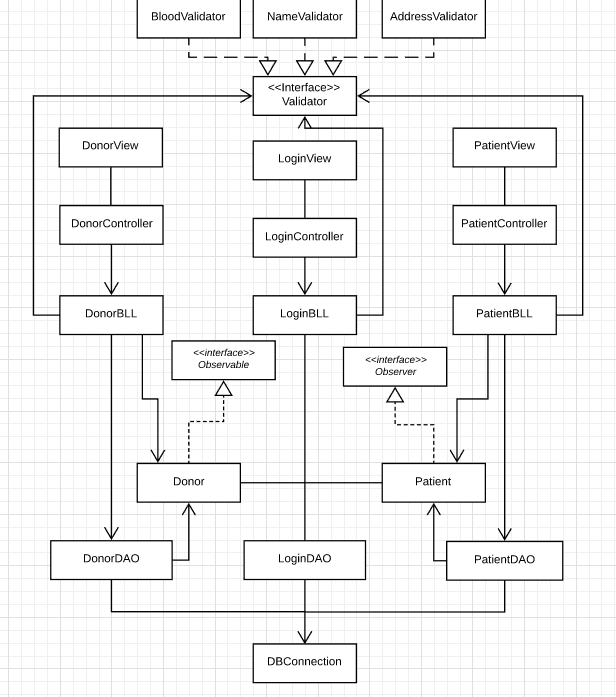
## Dynamic Behavior

Scenario: Search for a compatible donor



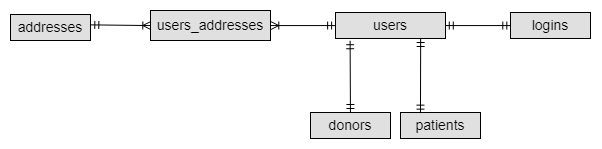


## Class Design



Observer Design Pattern will be used. The Donor will be the observable class from which the Observers (i.e. patients) will get data IF a compatible blood type encounters.

# Data Model



# Unit Testing

We will write tests to verify that a relatively small piece of code is doing what it is intended to do. Junit will be used for testing at method or class level, by checking that the actual output matches the expected output.

# 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