Health Track

Analysis and Design Document

Student: Sabou Cezara

**Group: 30431**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 17/03/2020 | 1.0 | Project Deliverable 1 | Sabou Cezara |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

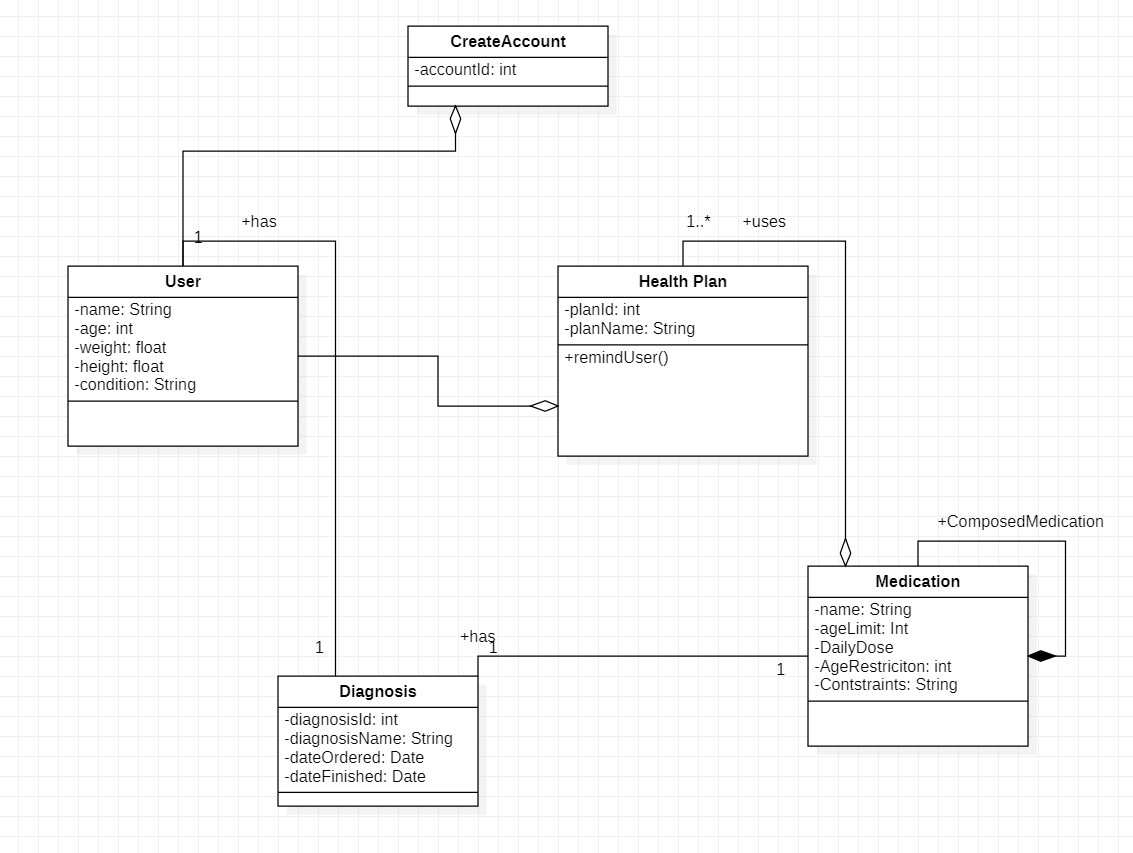
*For the Health Track Web Application, the programming language which will be used is Java. The application should focus on having a main user which is the patient, which would regularly input their condition, and based on it the system would respond by returning an efficient, safe and correctly implemented health plan which would remind the user to take their medicine on time. CRUD operation will be used to store all the information about the user and the medicine.*

# Elaboration – Iteration 1.1

# Domain Model

*The domain model is a conceptual model that incorporates both behaviour and data. Due to the fact that the project has not yet been developed, the classes are few and they are not yet correctly constructed. This is just a small representation about the idea. The conceptual*

*classes which currently suit the stage of the application will be presented in the following diagram:*

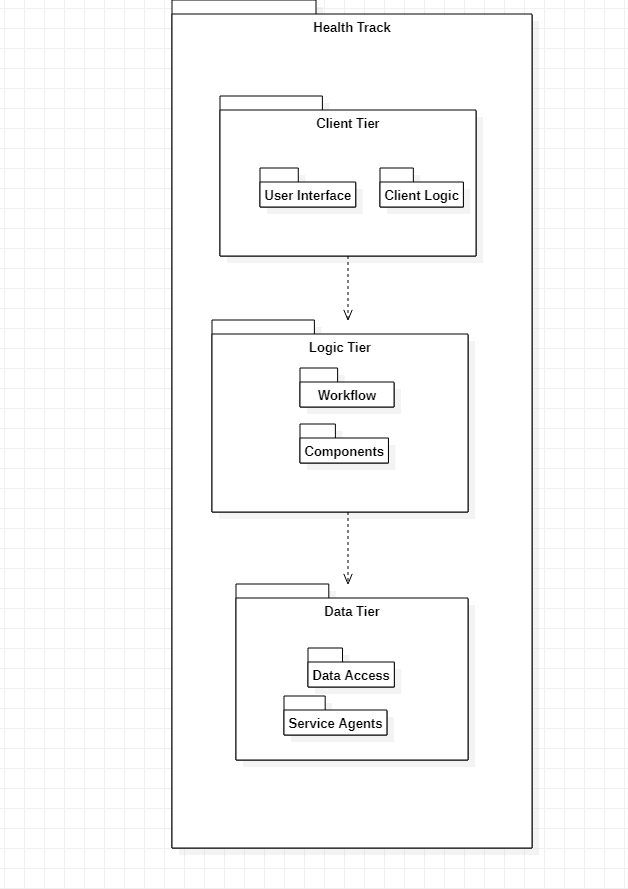
**

# Architectural Design

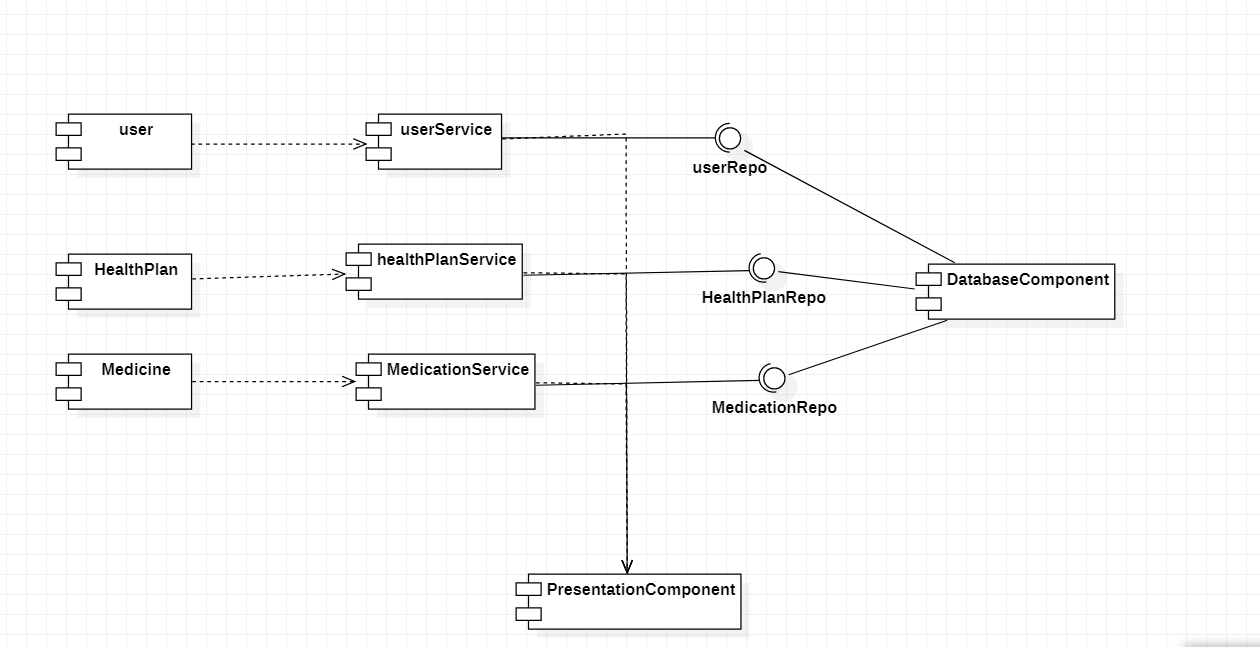
## Conceptual Architecture

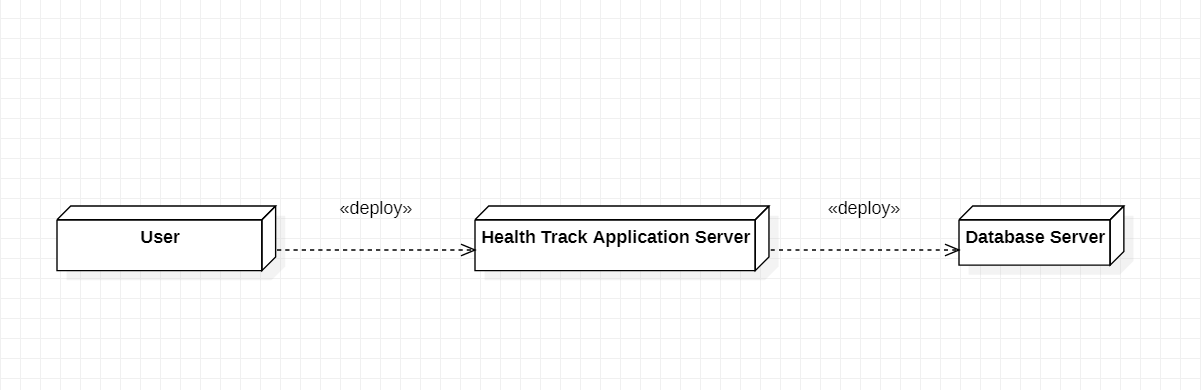
*The best architecture for the Health Track Web Application would be a layered architecture because it works best for web services. The 3-Tier architecture seems to be the best one to use as it is simple and easy to understand. For now there will be these three tiers : Client Tier, Logic-Tier which contains the Web Framework and the WorkflowSystem and the Data Tier, which contains the searchEngine and the Database access.*

## Package Design

**

## Component and Deployment Diagrams

**

**

# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# 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

*The test cases which were considered at the current state of the project are the following*

*Test cases:*

* *Login - the user should login successfully by using their username and password*
* *Registration - the user should be able to register by introducing their data such as: name, age, height, weight, condition, and after that the information will be added to the database*
* *Logout - if a certain button would be pressed by the user, the logout should be successful if they have already logged in*
* *Update condition - the user should be able to update their condition*
* *Update weight - user successfully updates weight*
* *Update height - user successfully updates height*
* *AddMedication - the user introduces the medication successfully and the database gathers data about it*
* *GetPlan - the system returns a functional plan based o the condition and medication introduced by the user and sends notifications whenever necessary*

# Future improvements

*This Health Track application has not yet been developed. Further modifications need to be done in the class diagrams, the component diagrams, the use cases, as well as the Testing area.*

# Bibliography